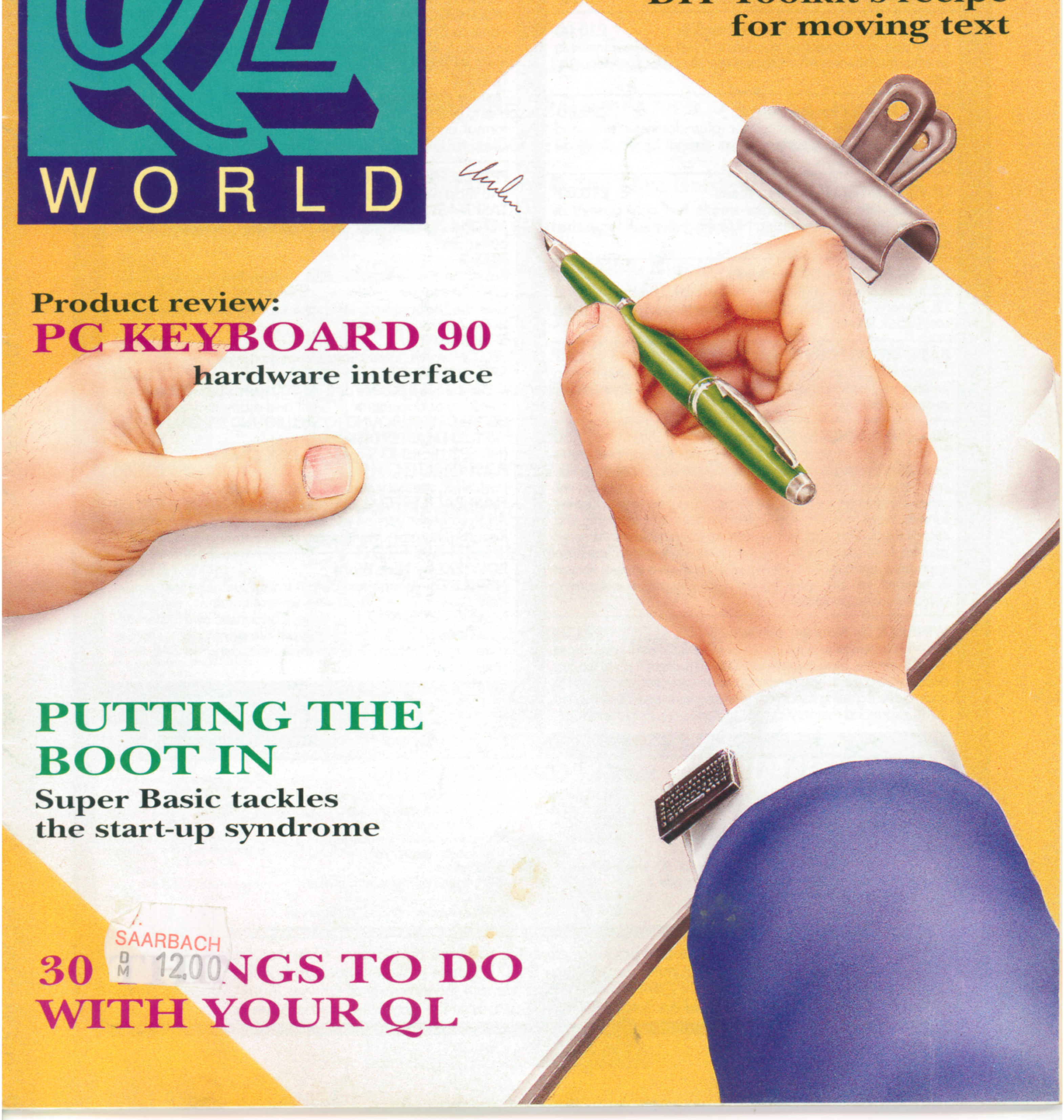
SINCLAIR QL WORLD

EVERY MONTH FEBRUARY 1991 £1.75



QL QLIPBOARD

DIY Toolkit's recipe for moving text

Product review:

PC KEYBOARD 90

hardware interface

PUTTING THE BOOT IN

Super Basic tackles the start-up syndrome

30 [illegible]NGS TO DO WITH YOUR QL

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL7 3YT
Tel: Bangor (0248) 354023

COCKTAILS WAITER by Imre Dominik **£10.00**
Over 400 cocktail drinks recipes, great for parties, Christmas and the New Year. Tell it what ingredients you have, it will suggest a drink! Needs 128k expansion to run.
EXTRA RECIPE SETS (over 300 each) **£5.00**

SUPER DISC LABELLER by Imre Dominik **£10.00**
Put a disc in a drive and it prints a label or sleeve insert in small print showing what files are on disc. Specify wildcards, etc.

HOME BUDGET by Joe Haftke **£20.00**
Personal income tax calculator plus domestic bills and accounts budgeting system. Does capital Gains Tax cost indexations too.

QUICK POSTERS by Dilwyn Jones **£10.00**
A handy little utility to make simple text only posters in minutes, large text, centering NLQ etc. NB check for printer compatability!

VISION MIXER by Dilwyn Jones **£10.00**
QL screen display utility, over 100 effects. Use mode 4 or 8 QL screens. Suitable for shop window display, video titling etc. (256k RAM).

BASIC REPORTER by Dilwyn Jones **£10**
SuperBASIC programmers aid, list names, lines, keywords, extensions, procedure/FN calls and so on. Indent BASIC program listings!

QL GENEALOGIST by Chris Boutall **£19.50**
* NEW * Record your family history with this comprehensive genealogy database program. It is suitable for a first family tree and the needs of the serious genealogist alike. This program will store, display and print your family record in a variety of different formats, keep track of your research data with indexing and search capabilities. Fast and responsive, compatible with Minerva and the Atari QL emulator. Multi-tasking. Please say if you have expanded memory or not when you order, as a special version is needed for an unexpanded QL. The price includes a comprehensive printed manual.

WINBACK by Norman Dunbar **£25.00**
NEW! A program to back up a MIRACLE SYSTEMS hard disc to any QL device, even microdrive!Only files altered since the last backup are copied. Specify directories to copy fromand device to copy to and if a listing is wanted. This program needs Toolkit 2 and 128K expanded memory to run. Includes a printed manual.

DAVE WALKER SOFTWARE
DISCOVER (NB min. RAM 256k) **£20.00**
QL to MSDOS/PCDOS disc format conversion aid. This is also the format used on the Atari ST. No cables or extra hardware needed!
MULTI DISCOVER NB min. RAM 256k) **£30.00**
Same as Discover, but transfers between more disc formats, including BBC DFS and ADFS, CPM (many CPM formats) and UNIX CPIO format.
TEXTIDY **£15.00**
Text file conversion utility. Convert "DOC" file to plain text files to plain text files, convert to DOS Quill format, Wordstar format and vice versa. Useful for preparing text files for Discover to transfer to MSDOS format wordproccessors.

PETER JEFFERIES SOFTWARE
(NB most need expanded QL - check with us!)
TASKMASTER **£25.00**
Multitasking front end utility, calculator, notepad, printer buffer, etc
FILES 2 **£12.00**
Based on the Taskmaster files system, this is a superb file copy, delete, view, rename, directory etc. Much enhanced over the Taskmaster system. Use by itself or as an upgrade to the Taskmaster system.
SPELLBOUND **£30.00**
30,000 words. Check-spelling-as-you-type.
SPELLBOUND SPECIAL EDITION **£50.00**
(2 Dictionaries) 50,000 and 30,000 word PLUS retrospective checking of documents in Quill and many new features.
UPGRADE SPELLBOUND TO SPELLBOUND SPECIAL EDITION - RETURN MASTER DISK PLUS **£30.00**
(NB SPELLBOUND S.E. IS ON DISK ONLY)
FLASHBACK **£25.00**
Fast, slick, memory resident database system
FLASHBACK SPECIAL EDITION **£40.00**
An even better version of Flashback, with new commands, Report generator, Mailmerge, Label printing and so on.

POLYTEXT by Nick Ward **£16.00**
NEW! Multi column text output from Quill. The text output from Quill can be placed into several columns, with many facilities such as NLQ, Pica, Elite, Condensed and Expanded as supported by an Epson compatible printer. Mix sizes in a document, retain Quill attributes etc. Send an SAE for a sample printout.

WE WOULD LIKE TO THANK OUR CUSTOMERS
AND WISH YOU ALL A MERRY CHRISTMAS
AND A HAPPY, PROSPEROUS NEW YEAR

SUNDRY COMPUTER SUPPLIES

3.5" DSDD Discs, unbranded	£0.50 each
20 or more	£0.45 each
Brand new microdrive cartridges	£3.00 each
Disc labels, per 100 on a roll	£2.00
Disc storage boxes: 10 (Ryford)£1.20; Holds 40: £4.50; Holds 80: £6.50	
Disc box headed subject dividers, blank on side 2 dry wipe plastic, set of 20	£3.00
Monitor stands, tilt/swivel, <=14"	£15.50

Address labels, on printer roll	£2.00/100
3.5" Sony/Mitsubishi discs	£0.75 each
20 or more	£0.70 each
Microdrive labels, printer roll	£2.00/100
Disc labels, printer roll	£2.50/100
5.25" discs	35p each
Wire frame printer stand, basket,	£9.00
Printer ribbons - phone, stating type.	

Please add £2.00 for U.K postage and packing, except to software only orders. Overseas, add the postage at cost, remember airmail is extra, minimum postage £2.00. Please make cheques (in British Pounds Sterling only) payable to DILWYN JONES COMPUTING. Goods remain the property of Dilwyn Jones Computing until paid for in full. Send an S.A.E. for further details of the programs and a price list.

CONTENTS

February 1991

SINCLAIR QL WORLD

Editor
Helen Armstrong

Production Controller
Jayne Penfold

Designer
Jeff Gurney

Advertising Sales
Jason Newman

Magazine Services
Sheila Baker

Advertising Production
Michelle Evans

Group Advertising Manager
Jean Dorza

Group Editor
John Taylor

Publishing Director
Ray Lewis

Managing Director
Peter Welham

Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

Unfortunately, we are no longer able to answer enquiries made by telephone. If you have any comments or difficulties, please write the The Editor, Open Channel, Trouble Shooter, or Psion Solutions. We will do our best to deal with your problem in the magazine, though we cannot guarantee individual replies.
Back issues are available from the publisher price £2 U.K., £2.75 Europe. Overseas rates on request.
Published by Maxwell Specialist Magazines, A Division of MCPC Ltd., Sinclair QL World is distributed by SM Distribution.
Tel: 081-677 8111. Subscription information from: MSM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN.
£21.00 U.K., £24.70 Europe, MiddleEast £25.80, Far East £27.60, Rest of World £26.20, U.S.A. $45.00.
Airmail rates available on request.
Typesetting by Ford Graphics,
8-10 Whitsbury Road, Fordingbridge, Hampshire. SP6 1BR. Tel: (0425) 655657
Printing by Cradley Print. (0384) 411411
Sinclair QL World is published on the third Thursday preceding cover date.

NEXT MONTH

THE USER GUIDE GUIDE

Mike Lloyd is working on the QL User Guide Guide and we now hope to have the first instalment for March. We hope that it will be a centre-spread pull-out as well, so that you can store it separately.

HARD DISK COMMENTS

New hard disk owner Bryan Davies shares his experiences so far with the modern medium.

PERFECTION
PERFECTION PLUS

NEW!

DIGITAL PRECISION LTD

Not just a word-processor - this one is THE word processor.

From the same inspired team who brought you classics like LIGHTNING SPECIAL EDITION and PC CONQUEROR, Digital Precision presents a product that will revolutionise the way you use your QL. Let us tell you how **PERFECTION** will do this.

Several hundred thousand QL users have grown familiar with the free word processor that was bundled with the QL. On the plus side, its use could be mastered in a few minutes thanks to its simple menu system, and it is reasonably WYSIWYG (What you see is what you get) in appearance. On the minus side, it is very slow, sometimes idiosyncratic (in what it prevents you from doing, or the roundabout way in which it forces you to go about things that should have been straightforward) and very many commands that we think should have been provided with it simply weren't. Valiant attempts to accelerate it by "patching" it have achieved only a 20% speedup.

There is, however, no getting away from the fact that the majority of QL owners still use Quill as their main program. They have grown used to the user interface of the bundled programs, and are reluctant to invest time in learning some totally incompatible system, whatever its claimed advantages might be. So - using their ingenuity, for QL people are an ingenious tribe - users have put up with the inadequacies and slowness, and enjoyed Quill's friendliness.

You are probably just such a QL user yourself....

Now here is a product CREATED JUST FOR YOU.

A word processor that you can master in just a couple of minutes.

A word processor whose user interface uses precisely those keys that you would expect it - from intuition or experience - to use.

A word processor that is menu-driven (multiple page menus) so there is absolutely no need for you to remember anything or ever refer to the manual: the menu is on the screen all the time. A word processor that is intuitively obvious to operate: even more obvious than was the bundled one. If you have become at all used to the F3 interface, you will love this enhancement! A word processor that is designed for Absolute Beginners and Advanced Users, for Complete Wallies and for Albert Einsteins alike.

A word processor that is delightful to use for letters and documents of but a few pages as well as for articles, journals, magazines, books, theses or manuscripts hundreds of pages long.

A word processor that can unleash the power of your printer, whatever its make, and squeeze the very best from it. A word processor with a very flexible user-configurable printer driver: one, however, that you should never have need to configure(!), as it works as shipped with Epson-compatibles and most non-compatibles, and if you have an esoteric printer, it can utilise your existing configured Quill printer data file, automatically, if you want it to! With this word processor you do not have to buy any extra printer drivers - you get everything you need right from the start.

A word processor with full on-screen indication of character mode - bold (i.e. emphasised) appears bold on screen, underline appears underlined, italics appear in italics, superscript and subscript appear superscripted and subscripted.... Other "special" type modes - dependent on the capabilities of your printer, like switching fonts, pitches, NLQ/draft mode, proportionality, double-strike, or anything else you choose - are indicated on screen by variations in ink/strip colour combinations, just as the most advanced PC word processors do. You can even make up your own "attributes" to be displayed on-screen in a particular ink/strip. Combinations of attributes are permitted - the display copes fine. Never before have things been so clear and simple....

That **PERFECTION** manages to do all this is remarkable. That it manages to do it at all fast would be amazing. But the truth is, in fact, much much better. **PERFECTION** is by far the fastest word processor for the QL, being DOZENS of times faster than Quill on many operations (a **minimum** of five times faster than it on everything), and - yes - many times the speed of our own beloved and excellent Editor, and far far ahead of all the others.

This may seem impossible to you. Two years ago, before we started work on **PERFECTION**, it would have seemed impossible to us as well! But a remarkable bit of software technology has enabled us to achieve this incredible acceleration. Of course you don't need to know or understand how we have accomplished all this in order to enjoy to the full the benefits of **PERFECTION** speed: if you want to know anyway, look at the technical section later on.

If you already use Cuill or ANY OTHER QL WORD-PROCESSOR, you will be overjoyed to know that **PERFECTION** can load your existing saved files (_doc or _lis or ASCII) directly, with no conversion process required. This - together with the automatic reading of existing printer driver data - takes all the trauma out of the move to an exciting new system! And **PERFECTION** files are usable with PC/ST/Amiga word processors too.

PERFECTION comes with a multi-function configurator that allows you - if you want - to tailor-make a version specific to your tastes. Practically everything that is settable at run-time is also pre- configurable, making **PERFECTION** comfortable to operate. As you become more familiar with **PERFECTION** and no longer need the menu options to be visible all the time, you can toggle the menu off, freeing all the screen for your document. You can configure **PERFECTION** so that on startup the menu is either visible or not. As you become even more familiar with **PERFECTION**, you can opt to bypass the menu system entirely, and use alternative direct keypress commands to access **PERFECTION**'s power even more rapidly.

PERFECTION natively multitasks (of course) which means that without any other tools you can run multiple copies of it simultaneously, as well as run it at the same time as other pieces of software. Even if you choose to run only one copy of the program, you still have the option to look at more than one part of the document at the same time. You can take a "snapshot" of part of the document, and keep that snapshot in view as you edit a totally different area of the document. Ideal for indexing or cross-referencing. You can set up macros so that making a glossary is easy. Also, you can have any number of blocks - not just one - defined in the document. You can undo/edit attribute changes with a single keypress - there's no need to laboriously "paint" over areas or navigate to the start and end of a highlighted area in order to adjust the attribute!

Being able to cope with human error is an important part of **PERFECTION** philosophy. For example, not only is there an Undo option, but you can also ESCape from any command. When you have right justification on **PERFECTION** will add pseudo-spaces to pad out the line. Pseudo-spaces look like spaces and print like spaces but when you left justify they are removed while real spaces - the ones you have entered (via the SPACE bar or TAB) are not. This means if you accidentally right-justify tabular or columnar data, a simple left-justify will get it back to its exact original state. Most other word processors do not distinguish between spaces you have entered and spaces they have inserted, and hence cannot auto-recover.

Many users need the uuse (sic) of a spelling checker with their word processor. Adequate spelling checkers already exist for the QL, and for users who either do not want a spelling checker, or who do not want one as yet, or who already have one and are on a tight budget, we supply a version of **PERFECTION** without any built-in checker. But to get the best out of **PERFECTION**, we also supply it bundled with a dedicated Spelling checker of unsurpassed speed. There are even two levels of dictionary supplied (you get both) - the larger one is 225,000 words (no more hassle of having a checker which doesn't know the words you use: this dictionary is about 400% larger than its nearest competitor!) and a compact one: use the latter if you are short of memory, or when your document is really huge. You can add new words to the dictionary as well as create new dictionaries. With either dictionary **PERFECTION** PLUS one checks as you type, or checks saved files, or - BEST OF ALL - spell-checks interactively from any one point in the document to another....

If you already have our Editor Special Edition and use it for documents, database work or programming, you will find **PERFECTION** a wonderful treat. **PERFECTION**'s WYSIWYG behaviour, greatly enhanced document facilities ("tells you everything" status line, available word/line/character counts, regular and forced page breaks, headers and footers), menu-driven options and VERY MUCH GREATER SPEED make it an ideal upgrade. There are hundreds of detailed changes - to give but one: paragraphs do not need to have a blank line between them in order to distinguish them any more . There remains an area, however, where Editor Special Edition remains supreme - the editing of "non-printable" data, the ability to handle the entire ASCII character set from codes 0 to 255. So if you are a technical or semi-technical user and do not have either Editor Special Edition or **PERFECTION**, your best buy is the two programs together (they can interact, coexist, work simultaneously and have fully-compatible file formats). You will then get Editor Special Edition at HALF PRICE (Special Offer - limited duration).

The characteristics of a good database are its ability to Store, Retrieve and Manipulate information rapidly. By this criterion, this word processor makes an ideal database system too, as it is blindingly fast and flexible. Forward and backward 'Search' takes at most a couple of seconds, even when you have a document that fills an 896K Trumpcard system to the brim! Cursor navigation is also unbelievably fast and smooth, with an accelerating rate of scrolling if you indicate impatience. And there are macros, programmability and more for the more advanced user. If you have been unhappy with the speed or complexity or non-programmability of your existing database, **PERFECTION** will solve your problems.

PERFECTION can even access your existing Archive export files.

And if you want full desktop publishing capabilities (the use of fonts that your printer does not possess, and graphics) interlinking **PERFECTION** with Professional Publisher is a doddle. Use **PERFECTION** for creating, editing and manipulating, and "pour" the result into Pro Publisher.

But first and foremost **PERFECTION** is a user-friendly, familiar user-interface, stand-alone WYSIWYG dual-control (menus or direct commands) word processor of enormous power and blistering speed, which (for the first time) makes output to printers hassle-free. There is nothing else like it or even remotely as good as it on the QL or on anything else. **PERFECTION** is our best yet.

PERFECTION is for you whether you hate your existing word processor, are indifferent to it or love it. **PERFECTION** will let you forget about all the technology and concentrate only on the writing.

PERFECTION costs just £79.95 including integrated printer drivers, ancillary programs and jargon-free, friendly but to-the-point documentation (that you will probably never need to read through!). **PERFECTION** PLUS comprises **PERFECTION** plus the dedicated Spelling checker with dictionaries and costs just £119.95.

TECHNICAL INFORMATION ON PERFECTION

You don't actually need to read or understand this.

PERFECTION gets its superb speed from two sources. Firstly, **PERFECTION** - unlike virtually any other word processor - is written entirely in 100% hand-written machine code. This gives us a considerable speed advantage over compiled alternatives. Had we written **PERFECTION** in a high level language it would have been 4 times slower, 6 times bulkier and taken us a great deal less time to produce. You reap all the benefits of our hard work.

The other source is design. There are two formats for internal data storage for character handling programs. Many store data serially, in a long stream of characters. Ones like Editor store data as lines scattered through RAM, with a table of pointers to the lines - a far more advanced method. The first format has the advantage that it is cheap to program - the user pays the cost in terms of performance, with sluggish block-defining/moving, navigation and insertion. The second format has advantages including instant random access to any line and quick insertions and deletions - the disadvantages may include heap fragmentation that will result from repeated grabbing of small chunks of space (garbage collection may be required periodically if space is short). Both formats share the disadvantage that "global" changes made to a part of the document - say a switch to bold at the top - will take a long time to filter down through the system to become visible on-screen at lines at the bottom.

PERFECTION uses a variant of the second format that does not have its disadvantages. Data is stored in RAM in optimally-sized chunks - a chunk being roughly the size of several screens. Each chunk has a control information area within it about the number of lines etc within it, the display status at the start of it (say bold on, italics/underline etc off). Whenever you are editing, the relevant chunk(s) are instantly loaded into a large work area that has slack space at both top and bottom. That means that you can add or delete a great amount of data instantly, without **PERFECTION** having to bother about updating anything but the work area. Only when you move over the edge of the work area will **PERFECTION** need to housekeep outside the work area: the housekeeping itself is then very very fast, as only control information areas need to be updated. There is no need for a general scan through all following text. There are many more speed and power advantages to our system. There is one big disadvantage - it is an absolute nightmare to design and implement! Fortunately for you, you don't have to know anything about it - it just works like clockwork, automatically and behind the scenes.

Other elements of **PERFECTION** design to enhance performance include lazy screen (when you keep a key pressed in order to get somewhere, we stop updating the whole screen and instead just scroll the line your cursor is on) and lazy attributes (where in a huge document of hundreds of pages you do a long jump - say from near the top to near the bottom, in one go, and we have not yet resolved the attribute status (say underline on) of the area you want to get to, we don't hold up the display for even one hundredth of a second while we are computing attributes, but display the new area immediately without any pause - the attributes will 'catch up' a second later: you will only see this if your document is very very big and you navigate in huge leaps). Also, there is a garbage job running all the time in the background, doing whatever internal tidying up and optimising is needed when you are not doing anything (with **PERFECTION**'s speed, even if you are typing at 200 wpm the program is sitting twiddling its thumbs for 90% of the time as it awaits input!). Consequently, **PERFECTION**'s internal tables are always in a PERFECT state. Both lazy screen and cursor acceleration are user-configurable, incidentally.

There are dozens of other more localised ways in which **PERFECTION** performance is obtained. For example, **PERFECTION** has built-in knowledge of statistical distribution of occurrence frequencies for the various alphabetic characters in English and other European languages. It uses this data as follows: if you ask **PERFECTION** to search for the word 'praxis' in your document, we won't look for the 'p' first. Instead, we automatically look for an 'x' (less occurrences of 'x') and having found 'x' then resolve whether it is embedded within an occurrence of 'praxis' (if not, we search for another 'x'). Obvious? We thought so. But no one else appears to be using this excellent trick. Or dozens of other tricks that we'd prefer to keep to ourselves...

For those with advanced needs, **PERFECTION** features include full programmability - more than just macros - with the ability to save and re-execute programs. There are over a hundred commands. You will be relieved to note that **PERFECTION**'s file format is very clean, containing one short header (giving the margin/TAB etc data for that document) and then exactly what you typed in (no mass of pointers or counters). Changes of attribute (bold, NLQ, underline etc) that you have opted for are stored as control characters (we document the structure) in the appropriate places in the file - note that while the control characters themselves (as opposed to their effects, which are WYSIWIG'd on-screen) are invisible when you are viewing the file, you can edit/delete them (search for the next or previous bold text, say!) and even program the access to them (swop all bold for double-strike plus underline)... You can even opt to Export so the header is suppressed, to enable its direct use as a programming or technical front end, or to allow its output to be read in by other word processors (QL, PC or whatever).

The net result of all this is that in terms of features and performance, **PERFECTION** running on a QL will beat most word processors even running on state-of-the-art £7000+ 486 PCs... In a nutshell, **PERFECTION** will blow your socks off.

LIGHTNING SPECIAL EDITION
LIGHTNING

Until the autumn of 1989 the fastest way of speeding up your QL display was to buy **Lightning**, which greatly accelerated QL text printing, graphics and maths, without affecting compatibility at all. NOW you can buy **Lightning** Special Edition, which is significantly faster than **Lightning** and does a lot more! **Lightning** Special Edition is simplicity itself to use. Once it is loaded ALL programs will AUTOMATICALLY benefit from the enhancements it provides. If you are using a QL without **Lightning** you are probably a little pale (quote from John Norton of Sector Software), you should get out and about more... Go to some QL shows or meetings where you will see **Lightning** in action - or take our word for it. If you don't have **Lightning** you are WRONG.

Lightning Special Edition works by automatically (I know we keep using the word, but it is the only one that is really correct here) and instantly replacing QL ROM code (or Minerva code, for that matter - Minerva and **Lightning** complement each other superbly) that has usually been optimised for space, with extremely high speed routines written by us that do the same job but much faster. Screen output speed gets accelerated by factors from over 1.5x to over 10x (about 2x-4x is representative), graphics are drawn twice as fast (points are plotted 5 times faster) and internal maths is speeded up by 2x-5x (you can even vary the precision). There is virtually no cost in RAM (for example, you can still run Quill with a fairly large document on an unexpanded QL with **Lightning** Special Edition).

The Special Edition is supplied on EPROM plus disk/cartridge: if you already have something precious plugged into the QL's EPROM socket (at the rear), there is no problem - all the EPROM's functionality is duplicated on the other medium!

Lightning Special Edition provides more than acceleration - you can dynamically adjust channel parameters - like ink, paper, font, screen position, use over 80 fonts, a null device, a character drain and all sorts of other interesting gadgets.

Lightning Special Edition installation has been totally automated, and will not present you with complications no matter how computer-naive you are.

If you cannot afford the Special Edition, get **Lightning**. Refer to its review in September 1988 QL World to see how effectively **Lightning** acquitted itself.

Both of these programs transform the QL into an altogether more zippy, business-like, efficient, enjoyable machine.

PC CONQUEROR WITH MS-DOS
PC CONQUEROR

Terrific though we know the QL to be, we do feel the pressure to be "PC compatible" in today's world. There is increasing demand to be able to bring home and run the programs we use at work (or the other way around!), and to have access to the vast storehouse of PC software: word processors, databases, spreadsheets, expert systems, accounts and financial modelling packages, vertical market applications, visualisation aids, graphics/CAD/PCB designers, languages/compilers, operating systems, environments, utilities, adventures - you name it, there are scores of each type readily available for the PC. And thousands of shareware/PD programs too, most for the cost of a blank disk plus postage.

If you buy PC **Conqueror**, you will be able to run these programs! To boot up PC **Conqueror** takes 10 seconds from the F1/F2 prompt: thereafter, your QL is a HIGHLY compatible PC clone (indeed, more compatible than some "real" PCs). **Conqueror** is all-software. There is no comparison in quality between **Conqueror** and its predecessor: **Conqueror** has ALL the features of Solution (read the details later in this ad if you are unfamiliar with Solution's legion facilities), but is almost TWICE as fast: this has come about by our careful rewriting and optimising of Solution's code. As if the colossal speedup was not "enough", **Conqueror** (unlike Solution) runs perfectly even with PC software that makes various "non-legal" calls to the PC operating system. **Conqueror** runs with virtually anything that will run on a PC: QL Worlds from December 1989 to March 1990 listed several hundred PC programs/utilities found to work with **Conqueror**. It is simpler to say that we have yet to find a program that runs fine on a standard PC that doesn't run with **Conqueror**: we are aware, however, of programs that will run with **Conqueror** but won't run on some PCs!

Because in **Conqueror** we've cracked the problem of detecting when the PC screen has been changed, we need not slavishly update the screen many times a second (taking precious time away from the main PC-emulation job) as did Solution. Instead, we update the screen instantly it needs to be updated. This simple to understand but very hard to implement modification gives **Conqueror** additional (over and above what we've already discussed), "tunable" acceleration, as well as absolutely smooth echoing of keyboard input to screen (Solution could be a bit jerky when you typed quickly).

Conqueror's new features include a more flexible configurator and a better diagnostic and supervisor option, an enlarged manual (**Conqueror** itself is more compact!) with a troubleshooting chart, and a new mode of operation (in addition to the "normal" one of reading/writing PC disks directly) which allows you to create mini PC environments - you select the size, location and name - on any QL device (including floppy, ramdisk, hard disk and even mdv) which look like files from QDOS (and can therefore be copied with SuperBASIC's COPY!) but are indistinguishable from PC drives from within MS-DOS...

If you do not have legal access to a copy of MS-DOS , you need to buy MS-DOS too - but we sell it (with GW-BASIC, Shell and all the system utilities thrown in) at about half the normal price.

Of course QLs are better than PCs - but QLs that are PCs as well are better still. We will leave the last word to people who have already bought **Conqueror**. All these sentiments are unsolicited.

"I wish to congratulate you on the excellent work you have done on ***Conqueror****. The improvements in performance over Solution are astounding. Well done!" B.C.Papegaiij, Netherlands. "I am highly delighted with this new emulator. (Apart from the speed-up) it also appears to be more tolerant." L.Chandler, Peterborough. "Congratulations on bringing such a fast PC emulator into the world - on it, even Wordperfect runs at a reasonable speed." R.Williams, London. "I'm impressed with the improvement in speed over Solution." P.Vervoort, Netherlands. "Thank you for your prompt service. I have* ***Conqueror*** *up and running, and congratulate you on an excellent piece of software." G.Leagas, Hartlepool. "On some benchmarks almost as fast as a PC." P.Johnson, Stoke on Trent. "****Conqueror*** *is still a whole lot faster (even) without Lightning than Solution is with the assistance of Lightning." P.Christie, Glasgow, who went on to praise* ***Conqueror*** *for running software Solution couldn't handle. "****Conqueror****, to which I upgraded from Solution, is a delight to use by comparison!" B.Gouldwell, Dunipace..... V.Pakanen, Finland sums it all up rather well with - simply - "Excellent."*

PROFESSIONAL PUBLISHER

To show you a little of what our **Professional Publisher** can do, we have prepared our last advertisement using it. Notice from our May 1990 advertisement how we can wrap the result around graphics or in fact anything, of any shape. When we wrote **Professional Publisher (PP)**, we knew it was a very special sort of program. **PP** can produce pages of quality - virtually indistinguishable from those prepared on professional typesetting kit, the only limiting factor might be your printer: however, while the very best output output from **PP** will be obtained from 24 pin models and lasers, you will be stunned by what **PP** can squeeze out of the humblest 9-pin machine. Great care was taken in the design of **PP** so we were absolutely sure that no actual knowledge of, or practice with, desktop publishers was required in order to use it (the '**Professional' in 'Professional Publisher**' refers to the output quality, not the level of operating skill required. When you use **PP**, you will notice that at each and every stage a menu is available (there are getting on for a hundred menus in total) with a list of options selected by using either the cursor keys and SPACE bar, or by pressing a digit key - use what suits you! There is context sensitive, on-screen help too. When you get more experienced with the program, you may select Command mode (using the Enter key) and choose operations directly, bypassing the menu system. **PP** is more user-friendly than any page-making program we have ever seen on any computer, period.

Let us talk you through how you might choose to produce a page or succession of pages. This is just one way you might proceed: **PP** does not impose any sequence of steps upon you, and you can omit certain operations altogether. You will have pre-configured **PP** to boot up with a generous lot of fonts (you select which ones you are likely to want - of course you can load in additional ones, or discard existing ones, at run-time too). You could then set the required page dimensions and orientation, as well as not-necessarily-symmetric margin, grid, gutter, column and navigation-guide positions (yes, half the **PP** manual is a glossary) - you could have pre-configured **PP** for these too, or loaded in alternative layouts (layouts are distinct from page contents) you've created in previous sessions. If you don't set layout we'll use the default, or the one used for the previous page. Now you would plan the page in detail. Laying out graphics (if any) comes next - you can create these in **PP** itself, with its superb rubber-banding, dozens of brushes, palettes, texture-fills and so on. Alternatively, you can load in screens created elsewhere, including Eye-Q, Easel, any other graphics programs or digitiser, into a cut and paste buffer where a dozen tricks (including resizing, slanting, scrolling and texturing) are available, and then take the finished product onto the page. This done, you might insert headlines or captions, selecting from the dozens of fonts available. Each font can be manipulated in billions of ways (yes, we mean thousands of millions): to give but two examples, you have a choice of 32 slopes for italics for the font, and dozens of aspect ratios are selectable. Now you might opt to get the main body or bodies of text down on the page. As fonts are defined to great accuracy (upto 2304 pixels PER CHARACTER!) jaggedness is a thing of the past, and visually the choice of fonts can only be described as stunning! You can do this either by directly typing it into cursor-dragged boxes (with all the options you would expect from a dtp system, and a few more besides), or by loading it in from file created by Quill, **PERFECTION**, Editor or other word-processor. The latter method is better (because you retain the text as a character stream rather than as pixels when you save the file). Highlights such as bold, underline etc which you may have inserted into the text are preserved. Indeed, you can control **PP**'s operation from within the text file itself. If you are an advanced user, you can even teach **PP** your own mnemonics, so that it switches between different styles and modes as it encounters instructions you put into your text file when you created it! The imported text file is editable within **PP**. It is up to you to decide where the text is to lie - **PP** places no restrictions on either the number or the shape of the windows into which the text is to flow: they need not be rectangular, and can have any irregular border, and can even overlap or be contained one inside another! You can freehand-draw (there's excellent rubber-banding to help you) the window borders as you choose, to get any effect you desire, to fill any space you wish and to avoid any existing material already on the page (or to reserve room for new material). Amazingly, within the window the text will all be perfectly micro-justified in the font(s) of your choice, however bent or contorted you made the border. Text will flow automatically from one window to the next either until you have run out of text or out of windows. There are many text formatting facilities: you can select word-wrapped, force-broken or hyphenated, and you can specify minimum numbers of "pre-hyphen" and "post-hyphen" characters so that absurd hyphenations are avoided (if no sensible hyphenation position can be found the word is wrapped instead). There are so many fine-tuning controls here that the rest of this ad could be devoted to describing them and would still leave things out! We will have to content ourselves with but one example: with micro-justification (pixel by pixel spacing, not crude character by character stuff) we even allow you to specify what % of padding space is to be allocated between characters and how much between words! Text work completed, you can then put in the final touches by adding borders, shadows, patterns or designs, overwriting or slipping under or combining these with existing material, repositioning parts of the page if necessary. The end result - be it for a letter, letterhead, document, manual, article, newsletter, magazine, book, thesis, ad - is far better than you have any right to expect from a piece of software costing under £2,500, let alone under £100...

PROFESSIONAL PUBLISHER TOOLBOX

NEW!

For Professional Publisher users - this useful addition not only supplies several man years worth of beautiful high definition fonts - including familiar types like Roman and Universal - but also contains many smaller fonts, more clipart and programs to load Sector Software clipart, filter text before importing into Professional Publisher, save parts of Professional Publisher pages as screens (for importing into any graphic program - like Eye-Q - or manipulating via SuperBASIC) etc. Excellent value.

FONT ENLARGER

For Professional Publisher users - loads of large fonts are automatically created by this multitasking utility, as and when you need them (or in advance), by enlarging existing smaller smaller fonts from PP itself and from Lightning Special Edition and hordes of other sources: with this there is NO jaggedness at all. A font editor for small and large (hdf) fonts is included.

GRAFIX

Scaleable output for all our desktop publishers on 9- and 24- pin printers: a useful alternative to the built-in drivers.

EYE-Q

There is no way to describe **Eye-Q** except as the best graphics program for the QL. This master is now four years old, and we have never felt the need to change anything. Its use is characterised by absolute simplicity, speed and power - it has that indefinable precision "feel" that is just right. All the expected manipulations are provided. Whether your needs are technical drawing, labelling, design, illustration, freehand work, copying - or just having fun, **Eye-Q** will not disappoint. Of course it is menu driven with context-sensitive help. The system takes 5 minutes to learn. The variable zoom and fill facilities, anti-fingerslip measures, cursor acceleration and so on make **Eye-Q** a classic in its own time.

ULTRAPRINT

To get the best printer output from Eye-Q or any other graphics program from any other source, **Ultraprint** delivers. An amazing 22 styles to choose from: enhance contrast (for line output) or gradation (for pictures) and vary magnification... A printer without **Ultraprint** is no printer at all.

MEDIA MANAGER SPECIAL EDITION
MEDIA MANAGER

MMSE is a joy to use. Whether something has gone wrong with a disk or tape ("Not found", "Not a valid Quill file", "Bad or changed medium", "Read/write failed" etc) or whether you want better control over your programs and data, **MMSE** should be to hand. Virtually any calamity can be recovered from automatically: all permutations (accidental deletion or part-overwriting, part-formatting, errors yielding: bad map but OK directory, bad

directory but OK map, bad map and directory, OK map and directory but bad file sectors, unknown fault, power glitch corruption and so on) have been carefully thought through and catered for. If nothing is wrong, but you just want to explore and understand more about your system, you can potter to your heart's content, assisted by the clear and packed-with-facts manual. Dozens of different diagnostic printouts can be produced. The whole system is menu-driven, with context-sensitive, on-screen help for every option. The speedy Sector Editor is a positive delight: the collector file facilities, bulk recovery, auto-navigation, skipping through the medium in physical, file (if map), logical (if no map) or uncollected/logical (if destroyed map, and because of "chequered" history with lots of overwriting/deletions no one-step recovery available) sequences must all be experienced to be believed. **MMSE** is extremely simple to operate, and assumes no advance knowledge whatsoever.
Alternatively, if you wish to tidy up your disks or cartridges, **MMSE** allows you to change volume format names, sort directories into alphabetic, date or size order, analyse file contents and histories, change case of filenames, move data/programs to/from alien-format disks, introduce or break copy-protection systems (illegal use prohibited!), **MMSE** can and will deliver the goods. It is absolutely superb.
The standard **Media Manager** is much less powerful, and less easy to use. It is only for those on a tight budget.

TOOLKIT III WITH ROM
TOOLKIT III

Virtually everyone with a disk system has Tony Tebby's fine TK2 Supertoolkit on board (usually built into the disk interface). **Toolkit III** - which works whether or not you have TK2 - takes off where TK2 ended, adding about 70 new commands and enhancing many existing QL and TK2 commands. **TK3** is for everyone with a QL. You can get this system on cartridge/disk, with or without a plug in ROM cartridge in addition. The documentation is complete and very comprehensive. Some of the added commands are:
ADIM * ADIMN * AND_L * ATYP * BASREF * BV_BASE * CHANNELS * CH_BASE * CINT * CLOSE% * DEVLINK * DIR_USE * DIT$ * DIV_L * EOR_L * EXTRAS * FACC * FLP_SEC * FLP_START * FLP_TRACK * FLP_USE * FRAC * ISFLT * ISINT * KEY$ * LARRAY * LOWER$ * MEMCOPY * MEMSWAP * MJOB * MJOB_W * MOD_L * NFS_USE * ODD * OM_INIT * ONPIPE * OR_L * PEEK_F * PEEK$ * PEND * PIPE_ * POKE * POKE_F * PRED * QDOS$ * QIN * QOUT * QTEST * QWAIT * RAM_USE * REPLACE * REPLACE$ * RESET * RJOB_A * ROUND * SARRAY * SEARCH * SETDIR * SETDIR_A * SETHOST * SETNET * SETRO * SETRW * SETSYS * SETUSER * SGN * SORT * SORT_I * SUCC * TK3_EXT * UPPER$ * USER * WN_BASE * WSETHOST * WSETNET * WSETRO * WSETRW * WSETSYS * WSETUSER

QFLICK CARD INDEX SYSTEM

Few users actually require all the facilities of a complicated database like Archive. **QFlick** presents a very convenient alternative - a very fast, simple to use card-file database, with easy to learn, snappy search and navigate commands and clean file-handling. You can move Archive data to/from **QFlick**. You can run multiple copies of **QFlick** too.

PERFECT POINTER TOOLS

This excellent program gives you an on-screen pointer (arrow) environment and all the tools you are likely to need to run it.

QKICK MULTITASKING SYSTEM

A pull-down menu controlled multi-tasking program, ideal for running in the background and giving you notepads, file-handlers, quick backup, clock, diary, mini-database, calculator etc.

DISKTOOL WITH QUICKDISK

An exciting way to accelerate disk access by upto 30%, add password protection to disks and to optionally increase disk storage capacity by 36K to 1512 sectors! All this works while still giving you full normal control of the disk.

DIGITAL C SPECIAL EDITION
DIGITAL C COMPILER

Superb C compilers these - fast in execution, they produce extremely speedy and concise code. No-nonsense documentation is included. The Special Edition has many more features, including pointers, long pointers, structures, >64K code sizes, direct access to traps and vectored utilities, and is twice as fast because of its more efficient C/QDOS libraries.

TURBO BASIC COMPILER + TOOLKIT

This state of the art system will automatically convert ordinary SuperBASIC programs - the sort you buy, write yourself or type-in from magazines - into machine code, the language of the 68008 CPU, the brain of the QL. Such pure machine code programs run "directly", without the need to be interpreted by any intermediary system. This direct execution makes them MUCH faster in execution than BASIC. **Turbo** also adds a host of useful high-speed commands (called "toolkit extensions" if you are fond of jargon). Here are some timings, all carried out on a JS Trumpcard QL, to give you a taste of just how much **Turbo** can improve things:

	Iterations	SuperBASIC	**Turbo**'d	Speedup
Empty FOR...END FOR Loop	30000	49 sec	1.3 sec	38x
Empty REPEAT Integer Loop	30000	151 sec	2.4 sec	63x
String concatenation	3000	448 sec	0.4 sec	110x
Search through memory	300000	1410 sec	1.5 sec	900x

Turbo's automatic conversion process, called compilation, is as simple as this: (1) Boot up with the **Turbo** disk (2) Load in or type in your BASIC program (3) Enter the word CHARGE, and watch the friendly front-end menu pop into view (4) Choose a filename for the machine code task that is to be generated and (5) Press the SPACE bar. **Turbo** does the rest! Compilation is a one-off process, and is very fast too - it takes little more time than LOADing the original program did! Once compilation is finished, you have a machine code version of the original program. Start this with EXEC, just as you used to invoke the original program with LRUN: besides the tremendous difference in running speed, you will notice that the program loading time is cut down to a few seconds at most (big SuperBASIC programs can take half an hour or more to load). The EXEC mechanism also allows you to multitask programs, something impossible with SuperBASIC, as well as manipulate their time-priorities, link them together, exchange data and even share parts of their code while executing.
If you are an advanced user, **Turbo**'s numerous fine-tuning facilities, 200-command toolkit (a terrific complement to the famous Supertoolkit) and 300+ page manual will be irresistable. If you are a beginner, you will wonder how you ever did without **Turbo**'s program diagnoses and auto-correction.
Turbo is more than a very clever optimising compiler. **Turbo** is magic. If you do not have it, you can have no conception of the experience you are missing and the power you are forfeiting.

SOLUTION WITH MS-DOS
SOLUTION

This program transforms your QL into a pretty compatible - albeit not fast - PC clone. **Solution** will run over 95% of the "big name" PC software you have read about, missing out only on programs that make illegal use of the PC's operating system. **Solution** works solely from software so you don't have to worry about ripping your QL to pieces to fit anything, or have anything hanging out of the back. Just boot up the **Solution** disk and you will be using a PC, which will then ask for a copy of MS-DOS (just as it would if you were using a "real" PC). End of story - you are now using a PC. There are very few restrictions: both mono and colour CGA graphics are supported. 479K is available for PC software on a 640K machine and 667K when using Trumpcard - more than you will get on a PC or XT! Speed can be increased by using Lightning Special Edition but in final analysis just can't compare with Conqueror's speed). Because your newly aquired PC is really a QL you can multitask two or three PC programs (try doing that on a "real" PC!). You can also run QL programs alongside PC programs (DON'T try that on a "real" PC!). Converting files (in either direction) between QL and MS-DOS is no problem and you can re-configure the QL keyboard if you wish.

PROFESSIONAL ASTROLOGER
PROFESSIONAL ASTRONOMER

Our use of the term "Professional" in the name of an application program does mean that the quality achieved will meet or surpass the highest professional standards for that application. The term does NOT mean that you have to have the knowledge of a professional in order to get the best out of the programs.
Astrologer teaches you astrology from scratch, and enables you to produce reams (if you are short of paper, you can choose exactly how much) of narrative printout giving a person's horoscope, personality delineation, year-to-year life overview, detailed day-to-day (in fact, minute-to-minute!) predictions, as well as two-person compatibility interpretations. Also provides all the technical readouts, charts and zodiacal wheels you would expect. It is extraordinarily fast (there is a great deal of very clever maths within it) and it performs the whole computation in under a second. The author of the manual is the author of this advert, so you can expect a lucid and humorous read! Whether or not you believe in astrology - indeed, especially if you do not - this program is one that you cannot afford not to have. Scores of detailed readouts for famous people are supplied, incidentally - very interesting reading they make too... Discover Mrs Thatcher's secret yearnings, explore yourself, play the Stock Exchange...
Astronomer is an extremely efficient solar system computer, with planetarium views, planet faces (with shadows/eclipses), five different co-ordinate systems, 1sec=1day cinerama, etc.
Astrologer + Astronomer is supplied at a very low combined price.

ACT SPECIAL EDITION

The **Adventure Creation Tool** is for every programmer or putative programmer. Whether or not you have any interest in adventures, you will find something useful here. Animated graphics, data compression, language design and parsing, maps, object-oriented control and much more, with an excellent educational manual too.

3-D PRECISION CAD SYSTEM

2-D and 3-D design and manipulation, at a speed sufficient to permit real-time animation! Whether or not your interest is serious, **3DP** will change the way you look at the world around us. The variation of viewpoint, perspective and magnification is very smoothIn addition to dot-matrix output, plotters are catered for.

SUCCESS

Run CP/M programs on your QL! What more is there to say, other than that after the PC family, no more common system exists than CP/M, with thousand of cheap programs... And **Success** is fast!

THE EDITOR SPECIAL EDITION
THE EDITOR

If your needs are for a technical Editor, or for full access to the entire ASCII character set (to handle machine code or compressed data files), or if your budget cannot stretch to **PERFECTION**, then this is the program for you. Editor is command-line driven and programmable. The Special Edition version is certainly better than the standard version: that is because the standard one contains only as many features as we could get to fit into an unexpanded QL. Both are fast and flexible, and very powerful indeed in the hands of the intelligent. Not a word processor, Editor's a way of life.

SPECIAL DESKTOP PUBLISHER
DESKTOP PUBLISHER

Both these WYSIWYG ("What You See Is What You Get" dtp systems are excellent in their own rights - it is only when you compare them with the stunning Professional Publisher that you become aware of their shortcomings. You won't get fonts as large or smooth as with PP, or wrap-around graphics, or as sophisticated a printer driver or text/graphics file import facility. You will get a very workmanlike tool, capable of producing output that the computer press described as fantastic and superb.... The standard edition is the ideal if you do not have a disk drive: if you do have one, go for the Special version, which correspondingly has more features including textures, large windows, better drawing and improved command entry. All upgrades are possible, and there is only a £10 penalty for doing it in two stages. So if you simply cannot afford PP, one of this pair is certainly for you.

SUPERFORTH COMPILER WITH REVERSI

Why not learn FORTH, the most logical computer language of all? This superb FORTH-83 compiler produces stand-alone multi-tasking code of speed comparable to C. **SUPERFORTH** source is even portable to other machines! The manual teaches you the language.

IDIS SPECIAL EDITION
IDIS

Machine code (from other people's programs, toolkits and the ROM) is unintelligible until you put it through **IDIS**, the intelligent disassembler. **IDIS** Special Edition automates everything it possibly can, and requires no human intervention. It even sorts out subroutines, replaces addresses with names, untangles data from code and so on. Standard **IDIS** contains as much as we could pack into an unexpanded machine, and is nearly as automatic.
If you want to find out how computers work, buy one of these two!

MONITOR

Check dynamic operation of programs - IDIS's ideal companion.

MICROBRIDGE

Never be short of a 4 for Bridge again. Superb bidding tutor included, based on random hands dealt with lightning speed. Manual a masterpiece.

SUPERCHARGE SPECIAL EDITION

If you have an unexpanded QL, or cannot afford Turbo, but want SuperBASIC programs to go faster, **Supercharge** is the answer. It has about half the speed of its big brother, is not as tolerant of badly-written programs, and lacks many of Turbo's features (like linking, program sizes >64K etc): nonetheless, it is the compiler about which we recieved over ONE HUNDRED happy letters from satisified users all using the word "Excellent" to describe it - and hundreds more who used other equally complimentary terms. The only gripe was about the Lenslok copy-protection, long since removed by us. So now **Supercharge** is wonderful....

SUPER SPRITE GENERATOR

SSG moves things about the screen rapidly, at machine code speed, directly from simple SuperBASIC. Any number of sprites (each with upto 16 frames for smooth realistic motion), 256 speeds, 256 planes, collision detection and dozens of special effects.

SUPER ASTROLOGER

A cut-down version of the Professional Astrologer - smaller horoscopes and manual, no interpretations for forecasting or compatibility testing. Still a marvellous buy at the price!

BETTER BASIC EXPERT SYSTEM

SuperBASIC is a super BASIC. If you want to improve your programs automatically, and learn as you do this, get **Better Basic**.

TRANSFER UTILITY

Copies files between devices, performing translates as it goes. Needs a ramdisk to run. Can move your microdrive material onto disk, so programs run from disk but you still have access to microdrives.

TERMS AND CONDITIONS)

* Our non-game programs are very comprehensively documented with A4 manuals averaging about seventy pages in length (the largest is 325+). They are 4-hole punched for easy binding and storage.
* UK purchasers - the quoted figures are all-inclusive. For the rest of Europe, add 5% (rest of the world, 10%) to the quoted figures to arrive at the VAT-free total (exports are zero-rated for UK VAT), inclusive of all freight and documentation charges.
* Acceptable forms of payment are sterling cheque drawn on a UK branch of a bank or building society, sterling postal order, Eurocheque made out in sterling, international money order in sterling, VISA / ACCESS / EUROCARD / MASTERCARD (specify expiry date), cash, foreign currency cheque (add 10% conversion charge), direct money transfer (write to us notifying us of the transfer, and ensure that all charges are paid your end, or add 5%) to A/C 50327808 DIGITAL PRECISION LTD at Barclays Bank PLC (Branch code 20-79-44), South Chingford Branch, 260-262 Chingford Mount Rd, London E4 8JN. We are very flexible, as you can see!
* To upgrade from one version of a program to a superior program, send us the cartridge/disk. Except in the case of upgrades between program versions both with the same name, send the manual too. The cost of an upgrade is £10 plus the difference in current advertised price between the two programs. So the upgrade from DIGITAL C to SPECIAL EDITION DIGITAL C costs £30, and you would need to return the old manual as the program names are different.
* Our programs are all user-transferable between cartridge and disk, are all free from ALL copy protection, and all work with all drives, toolkits, RAM add-ons and disk interfaces (except for early MCS interfaces, to which the emulators and media managers object). Users of the Microperipherals interface are recommended to buy the QFLP ROM upgrade from Care Electronics or Qjump. ST/QL Emulator owners will benefit from a c2.7x speed increase on all our products.
* If you want Eye-Q, but want to use it with a Gigamouse, with QRAM or on a THOR, specify your intended use with your order.
* Microcartridge users please note: don't panic! we have large stocks of microcartridges and we are NOT going to run out. Quality software on cartridge will continue to be available from us for the foreseeable future. You can buy cartridges from us at the rate of £19.95 for a set of five cartridges.
* Digital Precision (abbreviated by our admirers to DP!) is a trading name of DIGITAL PRECISION LIMITED, Company Registration No. 1833989, registered in England & Wales, VAT Reg. 420 1560 08.

All trademarks are acknowledged.

DIGITAL PRECISION LTD

PRICE LIST

	£	Key
PC CONQUEROR WITH MS-DOS	139.95	eT
PERFECTION PLUS WITH SPELL CHECKER	119.95	dT
PC CONQUEROR	89.95	eT
PROFESSIONAL PUBLISHER	89.95	cT
THE SOLUTION WITH MS-DOS	89.95	eT
PERFECTION	79.95	dT
TURBO BASIC COMPILER + TOOLKIT	79.95	aT
PROFESSIONAL ASTROLOGER WITH ASTRONOMER	69.95	aT
PROFESSIONAL ASTROLOGER	59.95	aT
LIGHTNING SPECIAL EDITION	49.95	+aT
DIGITAL C SPECIAL EDITION	49.95	aT
MEDIA MANAGER SPECIAL EDITION	49.95	dT
3-D PRECISION CAD SYSTEM	49.95	dT
SUCCESS	49.95	bT
THE EDITOR SPECIAL EDITION	49.95	dT
ACT SPECIAL EDITION	49.95	eT
TOOLKIT III WITH ROM	44.95	+a
IDIS SPECIAL EDITION	39.95	dT
EYE-Q	39.95	aT
THE SOLUTION	39.95	eT
SPECIAL DESKTOP PUBLISHER	39.95	ct
SUPERFORTH COMPILER WITH REVERSI	39.95	aT
MICROBRIDGE	34.95	d
PROFESSIONAL PUBLISHER TOOLBOX	29.95	eT
TOOLKIT III	29.95	a
QFLICK CARD INDEX SYSTEM	29.95	a
PERFECT POINTER TOOLS	29.95	a
SUPER SPRITE GENERATOR	29.95	a
SUPERCHARGE SPECIAL EDITION	29.95	a
PROFESSIONAL ASTRONOMER	29.95	fT
THE EDITOR	29.95	at
DIGITAL C COMPILER	29.95	at
GAMES COMPENDIUM (ALL FIVE GAMES)	29.95	a
QKICK MULTITASKING SYSTEM	24.95	a
IDIS INTELLIGENT DISASSEMBLER	24.95	at
LIGHTNING	24.95	aT
SUPER ASTROLOGER	24.95	f
DESKTOP PUBLISHER	24.95	d
MEDIA MANAGER	24.95	a
BETTER BASIC EXPERT SYSTEM	24.95	aT
FONT ENLARGER	19.95	eT
DISKTOOL WITH QUICKDISK	19.95	b
ULTRAPRINT	19.95	aT
MONITOR	19.95	aT
TRANSFER UTILITY	9.95	b
GRAFIX	9.95	eT
SUPER BACKGAMMON GAME	9.95	f
REVERSI GAME	9.95	f
ARCADIA GAME	9.95	f
BLOCKLANDS GAME	9.95	f
DROIDZONE GAME	9.95	f

KEY>As well as a cartridge or disk, you get a ROM	+
Available either on cartridge or disk	a
Available only on disk	b
Minimum 512K RAM:only available on disk	c
Minimum 256K RAM:either cartridge or disk	d
Minimum 256K RAM:only available on disk	e
Available only on cartridge	f
Compatible with THORs including the THOR XVI	T
Compatible with all THORs except the THOR XVI	t

SPECIAL LIMITED PERIOD OFFER - ACT NOW>>>>
* Buy ANY *TWO* programs, get **10%** off the total price!
* Buy ANY *THREE* programs, get **15%** off the total price!
* Buy ANY *FOUR* programs, get **20%** off the total price!
* Or buy *FIVE* or more to get **25%** off the total price!
****
UPGRADES COUNT AS PROGRAMS FOR COMPUTING SPECIAL DEALS
****

As PERFECTION is not a superior Editor version but instead is all new, it is not available at an upgrade price to Editor owners. However, Editor owners can claim a £5 reduction when ordering it.

DIGITAL PRECISION LTD

To: DIGITAL PRECISION LTD
222 THE AVENUE, LONDON E4 9SE

ORDER COUPON

Or telephone
081-527 5493

Please send me: ____________________
Name: ____________________ Address: ____________________
Postcode: ____________________
Enclosed: CHEQUE/PO/CASH/VISA/MASTERCARD/ACCESS/EUROCARD for £ __________ Signature: ____________________
Card No: ____________________ Expiry: __________
I use: (QL/THOR) (MDV/3.5" DISK/5.25" DISK) (360/720/1440 SECTORS)

DIGITAL PRECISION LTD

PSION ORGANISER

LZ64	£180	Using Psion Organiser	£10
Psion LX Book	£15	Spelling Checker	£30
Comms Link	£60	Organiser XP2 32K	£110
32K D Rampack	£55	Numbase 1	£40
Centronics Interface	£40	Psion Printer	£230
Organiser LZ	£150	File Handling Book	£15
128k Datapak	£100	32K Datapak	£35

Z88

ZBase	£69.00	Easy Money	£69.00
Spellmaster	£59.00	Z88 Computer	£230.00
Thinkz	£50.00	z-Term	£50.00
Cambridge Topper	£8.00	Superchip 4	£150.00
Centronics interface	£30.00	128K Eprom	£50.00
128K RAM Pack	£50.00	Z-Tape	£50.00

TK COMPUTERWARE

the **QL** stockist

Tel: 0303-81-2801

HARDWARE

768K Trump Card	£225.00
256K Trump Card	£135.00
768K Trump Card Dual 3.5" Disk Drives	£375.00
256K Trump Card and Dual 3.5" Disk Drive	£285.00
256K Trump Card and Single 3.5" Disc Drive	£210.00
Disk Card	£100.00

SPARES

JS ROM SETS	£30.00
Keyboard Membrane	£7.25
ULA ZX8302	£10.25
ZX 8301	£12.00
MAB 8049	£7.50
MP 68008	£20.00
Z9007E	£10.25
Repairs	phone
M/drive assemblies-complete	£27.50
Voltage regulator (high output)	£4.75
Power supply UK	£28.50
QL Test kit	£13.50
M/drive assemblies (xchange)	£20.00

BOOKS

Z88 Dabhand Guide	£16.00
Assembly Language Prog	£9
Z88 Ref Manual	£16
Z88 Magic	£16
Z88 Pipe dream	£16
Using Your Z88	£10
Z88 Computing	£10
Z88 Software DLV Guide	£40

SOFTWARE

BUSINESS

Perfection	£80.00
Perfection plus	£120.00
Cash Trader (PDQL)	£40.00
Decision Maker	£35.00
Stock Accounting System	£40.00
Sales Invoicer	£20.00
Home Finance (Buzz)	£22.00
Investment Monitor	£20.00
Project Planner	£29.00
Small Traders Pack	£25.00
Touch Typist	£12.00
QYTP	£23.00
QYTP 2	£30.00
QPAC II	£50.00

UTILITIES

Chocolate Emulator	£90.00
Archivist MP	£40.00
Archivist	£30.00
Cartridge Doctor	£18.00
Expert System	£50.00
Page Designer 2	£35.00
Professional Publisher	£90.00
Master Spy	£30.00
ArchRTM+ArchDEV	£20.00
ICE Eprom	£27.00
Locksmith	£15.00
Mailfile (256K)	£20.00
Archive Tutor	£20.00
QFLASH Ramdisc + t'kit	£19.00
QFLASH Ram Eprom	£25.00
QFLASH Ram utilites (c-d)	£10.00
Reconfigurable toolkit 2	£29.00
4 Matter	£15.00
Recover	£20.00
Turbo Quill Plus	£13.50
Perfection	£30.00
Media Manager Special Edn	£50.00
Super Media Manager	£30.00
Sidewinder Deluxe	£20.00
Conqueror	£90.00
Flashback Special Edition	£40.00
Media Manager (d)	£20.00
XRef 2.00	£20.00
Super Toolkit 2 (Eprom)	£29.00
Disc Over	£30.00
Success	£50.00
RPM	£15.00
Task Master (c-d)	£25.00
The Editor	£20.00
The Editor special edition	£49.00
Turbo (c-d)	£80.00
Vanilla Emulator	£40.00
Ultra Print	£20.00
SPY	£15.00
Lightning	£30.00
Quailsoft Terminal	£30.00
Lightning S.Edition	£50.00

CAD

Graphic QL Plus	£25.00
Eye Q	£40.00
3D Precision	£49.95
Technikit	£25.00
TechniQL	£50.00
3-D Designer	£38.00
PCB 1	£99.00
PCB 2	£50.00

LANGUAGES

Assembler (Comp One)	£20.00
Forth (Computer One)	£30.00
Better Basic	£24.00
BCPL Development Kit	£51.00
Digital C	£30.00
Digital C Special Edition	£50.00
MonQL	£30.00
Pro-Fortran-77	£99.00
Pro Pascal	£99.00
QMON	£20.00
Super Forth + Reversl	£39.00
IDIS	£25.00
IDIS Special Edition	£35.00

GAMES

Chess	£19.00
Super Croupier	£13.00
Flight Simulator	£19.00
Matchpoint	£15.00
Bridge Player 2	£15.00
Professional Astrologer	£70.00
Super Astrologer	£25.00
Crown of Sorkingnm	£17.50
Snooker	£13.00
Super Backgammon 3	£13.00
War in the East (set)	£39.95
Microbridge	£35.00
Heart of Gem	£15.50
Area Radar Controller	£13.00
Talisman	£19.00
The Prawn	£15.00
Blocklands	£10.00
Archanoid	£15.00
Death Strike	£15.00
Fictionary	£11.50
Firebirds	£13.00
Karate	£15.00
Mortville Manor	£15.00
Vroom	£13.00
Stone Raider	£13.00
Lands of Havoc	£19.00
Shang	£15.00
Aquanaught	£19.00
King	£13.00
Stone Raider	£15.00
Brain Smasher	£18.00
Patience	£18.00

• Please telephone for details of other products not listed here •

SUNDRIES

VDU cleaning kit	£7.50	Joystick + adaptor	£16.00
10 Microdrive cartridges	£23.50	Psion 2.35 Sets	£23.00
100 Microdrive labels	£3.75	QL-Z88 Link	£25.00
Mains filter	£24.00	10 3.5" Discs (unbranded)	£12.00
10 DS/DD discs 3.5" (brand)	£29.00	QL Dustcover	£6.00

PERIPHERALS

QL Scanner	£140.00
Centronics Interface	£29.00
TAN Data QCom set complete	£75.00

ALL PRICES INCLUDE VAT. U.K. POSTAGE £2.50 PER ORDER. - WE EXPORT WORLDWIDE
Phone for details of other QL products OR send SAE for our latest catalogue.
Credit card phone orders accepted, or send cheques, postal orders, Eurocheques, to:

**TK COMPUTERWARE, STONE STREET,
NORTH STANFORD, ASHFORD, KENT TN25 6DF**

TELEX
966676 PMFAB G

VISA · Access · MasterCard · AMERICAN EXPRESS · EUROCARD

FAX
0303 812892

QL SCENE

New from Van der Auweras

Progs Van der Auwera have released a new program, *Qractal,* which (oddly enough) generates Mandelbrot or Julia set pictures on the QL. Qractal is described as featuring fast, precise 2^{28} fixed point calculations, toggling between a Mandelbrot set picture and its Julia equivalent; saving, redrawing, recolouring of maps 'to discover the hidden structure of the set', 100% machine code and multitasking program using pointer environment and menu extensions; save option for created screens; inner set filling with attraction count of value; option to change co-ordinates of Mandelbrot centre; supports BDM or bnary decomposing, mode4 and mode8, variable maximum number of iterations; easy zoom; special clear colouring method; option to make parameter files. The program comes on three disks with sample charts, and costs £40 sterling (including conversion costs) or Belgian francs 2000 (post free in EC, post BEF 300 outside EC area).

The Van der Auweras have also updated *The Painter* to V4.00. The cost of the update is £14 (BEF 800) without binder, £17 (BEF 1000) with binder. Add £10 postage/bank conversion charge if sending a Sterling cheque (BEF cheques do not incur the extra charge).

Progs Van der Auwera are at Haachtstraat 92, 3020 Veltem, Belgium. Tel (local) (016) 488952.

Changes in Quanta Editor resigns

The November issue of the user group Quanta magazine has some updates on sub-group information, and details and maps relating to the November workshop in Nottingham. Apart from that, the A5 magazine is mainly full of tips and responses from members. It's entertaining to see the familiar *DIY Toolkit* headline suddenly pop up at the back in a full-page list of Simon Goodwin's Toolkit disk collection, available from CGH Services.

The November editorial comments that The Consumer Protection Office in Birmingham now has a number of complaints against supplier PDQL, along with an odd comment about the office's "new address". The address – the same one that appeared in QL World October 1990 – is given as **155-157 Corporation Street, Birmingham B4 6TH.**

We note that the Essex sub-group has been 're-chaired' by Ron Dunnett. Ron is also trying to set up an 'out of Quanta' public domain library and would appreciate support.

As we write, we hear that Sarah Johnson has resigned as Quanta editor.

German Group

The Sinclair QL User Group Ev (SQLUC) of Germany has nearly 1,000 members, mainly in Germany but also in Japan and other parts of Europe and has been in action for nearly six years.

SQLUC publishes a quarterly magazine in German, *Quasar*, and holds local meetings monthly. "The QL is not at all dead, but remarkably alive. . . anyone who roughly understands German should think about getting a membership in our non-profit organisation", says Foreign Contact Franz Herrman.

The Sinclair QL User Club EV are contactable via Franz at Talstrasse 21, D-W5460 Ochenfels, W. Germany. Tel. (local) (0) 2644-1855.

Draw

Congratulations to the following winners in the MSM November subscriptions renewal draw:

J A Pedder of Luton wins £150

A J Smith of Northampton wins £50

R D Knight of Essex wins £50

When subscribers renew, their names will automatically be entered in the monthly renewal draw. You can renew at any time. Call 091 510 2290. Good luck!

It's a Q

Qlympic, publishers of *Goblin's Quest,* have written to say that the instructions for the program can now be read on televisions as well as monitors, Qlympic also have a new telephone number (German local) 02134 96694.

We also apologise for printing Qlympic's name as 'Olympic' — we had a flag out to catch that very obvious mistake, but it got through. We hope it won't happen again.

Qlympic's address is **Qlympic Computer Systems, Quellenweg 18, 4220 Dinslaken, W. Germany.**

New Sub-group

The first meeting of the new Quanta sub-group in London was held on August 1 at the Marquis of Clanrickad pub near Paddington Station. Tony Firshman of TF Services gave a demonstration of the QBox bulletin board, Mark Knight demonstrated his Molecular Graphics chemistry program (available from the Microdrive Exchange) and Laurence Reeves of QView gave a talk on Minerva.

The subject of the September meeting was QPAC2, the rewrite of Qram.

There will be a hands-on opportunity to try QPAC2. Each meeting will have a "beginners' corner".

Subsequent meetings will take place on the first Wednesday of each month. For information contact **Jeremy Davis, 6 Elmcroft Crescent, Harrow, Middx HA2 6HN. Tel. 081 863 1631.**

OPEN CHANNEL

Open Channel is where you have the opportunity to voice your opinions in *Sinclair QL World*. Whether you want to ask for help with a technical problem, provide somebody with the answer, or just sound off about something which bothers you, write to: Open Channel, Sinclair QL World, 116/120 Goswell Road, London EC1V 7QD.

Mail

I'm a first-year computer science student at Brunel University, and I have seen appeals for QL news through Electronic Mail.

There must be a few more QL owners who are university based with access to a mainframe. If so I would welcome electronic mail from them at C589559 @ UK, ac, brunel, cct.

I'm in a sandwich course, so am going for industrial placement very soon but these computer-ese people are welcome to write to me at home.

Sunil Gupta
45 Marlborough Crescent
Grays
Essex

Survival

I have been going to write to you for some time to say that I think the QL World is a most excellent magazine.

It is perhaps invidious to pick out particular contributors but Mike Lloyd's series on Better Basic and SuperBasic have been immensely educative and enjoyable. Also, Simon Goodwin's DIY Toolkit has provided many helpful titbits along the way.

So glad to learn that the magazine has survived an upheavel that could have seen its demise.

J W Hill
Matlock
Derbyshire

Editor's notebook

As the new Year approaches, I contemplate the things that gone astray in the Christmas Rush – a page here, an author there, a supplier somewhere else. Why has Bill Richardson not collected his valuable monitor? What has become of Abacus aticle? Where is my letters file? How is Mike Lloyd to finish the first instalment of the QL User Guide for the March issue? Who plugged in the first fairy lights where my computer used to be?

Then I contemplate all the things that haven't gone astray.

Bryan Davies has had first try of Perfection and reports in Troubleshooter. Simon Goodwin has finished Qlipboard a valuable new function for the QL. Howard Clase has sent an edition of Technical Helpline all the way from Nova Scotia. Jeugen Falkenburg has produceed a new PC keyboard interface. At last we have room to run the concluding part of Andy Wright's Programming in C. And much, much more.

1991 will be a time of more concentrated action on QL hardware than we have seen for a number of years. Interesting times indeed!

Mail

I was interested to read Bryan Davies' comments about the upgrade paths (un)available to QL users (October 1990). As a QL owner since 1985, I was very interested to read about some of the ideas being thrown around at the moment.

Having already decided to buy another computer, I have been shopping around looking at IBM compatibles with the view to buying a '386 (SX) system for myself. However, I have yet to come across a system that gives me the flexibility of true concurrent programs (Yes, some of us do use it!) together with a very good interpretation of Basic that I currently have access to on my QL. I find myself in the position that I am not preapred to pay the sums of money required to buy a machine that, in almost every respect except speed, is inferior to my QL.

So, in answer to your solicitation for ideas for new machines, I have included my 'dream machine' spec.

In my sunny, happy (QL) world I would be able to buy a 'box', with say a ten slot backplane (obviously built to a non-standard spec!) and the following cards installed.

CPU Card: 68040 processor running at 25 MHz with an amount (expandable?) of cache memory. 2MB of SIMM memory onboard. QDOS+ multitasking operating system in ROM (basically a 32 bit amalgamation of QDOS, TK2, TK3 and DIY toolkit routines) including a 'TaskMaster' sort of program/multitasking/file manager shell. Keyboard driver chips.

Memory Card: An address selectable memory card expandable to 4MB per board using standard SIMM's. Each system would be able to address a maximum of 7 such cards to give the possibility of 28MB of 'flat' expansion memory.

Disc Interface Card: Interface allowing connection of 2 Floppy drivers and a hard drive. Formatting, copying and backup routines contained on 'local' rom to allow true background task disc operations.

Video Board: Resolution around double that of the present QL display, and with an expansion of the palette.

Application Board: An EPROM/ROM allowing the installation of up to 16 application EPROMS which could then provide for embedded application multitasking. Workstations could be produced just using a floppy for data storage, all applications being on board and running at boot-up time.

Other boards could include, modems, real-world interface and development boards etc, etc. The use of a standard (sic) bus and the availability of blank boards could start many people experimenting furiously, not least of all myself. I have deliberately refrained from including a price range for this 'dream machine' because we all know that dreams do not come cheap. This, however, would be one dream that I would sell (almost) everything for!

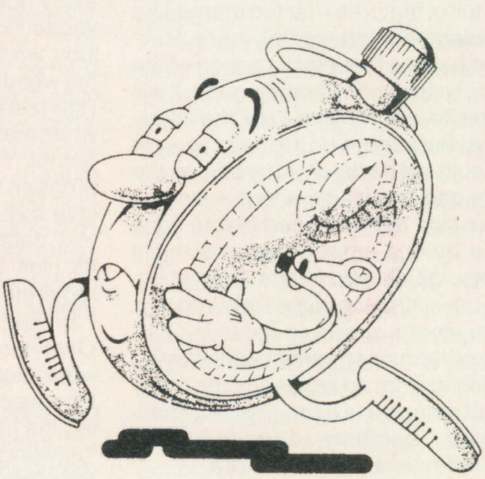
As for all the redundant QL's, I'm getting interested in simple multiprocessor applications so I'll take them off everybody for a fair price, although I'm sure many other can think of things to occupy a QL put out to graze! Any takers for my dream?

Gareth Williams
Shepperton
Middx

PRODUCT REVIEW

REALTIME CLOCK

The QL has always been short of a perpetual time and date function. Rich Mellor tries a battery-backed board which uses one of the rom sockets.



The QL has always lacked a means of retaining the correct time and date when it is switched off. In the past, there have been one or two systems which have managed to overcome this short-fall; but with the passage of time, these have become more and more difficult to obtain. This is a relatively recent system, which comprises a small circuit board, a small rechargeable battery and a few more Basic commands to load from disk.

The hardware element is quite simple to install. You open your QL (by removing eight screws), and remove two chips: the 8301 (which is a large chip next to the 68008) and one of the two rom chips. Next you push the main circuit board firmly into the empty rom socket, and push a through connector into the 8301 socket. Now all you have to do is replace the two chips by placing the rom and the 8301 chip into the supplied circuit board and through connector respectively, decide where to place the battery (there is room next to the expansion socket), and screw the QL back together.

File

Users who already have something else in one of the rom sockets, such as *Minerva,* or the Qimi interface, should not experience any problems, since the real time clock will quite happily sit in either rom socket. I did however find that the board supplied and the Minerva circuit board formed a very tight fit. Although I was able to install the board without too much trouble, some users may find that they need to file a little bit off one of the boards.

The instructions for fitting the Clock board are very straightforward and should not cause any problems. One thing however that the manual omitted to mention was that if you are unsure which way around the chips faced, there is a small notch on them which should always be towards the rear of the QL.

Once installed, the QL clock can be set to the same date and time as that kept by the real-time clock by using one of the four supplied Basic keywords. Should the stored date and time ever be incorrect (for example if the battery is running low) then you can set the QL clock manually and then use another new keyword to write this value to the battery-backed clock. Using this method means that the time on boot-up cannot be corrupted accidentally by software which uses SDATE for its own purposes, nor should it be corrupted by the QL being switched on and off.

Recharge

One advantage of this package is that there is never any need to replace the battery, since it constantly recharges itself whenever the QL is switched on. It should be able to keep the time and date accurately for up to one month without the QL ever being switched on, which should be more than ample for most QL users.

Extra

A small oddity with the package is that there is a spare 32 bytes of ram on board which is also preserved by the battery. This is accessed by using the other two new Basic keywords, but is so little as to prove not very useful, however the manual suggests that it could be used to store default baud rates or a default directory for *Toolkit II.* A more useful idea would be to use it to store a password for a software package; unless the unwelcome user knew that this memory existed, he would find it very hard to trace the password.

On the minus side, certain things plugged into the QL rom port can affect the real-time clock, since it is located in the last four addresses used by the rom port. This should not give many problems (certainly Toolkit II, *Ice* and *Lightning* all work okay), and can easily be checked before ordering the package by the following lines:

```
10 FOR a=1 TO 4
20 PRINT PEEK (1024*64-a)
30 END FOR a
```

If the values returned are all either 0 or 255, then you will have no problems. In the manual Phil states that he may be able to adapt the package for use with problem roms, but this will only be effective on JM and Minerva QLs.

Overall the package is extremely useful and considering the cost of making a hardware add-on in small quantities, is quite cheap. When balanced by the time and effort required to set the QL clock each time the QL is reset, the package is a bargain.

INFORMATION
Product: Real Time Clock (V1.20)
Price: £29.50
Supplier: Phil Gaskell, 16 Tennyson Rise, East Grimstead, West Sussex RH19 1SQ. Tel: 0342 312649

T

ROUBL

A PROBLE

Bryan takes a first look at DP's new word processor

The new WP program *Perfection* was advertised in the December issue. Those readers who have good spectacles will have noted in Digital Precision's advert that the program is full of features – far too many to be commented upon adequately here. Having been using a 'beta test' version of the program, I can confirm that it is good. It will take time to get familiar with the many functions, but users who just want a faster *Quill* should not be deterred by this, as the program can be used without a lengthy study session being needed before firing it up. To try and satisfy what one might categorise as *QL World* and *Quanta* users, the user interface comes in two forms: one with menus similar to those in Quill, and the other having combination keying more akin to what an editor has. As many users get along quite happily without having their spelling checked, the spell-checker is an optional extra.

Perfection

Although Perfection is in the same market area as *text*^{87}, it has a different feel to it and the two programs are actually somewhat complementary. Perfection does not set out to make the screen look like the printed page, but it has a more obvious bias towards the programmer than *text*^{87} has, in that many more commands are built-in for tailoring the behaviour of the program to the user's requirements. For example, the 'lazy screen' feature can be switched on or off. This means that text can be scrolled line-by-line, with all lines being shown at all times in their correct relative positions, or the whole display except the top or bottom line can be held static while just the one line changes (very rapidly). Not surprisingly, users of *The Editor* will find much to commend in the new program. Speed has been a target throughout the development of the program, and that aim has largely been achieved, with almost every function being performed briskly. As lack of speed is a major drawback of Quill, Perfection should prove a worthy successor. Having been assured that the version I am using is about to be replaced by a much faster one, I look forward to the latter being quite something.

Adding the new program to the system disk reminded me how long it has been since I had to make any significant change to the system. It's nice not to have to keep chopping and changing, but a change every now and then is healthy; it prompts you to think afresh and, maybe, do other jobs which had been forgotten. Despite the wonders of both Perfection and *text*^{87}, I still feel happier going back to The Editor when making changes to SuperBasic programs. You never quite seem to find one program that does everything for you, but give the new program consideration for text, database and SB programming use. Its high basic speed fits it for all three requirements.

Di-Ren, the producers of *Fleet Tactical Command,* are emphasising their hardware activities these days, and have introduced a small, cheap box for checking the health of QL networks. Called the 'network prover', this device sells for £3.50 (plus post and packing, and VAT). It is plugged in-line with the network lead between two QLs. A led on the box flashes when transmission is taking place over the network; when one QL is expecting data from another, but is not receiving it, the led comes on continuously. It was no surprise to find the network between my two QLs in a good mood when I came to test the device, but it was easy enough to simulate a fault condition by pulling one end of the network lead out, whereupon the led duly came on continuously.

Don't expect the led to flash all the time when the network is in use; if drives are running, data transmission is held up and the led won't flash, so there are only brief periods of flashing in-between long periods off, if you make copies between microdrives. The led does not come on continuously to indicate faults which are not the responsibility of the network; for example, pulling out a cartridge part-way through a copy operation is not a network fault, although it certainly halts any network operation that was in progress.

The cartridge used for checking the network prover was picked at random from a box-full of cartridges, none of which get any significant amount of use. The WCOPY command was used to copy files from flp1 on one system to mdv1 on the other. After much running of the cartridge, it became clear that something was amiss. It was not a network fault, so the led didn't come on. Being occupied with other things, I didn't check the QL for some minutes and, when I did, the fault turned out to be a defective cartridge. The tape was jammed, and my usual treatment – bouncing the cartridge off the workbench a few times – didn't relieve the situation. Every time I have to go back to using cartridges, I get trouble. Do other disk users find this? How did we ever get any work done in the 'good old days'?

As I've mentioned before, my personal experience of hard disk drives (on a PC, I hasten to add – not on the QL), leads me to feel floppies are a more reliable medium than either cartridges or hard disks (*hear jolly well hear – Editor*). Nevertheless, there's no way I would go back to using floppies as my main storage device, after a few years using hard disks, any more than a floppy user would go back to microdrive cartridges.

Expands

Computer equipment gradually expands to fill your available workspace, and one of the things you may need at some point is a switch for your printer(s). Maybe you have two computers using the same printer, or one computer using two printers; the problems is the same, but the necessary hardware may be slightly different. Before posting your mail-order cheque off, do a bit of 'thinking on paper'. Printer switches tend to be expensive, but you can get them for a reasonable figure. A 2-way, 36-line manual switch need cost no more than about £15; paying two or three times that much will very likely get you the identical switch to the cheaper one. (I know, since I've recently spent about £30 on one, because the supplier advertising the same switch at £15 had no stock, and I was in a hurry.) A 2-way switch will have three connectors on the rear, and the connector types vary with the purpose the switch is to be put to. Check the cables you already have, between computer(s) and printer(s). Do they have 9- or 25-pin "D" connectors at the computer end, and the larger, 36-pin type at the printer end? A printer with a parallel interface will normally have the 36-pin type, male connector; they are easily recognisable, usually being blue in the area of the connections, and the connections themselves being strips of metal rather than pins. Some printers have both serial and parallel interfaces, so you may find a 25-pin connector also. In general, it is simpler to use the parallel interface, because you as a user do not have to do any setting-up for it; you simply connect the cable, whereas you may have to do some configuring (setting hand-shaking, parity etc) when a serial interface is used.

The computer is most likely to have the 25-pin D type, unless it happens to be a QL, in which case it has the BT-like PCC connector, or a 9-pin D (if it is a German or US machine). Most users will have a serial-parallel interface as a separate device for their QLs, the Miracle interface being the common one. This being the case, you don't need to worry about the non-standard connector on the QL, when connecting the QL to a switch; the connector of concern is at the other end of the interface, and is the 36-pin printer type. Likewise if you have another computer, of a different type. Whatever the connector for its printer port, you should have a cable for it which has a 36-pin connector at the printer end. If you have two computers, then, of whatever type, the chances are that the connectors on their printer cables, which will have to fit to a 2-way switch, will both be of the 36-pin parallel printer type. The remaining connector on the switch is the one which will be used for the printer. If that connector is the 36-pin type, you are going to need a 36-to-36-pin cable between it and the printer, and that is not what suppliers will generally understand if you ask for 'a printer cable'. Specify just what connectors you want on each end of any cable ordered – both the number of pins and the connector type, and whether they are male or female orientation. If you already have the necessary cables, supply the same information when ordering the 2-way switch.

When you have printer problems of the missing-characters variety, always blame the printer connections in the first instance, With so many points of contact – especially when a switch is used – the chance of a bad connection is quite high. Separate all connectors, spray the pins with **contact** spray (*definitely not* any type of lubricating oil), and reconnect, making use of any locking devices on the connectors (try elastic bands to hold them together, if necessary). If the rotary connection wafers on a switch are accessible, spray the tracks on them, too.

The information which has been quoted in another magazine, that the President of Microsoft (who produce MS-DOS) is the 33rd-richest man in the world. So far as I am aware, the source of his wealth is the one company, and basically that one program (although the WP program *Word*, the multi-tasking program *Windows*, and a few other programs also sell well). Sales of MS-DOS have been quoted at about 11 million copies per year, which is nice money at maybe £30 average per copy (most are sold to dealers). Incidentally, Digital Research, the producers of CP/M, sell a 'compatible' replacement for MS-DOS, called DR-DOS. This is claimed to do all the things the original fails to do, such as giving access to hard disks larger than 32 MB, making 50 KB or so more space available for programs, and supporting expanded memory, MS-DOS 4.01 does provide some of these facilities now. *PC Conqueror* users might think about DR-DOS, if they haven't already got either MS-DOS or PC-DOS.

Readers' Letters

As recorded previously, my recent mail has not included complaints about the *Minerva* roms, but obviously some suppliers have been getting regular complaints. The target of the complaints is sometimes the supplier of Minerva (QView) and sometimes the supplier of software which has problems when run with Minerva. Most users will not be in a position to know whether software or hardware is causing the problem, although that does not seem to stop some of the taking a strong critical stand, one way or the other. One of the aims of the developers of the roms has been to remove known bugs in Qdos, the QL operating system. A laudable aim, but perhaps coming rather late in the day, since software writers have for many years either found ways around the major bugs or added their own fixes into their programs (or extensions files supplied with them).

Fixing known (and agreed) bugs in Qdos is one thing, but accusing software writers of having bugs in their programs is another, and one which undoubtedly raises the temperature quite a bit. This looks a bit like the situation with certain programs, where the proponents of Minerva are stating that they expect to see changes made to programs which have been sold for several years, and which have worked satisfactorily until the advent of (some versions of) Minerva. Some changes have been made, where it has been accepted that they were either necessary (regardless of Minerva), or could be implemented quickly and easily, to keep Minerva users happy. The program writers are not making other changes because they do not accept that the problems are caused by their code; that is, they feel the fault lies in Minerva. A little more consultation before some of the rom development took place might have prevented all the users' problems.

Here are some notes on reported problems. **Some** versions of Minerva don't get along with **some** of the toolkit functions with the *Turbo* or *QLiberator* compilers, in **some** QLs. **Some** versions of Minerva and *Professional Publisher* don't work too well together; specifically, the 'Edit Page' close-up window can be blank, and high-definition founts may be unusable. Version 1.81 Minerva prevents *Microbridge* loading fully; this may be solved by increasing the Dataspace allocated to Microbridge by 1 KB (if you have the Turbo dataspace routine available). Unspecified programs will not work unless the Minerva integer tokenisation function is switched off. *FlashBack SE* displays minor quirks when used with Minerva V1.81; *Touch Typist* does not load fully with the same version. The SEXEC command gives loading problems with some programs, when V1.82 Minerva is used.

Spencer Harvey reports a lack of response (as of mid-November) from **Sector Software** concerning a printer interface, returned by him as faulty in June 1990. It has been suggested that the interface never got back to Sector, but letters concerning it did, and it would seem reasonable to have had some reply to them by now. A QL user who seems to have regular problems with his equipment (hardware and software) is **J Roy Goodall,** and his location (Belize) makes sorting-out problems a rather difficult and protracted business. He eventually received a long-delayed printer interface from **Miracle Systems,** along with a cheque which was apparently a refund of the cost for it, and has also purchased a Trump Card. He tried out *TechniQL* on a QL he has recently obtained, but had 'the usual negative results'. Another acquisition is a PS/2-style keyboard, from **Keyboard Products**; neither of his QLs will give the F1/F2 start-up screen with this keyboard fitted, so he is now awaiting comment on this problem. He says his rechargeable batteries for the **Frequency Precision** power back-up unit are in need of replacement, but this device has seen him through "many blackouts, and also overcame overheating problems", over a period of two years.

MIRACLE SYSTEMS

MASTERCARD VISA

Microdrive users - read this ...

NEW TRUMP CARD

£225 inc. (£198 export)

RAM + Disk interface + firmware

We have re-engineered the TRUMP CARD 768K to use the new 1 Megabit DRAM memory chips. This new design runs about 20% faster (twice the speed of the QL's internal RAM) and uses less power than the previous one (still available in the 256K size). It holds the same firmware:

- TOOLKIT II which comprises more than 100 additions and enhancements to the QL's Superbasic and operating system including an on-screen alarm clock, wild card copying, accessing remote devices on other QLs equipped with a ROM-based TOOLKIT II via the network.
- a printer buffer which can be used to buffer the serial ports (the size of which is limited only by the amount of free memory) letting you get on with something else whilst the printer is printing.
- a screen dump facility to copy all or part of the screen image to most types of dot-matrix printer including some colour ones.
- a RAM disk that allows you to access the memory as you would Microdrives or floppy disks for fast file retrieval (please note that RAM disk contents are lost after switch-off or reset).
- a memory cut that resets the QL to appear as an unexpanded 128K type for the few early programs that refuse to run in expanded memory.

The disk interface can access up to 4 disk drives (e.g. our DUAL 3.5" plus our 5.25") and has the same commands as are used for Microdrive control. There is an additional command FLP_USE which can be used to divert all MDV accesses to FLP (the floppy disk interface device name). This makes the transferring of your software from unprotected Microdrive (i.e. the majority of QL software including Quill, Abacus, Archive and Easel) to disk a trivial task. A simple step-by-step guide for transferring Quill as an example is given in the comprehensive TRUMP CARD USER MANUAL supplied with the TRUMP CARD.

The TRUMP CARD 768K's RAM adds to the QL's own 128K giving a total of 896K. Like the firmware the extra RAM is installed automatically when the QL is switched on so that no installation procedure is necessary. The exception to this is TOOLKIT II which can be left uninstalled for compatibility if you have other toolkits; installation consists of simply entering the command TK2_EXT.

Fitting the TRUMP CARD 768K is easy - you remove the door at the left hand end of the QL and slide the TRUMP CARD into the expansion port. A "Beginners Guide" on pages 3 and 4 of the *TRUMP CARD USER MANUAL* will quickly get the novice and experienced user up and running.

TRUMP CARD 768K PACKAGE

£375 inc. (£333 export)

TRUMP CARD 768K + dual disk drive + 10 diskettes

This is the ideal upgrade path from obsolete Microdrives. The package comprises the latest TRUMP CARD 768K plus a QL standard floppy disk drive with 2 mechanisms plus ten 3.5" double-sided double-density diskettes. The only extra item required is the appropriate mains plug to suit the country where it is to be used.

Disks are more reliable than Microdrives, hold much more information (720K after formatting) and are several times faster. Besides these advantages they actually cost less. Our QL DUAL DISK DRIVE is fully boxed in a black metal casing and is mains (220V-240V AC) powered.

An EXPANDERAM 512K can be used as part payment against the TRUMP CARD 768K PACKAGE. Just send it to us together with £285 (£255 for overseas customers) remittance and we will send you the TRUMP CARD 768K PACKAGE.

This package transforms the unexpanded QL into a very powerful machine and is very easy to fit. We are confident that you will find this investment more than worthwhile as many QL users have already done so. If you are not fully satisfied with your purchase then by returning it to us within 14 days of receiving it we will return your money in full.

When ordering by phone it is sometimes easier to spell names and addresses using the phonetic alphabet

A - Alpha
B - Bravo
C - Charlie
D - Delta
E - Echo
F - Foxtrot
G - Golf
H - Hotel
I - India
J - Juliet
K - Kilo
L - Lima
M - Mike
N - November
O - Oscar
P - Papa
Q - Quebec
R - Romeo
S - Sierra
T - Tango
U - Uniform
V - Victor
W - Whiskey
X - X-Ray
Y - Yankee
Z - Zulu
0 - Zero

MIRACLE SYSTEMS Ltd
25 Broughton Way,
Osbaldwick, York, YO1 3BG,
U.K.
Tel: (0904) 423986

MIRACLE SYSTEMS

MASTERCARD VISA

14 day money back guarantee on all products
12 month warranty on all products
UK prices include VAT and P&P
(Export prices in brackets include P&P)

TRUMP CARD 256K
£135 (£120)

This is an ideal way to start expanding the bare QL for those who are not ready to purchase the full TRUMP CARD 768K. It comprises a disk interface, 256K of memory and utility software. The disk interface can control up to 2 double density drives. It can be made to control 4 drives by the addition of the DISK ADAPTER. The 256K memory adds to the QL's 128K giving a total of 384K. This memory is automatically installed at power-up such that QDOS cannot see the join. Programs running in the extra memory, eg Quill, run about 1.75 times faster. The memory can be upgraded to 512K or 768K by the addition of 8 or 16 memory chips of the type 41256. (Please note that we neither supply the chips nor do we do the upgrade.) The utility software includes TOOLKIT II, printer buffer, screen dump, RAM disk and memory cut.

NB Adding the DISK ADAPTER or extra memory chips will not increase the speed of this TRUMP CARD.

DISK CARD
£100 (£89)

This disk interface is intended for use with internally or externally expanded QLs. It can plug directly into the QL's expansion port or into the through connector on the EXPANDERAM. The circuitry is derived from the new TRUMP CARD 768K which can access up to 4 disk drives, and it includes TOOLKIT II, RAM disk, printer buffer, screen dump and memory cut software. There is no memory driver circuitry and memory cannot be fitted directly onto the DISK CARD.

DISK CARD PACKAGE
£250 (£224)

This package comprises a DISK CARD, a DUAL 3.5" DISK DRIVE, and 10 diskettes.

EXPANDERAM 512K
£99 (£88)

This card plugs into the QL's expansion port and increases the memory from 128K to 640K. QDOS recognises the extra memory during power-up so there is no need for the user to inform the QL that extra memory is installed. Programs run in the EXPANDERAM about 1.75 times faster than on an unexpanded QL. Quill users will not just see this speed increase but will also be spared the 'DEF_TMP' syndrome which occurs only on unexpanded QLs. A through connector is provided for connecting a disk interface such as our DISK CARD. The EXPANDERAM cannot be used with internal expansions or external memory like our TRUMP CARD since this would cause an address clash. Users wishing to upgrade to disks can use their EXPANDERAM 512K as part payment towards a TRUMP CARD 768K.

QL CENTRONICS
£29 (£28)

The simplest way to connect a parallel printer to the QL is by using this interface. It measures just 3" by 2" by 1" and plugs directly into the standard CENTRONICS port on the printer. Included is a 3 metre cable that plugs into either SER1 or SER2 on the QL. No setting up of the QL is required since the interface works at the QL's power-on default set-up of 9600 baud, 8 bit data, no parity and 2 stop bits. With two interfaces, two printers can be driven simultaneously - one from SER1 and the other from SER2.

TRUMP CARD 256K PACKAGE
£285 (£255)

This is all that's required to get disks up and running on the QL. The package consists of 3 things: the TRUMP CARD 256K, the DUAL 3.5" DISK DRIVE and 10 diskettes. The TRUMP CARD plugs into the QL's expansion socket and the DISK DRIVE plugs into the TRUMP CARD.

DISK ADAPTER
£15 (£15)

TRUMP CARDs purchased prior to March 1990 together with TRUMP CARD 256Ks use the original TRUMP CARD design which is able to access up to only 2 drives. However, the DISK ADAPTER contains a small amount of circuitry which allows access to be increased to 4 drives. It plugs into the disk drive socket on the TRUMP CARD and comes with a replacement ROM containing the latest version of TOOLKIT II. This is ideal for adding our QL 5.25" DISK DRIVE to a DUAL 3.5" DISK DRIVE system.

QL DUAL 3.5" DISK DRIVE
£175 (£155)

This is the drive supplied in the TRUMP and DISK CARD PACKAGEs. There are 2 mechanisms; each one is 3.5", 80 tracks per side, double sided, double density, with a formatted capacity of 720Kbyte per diskette. This defines the standard disk format for the QL and is probably more widely used than Microdrive cartridges. The key advantages that disks have over Microdrives are as follows:

a) the capacity is much greater
b) the speed is much higher
c) the media are much cheaper

Also 3.5" diskettes are widely available. The drive mechanisms are housed in a black metal case which also contains the mains power supply. The drive comes with the necessary cable to connect it with the disk interface. A disk interface such as our TRUMP or DISK CARD is needed to use it with the QL.

MIRACLE SYSTEMS

MASTERCARD VISA

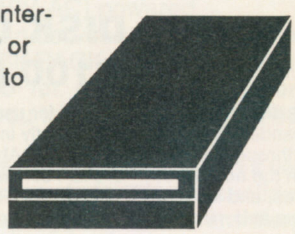
14 day money back guarantee on all products
12 month warranty on all products
UK prices include VAT and P&P
(Export prices in brackets include P&P)

QL 5.25" DISK DRIVE

(includes free DISK ADAPTER)

£95 (£88)

CONQUEROR and SOLUTION users will find this drive invaluable for loading PC compatible diskettes. It conforms to the standard PC format: 40 tracks per side, double sided, double density, and gives a formatted capacity of 360 Kbytes. The drive comes complete with a DISK ADAPTER so that nothing extra is needed to add it into a system with the old or new TRUMP CARD and DUAL 3.5" DISK DRIVE. Users upgrading from Microdrives are advised to consider our DUAL 3.5" DISK DRIVE as this is the standard disk system for the QL. The 5.25" DISK DRIVE requires a disk interface such as the TRUMP or DISK CARD in order to be connected to the QL.



QL HARD DISK

£449 (£405)

40 Megabytes of on-line storage is what this is about. To cope with large numbers of files a new sub-directory system has been implemented. Accessing the HARD DISK is approximately 5 times the speed of a floppy. The HARD DISK uses about 55K so memory expansion is essential. Also it is highly recommended that the QL system has disk drives fitted such as our TRUMP CARD 768K PACKAGE so that the HARD DISK can easily be backed up. The HARD DISK plugs into the QL's ROM port and has a through connector for ROM cartridges.

Tel: (0904) 423986

To place an order by phone please have your credit card ready. For overseas customers we charge the prices shown in brackets.

To order by post, please fill in the form opposite or write to us quoting your credit card number and expiry date, or enclosing a cheque payable to MIRACLE SYSTEMS Ltd

✂

To: MIRACLE SYSTEMS Ltd, 25 Broughton Way, Osbaldwick,
York, YO1 3BG, U.K. Tel: (0904) 423986

Please send me ____________________

I enclose a cheque to the value of: £ ____________________

Or debit credit card □□□□□□□□□□□□□□□□

expires □□/□□

Name ____________________ Signature ____________________

Address ____________________

THE ITALIAN JOB

The QL is still active in Italy, where the annual QL Users Meeting 1990 took place recently by Lake Garda. The verdict: a success, but rather hard to get there.

This is a user's-eye report from the Second Italian QL Users Meeting which was held at the Management Conference Centre at San Felice del Bennaco which is on the southern shores of Lake Garda, halfway between Milan and Venice.

The southern shores of Lake Garda are a beautiful part of the world to spend your time if you have any but there was no time for sightseeing at the Meeting.

Committee

The Meeting was as far as the Brits were concerned ably organised by Eros Ferrenzi, who is a student. The Users committee consisted of industrialists and enthusiasts from northern Italy and the company called SPEM, owned by Guido Masoero, who operates from Turin and who the principal QL trader in Italy.

Miracle Systems, who usually travel to QL meetings by bicycle, left their bicycles behind and travelled by car to Dover and then to Brescia by train from Calais. Bill Richardson of EEC Ltd chose to fly carrying several kilos of QL product in his baggage, but despite Al Italia hospitality onboard found himself incarcerated in the Customs sheds at Milan Airport for three hours, missing the opening stages of the afternoon, when not many people attended. Another trade attendee was Jochen Merz from Germany, showing an updated Version 3 *QD Editor* and an improved range of software.

Sunday was a beautiful morning and it would have been nice to have gone sightseeing rather than the Show but on arrival we found twenty to thirty QL enthusiasts waiting for us, mostly from local areas but several had travelled from as far as Rome and San Remo on the French/Italian border.

Audiovisual

During the course of the day several QL presentations were held in an excellent lecture hall well equipped with audio visual paraphenalia and material put over by the Italian delegates, Miracle Systems' Stuart Honeyball and Jochen Merz. All was efficiently translated into Italian by the members of the User Group. The opinion of the UK exhibitors, Stuart Honeyball of Miracle Systems and Bill Richardson of EEC Ltd was that, while sales were good and most of the stock taken by both companies was sold, sales takings were great in millions of Lire but condensed into pounds sterling it probably represented the kind of sales that one would have expected from a one day Show at the Horticultural Hall in London. Expenses were well exceeded, though, and it was felt that a number of orders will follow the visit. Quanta leaflets and back numbers on the EEC stand attracted a lot of attention and a number of new members will result.

Only about 100 visitors attended however and all exhibitors felt that the same faces seem to be around who we had previously met and it was thought that, bearing in mind the somewhat remote tourist area around Lake Garda, it would be nice to see other shows in the more populous areas of Italy such as for example, Turin or Rome and it was suggested by an attendee that a good place for a QL meeting be Florence in future.

SUPER BASIC

I

Most computer users occasionally write code to start software. Mike Lloyd provides a step-by-step guide.

Every computer system needs them, yet many computer users cannot create them or modify them. Some are remarkably simple, while others are unavoidably complex. Most of them are essential, but on occasion they are merely convenient. Some have a single purpose, others offer a wide choice of options. Some never change, but many need to be modified to suit particular needs. They are, of course, boot files: short programs required to load and execute other pieces of software.

Most of the requests I receive for help relate to boot files, particularly ill-considered amendments to them which need to be put right. A quick straw poll amonst software sellers and producers reveals that boot files come pretty high up their list of reaons for customer calls. It is not that the boot files supplied with the software cause problems; it is the enthusiastic but badly-planned adjustments to standard boot files which lead to problems.

Bootfile

This SuperBasic article and its successor are devoted to the techniques and principles of the humble boot file, including a complete review of the commands involved, a look at boot file editing methods, some hints and tips, and a simple but fully-functioning menu-driven boot file which can be adapted to suit your own software library. I hope it answers the vast majority of questions about the subject and reduces the number of calls for help to suppliers.

To simplify the text and to avoid saying everything twice, I have assumed that floppy disks are being used throughout the rest of the article. If you are still using microdrives, replace all occurrences of FLP1_ with MDV1_ and translate all mentions of 'floppy disk' and 'disk drives' to 'microdrive cartridge' and 'microdrive'.

A boot file contains a number of program commands which are carried out when a disk or microdrivé is accessed during the boot-up sequence when the computer is first started. This is incredibly useful because, instead of being faced by a blank screen and a flashing cursor, users can be presented with their favourite program loaded and running automatically.

Disk User

The main purpose of a boot file is to issue the commands needed to load and execute another, larger program. It might also contain some commands to change aspects of the QL's settings to meet the needs of the software it will launch. Disk drive users, who can place many substantial programs on a single disk, often develop a boot file with a menu so that they can select which software they wish to run. This is less likely to be the case with microdrive users because it is often not possible to place more than one large program on a single microdrive.

There is nothing magic about a boot file's contents. Usually, they contain the commands which would otherwise need to be typed in by the user directly whenever a piece of software was to be loaded. Generally, the commands in a boot file fall into three categories: file-loading commands, environment-setting commands and incidental commands. The file loading commands perform the job of loading programs from disk or microdrive and starting them off. Environment-setting commands might modify the screen settings, or establish parameters for the printer channel, so that the main program knows exactly how the QL will behave. Some users add extra commands to set up ram-disks, or make multi-tasking software available, in order to meet their own particular needs. Incidental commands include special screen displays, or perhaps a menu.

Likewise, there is nothing very special about the file itself. It is simply a Qdos convention that when a microdrive is present in mdv1_, or a floppy disk is in flp1_, during the start-up sequence the operating system looks for a file called 'boot'. If it does not exist, nothing happens, but if it does it is assumed to contain SuperBasic commands which are executed. If you forget to put a microdrive or disk in place when starting up the computer, type LRUN FLP1 BOOT in the command window and press Return. LRUN is a SuperBasic command which means "Load and RUN" and can be used to launch any SuperBasic program.

Incidentally, boot files are not particular to the QL: all modern operating systems have them. In MS-DOS, they are called AUTOEXEC. BAT files and in Unix they are called .PROFILE files, but their purpose is the same as the QDOS BOOT file. Of the three, the Qdos name is by far the most meaningful.

Boot files are always standard SuperBasic program files, rather than being machine code files. This ensures that they are easily readable and adjustable. To load a boot file without executing the commands it contains, start up the QL without a disk or microdrive in place. The QL screen will be set to the default according to whether the start-up sequence was initiated by an F1 or F2 keypress. The screen will contain three windows, although in the tv display mode only two are immediately apparent. The lower window is the command window, where SuperBasic commands are typed and program lines are modified. Above the command window are the listing window and the default window. In Monitor mode these windows are side by side whereas in TV mode the default window is superimposed over the listing window.

To view a boot file, place the appropriate disk in the flp1_ drive and type:

```
COPY FLP1_ BOOT TO SCR_
```

The boot file will then be displayed in the centre of the screen. Press the F5 key to pause scrolling if the beginning of a long file looks in danger of scrolling off the screen. If each line begins with a number then the file can be loaded directly into memory. If there are no leading line numbers we must add some before we are able to edit the boot file.

To load a numbered boot file, place its disk in the appropriate drive and type in the command to load it into the QL's memory:

```
LOAD FLP1_ BOOT
```

The disk drive will spin into life and if a file called BOOT exists it will be loaded and the cursor will flash ready for the next command. To view the file's contents, type:

```
LIST
```

Sometimes, software developers write boot files which do not have line numbers, for reasons best known to themselves. If you wish to edit such a file, you can copy the details of the file from the screen (or a printout) and type them in again with line numbers, or you can use the short routine at **Listing** one to do the job automatically. The file it creates is called "NEWBOOT". When this file has been edited, delete the old BOOT file and save the amended version as "BOOT".

Before we can begin to modify these commands their purpose must be understood. The heart of any boot file comprises the instructions to run the main program or programs, so let us look at those first.

> **WARNING:** Always edit a copy of the original boot file, not the original itself. That way, if you make a mistake it can be rectified by taking a new copy of the original boot file and starting again.

There are three ways of causing a program to be run from file on the QL: the LRUN command for SuperBasic programs, the RESPR/LBYTES/CALL sequence of commands for machine code programs and the EXEC family of commands for multi-tasking programs. Variations of these commands are available in toolkits, and will be covered at the appropriate points below.

It is unusual for SuperBasic programs to be launched from boot files, but it can be convenient to do so and therefore the method is detailed here. The command is LRUN followed by the file name. Note that there is no point in following the LRUN command by any other commands as they will never be carried out: the incoming SuperBasic program completely overwrites the commands loaded from the boot file.

Very little software for the QL has been produced in conventional machine-code format because it is easier and more advantageous to make such programs multi-tasking. However, it is very common to see machine-code format programs which set up a multi-tasking program of some sort. Perhaps the prime example is Qram, the Multi-tasking utilities package. It is machine code programs such as these which lead to the majority of problems encountered by *QL World* readers.

Three Stages

With standard Qdos and SuperBasic, loading a machine code program is a three-stage process. An area of the QL's random access memory (ram) must be specially designated for the program, the code must be copied from the disk into the designated memory area and, finally, the code must be started off.

Memory is designated using the RESPR (REServed PRogram space) function. You ask the computer to reserve a memory area of so many bytes and the computer in turn tells you where that memory areas starts. The syntax for the command is:

```
LET base = RESPR (2048)
```

Here, the request is for a 2K chunk of memory and the first byte in that chunk is at the address number which Qdos will have placed in the "base" variable. You could just have easily typed:

```
PRINT RESPR (2048)
```

which would have reserved a space and printed its first address on the screen. Putting the address into a variable saves the effort of writing it down and typing it directly into subsequent commands.

You can find out how much space to reserve for a particular program by referring to the product's manual or by using the WSTAT Super Toolkit command. Memory blocks are allocated in lumps of 512 bytes, so there is no requirement to reserve exactly the right amount of space, provided that there is enough roped-off memory available. If you intend to load two or more machine code programs sequentially, you can reserve a single space to contain them all, provided that each is loaded beginning at an address which ensures that none are overwritten.

RESPR will fail if there is insufficient ram available, or if there is a multi-tasking program running when it is called. The first restriction can be solved by removing any other reserved memory blocks which might be present, or by buying extra memory for your QL. The second restriction has been overcome by replacing the RESPR command with toolkit functions such as ALLOCATION, from the Turbo toolkit.

Having reserved space in memory, the machine code program must be loaded into it. The command for this is LBYTES, followed by the file name and the address of the first byte of reserved memory space. To follow on from the RESPR command above, a typical command might be:

```
LBYTES flp1_ little_ prog, base
```

If a single ram area has been reserved for a number of machine code files, careful control must be maintained of the starting address for each LBYTES command if problems of over-writing are to be avoided. Additionally, it is a Qdos rule that machine code programs must begin at an even-numbered address in memory. Let us take as an example the following Qram files, with their approximate sizes:

Sharing

flp1_ wman (5006 bytes)
flp1_ ptr imi (9654 bytes)
flp1_ ramprt (1024 bytes)

Memory space can be reserved and occupied by the following commands:

```
LET progA = RESPR (15684)
LBYTES flp1_ wman, progA

LET progB = prog A + 5006
LBYTES flp1_ ptr_ imi, progB

LET progC = progB + 9654
LBYTES flp1_ ramprt
```

The final part of the sequence is a CALL command which activates the code loaded

LISTING 1

Note: this listing is a stand-alone program which should be saved (eg SAVE flp1_editboot) before being run. Do not type the italic remarks.

```
100 OPEN_IN#3, FLP1_BOOT              | Open existing boot file
110 OPEN_NEW#4, FLP1_NEWBOOT          | Open a new file
120 LET NUMBER = 10                   | Set line number value
130 REPeat LOOP                       | LOOP:-
140   IF EOF(#3): EXIT LOOP           |  Stop if file end reached
150   INPUT#3, A$                     |  Read next line from BOOT
160   IF A$(1) <= "9": EXIT LOOP      |  Stop if its already numbered
170   PRINT#4, NUMBER & " " & A$      |  Add numbered line to NEWBOOT
180   NUMBER = NUMBER + 10            |  Increment the line number
190 END REPeat LOOP                   | END LOOP
200 CLOSE#3                           | Close the BOOT file
210 CLOSE#4                           | Close the NEWBOOT file
220 PRINT "COMPLETED"                 | Signal end of program
```

by the previous commands. The CALL command needs one parameter, the address of the first byte of executable code. Strangely, this might not always be the very first byte of the reserved area, in which case the software manual will advise you to use a command such as:

```
CALL base + 12
```

Fortunately, for everyone who finds this three-command sequence a bit of a bind, Tony Tebby's *Super Toolkit 2* contains a single command which performs all three tasks and which does not require the user to worry about start addresses, even-numbered bytes and 512-byte memory chunks. The command is LRESPR, and it is followed by a single parameter — the program's file name. Therefore, instead of typing:

```
LET base = RESPR (1024)
LBYTES flp1_ program, base
CALL base
```

the following single command can be used instead:

```
LRESPR flp1_ program
```

The final category of runnable files comprises those that can be multi-tasked together. To take full advantage of multi-tasking software it is best to purchase *Qram* or *Qpac II* because these products modify Qdos to improve the multi-tasking environment. Almost inevitably, if you intend to load more than one substantial executable program at a time your QL will need memory expansion.

The commands to load and initiate an executable program are quite straightforward. If the program is to work at the same time as other multi-tasking programs the command is:

```
EXEC flp1_ filename
```

If a number of programs are launched in this way they will share processing time and appear to be working independently and simultaneously. For programs which need keyboard input this can be a problem — which of the programs should respond to a particular set of keypresses? Which program is going to have first access to the printer, or the disk drive? Qdos has no simple answers to these questions, and so its multi-tasking environment needs to be tamed.

The easiest way of preventing input and output clashes is to tell Qdos that the program being run takes full control of all device channels and processing time until it is finished. The command to do this is EXEC_ W, and its syntax is exactly the same as for its EXEC_ relative. The W extension means "Wait until terminated". For the vast majority of executable programs commercially available the EXEC_ W keyword should be used as a matter of course. It is nervertheless worthwhile to be aware that under Qram and Qpac II the QL's multi-tasking environment is changed so that EXEC and EXEC_ W have the same effect.

Super Toolkit 2 expands the EXEC family to include two more keywords, but their full potential is of more benefit to knowledgeable programmers than for non-programmers. However, for convenience use can be made of:

EX Exactly the same as EXEC
EW Exactly the same as EXEC_ W

Turbo Toolkit provides EXECUTE, EXECUTE_ A and EXECUTE_ W as other alternatives. These commands all allow more parameters than just a filename, but their use lies beyond this article.

There are advantages which can be gained from the features of a multi-tasking program without necessarily running it simultaneously with another program. The Psion business utilities bundled with the QL are all multi-tasking executable files which can all fit on a single disk, but only one of which can be loaded into a standard QL ram at one time. Nevertheless, a form of task-switching can take place provided that an appropriate boot file is constructed. The details of this boot file will be printed next month, along with many more useful and practical hints about boot files.

SOFTWARE FILE

DOUBLE BLOCK

Rich Mellor goes Russian in an old game brought up to date.

This game is based upon a popular Russian game called 'Tetris' which is in itself a simple, highly addictive game. The idea of the game is that several different-shaped multi-coloured blocks fall from the top of a 'pit' on screen. These must be rotated and moved left or right so that they form a full line of blocks across the screen. Once a line has been completed, it disappears from the screen so that more lines may be formed within the 'pit'. Should any line have a gap (or two) in it, this will remain on screen, with the pit gradually filling up with incomplete lines. Once there is no more room in the pit to allow another block to be dropped, the game is lost.

With the idea behind the game being so simple, the keys must also be simple. The left and right cursor keys move the falling blocks left or right, the up cursor rotates the block, and the down cursor makes the block drop more quickly. This is a straightforward combination which allows a joystick to be used if necessary. However, this version of the game also incorporates a two-player mode, where both players have a pit on screen, each with its own set of falling blocks. One player can use the cursor keys, while the second player really needs a joystick plugged into CTL2 (although the function keys F1 to F4 can also be used).

Colour

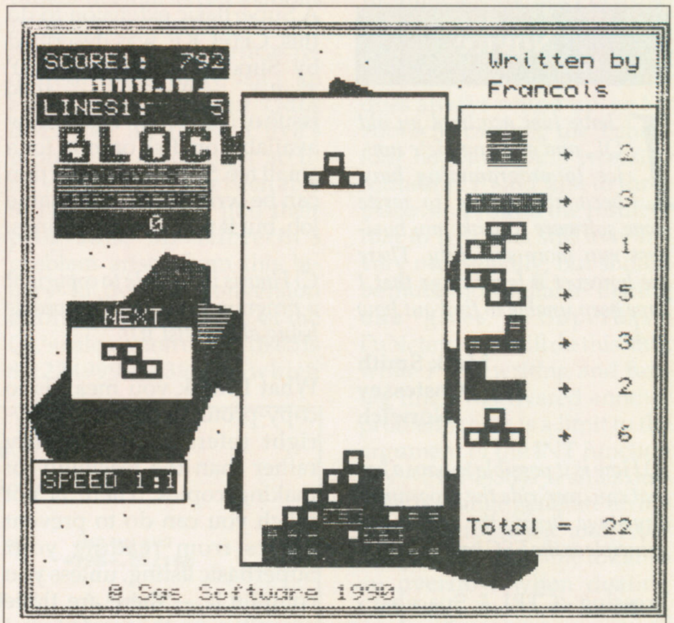
The display on screen is well laid out with lots of colourful graphics, and even a little sound to keep you amused. Also shown on screen is your present score. There is an option in the one-player mode to show the next block which will be dropped, so that you can plan ahead. Peculiar to this version of the game is an extended graphics mode, where the blocks can be any one of 16 shapes (the normal set features only seven). This extended graphic mode certainly makes the game much more difficult and should prove a challenge to the most practised Tetris player.

Points are scored in this game for each block which falls and each line which is formed. The game maintains a high score table, which on start-up contains some high scores achieved by the author: these are depressingly high however, and you will be glad to learn that you can get rid of the current high scores at any time merely by deleting one file from your disk.

Faster

The game also features 10 different speed levels; these can be different, if you are playing in two-player mode, for each player. However as you play the game, the speed of the falling blocks increases, so that with the screen slowly filling with incomplete lines, you soon begin to realise that your reaction times are much too slow. It was however nice to see that the speed increase is so set that in the two-player mode, if one player begins at speed 0 and the other at speed 6, you will both reach speed 7 at the same time! Still, if it all gets too much for you, a press of the 'P' key will pause the game while you get your breath back and decide just what to do with the mess you have made on screen.

In all the game is a most welcome rest from the normal type of arcade game which appears all too readily on home computers. This game requires thought, quick reflexes and an eye for opportunity. It reminds me of that old children's educational toy where they have to fit the different shapes into the different holes, the only difference being that here much more speed is needed, and a better eye for patterns.



SCORE1: 792

LINES1: 5

BLOCK

0

NEXT

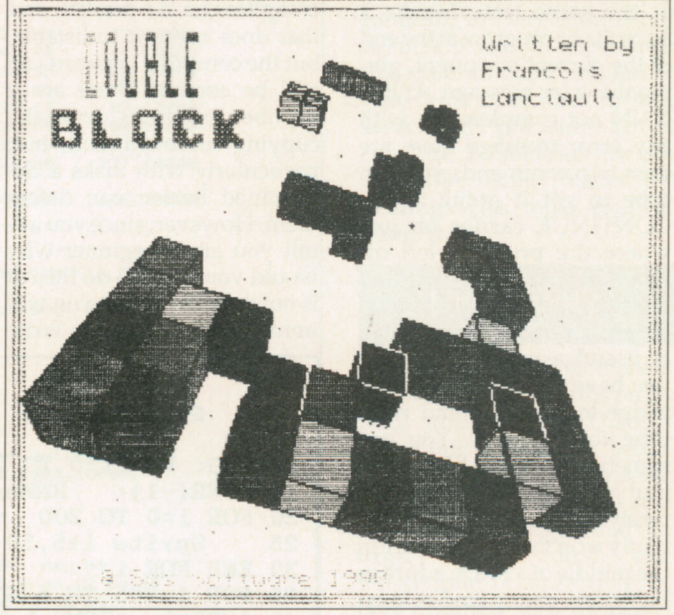
SPEED 1:1

Written by
Francois

+ 2
+ 3
+ 1
+ 5
+ 3
+ 2
+ 6

Total = 22

© Sas Software 1990



DOUBLE
BLOCK

Written by
Francois
Lanciault

© Sas Software 1990

INFORMATION
Product: *Double Block*
Price: £9.00
Supplier: CGH Services,
Cwm Gwen Hall, Pencader,
Dyfed, Gymru SA39 9HA.
Tel: 0559-384574.

Technical Helpline

Howard Clase picks up some readers questions about technical aspects of the QL.

Novice

I have just acquired an old QL, and as a complete novice to programming have been attempting to write some software to make my business run more smoothly. There are however a few things that I have been unable to find out how to do.

Mark Smith
New Costessey
Norwich

A) How is it possible to have a key that can over-ride the program at any stage during the run, (same principle as the ESC key in Quill)?

Pressing <CTRL & Spacebar> at the same time breaks a SuperBasic program at the end of the current statement, giving the error message At line NNN not complete. As with any error message there are then two commands you can type to get it going again: CONTINUE carries on just where the program left off while RETRY goes back and does the last statement again before continuing — this can be useful, as when you realise you have the wrong mdv cartridge in the drive and want time to change it. You can sometimes actually EDIT a line and then RETRY, but this rarely works properly; it certainly won't work if you are in the middle of a PROCedure or FuNction, or any kind of loop; the program tends to forget where it was in these when you EDIT.

B) Is it possible in one line to check that an input is numeric? (In order to stop a program crashing if a non-numeric amount is entered in a numeric variable.)

A SuperBasic extension to do this, CHECKF, was described by Simon Goodwin in *DIY Toolkit* in the January 1989 issue of *QL World*. This is now available on disk or mdv for a small fee. SuperBasic function can be written to do the same job, but it takes about 35 lines.

C) How is it possible to copyright a program (to stop other people being able to list it)?

What I think you mean is to copy-protect a program copyright refers to the legality rather than the feasibility of making copies. There is not much you can do to prevent others from reading your SuperBasic listing, unless you compile it — there are three compilers on the market, and that does make it unlistable, but the compiled versions can still be copied. There are a number of tricks to make copying more difficult, but, particularly with disks a determined hacker can defeat them. However, since you admit you are a beginner why would you want to do this? It is not very likely that you will immediately be able to write the ultimate QL program, and make your fortune! Fortunately many competent programmers are prepared to rely on the law of copyright and allow users access to their code. That is one of the ways a beginner can learn: study other people's programs from wherever you can get them — MDX exchange, Quanta library if you are a member (and any serious QL user should be), QL World listings, etc. and see how they do it. Once you find yourself saying, "I can see a better way to do that," you are on your way.

Background

Please can you tell me the command which changes which part of the ram the screen is, also can you tell me how games writers stop the background graphics being erased when the sprites walk over them? I am hoping to include these in a Basic program.

Peter Smith
Middlesborough
Cleveland

Although Sinclair planned a two screen mode for the QL he never implemented it in any of his versions of the rom. However QView's *Minerva* rom does have this facility along with many other improvements.

True sprites require considerable expertise in writing machine code, but you can make use of the OVER -1 command in SuperBasic to get a similar effect. This has the effect of XORing any printing or graphics in a window against the background until the OVER command is used again. So that printing the same thing twice at exactly the same spot causes it to appear and disappear again, restoring the original background. Because it is in SuperBasic it won't be very fast. The short program in the listing illustrates the principle.

The 'sprite' here is a simple cross which moves in a sinuous path over the text, returns straight to the starting point and then vanishes. The more complex your sprite the slower will be the movement. I've drawn the sprite twice within the PROC; this is faster than calling it twice from within the loop. It is best to design your sprite with the BLOCK command if you can rather than with the graphics commands because they are also slower. Changing the parameter of the PAUSE command in line 65 can be used to adjust the time the sprite spends on the screen, for maximum speed remove it completely. Use the variable c in the procedure to experiment with different colours.

Sprite Listing

```
10 CLS: FOR i=0 TO 208: PRINT"Fred";
15 OVER(-1):    REMark XORs screen output
20 FOR i=0 TO 200
25    Sprite i+5,50+50*SIN(PI*(i)/50)
30 END FOR i
35 FOR i=205 TO 5 STEP -2: Sprite i,50
40 OVER 0:  REMark Normal printing again
45 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
50 DEFine PROCedure             Sprite(x,y)
55 LOCal c: c=7: REMark colour of sprite
60 BLOCK 2,10,x,y,c:BLOCK 10,2,x-4,y+4,c
65 PAUSE(1):        REMark To adjust speed
70 BLOCK 2,10,x,y,c:BLOCK 10,2,x-4,y+4,c
75 END DEFine
```

Rounding Up

A German reader, Wilhelm Albrecht, has noticed that the

Listing 1

```
100 CLS
110 n=4.5*18.05
120 PRINT Round(2,n)
130 STOP
140 REMark ^^^^^^^^^^^^^^^^^^^^^^^^^^^
150 DEFine FuNction          Round(p,n)
160 REMark p = No. of decimal places
170   LOCal m
180   m = INT(n*10^(p+1)+.5)/10^(p+1)
190   RETurn INT(m*10^p+.5)/10^p
200 END DEFine
```

standard 'rounding off' routine behaves erratically when the digit being dropped is exactly 5. The SuperBasic formula for rounding a number N to p decimal places is: INT(N*10^p + 0.5)/10^p, and Mr Albrecht has discovered that when N = 4.5*18.05 rounding to two places the result (81.225) is rounded down to 81.22 instead of up to 81.23 as he would like. He suggests adding 0.5001 instead of 0.5, but while this corrects this particular problem it would create new ones, rounding up numbers like 18.7499999 when it ought to round them down.

Like Mr Albrecht I was taught to always round up under these circumstances, but here in North America the convention — and that is all it is — is to round a 5 to an even number. The argument is that this will randomly result in an equal number of up and down roundings, so that the total of a large number of rounded numbers will be more accurate than the total of a set produced by the European method.

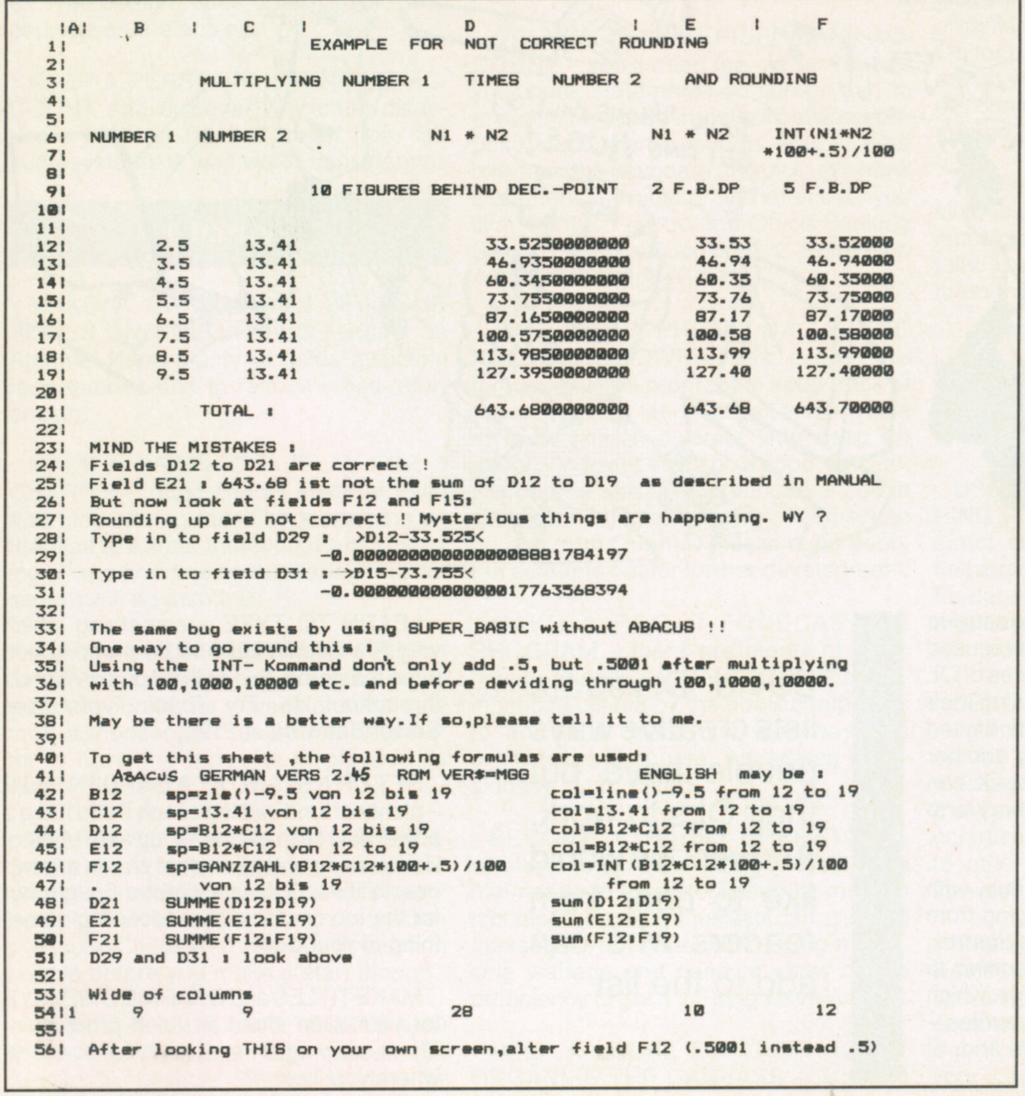
Binary

The QL, like all digital computers, does all its calculations in binary, converting decimal numbers back and forth as required; often there isn't an exact conversion, but Qdos does live up to Sir Clive's promise of seven digit accuracy, and Psion have been able to build twice that into their programs. Mr Albrecht's problem arises from this binary to decimal conversion error. The actual result of the QL's calculation of 4.5*18.05 is 81.22499994039536 which Qdos correctly prints as 81.225 to seven digit precision. Note that this is accurate to 9 figures. (How did I find this out? Try PRINT 4.5*18.05 - 81.225 and you will see the difference.) The situation is similar in Abacus except that the arithmetic is "double precision". In both cases the INT function operates on the number actually stored in the QL's memory and not the rounded value however meaningless the last 7 or 8 digits may be.

When presented with a true exact 5 to round the formula does always round up, so Mr Albrecht can get the behaviour he wants by a two step rounding: round first to three places and round the result of that to two; this still may fail very occasionally, but should be much more reliable. **Listing one** gives a SuperBasic FuNction that will do this. But, while I was writing and testing this, I discovered another problem, there is a limit to the argument of the INT function of +/-2147483647; values outside this range generate errors both in SuperBasic and Abacus. It is thus possible to cause an overflow when dealing with large numbers or when rounding to a large number of decimal places. In SuperBasic this shouldn't be a problem if you remember the normal seven digit precision, but it could be a limitation for accountants using Abacus for dealing with multi-million budgets!

Uncertainty

The real point it that there is no correct answer to this problem, and in fact Qdos's behaviour is probably better (more random) than either convention. The fact that you want to round implies that the digits being lost are innaccurate or unreliable (if they originate in a measurement) or irrelevant (if they result from an exact calculation). You have to accept that some uncertainty still exists in the last figure and to minimise this you must work at least one more significant digit than you really want, rounding only at the very end.



```
 |A|  B    |   C   |              D              |    E     |     F
  1|                  EXAMPLE  FOR  NOT  CORRECT  ROUNDING
  2|
  3|          MULTIPLYING  NUMBER 1   TIMES   NUMBER 2    AND ROUNDING
  4|
  5|
  6|  NUMBER 1  NUMBER 2  .          N1 * N2            N1 * N2    INT(N1*N2
  7|                                                              *100+.5)/100
  8|
  9|                   10 FIGURES BEHIND DEC.-POINT   2 F.B.DP     5 F.B.DP
 10|
 11|
 12|        2.5     13.41                33.5250000000     33.53    33.52000
 13|        3.5     13.41                46.9350000000     46.94    46.94000
 14|        4.5     13.41                60.3450000000     60.35    60.35000
 15|        5.5     13.41                73.7550000000     73.76    73.75000
 16|        6.5     13.41                87.1650000000     87.17    87.17000
 17|        7.5     13.41               100.5750000000    100.58   100.58000
 18|        8.5     13.41               113.9850000000    113.99   113.99000
 19|        9.5     13.41               127.3950000000    127.40   127.40000
 20|
 21|          TOTAL :                   643.6800000000    643.68   643.70000
 22|
 23|  MIND THE MISTAKES :
 24|  Fields D12 to D21 are correct !
 25|  Field E21 : 643.68 ist not the sum of D12 to D19  as described in MANUAL
 26|  But now look at fields F12 and F15:
 27|  Rounding up are not correct ! Mysterious things are happening. WY ?
 28|  Type in to field D29 :  >D12-33.525<
 29|                   -0.00000000000000008881784197
 30|  Type in to field D31 : >D15-73.755<
 31|                   -0.0000000000000017763568394
 32|
 33|  The same bug exists by using SUPER_BASIC without ABACUS !!
 34|  One way to go round this :
 35|  Using the  INT- Kommand don't only add .5, but .5001 after multiplying
 36|  with 100,1000,10000 etc. and before deviding through 100,1000,10000.
 37|
 38|  May be there is a better way.If so,please tell it to me.
 39|
 40|  To get this sheet ,the following formulas are used:
 41|    ABACUS  GERMAN VERS 2.45   ROM VER$=MGG              ENGLISH  may be :
 42|  B12     sp=zle()-9.5 von 12 bis 19          col=line()-9.5 from 12 to 19
 43|  C12     sp=13.41 von 12 bis 19              col=13.41 from 12 to 19
 44|  D12     sp=B12*C12 von 12 bis 19            col=B12*C12 from 12 to 19
 45|  E12     sp=B12*C12 von 12 to 19             col=B12*C12 from 12 to 19
 46|  F12     sp=GANZZAHL(B12*C12*100+.5)/100     col=INT(B12*C12*100+.5)/100
 47|  .          von 12 bis 19                          from 12 to 19
 48|  D21     SUMME(D12:D19)                      sum(D12:D19)
 49|  E21     SUMME(E12:E19)                      sum(E12:E19)
 50|  F21     SUMME(F12:F19)                      sum(F12:F19)
 51|  D29 and D31 : look above
 52|
 53|  Wide of columns
 54|1     9         9                  28                10          12
 55|
 56|  After looking THIS on your own screen,alter field F12 (.5001 instead .5)
```

30 THINGS TO DO WITH A QL

In the same way that many expensive high-spec cameras are only ever used for taking holiday snaps, scores of QL computer owners are only using their superb machines to carry out a limited range of work. It is worth taking another look at some of the many tests the QL can perform – and here are thirty of them just to start the ball rolling.

PLAY GAMES – the obvious one, with plenty of software available ranging from the many Quanta freebies such as *Startrek, Dungeons, Duckshoot* or *Mastermind* to commercials such as *Microbridge* which could set you back £35. Old favourites – but still unbeaten – are *Scrabble* and, of course, *Psion Chess.*

David Drysdale lists creative ways people have put their QLs to work. And yes, we would like to hear from readers who can add to the list.

LEARN TO TYPE – something which very few people learn to do properly yet, once mastered, this skill will pay dividends throughout life. Try *Touch Typist* from Sector Software at £12.

SET UP A STAND AT A GARDEN FETE – plenty of fund-raising scope if you have a power source to get the QL up and running. Horoscopes and biorhythm charts are two possibilities. Software can be bought but for the fun occasion try concocting something of your own.

MAKE TITLES and GRAPHIC DISPLAYS for your slide show or video production. There are programs, at every level, everywhere you look.

GET TO GRIPS WITH ARCHIVE – it really is an amazing piece of software once you get used to the procedures and Sedit screens. All will be revealed in Mike O'Reilly's book *Database Management on the QL* if you are lucky enough to find a copy. Or try the *Archive Tutor* program from TK Computerware.

PLAN YOUR HOUSEHOLD BUDGET FOR THE YEAR – there's no finer way to learn the wonders of the Abacus spreadsheet. Once your budget is up and running ask some 'What if?' questions about your personal spending. The spreadsheet's lightning recalculations may throw up some surprises.

MAKE A COMPUTERISED CHRISTMAS CARD LIST – on Archive, of course. You can bring it on screen each year and update it. Remember to record gifts sent as it is all too easy to send the same type of gift year after year automatically.

BECOME A PRINTER – the variety of desk top publishing programs offered in Sinclair QL World should enable you to print like a professional. It won't happen however without a printer's trained eye for type design. Look out for books and courses on the subject.

KEEP A PERSONAL SPENDING ACCOUNT and discover why cash disappears so fast. *Bank Account* from SD Microsystems is well worth considering.

Charity

MAKE A CARD INDEX OF YOUR SLIDES AND NEGATIVES – again on Archive. Keep plenty of fields available for searches and to reduce cross-referencing.

MAKE A GRAPHIC DISPLAY FOR A CHARITY – most charity stalls at fetes and fund raising fairs like something on the stall to attract attention. Ever changing charts and graphs appearing on your screen will do just that.

MAKE AN ADVERT DISPLAY FOR A SHOP WINDOW – if you can stand being without your QL for a week or so at sale time, a shopkeeper friend would be grateful to have the bargains displayed this way.

HELP A CLUB WITH ITS MEMBERSHIP LIST – clubs and small charities are always grateful for a computerised membership list that can give up-to-date printouts. Check with the Data Protection people before you make a start though.

MONITOR YOUR UNIT TRUSTS AND PRIVATISATION SHARES – *Investment Monitor* by Michael Slatford is ideal for this. (TK Computerware £20). And don't forget *Easel* for share performance graphs.

DEVISE A STOCK CONTROL SYSTEM FOR YOUR FREEZER – with shopping lists and an updatable printout to fix on the freezer door with a fridge magnet.

MAKE A FAMILY MEAL PLANNER – with rotational recipes, and save the cook's constant worry about what meal to serve.

GO ONLINE TO PRESTEL – you'll need a modem of course; look out for the Tandata three stack modem which can sometimes be obtained from Quanta at an advantageous price. There are hundreds of information providers plus computer clubs, an agony aunt, an AIDS information service and a Gay line.

Prestel

BOOK A CHEAP HOLIDAY – again through Prestel. Many travel agents have cancellations and last minute bargain offers and Prestel is the ideal medium for picking them up.

OPEN A COMPUTERISED BANK ACCOUNT – you'll need the modem to see your bank statements on screen and to make instant cash transfers between deposit and current accounts. Pay all your bills from the keyboard and pick up a bank statement from your printer whenever you like. Contact: Home and Office Banking Service, Bank of Scotland, 2 Robertson Ave, Edinburgh EH11 1PZ.

TIME YOUR PHONE CALLS AND KEEP BILLS DOWN – lots of scope here for designing a program to keep track of the time of day, length of call and the cost of each unit. A discreet little beep for every 50p spent might be a good idea too. For people who are not too advanced in programming a ready made call timer can be had from Quanta. This is quite good, but screams out for further development.

MAKE A POOLS FORECASTING PROGRAM – there are plenty of basic coupon marking systems around including those issued by the pools companies to act as a basis. You can buy one readymade of course. Advertised occasionally in QL World.

BECOME A RADIO OPERATOR – wireless telegraphy using morse code has not been totally eclipsed by modern techology. Radio enthusiasts can still send messages round the world using minuscule wattage and need to show morse proficiency to get a full long wave licence.

HELP A LOCAL ELECTION AGENT ORGANISE THE CANVASS – lists and more lists, of course, both for pinning on the walls and handing out to the canvassers. Archive, especially with a disc drive, should be able to handle it.

BECOME A MUSIC COMPOSER – the QL is possibly not the best computer on the market for this purpose but Miracle System's Midi music interface is said to be having software redevelopment and could be on the market soon.

CHECK YOUR WINNINGS – write a program for checking betting shop winnings. Programs can be bought of course. Especially useful if you bet by phone and need the information instantly available.

BECOME SELF EMPLOYED – Quill, Abacus and Archive have all the capacity needed to handle your paperwork. For a one-man business *General Ledger* from SD Microsystems makes accounting a doddle. If you are on the dole don't forget the Enterprise Allowance. Freebie help is available too in deciding what business to start.

MAKE A DAY PLANNER WITH AUDIBLE REMINDERS – no reason why this job which is done by electronic pocket organisers should not be done on a static QL for people who work from home. You are on your own with this one however as no commercial program seems to be around.

DESIGN A CALORIE COUNTER – slimmers can find plenty of calorie data to input into a program to count up the day's tally. There is probably a program around to do this but I haven't spotted it.

Shopping

USE YOUR MODEM TO GO SHOPPING – Compustore Online provides a direct cut price shopping service with delivery to your door (ring 0753-868191 for details). Prestel also lists businesses that supply everything from vitamin tablets to home-delivered flowers.

SEND A TELEX ACROSS THE WORLD – again through Prestel. Their Telex Link service allows your computer to output a message that can be delivered anywhere that the teleprinter network reaches. You can also send freebie messages directly to other Prestel users by the mailbox service.

Finally, as an extra suggestion, how would you like to ADD TO THIS LIST. The pooled knowledge of readers of QL World could probably increase it a hundredfold. The Editor, I am sure, would like to see your ideas.

DIY TOOLKIT

Simon Goodwin shows how to transfer text between console windows, and demonstrates the CLIP functions for the QL and Thor.

I shall now reveal the inner workings of the CLIP functions, introduced last issue, suggesting more uses as I explain the code and its ramifications.

For years I have wanted a 'clipboard' that can read characters from the QL screen, and type them into any other–a kind of OCR for QL Windows, to save me from ever having to read and re-type what the QL already 'knows'. At last I've done it, and the listings show how. After years of conditioning I couldn't resist the tradition of QL puns, and dubbed it *Qlipboard.*

Qlipboard can scan Qdos or Argos to find and identify all screen and console windows. It reads special attributes like character-size and fonts from the window. With this information the scanner can read or test lines or individual characters, and scoot around any window.

The editor lets you check or change the clipped text; it can be called up at all times, in its own pop-up window. Text is held in string pages of up to 32K; the routines listed share data space between ten pages.

The only important thing about *Qlipboard* is to remember to load it. You will keep new finding ways to use it. I mis-typed a line-number while programming it, and lost a long PRINT line from the listing. I still had the latest version in the output window, so I just clipped the characters and fed them straight back into ED.

Squirt

To copy various bits into one *SuperBasic* procedure, LIST the lines containing the bits, clip them up and squirt them back into Basic somewhere else. It's similarly easy to squirt text into *Quill, Archive,* editors, emulators and just about any other task that lets you type lines into it.

Qlipboard was written in response to years of suggestions from QL users. It is not a replacement for existing facilities, but can be an alternative. Often it fills a gap in the QL armoury, allowing figures, code and text to be transfered accurately between tasks. As well as text, it can easily transfer hex data, Archive procedures, *Abacus* formulae, SuperBasic or other programs, *Easel* labels and so on...

Qlipboard can envigorate slothful editors. Psion and Metacomco block copy operations are painfully slow and limited. It is often easier to clip characters from the screen, go to the new place directly, and spit the text back into the file.

Qlipboard does not obscure the screen while scanning, but it only reads one screenful at a time. When you've scanned all you can see without satisfaction you must page the task's display and scan again. Swapping can use a single keypress. The editor can cope with up to 32K, so you can extract 500 lines or more, a screen at a time.

Qlipboard can pick up past lines from the command window, but *Super Toolkit's* Alt Enter may remember more, depending on line length. It works if the text is still visible, even when Alt Enter would be confused by single key choices, arrows or deletions, or ESC is pressed accidentally on *Minerva,* denying the line to ALT Enter. I use both, and find Qlipboard most useful when past attempts warn me that Alt Enter has reached its limit.

CLIP% and CLIP$ can recognise user-defined graphic characters as well as text. They are useful accessories for games writers and converters who miss Spectrum functions like SCREEN$ and ATTR, often used to detect collisions in interactive animated sequences.

The Clip functions are also suitable for quick printouts of text displays. Now you can scan characters from the screen, and translate their codes before printing, if need be. Most printers give best results from a stream of Ascii text. Shaded bit-image graphics tend to be slow and streaky by comparison.

Routines to search fonts quickly and pack any character from the screen into a nine byte font pattern are imbedded in **Listing two.** You could scan graphics into a font, and PRINT two colour patterns of any width, up to nine (or 18) pixels high.

Qlipboard recognises four extra keystrokes while other tasks are using the keyboard. The codes are configurable by editing; the names are stored in Ascii text in the compiled task, with easily-recognised labels like '*=EDIT' after each decimal code number. Even after compilation you can change the codes with utilities like *The Editor, Spy or FEDIT,* by patching new digits into the file. You could use LBYTES, PEEK, POKE and SEXEC, but SEARCH_MEMORY would save time. Just replace the stored codes with your own choices, 0-255.

Codes

I prefer to assign the Qlipboard to 'control' codes, 0-31, or non-display codes 192-255, as most tasks ignore them. This discussion assumes the keys set in Listing 1.

Control E calls SHOW_BUFFER, the clipboard Editor, saving the previous screen if RAM permits, before displaying various options and the current clipped text. You'll need to change this code if you need *Spellbound* turned on at the same time, as Control E calls up its examples.

Control P types the next line from the clipboard page into the current task, as if you typed them yourself. The first line is 'next' after the last. The computer burps briefly if it can't find anywhere to put the characters.

Each page can hold up to 32K of text, about 400 lines, but 3K is usually plenty. You can have many pages on one board, and this helps me keep text from different windows apart. This program allocates ten pages, unless you change line 1140 in **Listing one**. To swap between pages, call up the editor, press P then enter a digit, 0-9

Control W highlights each window in turn, so you can select a default. The highlight changes the colours on the screen, but Qlipboard can still recognise them all. Change "*=HIGHLIGHT" (Listing one, line 1330) if you don't like my choice; the range is 0 to 255, and the effect depends entirely on the background colours. 7 and 2 are good alternatives; 17, 34 and 222 are interesting. Press Space to highlight another window, Enter to pick this one, or Esc to escape changes.

Control G turns on a 'clip cursor' in the default window so that you can clip text from it. If the default is not the window you want, press Enter or Esc to turn off the clipper and pick another window with Control W.

You may clip text while a window is highlit, but you are meant to use Control W to pick a default. Visibility is improved and the cursor size is updated if you press Enter to confirm the selection, then Control G to Get text from the window while it is displayed in its usual colours.

Qlipboard has its own cursor; you specify the characters that you want to clip by moving the cursor over them. This is controlled by the DIALOGUE routine, which recognises a score of auto-repeating keystrokes.

Cursor

The four arrow keys move the cursor one space at a time, in character-sized steps. You can't move the cursor outside the current window. Alt-Left and Alt-Right move quickly to the start or end of the line.

Shift keys control reading. Shift Right reads the character under the cursor, if possible, and adds it to the current clipboard page, then moves the cursor one place to the right. Control Left and the Thor's grey arrow key move to the left and remove one character from the buffer. Control E displays the contents of the current Qlipboard page.

Shift Enter scans all characters to the end of the line, and puts them on the clipboard, followed by an Enter character. Shift Space is similar but adds a space at the end of the line. In either case the cursor ends up at the start of the next line, if any.

Last month's demonstration routine stripped out duplicate spaces in a line, so that text clipped from a *Quill* screen was not plastered with spaces added by justification or centring. Sometimes margins, tabs and 'extra' spaces are significant, so SCAN_LINE in this month's listing illustrates verbatim scanning, and recognises extra spaces.

The disk Qlipboard has both routines, so you can retain or ignore spaces after the first in each group. If the scanned line consists of nothing but spaces it is condensed to a single Enter character, which gives the same result more economically. Sometimes the default cursor grid does not match the position of text; you may find this if you try to read pixel-positioned labels from diagrams. If text has scrolled or been placed with CURSOR you might have to nudge the cursor to get it right on

```
1000 REMark QL QLIPBOARD by Simon N Goodwin and Phil Spink
1010 REMark Version 6.7; Exts: TURBO TOOLKIT 2, QBASE, CLIP
1020 IMPLICIT% w,c,key,windo :DATA_AREA 32 :REMark Needs #1
1030 :
1040 REMark **** QL & Thor keyboard-specific stuff
1050 sv_arbuf=SYSBASE+138 :REMark Key store address
1060 sv_keyq=SYSBASE+76 :REMark Pointer to key queue
1070 REMark Keycodes are ASCII for ease of patching
1080 pickey%="23  *=Windows" :REMark Control W
1090 stuffkey%="16  *=Paste line" :REMark Control P
1100 clipkey%="7   *=GET text" :REMark Control G
1110 edkey%="5   *=EDIT" :REMark Control E for Editor
1120 REMark **** Window & page configuration
1130 underline%=0 : getsize=1 : page%=0 : max_windows%=80
1140 maxpage%=9 : REMark MAXPAGE%+1 pages
1150 IF COMPILED : maxline%=2300 : ELSE maxline%=120
1160 :
1170 REMark **** Grab space for data & window details
1180 DIM xshift%(max_windows%),yshift%(max_windows%)
1190 DIM height%(max_windows%),width%(max_windows%)
1200 DIM base(max_windows%),enter$(1),space$(1),newline$(1)
1210 DIM Esc$(1),Line$(maxline%),Page$(maxpage%,maxline%)
1220 enter$=CHR$(10) : space$=CHR$(32) : Esc$=CHR$(27)
1230 newline$="\" :REMark Symbolic Newline, suits IO.EDLIN
1240 :
1250 WINDOW 512,256,0,0
1260 WINDOWS :FIND_MYQ :Line$="Qlip 6.7\":cpos%=1:windo=0
1270 REPeat indefinitely : PICK_WINDOW : POLL
1280 :
1290 DEFine PROCedure PICK_WINDOW
1300 LOCal paper%,selected,prior
1310 REMark Result is global WINDO {0 .. global CH% - 1}
1320 WINDOW 512,256,0,0 : selected=0 :REMark Clip Window
1330 paper%="4   *=HIGHLIGHT" :REMark XOR 222, 2, 17, 34
1340 WINDOWS
1350 IF windo<win% :first=windo :ELSE first=0 :REMark Guess
1360 REPeat tilpik
1370   FOR windo=first TO win%-1
1380     SHOW_WINDOW windo,paper%
1390     DIALOGUE windo,selected
1400     SHOW_WINDOW windo,paper%
1410     IF selected<0 AND windo<>first
1420       SHOW_WINDOW first,paper%
1430       SHOW_WINDOW first,paper%
1440       windo=first :REMark Reminisce
1450     END IF
1460     getsize=(selected<>0) :REMark New CSIZE?
1470     IF selected :EXIT tilpik
1480   END FOR windo
1490   first=0
1500 END REPeat tilpik
1510 END DEFine PICK_WINDOW
1520 :
1530 REFERENCE happy
1540 DEFine PROCedure DIALOGUE(win,happy)
1550 LOCal key,endx%,endy%,x%,y%
1560 REPeat getspec
1570   IF getsize
1580     stepx%=PEEK_W(base(win)+38)
1590     stepy%=PEEK_W(base(win)+40)
1600   END IF
1610   CSIZE (stepx% DIV 4)-1,stepy%>19
1620   POKE_W my_window+38,stepx% : UNDER underline%
1630   POKE_W my_window+40,stepy% : SET_MYQ
1640   POKE_L my_window+46,PEEK_L(base(win)+46)
1650   POKE_L my_window+50,PEEK_L(base(win)+50)
1660   endy%=yshift%(win)+height%(win)-stepy%
1670   endx%=xshift%(win)+width%(win)-stepx%
1680   x%=xshift%(win) : y%=yshift%(win) : REMark Top Left
1690   REPeat seek
1700     CURSOR x%,y%:CURSOR_ON #1!:key=CODE(INKEY$(#1,-1))
1710     SELect ON key
1720       =192,194 :REMark Move left and maybe delete
1730         IF key=194
1740           IF LEN(Line$):Line$=Line$(1 TO LEN(Line$)-1)
1750         END IF
1760         IF x%-stepx% >= xshift%(win) : x%=x%-stepx%
1770       =200,204 :REMark Move right and maybe clip
1780         IF key=204 : Line$=Line$ & CLIP$(#1)
1790         IF x%+stepx% <= endx% : x%=x%+stepx%
1800       =208 :REMark Move up one character line
1810         IF y%-stepy% >= yshift%(win) : y%=y%-stepy%
1820       =216 :REMark Move down one line
```

```
1830         IF y%+stepy% <= endy% : y%=y%+stepy%
1840       =197 :REMark Pixel shift left
1850         IF x%>xshift%(win) : x%=x%-1
1860       =205 :REMark Pixel shift right
1870         IF x%<endx% : x%=x%+1
1880       =213 :IF y%>yshift%(win) :y%=y%-1 :REMark Bit up
1890       =221 :IF y%<endy% :y%=y%+1 :REMark Down a bit
1900       =193 : x%=xshift%(win) :REMark Zoom to start/end
1910       =201 : x%=endx% - (width%(win) MOD stepx%)
1920       =252,254 : SCAN_LINE
1930       =63 : BEEP 500,0 : temp%=CLIP%(#1) : REMark "?"
1940         REPeat Intel : IF NOT BEEPING : EXIT Intel
1950         IF temp%<0 : BEEP 1000,-temp%
1960       =232,edkey% : REMark F1/Edkey = Editor & options
1970         CURSOR_OFF : SHOW_BUFFER : NEXT getspec
1980       =10,27,32 : REMark Enter confirms, SPACE = Next
1990         happy=(key=10)-(key=27)
2000         CURSOR_OFF : EXIT getspec
2010     END SELect
2020   END REPeat seek
2030 END REPeat getspec
2040 END DEFine DIALOGUE
2050 :
2060 DEFine PROCedure SCAN_LINE
2070 REPeat scan
2080   IF LEN(Line$)<maxline% : Line$=Line$ & CLIP$(#1)
2090   IF x%+stepx% <= endx% : x%=x%+stepx% :ELSE EXIT scan
2100   CURSOR x%,y% : CURSOR_ON #1!
2110 END REPeat scan
2120 IF key=252:Line$=Line$&space$ :ELSE Line$=Line$&enter$
2130 IF y%+stepy% <= endy% :x%=xshift%(win) :y%=y%+stepy%
2140 END DEFine SCAN_LINE
2150 :
2160 DEFine PROCedure SHOW_BUFFER
2170 REMark LOCal mem; DEALLOCATE MEM after BREAK tests
2180 vdu_lines=256 :vdu_start=131072 :REMark Screen 0
2190 mem=ALLOCATION(vdu_lines*128) :REMark Screen buffer
2200 IF mem>0 :MOVE_MEMORY vdu_start TO mem,vdu_lines*128
2210 CLS :UNDER 0 :INK 4 :CSIZE 1,0 :AT 1,0 :PRINT Line$
2220 REPeat opt
2230   SET_MYQ : OPTIONS editor : IF editor=0 : EXIT opt
2240   XCHANGE Line$,enter$,newline$:CSIZE 1,0:INK 7:AT 1,0
2250   CLS 3:CLS 2 :Line$=EDIT$(#1,Line$,maxline%) :cpos%=1
2260   XCHANGE Line$,newline$,enter$
2270 END REPeat opt
2280 IF mem>0
2290   MOVE_MEMORY mem TO vdu_start,vdu_lines*128
2300   DEALLOCATE mem
2310 END IF
2320 END DEFine SHOW_BUFFER
2330 :
2340 REFERENCE txt$
2350 DEFine PROCedure XCHANGE(txt$,old$,new$)
2360 LOCal p%
2370 REPeat swap
2380   p%=old$ INSTR txt$ : IF p%=0 : EXIT swap
2390   txt$(p%)=new$
2400 END REPeat swap
2410 END DEFine XCHANGE
2420 :
2430 DEFine PROCedure POLL
2440 LOCal key,keyq,pos%,not_space
2450 REPeat doze
2460   key=PEEK_W(sv_arbuf)
2470   IF key=pickey% : SET_MYQ : EXIT doze
2480   IF key=edkey% OR key=clipkey%
2490     keyq=PEEK_L(sv_keyq)
2500     REPeat clipk
2510       IF key=edkey% : SHOW_BUFFER : EXIT clipk
2520       DIALOGUE windo,not_space:IF not_space:EXIT clipk
2530     END REPeat clipk : REMark Await ENTER/ESC
2540     IF NOT windo_offset : POKE_L sv_keyq,keyq
2550   END IF
2560   IF key=stuffkey% AND LEN(Line$)>0
2570     POKE_W sv_arbuf,0 : REMark Cancel auto-repeat
2580     nextpos%=enter$ INSTR Line$(cpos% TO LEN(Line$))
2590     IF nextpos%>0
2600       STUFF Line$(cpos% TO cpos%+nextpos%-1)
2610       cpos%=nextpos%+cpos%
2620       IF cpos%>LEN(Line$) : cpos%=1
2630     ELSE
2640       STUFF Line$(cpos% TO LEN(Line$)) : cpos%=1
2650     END IF : NEXT doze : REMark Dream furiously
```

top of a character, so that CLIP can recognise it. Press Shift and Alt at the same time as any arrow to position the cursor precisely in the window.

The smallest text normally has one blank column on the left of each character, and one blank row at the top. You can test a position by pressing "?"; Qlipboard tries to read the character under the cursor, and bleeps once if successful; two beeps mean that the pattern was not recognised in either font for that window; you may need to move the cursor slightly.

If the window uses more than one CSIZE you may need to over-ride the default for Control G, which is the most recent size set. Call up the editor and type X to change the width, or Y to set the height. A cursor appears above the old value. Standard values are 6, 8, 12 and 16, for **X**, and 10 or 20 for Y. The new setting persists until you change it or select a window with Control W and Enter.

Once you have finished clipping text, press Enter or ESC to turn off the clip cursor. If you forget to turn off the scanner you may confuse it with another task, and have to skip past it when swapping cursors later. Control P will not work while the clip cursor is active, but you can call the Editor directly from the clipper by pressing Control E, or F1 for 'help'.

Taskforce

Users running *Taskforce* or other tools that use standard Qdos windows will find that Qlipboard swaps keyboard queues automatically. Pick the task's cursor and press Enter after loading, so Qlipboard can identify its own queue and get started. POKE SYSBASE+146,9 if you prefer to use TAB instead of Control C.

Qlipboard works very well with *Taskmaster 2.35* despite its quirks, like nine pixel tall names in the 'Files' display. You can clip from all the windows, as long as characters and border lines don't overlap. Note that the 'Info Box' display puts the date on line 6 of the Qlipboard editor, and the 'Notebook' accepts control codes.

Systems with 'extended' console software must rely on their own devices when it comes to queue swapping. Qlipboard needs to be able to move a cursor over the windows of other tasks, but *Qpac-2* prevents this by 'locking' windows so you cannot see through them.

This makes trouble for DIALOGUE (1540), even when 'EXEP""U' ensures an unlocked window. Qlipboard takes careful account of QPac's changes to Qdos channel usage; the editor and 'stuffer' work fine, and the Clip functions happily read a task's own windows with Qpac loaded. It is a pity that current versions separate task displays in such an inflexible way.

The Thor is more accommodating, if you POKE SYSS_VARS +133, -1 before starting, to discourage Argos from grabbing the display and cursor at odd moments. Press the Sys Req key to swap keyboard queues as necessary; two taps returns to *Xchange* from Qlipboard, with Thor Basic running alongside.

The Thor's PC keyboard gives standard codes whether or not you press Shift with Space or Enter. This eases typing but means Argos users must press Control Enter or Control Space to scan lines; these return the same codes as their shifted QL equivalents. Qdos treats Control Enter like Enter, or Control J, while Control Space corresponds to the Thor's Break key.

Qlipboard clones the fonts in each window, and uses them in the editor, so you see the same characters in the editor as you saw on the task display, as long as the task has not gone, taking the font with it! This feature suits the multi-fonted Thor XVI, and also helps to decode horrid games fonts. If you must know what it really says, clip the lines, pick a standard window, and consult the Qlipboard , to see the text in normal letters and colours.

You could add code to load Psion's subscripts font, or others, when several fonts are used in a window and you can't be sure which is current. MODQUILL, on Quanta's *Psion 1* library disk, should help, but I have been scolded for 'doing it all', so I'll stop there.

The Qlipboard editor uses CSIZE 1,0 text to suit all Qdos fonts. The full window can be reduced for a tv, but Listing one uses the same window for clipping as for editing. The menu needs MODE 4, unless you change the column numbers in OPTIONS or CHOOSE.

TurboQuill Plus uses fixed internal fonts which match the Thor default, rather than the characters indicated by QL channel definitions. The letter f is narrower than on the QL, and the second column of k is lower. This means that CLIP% and CLIP$ cannot recognise the letters k or f, or some non-Ascii characters, from QL TurboQuill Plus displays. The Thor is fine, as its fonts always match the screen.

Minerva

A keen Minerva user could fix this by extracting the TurboQuill characters with the A1 return from CHAXTOP, which returns the font pattern for any character on screen. Build up and set new default system fonts, if you are determined to Qlip art from TQ+ without nagging doubts!

Type X, Y or U to change the character options after calling up the editor. Set UNDER 1 to read underlined text, or UNDER 0 to distinguish underscores from spaces.

Normally Qlipboard picks up the character-size from the window channel definition; thus it is not limited to the eight standard CSIZEs. In fact it copes with single or double height text with any Y step 0 to 49, and X steps from 4 to 19. If windows have

```
2660   END IF
2670   SUSPEND_TASK 15 : REMark Encourage other tasks
2680 END REPeat doze
2690 END DEFine POLL
2700 :
2710 DEFine PROCedure STUFF(text$)
2720 LOCal status%,pos%
2730 pos%=1
2740 REPeat q_all
2750   status%=QUEUE%(text$(pos% TO))
2760   IF status%=0 : EXIT q_all
2770   IF status%=LEN(text$) : BEEP 500,127 : EXIT q_all
2780   IF status%<0 : status%=-status% : REMark Stuffed
2790   pos%=1+LEN(text$)-status%
2800 END REPeat q_all
2810 END DEFine STUFF
2820 :
2830 DEFine PROCedure SHOW_WINDOW(w,c)
2840 LOCal half%
2850 OVER -1 : IF w<=511
2860   BLOCK width%(w),height%(w),xshift%(w),yshift%(w),c
2870 ELSE
2880   half%=w DIV 2 :REMark Avoid Sinclair's BLOCK 512 bug
2890   BLOCK half%,height%(w),xshift%(w),yshift%(w),c%
2900   BLOCK half%,height%(w),xshift%(w)+half%,yshift%(w),c
2910 END IF : OVER 0
2920 END DEFine SHOW_WINDOW
2930 :
2940 DEFine PROCedure WINDOWS
2950 LOCal chbas,chmax,ch,chp,addr
2960 REMark Fill arrays etc. with 0..WIN%-1 window details
2970 windo_offset=CHANNEL_ID(#1) :REMark Check channel
2980 windo_offset=windo_offset-INT(windo_offset/2^16)*2^16
2990 my_window=CHBASE(#1) : chbas=PEEK_L(SYSBASE+120)
3000 channel0_driver=PEEK_L(PEEK_L(chbas)+4)
3010 windo_offset=chbas+4*windo_offset
3020 windo_offset=my_window-PEEK_L(windo_offset)
3030 chmax=PEEK_W(SYSBASE+114) : win%=0
3040 FOR chp = chbas TO (chbas+chmax*4) STEP 4
3050   IF PEEK(chp)<>255
3060     addr = PEEK_L(chp)
3070     IF addr+windo_offset=my_window:NEXT chp:EXIT chp
3080     IF PEEK_L(addr+4)=channel0_driver
3090       addr = addr+windo_offset : REMark Skip extras
3100       base(win%) = addr
3110       width%(win%) = PEEK_W(addr+28)
3120       height%(win%)= PEEK_W(addr+30)
3130       xshift%(win%)= PEEK_W(addr+24)
3140       yshift%(win%)= PEEK_W(addr+26)
3150       IF win%<max_windows%
3160         win%=win%+1
3170       ELSE EXIT chp
3180       END IF : REMark Room for details
3190     END IF : REMark Something like #0
3200   END IF : REMark Open channel
3210 END FOR chp :REMark WINDO_OFFSET, WIN%, MY_WINDOW set
3220 END DEFine WINDOWS
3230 :
3240 DEFine FuNction CHAN_ID(num%)
3250 RETurn CHANNEL_ID(#num%) : REMark TURBO Toolkit
3260 REMark DIY TK: RETurn BPEEK_L(BPEEK_L(48)+num%*40)
3270 END DEFine CHAN_ID
3280 :
3290 DEFine PROCedure FIND_MYQ
3300 PRINT "Qlipboard:";
3310 REPeat snoop
3320   CURSOR_ON #1! : Line$=INKEY$(#1,-1)
3330   myq=PEEK_L(sv_keyq) : IF myq : EXIT snoop
3340 END REPeat snoop : CURSOR_OFF
3350 END DEFine FIND_MYQ
3360 :
3370 DEFine PROCedure SET_MYQ
3380 CURSOR_ON #1!: IF NOT windo_offset :POKE_L sv_keyq,myq
3390 END DEFine
3400 :
3410 REFERENCE edflag
3420 DEFine PROCedure OPTIONS(edflag)
3430 LOCal k$(1)
3440 STRIP 4 :CSIZE 0,0 :INK 0 :AT 0,0
3450 PRINT " Qlipboard 1990 SNG+PRS "; :STRIP 0 :INK 4
3460 PRINT "  Edit  Page=";page%;"  Quit  RAM=";
3470 PRINT maxline%-LEN(Line$);" ";:AT 0,59:PRINT "Under=";
```

```
3480 PRINT underline%;"  X=";stepx%;"  Y=";stepy%;"  Zap ";
3490 REPeat keys
3500   AT 0,83 : CURSOR_ON #1! : k$=INKEY$(#1,-1)
3510   IF k$="q" OR k$=Esc$ OR k$=enter$:edflag=0:EXIT keys
3520   IF k$=="e" OR k$=space$ :edflag=1 : EXIT keys
3530   IF k$=="u" : CHOOSE 65,underline%,0,1
3540   IF k$=="p"
3550     Page$(page%)=Line$ : CHOOSE 39,page%,0,maxpage%
3560     Line$=Page$(page%) : edflag=1 : EXIT keys
3570   END IF
3580   IF k$=="x" : CHOOSE 70,stepx%,4,19 : getsize=0
3590   IF k$=="y" : CHOOSE 75,stepy%,0,49 : getsize=0
3600   IF k$=="z" : Line$="" : edflag=1 : EXIT keys
3610 END REPeat keys : CURSOR_OFF
3620 END DEFine OPTIONS
3630 :
3640 REFERENCE thing%
3650 DEFine PROCedure CHOOSE(place%,thing%,lo%,hi%)
3660 LOCal entry$(2)
3670 REPeat get_entry
3680   AT 0,place% : INPUT entry$;
3690   IF entry$="" : EXIT get_entry
3700   thing%="0" & entry$ : IF thing%<10 : PRINT " ";
3710   IF thing%>=lo% AND thing%<=hi% : EXIT get_entry
3720 END REPeat get_entry
3730 END DEFine CHOOSE
```

moved, remember that a window of 512x256a0x0 can scan any Qdos text, anywhere.

Listing one shows Qlipboard 6.7, about 16K of interpreted or compiled SuperBasic. It uses *Turbo Toolkit* plus DIY Toolkit's QUEUE, BASE and CHAN functions. I have tried to use 'core' Qdos facilities whenever possible, so programmers should be able to get it working with any common mixture of Toolkits.

Line 1020 holds Turbo compiler directives, like the REFERENCE lines before some definitions. The program can be interpreted without them. Line 1150 adjusts the size of the 'page' buffer; the interpreter needs MAXLINE%=120 to keep EDIT$ or EDLINE$ inside the minimum size of the SuperBasic buffer. Load or enter a long line to expand the area–or compile the program.

Windows

WINDOWS fills four arrays with the position and size of every window, as well as the base address of each channel's definition *(DIY Toolkit Volume C).* Adjust MAX_WINDOWS% on line 1130 if you need more than 80 windows at once.

DIY Toolkit's CHBASE returns the apparent base of channel details as seen by extension code. WINDOW_OFFSET is zero on a standard Qdos machine, but increases as windowing extensions expand the channel table.

If Qdos seems intact Qlipboard takes over cursor swapping, so you don't have to keep typing Control C. FIND_MYQ and SET_MYQ swap queues by PEEKing and POKEing the system keyboard queue pointer, SV_KEYQ, which steps between the queues at the end of console channel definitions, after each Control C or whatever.

I need to scan characters from window O and others like it, ignoring file and serial channels. Line 30BO sifts out channels with the same IO.DRIVER vector as O, which is always the first entry in the Qdos table. They use the same TRAP 3 input and output code as SuperBasic's channel 0, so it seems safe to assume that they define windows.

Second

WINDOWS skips its own default channel, 1, so it takes a second Qlipboard task to scan characters from Qlipboard pages. SuperBasic can't run the scanner and the editor simultaneously, but that seldom matters as you can direct lines from the page anywhere. In 1986 I designed *Turbo's* LINK_LOAD to run several modules of a program in parallel; users must compile each 'thread' separately and declare appropriate globals.

CHAN_ID is used to find the index number of 1 in the Qdos table. Unless you've clobbered window 1 this value is always 1 in interpreted Basic, so RET 2^16 will suffice.

Highlight

PICK_WINDOW uses the WIN% array entries to highlight each window in turn. Unless the cursor size is over-ridden DIALOGUE reads it from the channel being clipped, setting the scanning window 1 to use the same CSIZE and fonts.

Line 1930 puts the second processor in the spotlight for a moment. The main chip twiddles its thumbs till Sinclair's 8049, or the Thor XVI's 6802, has played finished the first note.

CURSOR_ON 1! matches Toolkit 2's CURSEN; likewise CURSOR_OFF corresponds to CURDIS. PAN 0,115 and PAN 0,116 do the same jobs for most QL and Thor Basics, a little more opaquely, unless you read *QL World* in February 1989.

Control E calls SHOW_BUFFER, which may reserve buffer memory with ALLOCATION and DEALLOCATE. Toolkit 2 alternatives are ALCHP and RECHP; DIY Toolkit Volume H has RESERVE and DISCARD. Note that ALCHP returns zero (ERR.OK!) if it **fails** to find the memory requested.

It there's enough free RAM, MOVE_MEMORY saves and restores the default QL screen (32K at 128K). You can do this more slowly with a loop of PEEKs and POKEs, or leave it out if you don't mind overlapping windows.

Editor

As listed SHOW_BUFFER uses Turbo Toolkit's EDIT$, but you could use DIY Toolkit's EDLINE$, or EDLIN from the MCS disk rom. The compiled editor lets you select different lines by moving the cursor, but this always winds back to the start, with CPOS%=1. The editor used by INPUT and EDIT is puzzled by lines containing Enter characters, so XCHANGE translates them into backslashes (top right on a QL keyboard). Thus you can spilt and join scanned lines, even with the rom editor.

STUFF makes determined use of QUEUE% to enter characters; QUEUE% featured in December 1989's *QL World* (DIY Volume Q). You could use Turbo's TYPE_IN instead of QUEUE% to enter characters, but might get stuck or lose text; TYPE_IN is designed for short strings, and does not report errors.

The peppering of calls to CHBASE and SYSBASE should ensure that the code suits all Qdos systems and windowing variations. SYSBASE returns 163840 on most systems, but progams may not run on other machines if they assume that address for the system variables. Thor users can call SYS_VARS instead.

The machine code source in listing two was assembled using HiSoft's *Devpac.* Like all recent DIY Toolkit listings, you can obtain my source, to adapt it or incorporate it into a larger program.

The START routine points A1 at the table of details at the other end of the listing, and calls BP. INIT, the rom vector that adds new functions to SuperBasic. This routine is at the start so it is easy to find after loading the code. It is not used later.

The first big lump is fairly normal parameter-handling, which creates a nine byte buffer on the A7 stack and passes its address, and a channel number, to SD.EXTOP. This points AO at the channel details and calls CHAXTOP, where things get interesting.

First we exclude MEM channels, the only others that allow EXTOP, and check there's room for a cursor at the current window position. Notice how the cursor check processes X and Y words together. The QL rom does this, and similar tricks can add four bytes to four others in one step; beware of overflows.

Next I set up registers, ready to scan a grid six pixels wide and nine high, in various steps to reflect the CSIZE. The table shows the register allocation just after the 'strip' background pixel has been read, at SCANLINE. I qlipped the table from my notes in *Devpac*.

Efficient machine code uses registers as much as possible. Values in registers can be accessed much faster than from memory, but optimum allocation is a big problem, complicated by processor quirks. This is where human programmers can excel over compilers. The scanner uses all 16 68008 registers; indeed, I run out of data registers and use A4 as a word data value, holding the background colour. This works, as long as you don't want to do much arithmetic with the value.

The code to extract each pixel, around SCAN_ROW, can be faster than in my PIXEL% reading function, as it need not combine the two colour bytes; it is enough to align them with the STRIP data, which is also a pair of bytes. The scanner builds a font pattern in the buffer addressed by (A1).

Pattern

LOOKUP is potentially another EXTOP routine, taking a buffer pointer in A1 and returning a code in D1. It looks for the pattern at (A1) in both fonts, checking all eight bits. As soon as a mis-matched line is found, it steps to the next character. This saves a lot of time, especially if the first line differs from the pattern. DESCENDERS checks the channel's flags and ignores the penultimate line of the pattern, if UNDER 1 is indicated.

I analysed the standard rom fonts in case it would be faster to check lines in some less obvious order, and found that the third and seventh lines in either font most often distinguish characters apart. The difference was small: out of 12,800 comparisons for each row of 161 characters, those rows distinguished 90 per cent, versus 85 per cent for row 0.

The optimisation might help measurably when scanning lots of small letters, but I kept the simpler code. At present pixel scanning takes longer than font searching unless the character is far down font 2, so I think the balance is about right.

The table labelled DEFINE indicates the names and addresses of the commands. I put this last to ease name changes. The assembled code will still work as long as changes are confined to the end of the file.

A configurable compiled Qlipboard task is available, with the programs and text from this month and last, and various bells and whistles besides. Ask for DIY Toolkit *Volume S* - for Screen Reading. Single

```
* QL WORLD DIY TOOLKIT - QDOS character CLIP functions
* Version 1.1, Copyright 1990 Simon N Goodwin.
*
start       lea.l     define,a1
            move.w    $110,a2             BP.INIT vector
            jmp       (a2)
*
* CLIP% & CLIP$ code - process channel parameter
*
clip_int    moveq     #-1,d7              Flag integer return
            bra.s     clip_param
clip_str    moveq     #0,d7               Flag string result
clip_param  move.w    $118,a2             CA.GTLIN Vector, ensures
            jsr       (a2)                4 bytes of stack space
            bne.s     bad_exit
            subq.w    #1,d3               One parameter expected
            bne.s     bad_param
            move.l    0(a1,a6.l),d0       Get BASIC channel number
            addq.l    #2,a1               Leave room for one INT
            move.l    a1,$58(a6)          Set maths stack BV.RIP
            mulu      #40,d0              Channel table size
            add.l     $30(a6),d0          Add base offset
            cmp.l     $34(a6),d0
            bge.s     what_chan           Past end of table?
            move.l    0(a6,d0.l),d0
            bmi.s     what_chan           Negative if closed
do_extop    move.l    d0,a0               A0 is channel ID
            lea.l     chaxtop,a2          Address of CH routine
            moveq     #0,d3               ERR.NC if window is busy
            lea.l     -10(a7),a7          Protect some stack space
            movea.l   a7,a1               A1 -> Pattern buffer start
            moveq     #9,d0               SD.EXTOP key
            trap      #3                  Call the device driver
            lea.l     10(a7),a7           Discard pattern (this time)
            move.l    $58(a6),a1          Retrieve maths stack BV.RIP
*
good_try    tst.l     d0                  Channel error?
            bpl.s     return_val          No
            move.l    d0,d1               Return error code in D1
            moveq     #0,d0               Disregard EXTOP error code
return_val  tst.l     d7                  Integer or string result?
            bmi.s     return_int
return_str  tst.w     d1                  Did the lookup succeed?
            bmi.s     return_nul          Return null "" if not
            move.b    d1,0(a1,a6.l)       Stack character code
            subq.l    #2,a1               Room for string length
            move.l    a1,$58(a6)          Update BV.RIP
            move.w    #1,0(a1,a6.l)       Stack length
            bra.s     got_str
return_nul  clr.w     0(a1,a6.l)          Stack a null string
got_str     moveq     #1,d4               Indicate string result
            rts
return_int  move.w    d1,0(a1,a6.l)       Put result in space
            moveq     #3,d4               Indicate type is INT
            rts                           Return EXTOP error code
what_chan   moveq     #-6,d0              CHANNEL NOT OPEN error
bad_exit    rts                           Error code is in DO
bad_param   moveq     #-15,d0             BAD PARAMETER error
            rts
range_err   moveq     #-4,d0              Out of Range report code
            rts                           (i.e. no room for cursor)
*
* CHAracter eXTended OPeration: A0-> Channel, A1-> Buffer
*
chaxtop     moveq     #40,d0              Size of a MEM block
            cmp.l     (a0),d0             Exclude EXTOP to MEM as
            beq.s     bad_param           it does not suit CHAXTOP
            move.l    34(a0),d0           DO.L := XPOS & YPOS
            bmi.s     range_err           Reject if XPOS < 0
            tst.w     d0                  Test Y (in lower word)
            bmi.s     range_err           Reject if YPOS < 0
            add.l     38(a0),d0           DO := DO + XINC & YINC
            cmp.w     30(a0),d0           Is the window too short?
            bhi.s     range_err
            swap      d0                  Move X to the lower word
            cmp.w     28(a0),d0           Is the window too narrow?
            bhi.s     range_err
find_pixel  move.w    34(a0),d1           D1 := X
            add.w     24(a0),d1           Add window offset to X
            move.w    36(a0),d2           D2 := Y
            add.w     26(a0),d2           Add window offset to Y
            movea.l   50(a0),a2           Get screen RAM base address
            lsl.w     #7,d2               1 line uses 2^7 = 128 bytes
            adda.w    d2,a2               Get address of start of line
pixel_mask  move.w    #$8080,d7           Presume MODE 4 pixel mask
            btst      #3,52(a6)           Check MODE (SV.MCSTA)
            beq.s     mask_ok             MODE 4, no worries
            bclr      #0,d1               MODE 8 or 12, X must be EVEN
            btst      #2,52(a6)           Test Thor MODE 12 bit
            beq.s     not_a_thor          This must be the QL's MODE 8
            move.w    #$COCO,d7           Use the Thor 16 colour mask
not_a_thor  move.b    #$CO,d7             Adjust MODE 4 mask for 8 colours
mask_ok     movea.w   #128,a5             Depth of one QDOS video line
            btst      #4,66(a0)           Double height attribute?
            beq.s     no_taller
            adda.l    a5,a5               Double Y step to suit
no_taller   moveq     #1,d5
```

```
                btst      #6,66(a0)         Double width attribute?
                beq.s     no_wider
                moveq     #2,d5             Boost X step
no_wider        move.l    d1,a3             Preserve character X edge
                moveq     #7,d2             Mask bottom 3 bits of X
                and.w     d1,d2             D2 := pixel # in word, 0-7
                lsr.w     #3,d1             D1 := line byte offset, X DIV 8
                add.w     d1,d1             Address only even words 0-126
                move.w    0(a2,d1.w),d4     D4.W := background screen word
                lsl.w     d2,d4             Shift pixel data to high bits
                and.w     d7,d4             Mask out a reference pixel
                move.l    d4,a4             Save background for later
*
* Scan all nine lines and compare them with the STRIP colour in A4
*
                moveq     #8,d2             DBRA through 9 lines
scanline        add.l     a5,a2             Advance to next line
                move.w    a3,d1             Reset X to left edge
                moveq     #0,d6             Clear fount pattern
                moveq     #5,d3             DBRA count for X (5-0)
*
* Scan a row of six pixels. A2 -> video RAM line, D1 is pixel X, 0-511
*
scan_row        moveq     #7,d4             Mask for bottom bits of X
                and.w     d1,d4             X && 8 is pixel # in word
                move.w    d1,d0             Copy old X before massage
                lsr.w     #3,d0             Offset on line is X DIV 8
                add.w     d0,d0             Address only even words 0-126
                move.w    0(a2,d0.w),d0     D0.W := relevant screen word
                lsl.w     d4,d0             Align pixel in Most Sig bits
                and.w     d7,d0             Mask out the background
                cmp.w     a4,d0             Does pixel match the strip?
                beq.s     roll_on           If yes, leave bit unset
                addq.w    #2,d6             Set one fount bit
roll_on         add.w     d6,d6             Shift pattern along
                add.w     d5,d1             Advance to next column
                dbra      d3,scan_row
                move.b    d6,(a1)+          Stuff fount line in buffer
                dbra      d2,scanline
                lea.l     -9(a1),a1         Retrace to start of buffer
*
* Look up the character pattern at (A1) in both founts
*
lookup          moveq     #0,d1             Clear high bytes of code
                move.b    (a1)+,d4          Pre-read the first line
                move.l    42(a0),a2         A2 -> Fount 1 base
                bsr.s     scan_fount
                beq.s     result_d1         Match?
                move.l    46(a0),a2         A2 -> Fount 2 base
                bsr.s     scan_fount
                beq.s     result_d1
                moveq     #-7,d1            NOT FOUND error code
result_d1       moveq     #0,d0             Signal no QDOS error
                rts
*
* SCAN_FOUNT; corrupts D0, D1, D2, A2, A3, A4; Exit: NZ, D1=ASCII
* Entry: A2 -> Fount base, A1 -> Pattern+1, D4 = start of pattern
*
scan_fount move.b    (a2)+,d1          Get first pattern # in fount
           moveq     #0,d0             DBRA pattern count in bytes
           move.b    (a2)+,d0          D0 := Number of patterns - 1
test_start cmp.b     (a2)+,d4          Exclude 86-88% of matches
           bne.s     zoom_on           Ignore the rest if mismatched
           moveq     #5,d2             DBRA count for more lines
           move.l    a1,a3             Scan pattern at (A3), saving A1
           move.l    a2,a4             Check versus (A4) preserving A2
scan_all   cmpm.b    (a4)+,(a3)+
           dbne      d2,scan_all       Check each line of pattern
           beq.s     descenders        First 7 lines are OK...
zoom_on    addq.l    #8,a2             Skip remainder of pattern
           addq.b    #1,d1             Assume next ASCII code
           dbra      d0,test_start
           moveq     #-7,d0            Not found, exit with Z reset
           rts
*
descenders btst      #0,66(a0)         Check for UNDER 1
           bne.s     last_line
           move.b    (a4),d2           UNDER 0, Check line 8
           cmp.b     (a3),d2
           bne.s     zoom_on
last_line  move.b    1(a4),d2          Line 9 (last in pattern)
           cmp.b     1(a3),d2
           bne.s     zoom_on           So near and yet so far...
           rts                         MATCH!  Exit with Z set
*
* Routine names and addresses
*
define     dc.w      0,0               No procedures
           dc.w      2                 Two functions
           dc.w      clip_int-*
           dc.b      5,'CLIP%'
           dc.w      clip_str-*
           dc.b      5,'CLIP$'
           dc.w      0                 End of list
*
           end
Sinclair QL World DIY TOOLKIT February 1991. Listing 2 - CLIP code
```

volumes cost £7, recorded on disk or microdrive. Each further volume ordered at the same time costs £3 extra; there are now 14 volumes. CGH supplies disks, but tape users must send one formatted cartridge for each volume required. Call Richard Alexander on (0559) 384574 or write to *DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA.*

Fast

The disk Qlipboard incorporates Phil Spink's fast windowing editor, with adjustable tabs, word-wrap, paging, filing and other civilised features. That editor was enhanced and tested for *Quickfax* more than a year ago. That project has stalled but Phil's editor deserves an airing. It has been carefully and intuitively designed, and recognises a superset of popular QL control keys.

Character scanner register utilisation table

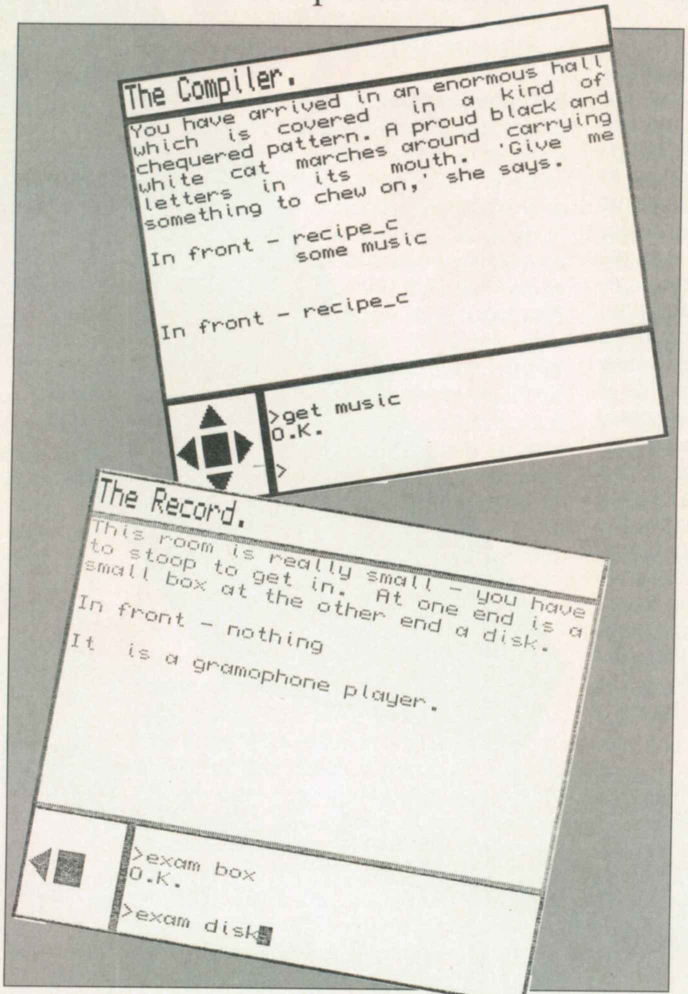
*D0 - Scratch
*D1 - Current pixel X co-ordinate
*D2 - line count, 8-0 top to bottom
*D3 - column count, 5-0 left to right
*D4 - Pixel bit offset in video word, 0-7 left to right
*D5 - X step between pixels: 1=CSIZE 0/1, ?; 2=CSIZE 2-3,?
*D6 - Accumulated caracter form for this line
*D7 - Pixel mask for this MODE (WORD)
*
*A0 - Pointer to channel definition block
*A1 - Pointer to byte buffer of 9 lines in QL fount format
*A2 - Pointer to start of current video RAM line (Y2)
*A3 - Leftmost pixel X co-ordinate
*A4 - Background pixel colour, normalised
*A5 - Y step between lines: 128=CSIZE ?, 0; 256=CSIZE ?,1
*A6 - Pointer to QDOS or Argos system variables
*A7 - SUPERVISOR stack pointer

SOFTWARE FILE

ANELPUM QUAT

INFORMATION
Program: Anelpum Quat
Price: £8.00 disk or mdv (& 80p for p&p etc.). All QLs.
Publisher: *CGH Services, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA.*

Rich Mellor gets inside the last word in adventure, where the compiler is a black and white cat



```
The Compiler.
You have arrived in an enormous hall
which  is  covered   in  a  kind  of
chequered pattern. A proud black and
white  cat  marches around  carrying
letters  in  its   mouth.  'Give  me
something to chew on,' she says.

In front - recipe_c
           some music


In front - recipe_c

>get music
O.K.

>
```

```
The Record.
This room is really small - you have
to stoop to get in.  At one end is a
small box at the other end a disk.
In front - nothing
It  is a gramophone player.

>exam box
O.K.
>exam disk
```

This game is a new adventure from the CGH stable. This represents yet another new direction in adventures, being set inside the very computer you are using. The manual (as usual) is short and gives just a little insight into the game, before you enter your first location.

As soon as you load the game, some of the differences in this adventure will become apparent. The screen is divided into four, with the name of the current location at the top of the screen, below which appears a description of the location. At the bottom of the screen is a graphical representation of the current exits, and another window where you enter your commands. The method of entry of commands is distinctive in that the parser will only allow you to type words which it recognises. It does this by checking every letter as it as typed, and if the given letter does not fall into a recognised word, a beep is issued and the letter is not shown on screen. This may speed up checking of the commands, and also prevents any 'not understood' errors. However, the method of deleting letters is rather cumbersome in that instead of the previous letter being deleted, a back arrow is printed on screen. This can lead to confusion, but thankfully pressing ESC abandons the command.

One soon becomes used to the new method of entry of commands, and with the program's ability to understand multiple commands, one soon begins to zip around the locations. Movement can be done by the normal east, west, north, sound commands, or simply by using the cursor keys if preferred. I must admit that I would have preferred a normal means of command entry, especially when you begin and are unsure which commands the computer will understand.

This first location actually presents the first major problem. You are outside the computer which is not working, and need to get inside it to fix it. Without a screwdriver or manual to hand, you soon discover that there is no easy method of entry; more 'direct' methods must be used. Once inside the computer, you will find some text telling you that the computer is in a right state with lots of unfinished tasks and half-written routines. As you go around, you will discover just what these tasks are.

Unfortunately at first appearances there do not seem to be very many locations, and even fewer objects. Looking around you will find lots of different characters which make up the different parts of the computer, ranging from compilers to viruses. Each of these has a different function, and while some will be able to help you, others are there to hinder you (or even kill you).

The compiler (a black and white cat) is a great help during the early part of the game, although presenting him with something to compile makes him ask you a cryptic question which must be answered to get anything useful for him. Answering some of these questions will open new areas of the computer up for you to search, while others will provide you with the means to finish tasks. A sign near the entry point to the computer tells how well you are doing by presenting a reading of a failure rate. I have only achieved an 82% failure rate so far in several hours of play and so I have a long way to go yet!

Many of the different tasks involve plays on words to give you a hint as to what is needed (there is no help feature in this adventure). For instance, the stack is missing its pointer - do not try looking for a computer-type pointer!

The puzzles within the game are very well thought out and involve a logical means of solving them. It provides a little insight into the workings of a computer as well as being an entertaining game. Fortunately there is no need to have an in-depth knowledge of computers, although a *Roget's Thesaurus* can be a big help. Thankfully the game can be saved onto any device to return to at a later date, since I can see that a pen and paper may be just as useful as a computer keyboard. Despite one or two minor criticisms about the method of command entry, I must say that the program is well worth getting one's teeth into.

PRODUCT REVIEW

PC KEYBOARD CARD

Many QL users prefer to use a PC-style keyboard. MIke LLoyd tests a new keyboard interface card.

Some people in the computer market must by psychic. Just as my trusty Sandy keyboard was beginning to show signs of its venerable age the postman brings a saviour to the rescue – a new piece of electronic wizardry from Jeurgen Falkenberg. Falkenberg, you will remember, was responsible for the excellent scanner reviewed in *QL World* some months ago. His latest offering is not a keyboard, but a chip-laden link to a standard PC-style keyboard, which gives users a choice of the full range of PC keyboards available in the UK today at very reasonable prices.

The original QL keyboard came in for a lot of criticism early in its life. One journalist wrote "Hands up all those who like the new QL keyboard . . . what a lot of funny-looking hands". Designed to be cheap to make and better to use than the Spectrum's rubber keys (and, incidentally, much better than the awful keyboard of the IBM PC jr.), the QL keyboard was still not as good as a "grown-up" PC keyboard. I

touch-typed my way through the manuscript of my first book using nothing but a QL keyboard, and I know of other authors who did the same, but many good typists could not get used to its tacky feel.

Smart money

One solution was to replace the top of the QL with one possessing better-sprung keys, and this proved to be a much more acceptable improvement than might have been thought. Nevertheless, the smart money went on add-on keyboards which were linked to the QL via a cable and which had a separate numeric keypad and additional keys laid out in PC XT style. As the owner of one of the original Sandy keyboards I can vouch for its ease of use and considerable endurance. Giving a journalist a keyboard is rather like giving a six-inch nail to a sixteen-stone carpenter – it tends to get thoroughly hammered.

Recently, as I remarked earlier, the Sandy keyboard began to show signs of senility. It would forget some of the keypresses, and occasionally had a stroke which would paralyse one side completely. The ribbon cable which lay across my desk like the Channel Tunnel rail link across Kent would snag papers and books so that a slight movement of the keyboard would partially dislodge it, rendering many of the keys useless until the socket had been pushed back into place. Keyboard bounce – the QL's annoying habit of printing random letters twice – was not eradicated by the Sandy keyboard, although it appeared to be reduced. All of these weaknesses could be lived with, but the keys in the centre of the board were beginning to wear out so that a delicate press was ignored and only a table-shaking stab would produce a response. After four years of unrelenting toil, it was time to find a new keyboard.

Letterbox

With impeccable timing, Jeurgen Falkenberg's gizmo was thrust through my letterbox. Mounted on a card some 12cm by 6cm, a cluster of eleven silicon chips gave some clue as to the computing power required to accept input from a PC keyboard and translate it so that it was fully compatible with what the QL expects. At one end of the keyboard a short cable was mounted, ending in a DIN plug familiar to anyone who has examined the back of a PC, or a QL for that matter. At the other end of the small card was an empty mounting slot. Unfortunately, the Jiffy bag contained nothing else – no fitting instructions, no user guide, and no invitation for a week's expenses-paid holiday in the Rhine valley.

The purpose of the card was obvious enough – but what about fitting it? The QL's keyboard is managed by an interesting little chip called the Intel 8049. It was perhaps the first "computer on a chip", in that it has its own arithmetic logic unit (alu), central processing unit (cpu), a read-only memory (rom), and a tiny amount of access memory (ram) all built on the same wafer of silicon. For a chip so comprehensively equipped, it is a little disappointing to find that it is not very bright. In the QL its task is to relieve the M68008 cpu from much of the mundane work of listening out for keypresses and controlling microdrive and diskdrive activity. The empty mounting slot on the keyboard controlling card was exactly the size for an Intel 8049, and small cutouts in the card fitted exactly with the topology of that part of the QL's motherboard where the Intel 8049 is situated – towards the front left, immediately beside the microdrive housings.

With only a little trepidation I opened the QL's case, carefully avoiding dislodging the wires which supply current to the various leds on the computer's top. The ribbon cable to the Sandy keyboard was quickly removed and, using a plastic chip-removing tool and a lot of patience, so was the Intel 8049. I pressed the chip into its new home on the piggy-back card and then began the process of gently persuading the pins underneath the card to slot into the holes left by the Intel chip. The secret is to slowly increase pressure across the entire width of the mounting so that pins do not become bent. It is surprising how much weight is actually required to press home the pins.

The final problem was how to get the cable out of the QL case so that a keyboard could be plugged into it. After seriously considering the removal of the second serial socket for a few moments I finally decided to unplug the expansion card, thread the cable through the huge hole in the QL's left side and then slot the expansion card back in place.

Some days later I received a call from Falkenberg's UK agents to say that the QL-Keyboard-90 card was also available in a form which simply plugs into the QL's rom port. This is a much simpler method of upgrading the QL and one which is recommended to anyone frightened of taking a screwdriver to his or her beloved computer.

Simpler method

The rest was easy: simply rob a passing PC of its keyboard and pug its DIN plug into the waiting socket. Nevertheless, powering up the QL was a little nerve-wracking. What if this wasn't a keyboard controller after all, but some other arcane device which I had completely misidenified? All was well, though. The keyboard lights flickered, the QL booted normally and I began typing away.

It was quite unusual to find myself using a QL with a fully-functioning PC keyboard. Deleting was performed not by the Ctrl-left arrow key combination but by the PC's backspace key. Similarly, the PC's delight key performed the action normally accomplished by pressing Ctrl-right arrow. The function keys, numeric keypad and cursor keys worked exactly as would be expected. Rather surprisingly, so did the Escape key: it acts like Ctrl-Space to halt a SuperBasic program or exit a line-editing sequence. Within a few moments the feel of the new Keyboard lost its alien nature and I quickly adjusted to the new key positions.

Free to choose

QL-Keyboard-90 is for sale in the UK at £75, which includes VAT, and it is available from TK Computerware in Kent. At the price it is not cheap, and many users without their own PCs will have to consider buying a keyboard as well, so can it compete in the keyboard add-on stakes?

The answer must be a definite yes. The keyboard mapping vastly exceeds in power and ease of use anything previously offered on the QL market. By buying Falkenberg's card you are then free to choose any of the myraid of PC keyboards available through mail order in Computer Shopper and other similar magazines. Good PC keyboards start at around £50, although some de luxe models can cost nearer £100. For many users, though, it will be a case of borrowing a keyboard from a PC, or maybe buying a second-hand keyboard attached to a broken PC, reducing the costs. Two simple jumpers on the card allow the firmware to be configured for old or new styles of XT and AT keyboards.

With full documentation and in its final release version, QL-Keyboard-90 has the feel of a quality product and it will undoubtedly improve your typing. Running Digital Precision's excellent *PC-Conqueror* emulator couldn't be easier if you are using a real PC keyboard. People with a desk cluttered by too many keyboards can now make one keyboard do the work of two. All in all, QL-Keyboard-90 is a complex product which does a very good job without bothering the user and if it suits your circumstances, and your pocket, I recommend that you give it serious thought.

INFORMATION

Hardware:	QL-Keyboard-90 PC-compatible add-on keyboard controller
Price	£75 inc VAT
Manufacturer	Jeurgen Falkenberg West Germany
Available from	TK Computerware, North Stanford, Ashford, Kent. Tel: 0303 81 2801

Programming in C

Andy Wright continues his beginners' guide to C with arrays and file processing.

In the last instalment we considered symbolic constants, variables, sequence control and conditional statements. There is another conditional statement (the switch statement) that we will look at in a future article.

The two main topics this month are handling arrays and basic file processing. There are three program listings included to illustrate the new concepts. A few other new ideas are also mentioned in passing.

Listing one illustrates very simple array processing. It takes input from the keyboard and counts the number of times that each of the different keys is pressed. When you press the ESC key it displays the results.

The program begins by defining a few symbolic constants. SIZE_ASCII represents the highest character code value used in the QL character set — all character codes are in the range zero through 255. NUM_LINES represents the number of lines of data that can conveniently be displayed in the window in which Digital C programs run.

Digital C supports int and char data types. You can also declare arrays of integers and characters. The syntax is:

```
data_type variable_name
[array_size];
```

The array_size must be enclosed within square brackets. The number of elements in the array is one more than the array_size. In our program the array contains 256 elements (SIZE_ASCII + 1).

Arrays in C are similar to arrays in SuperBasic, but there are a few differences. As they say, variety is the spice of life.

Each element of the array is referenced by specifying a subscript enclosed in square brackets (SuperBasic uses round brackets). For example, the first element in the array is referenced as variable_name [0], the second as variable_name [1] etc.

The program in listing one declares an int array named num_char. This array consists of 256 elements numbered from [0] through to [255] (SIZE_ASCII is 255). Each element of the array can hold an int value. The array is used to represent the 256 possible Ascii characters.

After printing a few lines of introductory text the program executes a for loop. At the beginning of the loop the value of sub is set to zero. The loop keeps repeating so long as sub is less than or equal to the highest subscript value in the num_char array (255). At the end of each pass through the loop the value of sub is incremented by one.

The body of the loop consists of a single statement which sets the value of num_char [sub] to zero. If the elements in the array were not initialised to zero they would contain garbage values.

An experienced C coder might have coded the for loop that initialises the array as:

```
for ( sub = 0;
      sub <= SIZE_ASCII;
      num_char [sub++] = 0 )
      ;
```

In this new version the body of the for loop includes no statements — the single semicolon at the end indicates this. At the end of each repetition of this non-existent body of statements, the following is executed:

```
num_char [sub++] = 0
```

The position of the ++ symbol is important. It tells the system to set num_char [sub] to zero and then increment the value of sub (ie it uses the current value of sub and then increments it).

Num_char [++sub] = 0 would have incremented sub first and then set num_char [sub] to zero.

The new version of the for loop is more concise and, in the eyes of some, more elegant. Such shorthand techniques are common in C programs. People who are new to C sometimes find it confusing. People who are experienced C programmers occasionally take satisfaction in how simple it is when you get the hang of it but how confusing it all can be to the uninitiated.

The next part of the program uses a while loop to process the next key typed in at the keyboard. The Ascii code of the key is stored in the variable inp_char.

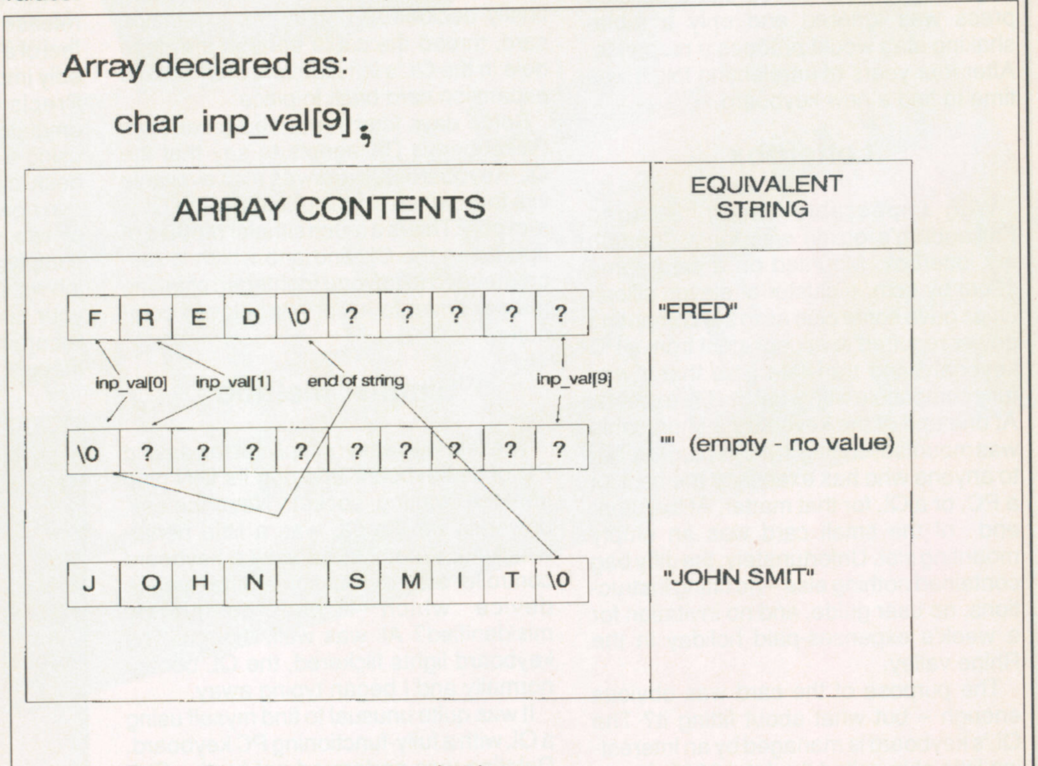
The loop stops when the (ESC) key is pressed. Each pass round the body of the while loop adds one on (using the ++ operator) to the num_char array element whose element number is the same as the value of inp_char. So, if you press A the value of num_char [65] is incremented by one — the Ascii code for an upper case A is 65. The statement:

```
++num_char [inp_char];
```

illustrates that char variables can be used as integers since they actually contain integer Ascii character codes.

The use of the putchar function within the loop makes sure that the character that you type is also displayed on the screen.

The last part of the program



Array declared as:

char inp_val[9];

ARRAY CONTENTS										EQUIVALENT STRING
F	R	E	D	\0	?	?	?	?	?	"FRED"
\0	?	?	?	?	?	?	?	?	?	"" (empty - no value)
J	O	H	N		S	M	I	T	\0	"JOHN SMIT"

Fig 1: Character arrays and strings

uses a for loop to print out the values held in the array. Initially the variables sub and count are set to zero. The variable sub represents each element number in the array — corresponding to the Ascii characters 0 through 255. The count variable is used to keep track of how many lines have been printed on the screen. The loop keeps going while sub <= SIZE_ASCII — ie until all of the elements in the array have been processed. At the end of each pass round the loop, both sub and count are incremented.

The printf function prints two versions of sub. One is a decimal version of the Ascii character code (using the %3d format), the other is the QL character which has the corresponding Ascii code (using the%c format). On each pass round the loop sub represents and Ascii character. The value of num_char [sub] represents the number of times that the corresponding keyboard character was typed. This is printed as a decimal (%5d) value.

Whenever the value of count is greater than NUM_LINES you will be prompted to press a key. This stops the display scrolling off the screen before you can read it. If you press the ESC key, a break statement is executed. This breaks (jumps) out of the loop that you are in — even though the condition defined in the for statement is still true. Program control passes to the statement at the end of the loop (the printf function in our program). Similar facilities are provided for getting out of loops in SuperBasic. If you press anything other than ESC the value of count is reset to zero and the for loop carries on with the processing.

Right at the end of the program, getchar is used to make sure that the display remains on the screen until a key is pressed. Note that the value returned by getchar is thrown away or discarded since no variable name has been specified to hold the value.

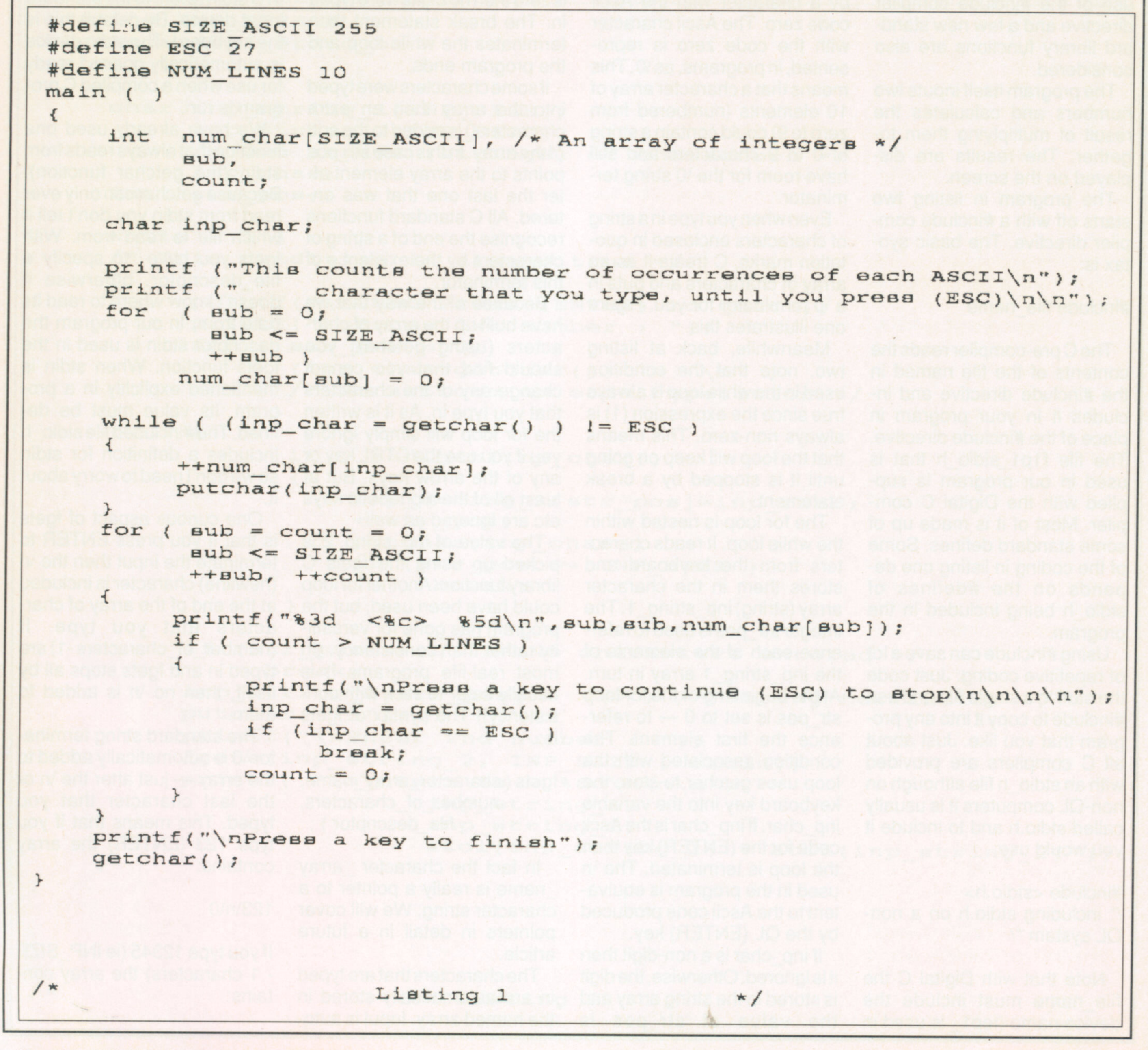
Exercise One

1. Change the program so that the array elements are not initialised to zero and run it a few times.

2. What would happen if you tried to update num_char [256] — or any other element beyond number 255? If you try this it will have unpredictable and potentially nasty results since it will update bits of the computer's memory which lie somewhere beyond the end of the array. Accessing non-existent elements in an array is a sure-fire (and fairly straightforward) way of guaranteeing that your program produces garbage — or worse.

3. Try using the alternative for loop for initialising the array elements. Code it correctly and then try coding it incorrectly (putting the ++ operator in the wrong position).

4. Re-write the loop that displays the results so that it doesn't print anything at all for array elements with a value of zero — ie those representing



```
#define SIZE_ASCII 255
#define ESC 27
#define NUM_LINES 10
main ( )
{
   int  num_char[SIZE_ASCII], /* An array of integers */
        sub,
        count;

   char inp_char;

   printf ("This counts the number of occurrences of each ASCII\n");
   printf ("     character that you type, until you press (ESC)\n\n");
   for   ( sub = 0;
           sub <= SIZE_ASCII;
           ++sub )
         num_char[sub] = 0;

   while ( (inp_char = getchar() ) != ESC )
   {
         ++num_char[inp_char];
         putchar(inp_char);
   }
   for ( sub = count = 0;
         sub <= SIZE_ASCII;
         ++sub, ++count )
   {
         printf("%3d   <%c>   %5d\n",sub,sub,num_char[sub]);
         if ( count > NUM_LINES )
         {
               printf("\nPress a key to continue (ESC) to stop\n\n\n\n");
               inp_char = getchar();
               if (inp_char == ESC )
                   break;
               count = 0;
         }
   }
   printf("\nPress a key to finish");
   getchar();
}

/*                   Listing 1                   */
```

keys that were pressed zero times.

5. You should find that the display is messy for the character whose Ascii code is 10 (the ENTER or newline key). This is because printing the character (%c) version forces the screen display to do a linefeed. Try to stop this from happening. You will need to include an if statement to change what happens when sub is equal to 10.

6. What would happen if the same key was pressed more than 32767? Change the code so that this cannot happen.

7. Try printing the value of num_char without specifying an element number.

Listing two illustrates the use of a character array in C. The use of the #include compiler directive and a few new standard library functions are also considered.

The program itself inputs two numbers and calculates the result of multiplying them together. The results are displayed on the screen.

The program in listing two starts off with a #include compiler directive. The basic syntax is:

```
#include file_name
```

The C pre-compiler reads the contents of the file named in the #include directive and includes it in your program in place of the #include directive. The file f1p1_stdio_h that is used in our program is supplied with the Digital C compiler. Most of it is made up of some standard defines. Some of the coding in listing one depends on the #defines of stdio_h being included in the program.

Using #include can save a lot of repetitive coding. Just code the stuff in a single file and use #include to copy it into any program that you like. Just about all C compilers are provided with an stdio_h file although on non-QL computers it is usually called stdio.h and to include it you would use:

```
#include <stdio.h>
/* including stdio.h on a non-
QL system */
```

Note that with Digital C the file_name must include the device name (fep1_ is used in our program; if you are using microdrives then use mdv1_ or mdv2_). On many other implementations of C the device name (or its equivalent) is not needed if you enclose the name between < and > symbols.

The program defines two arrays of characters — each with seven elements — numbered from zero to INP_SIZE.

In C all strings are made up of arrays of characters. This is also true of SuperBasic. However, it is often possible to ignore this fact when writing programs in Basic — you can never really afford to forget it when using C.

One area where C character arrays (or strings) are significantly different from SuperBasic is that in C the end of a string is always represented by a character with the Ascii code zero. The Ascii character with the code zero is represented, in programs, as \0. This means that a character array of 10 elements (numbered from zero to 9) could contain a string of 0 to 9 characters and still have room for the \0 string terminator.

Even when you type in a string of characters enclosed in quotation marks, C treats it as an array of characters and puts in a \0 terminator for you. Figure one illustrates this.

Meanwhile, back at listing two, note that the condition used in the while loop is always true since the expression (1) is always non-zero. This means that the loop will keep on going until it is stopped by a break statement.

The for loop is nested within the while loop. It reads characters from the keyboard and stores them in the character array (string) inp_string_1. The integer str_pos is used to reference each of the elements of the inp_string_1 array in turn. At the beginning of the for loop str_pos is set to 0 — to reference the first element. The condition associated with the loop uses getchar to store the keyboard key into the variable inp_char. If inp_char is the Ascii code for the (ENTER) key then the loop is terminated. The \n used in the program is equivalent to the Ascii code produced by the QL (ENTER) key.

If inp_char is a non-digit then it is ignored. Otherwise, the digit is stored in the string array and the value of str_pos is incremented. If the array has only one unused element left (ie the value of str_pos is equal to the array_size) the break statement is used — this terminates the for loop immediately. If this check on the value of str_pos was not included then you would be able to type in a potentially unlimited number of numeric digits. As str_pos went beyond the end of the memory allocated to the array, the contents of other areas in the machines memory would get corrupted. Notice that if str_pos equals INP_SIZE the final array element has not been used — the last element is needed so that we can store the C string terminator (\0) at the end of the array.

At the end of the for loop the value of str_pos is checked. If it is zero then no digits were typed in. The break statement then terminates the while loop and the program ends.

If some characters were typed into the array then an extra character \0 is added to the end of the array. In this case str_pos points to the array element after the last one that was entered. All C standard functions recognise the end of a string of characters by the existence of this terminator.

Because of the way that we have built up the array of characters (using getchar) you should find that you cannot change any of the characters that you type in. As it is written the for loop will simply ignore you if you use the CTRL key or any of the arrow keys. But at least all of the alphabetic keys etc are ignored as well.

The value of inp_string_2 is picked up using the fgets C library function. Another for loop could have been used, but the program has gone for versatility rather than consistency. (In most real-life programs it is usually safer to stick with consistency!) The syntax of fgets is:

```
fgets ( character_array_name,
        number_of_characters,
              file_descriptor ) ;
```

In fact the character _array _name is really a pointer to a character string. We will cover pointers in detail in a future article.

The characters that are typed in are automatically stored in the named array. Input is automatically terminated either when the you press ENTER or (number_of_characters - 1) keys have been pressed. You should find that the way that fgets works on the QL allows you to make changes to the string that you type using the arrow and CTRL keys. Whatever you type is also automatically displayed on the computer screen without you having to use putchar.

The fgets function can read from any open device or file. This means that you must tell it where to read data from. The device is specified using a thing called a file_descriptor — somethimes known as a file pointer. In listing two the file _descripter stdin is used. A file_descriptor is a bit like a channel number in SuperBasic. In C stdin refers to the standard input device. By default this is the keyboard. The stdin device is automatically opened ready for use when a compiled C program is run.

We have already used one function that always reads from stdin (the getchar function). Because getchar can only ever read from stdin you don't tell it which file to read from. With fgets you have to specify a file_descriptor, otherwise it doesn't know where to read its data from. In our program the descriptor stdin is used in the fgets function. When stdin is mentioned explicitly in a program, its value must be defined. The #included file stdio_h includes a definition for stdin so we don't need to worry about it.

One curious aspect of fgets is that if you press ENTER to terminate the input then the \n (newline) character is included at the end of the array of characters that you type. If (number_of_characters - 1) are typed in and fgets stops all by itself, then no \n is added to them.

The standard string terminator \0 is automatically added to the array — just after the \n or the last character that you typed. This means that if you type 123 (ENTER) the array contains:

```
123\n\0
```

If you type 12345 (ie INP_SIZE - 1 characters) the array contains:

12345\0

There is nothing in the fgets function to stop you typing non-numeric characters. These will just be included inside the array.

The printf statement at the end of listing two prints the input strings. Both inp_string_1 and inp_string_2 are printed using the %s format specification. %s is used to print character arrays (strings). It starts at the beginning of the array and stops when it gets to the string terminator \0.

The %6d format specification is used to print the result of the expression:

atoi (ino_string_1 * atoi (inp_string_2)

The atoi funtion converts a \0 terminated string of Ascii characters into an integer. The conversion stops when any non-numeric character is found within the string (for example a \0 string terminator or a \n newline character). This would mean that if you typed 12a4 the integer calculated by atoi would be 12. If you typed a3 then atoi would return a value of zero!

Note that the * included between the quotes in the printf statement is printed as an asterisk. The * in the list of expressions that are to be printed performs the multiplication of the integer version of each of the input strings. The result of this is printed using a %6d format specification.

After the results have been printed, the end of the while loop is reached and you should be prompted to enter another number.

Exercise two

1. Change the program so that no \0 is added to the end of the character array at the end of the for loop. See what difference this makes to the way that the program works.

2. Check to see if fgets and atoi cope with negative numbers.

3. What happens if you type in numbers that are too large? Can you change the code easily to stop this from happening?

4. You should find that the display of inp_string_2 produced by the printf function is not always consistent. It depends on whether or not the fgets function added a \n character to it. Change the code so that any \n is replaced by a \0 before the value of inp_string_2 is printed.

5. Try to re-write the statements:

inp_string [str_pos] = inp_char;
++str_pos;

as a single statement which achieves the same overall effect.

6. Try to replace the bits of the printf statement that prints the value of inp_string_1 by a for loop that uses the putchar function to print the contents of the inp_string_1 array — it should stop as soon as it gets to the \0 character.

What would happen if you

```
#define INP_SIZE 6
#include flp1_stdio_h
main ( )
{
     char inp_string_1[INP_SIZE],
          inp_string_2[INP_SIZE],
          /* Two arrays (strings) of characters */
          inp_char;
     int  str_pos;

     while ( 1 )
     /* This condition is always true */
     {
          printf("\nJust Press (ENTER) to quit");
          printf("\nEnter 1st number : ");
          for ( str_pos = 0;
                (inp_char=getchar()) != '\n' ;
                /* No actions on each repetition */
              )
              {
                  if ( isdigit(inp_char))
                  {
                       inp_string_1[str_pos]=inp_char;
                       ++str_pos;
                       putchar(inp_char);
                       if (str_pos == INP_SIZE )
                            break;
                  }
              }
          if (str_pos == 0) /* No characters entered */
               break;
          inp_string_1[str_pos] = '\0'; /* String terminator */
          putchar('\n');
          printf("\nEnter 2nd number : ");
          fgets(inp_string_2, INP_SIZE, stdin);
          printf( "\n\n%s * %s = %6d\n\n\n",
                  inp_string_1,
                  inp_string_2,
                  atoi(inp_string_1) * atoi(inp_string_2) );
     }
}

/*                    Listing 2                          */
```

forgot to make it stop at the end of the string (if you try it out you will have to reset your QL before carrying on — unless you can kill off the program somehow — you have been warned.

7. What happens if you forget to include the stdin file descriptor in the fgets function?

Listing three illustrates how you can process data from files other than the standard input and output devices.

The program given in listing three prompts you to supply a file name and then tries to display the contents of the named file on the screen.

The first few lines of the program are made up of #include and #define compiler directives. MAX_FILE_NAME is defined to be big enough to hold the largest sized file name available on the QL (in fact it is slightly larger than necessary). LINES_PER_PAGE is the number of lines from the input file that are displayed at a time. CHARS_PER_LINE is the number of characters that fit on a line inside the window that Digital C programs usually run in.

The other symbolic constant that is defined is FILE — it is given the value int. Its use is described below.

As we have already seen, files in C are accessed by using file descriptors. These file descriptors are a bit like SuperBasic channel numbers. In fact, file descriptors are used to access files and devices such as the keyboard or windows on the screen. So are SuperBasic channels.

Before a file or device can be accessed in C, it must be opened and linked to a file descriptor. Like other implementations of C, Digital C provides some devices that are automatically opened and linked to file descriptors. The file descriptors for the devices that are opened automatically are defined for you inside stdio_h.

They are:

```
stdin
stdout
stderr
```

By convention, stdin is linked to the keyboard whilst stdout and stderr are linked to the screen. Stderr is usually used to display error messages.

Digital C is rather curious in that all three of the default file descriptors are linked to a single QDOS channel. This channel is a console. You can read from a console (it picks up data from the keyboard) and write to a console. (the ouput appears in a window on the screen). This means that you can get away with writing programs that read from stdout and write to stdin. Even though you can get away with doing this — you should try to overcome the temptation. Attempting to read from an output device just seems wrong!

A few of the standard C library functions that we have used have written to the default devices. For example, getchar reads a character from stdin, while printf writes to stdout.

In listing two we saw that the fgets function can read from any file. The cost of this extra versatility in fgets is that you have to tell it which file or device to read from. We forced it to read from stdin.

The program in listing three defines its own file descriptor and uses it to read data from a file on a disc or microdrive.

The first line of the main function declares the file descriptor.

It uses:

```
FILE * file_d;
```

The word FILE is a symbolic constant — defined as int, so the declaration of the file descriptor could have been written as:

```
int * file_d;
```

so why wasn't it?

Most implementations of C define a FILE data type. Digital C doesn't. In most C implementations there would be no need to define FILE as a symbolic constant because it is already defined for you.

In Digital C you could edit your own stdio_h file to include the definition of FILE as int — then you wouldn't have to define it within the program. Just include stdio_h. This would make the whole program look more like a C program as coded on, say, a computer running Unix or MS-DOS. In Digital C, file descriptors should be of data type int, but this isn't necessarily true of other systems. Using the data type FILE gets round this problem fairly neatly because it can be defined differently in stdio_h on different machines and it doesn't affect your program code.

If the last few paragraphs bother you, it doesn't much matter if they don't make a great deal of sense if you are only interested in using C on a QL with the Digital C compiler. If you are keen on writing C programs that can easily be compiled and run correctly on other computers using different C compilers then it is important to see the significance of what has been said!

The other outstanding point it — why is there an asterisk (*) in front of the file descriptor name file_d? One answer to this question is that there has to be an asterisk in front of it! File descriptors are, in reality, pointers to things. This means that the file descriptor points to something that tells the system where a file is. File descriptors are also known as file pointers. When any kind of pointer is declared in a C program its name must be preceded by an asterisk. We will look at pointers in a lot more detail in a future article. For the time being just include the asterisk.

After the file descriptor declaration, a few integers and a character array are declared. The character array is used to hold the name of the file that is to be processed by the program. The array is declared to be at least large enough to hold the longest file name that you would find on a QL.

A while loop is used get the file name. Just before the while loop is entered, the file_d descriptor is set to NULL. The symbolic constant NULL is defined in stdio_h (in Digital C it has the value of -32768, on other systems it may be different but it doesn't matter!)

So long as the value of file_d is equal to NULL the while loop keeps repeating and you keep getting asked to enter a file name. We shall see what is so special about NULL in a few moments.

Note that the message that tells you to enter a file name is produced using the fprintf function. This is like printf except it can write to any open file or device. In listing three the message is sent to the stdout device. By default this is the computer screen.

The fgets function is used to read the filename from stdin. Immediately after the fgets function there is a check to see if any characters were entered. Just to be different, the strcmp function is used. Its basic format is:

```
strcmp              (
character_array_name or
characters in quotes,

    character_array_name or
characters in quotes );
```

The parameters (or arguments) to strcmp are really pointers to strings. Character array names and strings of characters enclosed inside quotes are actually pointers to strings — but more on pointers later.

The strcmp function compares the strings and /or character arrays. If they are equal to each other it returns a value of zero. See the Digital C documentation for other possibilities.

The if statement uses strcmp to compare the inp_name character array with the string \n. Remember that if you just press ENTER without typing any characters then fgets will insert the characters \n\0 into the character array. If the result of the strcmp is zero then the exit function is used. The exit function terminates the program and closes any open files.

The strcmp part of the program could have been coded as:

```
/* assuming inp_val has been
declared as an int */
int_val = strcmp ( inp_name,
"\n");
if ( int_val == 0 )
    exit () ;
```

The method used in listing three is more compact since it doesn't need to make use of the variable int_val.

Yet another way would have been to adapt the approach used in listing two:

```
if ( inp_name [0] == '\n' )
    exit () ;
```

Note that the == operator cannot be used to compare strings of characters — it only compares ints and char in Digital C. This means that the \n must be enclosed in apostrophes — if it were enclosed in ordinary quotation marks it would be a string! Inside apostrophes it means the Ascii code of the newline character.

If some characters were typed into inp_name the next part of the program uses a for loop to replace any \n character in the string by a \0. As we have already noted, fgets likes to put a \n at the end of the string that it gets. The for loop in the program processes each character in the array until it reaches the end of the string (marked by the \0 character). Any \n that is found within the string is replaced by a \0. This effectively gets rids of the \n and marks a new end to the string. For example:

f1p1_fred\n\0

becomes

f1p1_fred\0\0

So far as C is concerned, the new string simply contains the characters f1p1_fred since the first \0 marks the end of the string.

Once this has been done an attempt is made to open the file. This is done using the fopen function. The basic syntax is:

```
file_descriptor_name = fopen ( file_name,
            open_mode ) ;
```

The file descriptor must be declared appropriately at the beginning of the program. The file_name and open_mode can each be either the name of a character array or a string of characters enclosed in double quotes. (If you guessed that they should be pointers to strings — you guessed right).

The open_mode can be one of the following:

r - opens the file to read data from it

w - opens the file to write data to it, if the file already exists then its contents are deleted

a - opens the file to add data to the end of it, if the file doesn't already exist then a new empty file is created

The mode could be the name of an array containing the characters r\0 or w\0 etc or it could be "r" or "w" etc (enclosed in quotes).

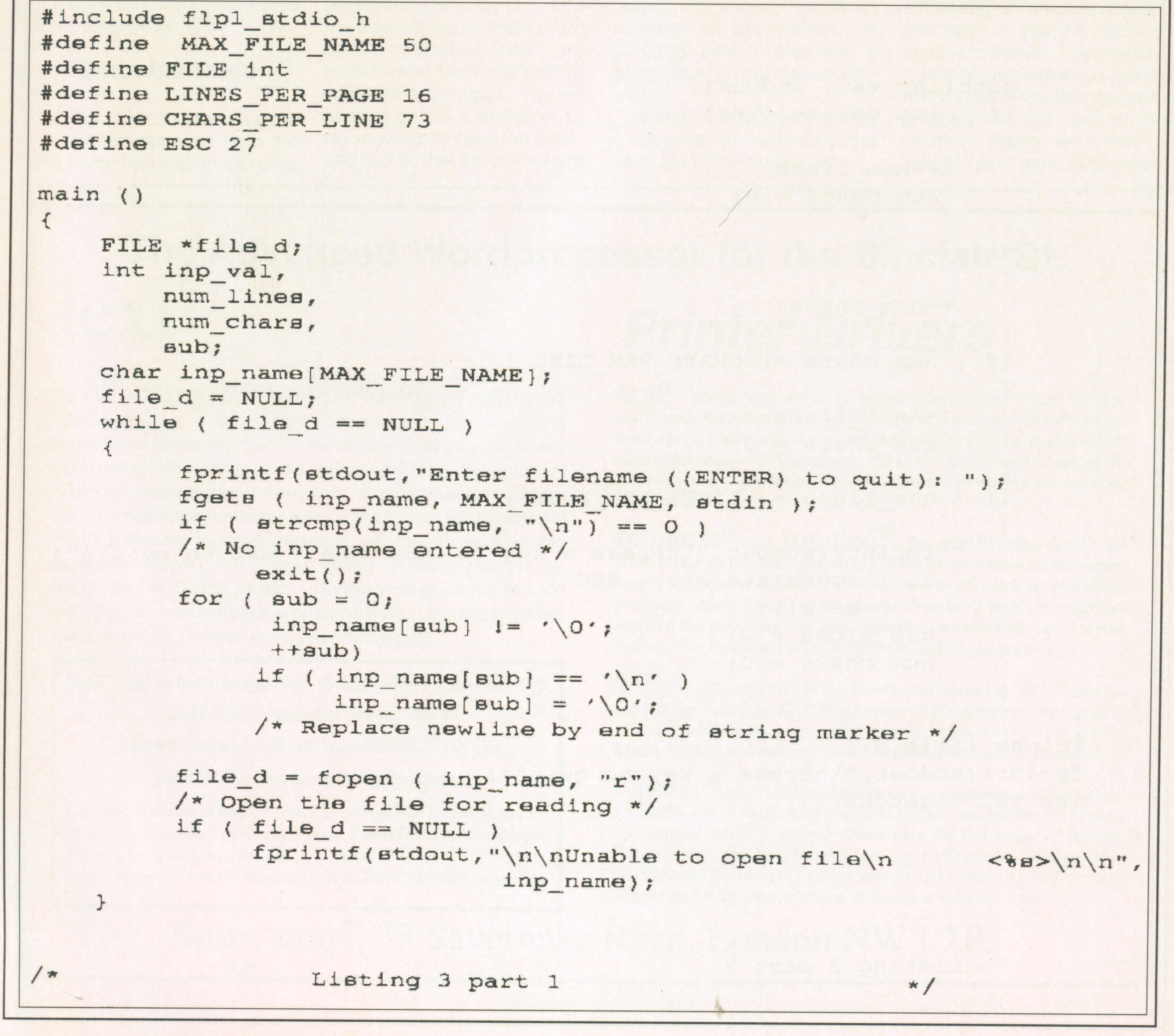
If the fopen statement fails then the file_descriptor is set to NULL. If fopen succeeds it sets the file_descriptor to a non-NULL value.

The fopen function will fail, for example, if you specify an invalid device name at the beginning of the file name (eg md1_ instead of mdv1_) or if there is no disk in the drive or if you try to open a file for reading and the file doesn't exist.

Our program opens the file for reading. If fopen fails and the file_descriptor is set to NULL then a message is displayed. At the end of the while loop, the descriptor file_d is still NULL so it goes round the loop again.

Once the file has been successfully opened and file_d is not NULL the while loop terminates.

The file's contents are then processed within a for loop. The variables num_lines and num_chars are set to zero — since nothing has been displayed yet. The loop is repeated



```
#include flp1_stdio_h
#define  MAX_FILE_NAME 50
#define FILE int
#define LINES_PER_PAGE 16
#define CHARS_PER_LINE 73
#define ESC 27

main ()
{
    FILE *file_d;
    int inp_val,
        num_lines,
        num_chars,
        sub;
    char inp_name[MAX_FILE_NAME];
    file_d = NULL;
    while ( file_d == NULL )
    {
         fprintf(stdout,"Enter filename ((ENTER) to quit): ");
         fgets ( inp_name, MAX_FILE_NAME, stdin );
         if ( strcmp(inp_name, "\n") == 0 )
         /* No inp_name entered */
              exit();
         for ( sub = 0;
                inp_name[sub] != '\0';
                ++sub)
              if ( inp_name[sub] == '\n' )
                   inp_name[sub] = '\0';
              /* Replace newline by end of string marker */

         file_d = fopen ( inp_name, "r");
         /* Open the file for reading */
         if ( file_d == NULL )
              fprintf(stdout,"\n\nUnable to open file\n     <%s>\n\n",
                              inp_name);
    }

/*                Listing 3 part 1                          */
```

so long as the following expression is true:

(inp_val = getc (file_d)) != EOF;

The getc function is used to read a character from the file associated with the descriptor file_d — this is the file that has just been opened. The getc function is rather like getchar except you must use a file descriptor to tell it where to read data from. The value that is retrieved is stored inside the variable named inp_val. If you check the program listing you should find that inp_val is declared as an int — not as a char. This is not a mistake — it is absolutely vital!

The for loop carries on repeating so long as inp_val is not EOF. What is EOF? Perhaps you've worked out that it must be a symbolic constant that is defined for you in stdio_h. It is! In Digital C EOF is given the value -1. (Some other implementations of C use different values, but this doesn't matter.)

When you use getc to read data from a file and there is no data left (because you have reached the end of the file) Digital C arranges it so that a value equal to EOF is passed back to your program. The value of EOF has to be different from all possible real character values — so it has to be outside the range 0 through 255. If EOF was the same as a valid Ascii character code it would be very difficult to work out if you'd reached the end of the file or just a funny character part way through the file. This is why inp_val must be declared as an integer. If it was defined as a char you would get problems. Char variables can only hold values in the range 0 through 255 — they cannot hold -1 (EOF). This means that if EOF was passed back to a char variable it would end up being converted to a value somewhere in the range 0 to 255. Because of this it would always not equal EOF — and the for loop would never end.

It is easy for a newcomer to C to declare characters that are being read from files as char instead of int, but if you do then you can expect problems! You won't be able to spot when the end of the file is reached.

Inside the for loop the putc function is used to copy the character that was read from the file to the stdout device. The rest of the for loop is concerned with stopping the display after a screen full of data has been displayed - rather than letting it all scroll off the top of the screen. The basic logic used is as follows:

```
   putc('\n',stdout);
   for ( num_lines = 0, num_chars = 0;
          (inp_val = getc(file_d)) != EOF;

        )
    {
         putc(inp_val, stdout);
         if ( inp_val == '\n')
         {
              ++num_lines;
              num_chars = 0;
         }
         else
         {
             ++num_chars;
         }
         if ( num_chars == CHARS_PER_LINE )
         {
              ++num_lines;
              num_chars = 0;
         }
         if ( num_lines == LINES_PER_PAGE )
         {
              fprintf(stdout,"\nPress a key to continue - ESC to quit\n");
              if ( fgetc(stdin) == ESC )
                   exit();
              num_lines = 0;
              num_chars = 0;
         }
    }
    fclose (file_d);
    fprintf(stdout,"\nPress a key to quit");
    inp_val = getchar();
}

/*         Listing 3 part 2                        */
```

1. If the character is a newline (\n) add 1 to num_lines and set num_chars to zero - since no characters have been displayed on the new line.

2. If num_chars equals CHARS_PER_LINE the line is full so add 1 to num_lines and set num_chars to zero.

3. If num_lines equals LINES_PER_PAGE then the screen is full. Output a suitable message and wait for a key to be pressed. If ESC is pressed then exit the program, otherwise set num_lines and num_chars to zero and carry on around the loop to display another screenful.

At the end of the loop the file is closed using fclose. The syntax is:

```
fclose ( file_descriptor ) ;
```

Once the file has been closed the file descriptor could be used to open another file.

Exercise three

1. What would happen if the \n was not removed from the file name that was input using fgets? Try it out.

2. What happens if you miss out the quotes from the "r" in the fopen function?

3. What happens if you try to get characters from a file that has not been successfully opened?

4. the fclose function actually returns an integer value. A value of zero means that the file was successfully closed, -1 means that it wasn't. Amend the coding so that it checks that the file was closed properly. One way of checking that this works is to write a program that tries to close a file that has not been opened. Picking up the value returned by fclose involves using:

```
integer_variable = fclose (
file_descriptor );
```

5. In listing three the final for loop reads characters and displays characters. Try to re-write the whole thing so that it reads lines and displays lines using fgets to read them and fprintf to write them out again. You will need to define a character array in which to read the lines - and fix it so that the call to fgets never reads more than, say, 73 characters (a screen line full). You need to know that the fgets function returns a value of NULL if the end of file was reached.

```
integer_variable = fgets ( .... ) ;
if ( integer_variable == NULL )
{ /* END OF FILE REACHED */
}
```

6. Change the program so that it counts the number of words and lines in the input file.

7. Write a program that does the following:

a) Gets an input file name from stdin and opens the file for reading.

b) Gets an output file name from stdin and opens the file for writing.

c) Copies the whole of the input file into the output file.

Be careful when you test this program - remember that if the output file already exists its contents will be deleted. It is probably best to play with a blank disk or microdrive cartridge.

Change the program so that it appends the contents of the input file to the output file.

C is a much more versatile language than we have so far seen. Some of this extra versatility will be covered in future articles.

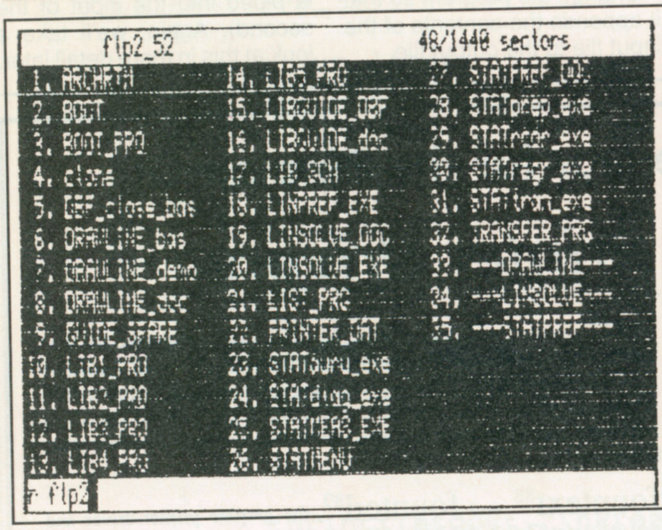
For example it is possible to get a C program to read stdin data from a file instead of from the keyboard - and to write stdout data to a file rather than to the screen. All of this can be done without having to re-write the program at all. This process is sometimes called redirection (reading stdout from somewhere different from usual and writing stdout to somewhere different). One of the few areas in which the MS-DOS operating system is more convenient than Qdos is in the way it allows the redirection of stdin and stdout. In a future article we'll look at how this can be achieved using Digital C.

It is also possible to take the stdout output from one program and pass it into the stdin of another program. This process is sometimes known as piping (the output of the first program is piped into the input of the second). Again, we'll take a look at this in more detail later.

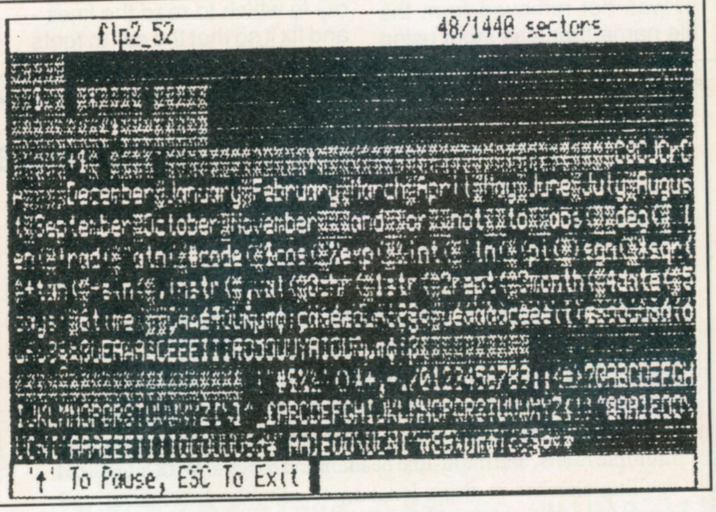
FILE EXECUTIVE

Bryan Davies tries a thoughtfully-constructed new file-handling utility with a reasonable price-tag.

INFORMATION:
Program: File Executive V1.00
Price: £12
Supplier: J. O'Connor
16 Parsons Way
Wooton Bassett
Wiltshire SN4 8DA



```
     flp2_52                     48/1440 sectors
1. ARCHRTH         14. LIB5_PRG          27. STATPREF_DOC
2. BOOT            15. LIBGUIDE_DBF      28. STATprep_exe
3. BOOT_PRG        16. LIBGUIDE_doc      29. STATrcor_exe
4. clone           17. LIB_SCH           30. STATregr_exe
5. GBF_clone_bas   18. LINPREF_EXE       31. STATtran_exe
6. DRAWLINE_bas    19. LINSOLVE_DOC      32. TRANSFER_PRG
7. DRAWLINE_demo   20. LINSOLVE_EXE      33. ---DRAWLINE---
8. DRAWLINE_doc    21. LIST_PRG          34. ---LINSOLVE---
9. GUIDE_SPARE     22. PRINTER_DAT       35. ---STATPREF---
10. LIB1_PRG       23. STATguru_exe
11. LIB2_PRG       24. STATdiag_exe
12. LIB3_PRG       25. STATMEAS_EXE
13. LIB4_PRG       26. STATMENU
r flp2
```



```
     flp2_52                     48/1440 sectors
'↑' To Pause, ESC To Exit
```

File Executive is a file-handling utility, and comes on $5^1/_4$in or $3^1/_2$in disk only. Instructions for use (11 pages) are supplied in both a *Quill* and an *Editor* document on the disk with the program. The program file is compiled *Super Basic* and requires a *Turbo Toolkit* extensions file to be loaded before it. A run-time set of extensions is loaded automatically by the boot, unless you choose to by-pass it because you already have an equivalent file loaded. The provided commands work with microdrives or floppy drives, but not with hard disk.

The instructions are quite comprehensive, and the available functions are what one would now expect, after having had the use of Toolkits for several years; some of the functions are a bit unusual, though. The command format is much similar to that used in The Editor — mainly mnemonic, single-letter, with parameters where appropriate. The interface with the user is through three windows, the main one to display requested information, and smaller, single-line ones above and below it to give basic information on the medium in use and to allow the input of commands to give error messages.

The program file is about 36KB in size, and takes a total of around 50KB when in use. This size limits it to being the only job on a basic 128KB QL, but it shouldn't restrict normal operations on a system with plenty of expansion memory.

The similarity of style to The Editor extends to the start-up and Help screens. Help is clearly laid out, and sufficiently descriptive without occupying too much screen space. You can manage without the written instructions. The help file is not loaded into memory, so the program disk needs to be in the appropriate drive for help to be available.

The commands available include the usual Directory, Copy, Delete and Format, but not Rename. In addition, you can View the contents of files, Find and Exchange strings within files, Get information on a file, Set the Dataspace allocated to a program file, View the Status of a medium, and Print the Current Directory. Command files (as in Editor) can be used, commands can be chained one-after-the-other, and they can be repeated.

The single-letter commands are easy to remember, being in mnemonic form. For example, the format of the command for displaying a directory on the screen is:

```
r [device] {[specifier]}
```

where "r" indicates read, "[device]" is the device name in the usual form — flp1 or flp1_ will do — and "{[specifier]}" is for a string within a file or files. If a specifier is present, the directory displayed will be of only those files which contain the string. The instructions are a bit misleading here, in that the specifier string has to be a part of the file name, rather than being part of the file itself as implied by the description of the function. It is a pity the square brackets are used in the opposite sense to that many users will be familiar with; that is, the presence of [] indicates a mandatory parameter, rather than the more-usual optional one. The curly brackets are used to enclose optional items; using square brackets within curly ones seems unnecessary. Several commands can be entered in sequence, separated by semi-colons. For example:

```
r/flp2;bz
```

will cause a directory of all

files on flp2 to be displayed, followed by a buzz from the QL's magnificent speaker. Commands — including several on one line — can be repeated by preceding them with the figure for the number of repeats, as in <10(bz;r/flp1)> to make the speaker buzz ten times, with the directory of flp2 being displayed after each buzz. An indefinite repeat is obtained by using <rp> — <rp bz> will keep the speaker buzzing continuously. All this will be familiar to the users of The Editor.

The QL function keys are used for familiar operations; F1 brings up help information, F2 recalls the last command used, F4 rewrites the screen, and F5 enables the program main window size to be altered and the position to be changed, using the cursor keys, alone and with ALT. Commands can be aborted during execution by pressing ESC.

Parameters can extend the utility of commands considerably, and a decent selection is offered. Numbers, text or strings can be used, strings having to be enclosed in quotation marks. Quotation marks are also needed when characters are used which the program would otherwise recognize as "for other purposes"; for example, a name which includes spaces must be enclosed by quotes. Files can be identified by Ranges, such as 1,3,5 or 10-20, or a mixture of the two forms. Parameters can be separated by full stops, spaces, commas or slashes.

The Copy command includes the Wild Card option, and the standard Y/N/A/Q (yes/no/all/quit) query, including when overwriting is necessary. You can specify a range of files (by number) to be copied; when only one file is specified, you can also give the file copy a different name from the original. The same format is used with the Delete command, Files can be viewed; they normally pass through the window in a continuous stream, but the up-cursor key pauses the flow, and ESC stops the operation. Finding and Exchanging Strings within files is a similar operation to Viewing files, but the specified string will be displayed in inverse video whenever it is encountered, and the number of occurrences of the string will be displayed when the operation has been completed. The format for a find and exchange is:

```
es [filenumber] [device]
[source] [dest]
```

where "[source]" and "[dest]" are the string to be found and the one to replace it by, respectively.

The Getting and Setting Information function reads the file header and displays the file name, length, date of creation, type (job or data), and dataspace (if the file is a job). Combined with this command is one to permit the user to alter the allocated dataspace for a job. The Format command is basic, and accepts no parameters other than the device name; you are asked to confirm that you wish to proceed with the format.

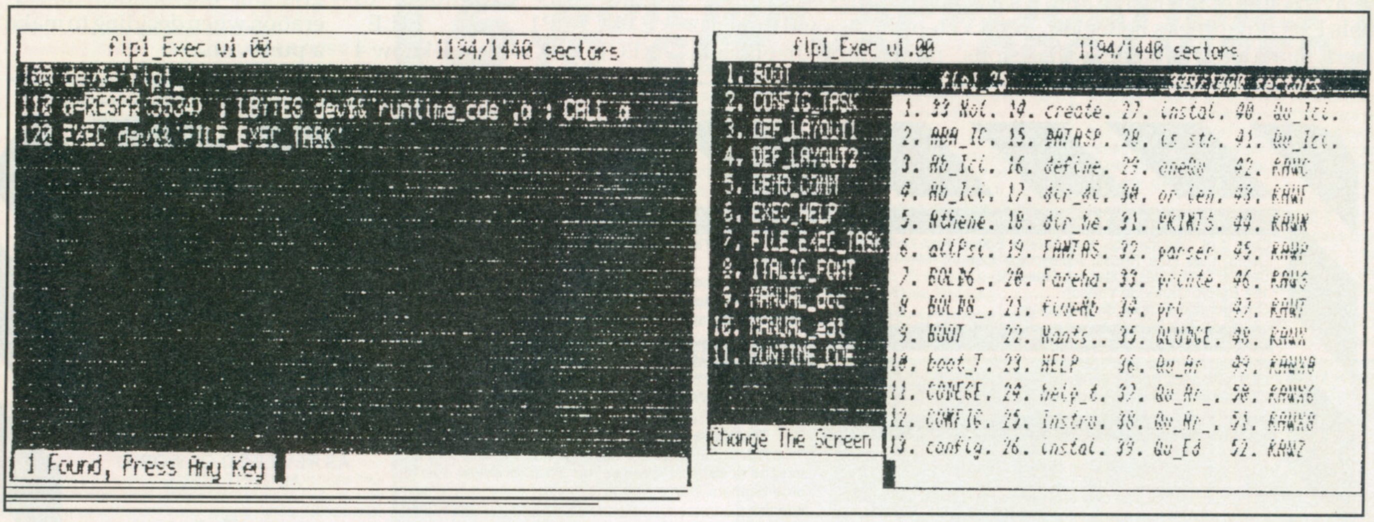
It is a pity no provision was made for giving the medium a Volume Label, nor for formatting to anything other than a default number of sectors. Viewing the Status of a medium produces a display of the medium label, write status (protected or not), total and available storage space in sectors and kilobytes, number of files present, number of files selected (by a previous command), search string (specified with the Read/Directory command).

Command files can be written with any WP or editor program, provided you can ensure the resulting file does not contain unwanted control codes. Quill is suggested as being usable for this purpose; presumably you would have to Print the file to disk/cartridge with no printer-driver present. The supplied demonstration command file is shown in the illustration, which also shows the alternative display fount and another window position. As can be seen, several lines of commands are possible. Comments can be inserted into a command file by prefacing the line with an asterisk. You can also display a prompt such as "Insert disk into flp1" in the command window, and cause the command file to be paused until the key is pressed.

There is a Zap command, which çlears information on the current medium from memory (without doing anything to the medium!) The currently -displayed directory can be sent to a printer (the port can be specified) and printed in the same format as it is displayed, using an 80-character width. The buzzer command (bz) could be handy if the user wants warning of when a long command sequence has been completed. The number of columns in the window display defaults to three, which is probably the best choice for most users, but the illustration shows four displayed, with truncated file names being terminated with a full stop to show that they are incomplete. The display colours can be altered within the basic QL range of 256 "colours"; this is, perhaps, only a frill, but continuous use of a utility can lead to irritation with certain colours or combinations. Somewhat unexpected in a cheap utility is the facility to change the case of file names, the options being lower-, mixed- or upper case. The Sort facility allows files to be sorted into alphabetical order when a directory is read. Subsequent copying will be done in the displayed order, not the actual one on the medium.

A routine CONFIG_TASK is provided, to enable the user to specify the name of the defaults file, which is read when the program starts running, and the location of the help file. Error messages from the program, and from QDOS, are displayed in the small, lower window. The instructions explain the program messages (11 of them) and you are referred to The QL *User Guide*, Concepts section, for explanation of QDOS messages. The program did not "fall over" when errors occurred during the review.

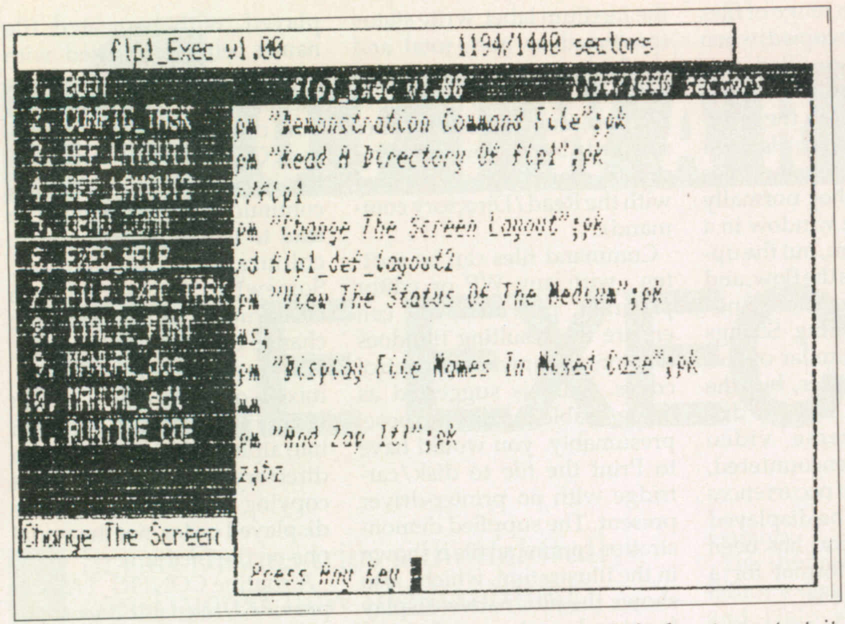
File Executive worked alongside several other EXEC-ed jobs on the review system without apparent interaction. It is not a pop-up job, as the comparable program Files (from Sector Software) is, but you can use CTRL-C to return to it from the current application, rewriting the display using F4 if necessary. The displaying of a basic directory is done quickly, the default display being three columns wide by 13 rows deep



```
flp1_Exec v1.00                1194/1440 sectors
100 dev$='flp1_
110 a=RESPR(5534) : LBYTES dev$&'runtime_cde',a : CALL a
120 EXEC dev$&'FILE_EXEC_TASK'

1 Found, Press Any Key
```

```
flp1_Exec v1.00                1194/1440 sectors
1. BOOT             flp1_25                  393/1440 sectors
2. CONFIG_TASK     1. 33 Rol.  14. create.  27. instal.  40. Qu_Ici.
3. DEF_LAYOUT1     2. ABA_IC.  15. DATASP.  28. is str.  41. Qu_Ici.
4. DEF_LAYOUT2     3. Ab_Ici.  16. define.  29. oneQu    42. RAWC
5. DEMO_COMM       4. Ab_Ici.  17. dir_di.  30. or len.  43. RAWF
6. EXEC_HELP       5. Athene.  18. dir_he.  31. PRINTS.  44. RAWN
7. FILE_EXEC_TASK  6. allPsi.  19. FANTAS.  32. parser.  45. RAWP
8. ITALIC_FONT     7. BOLD6_.  20. Fareha.  33. printe.  46. RAWS
9. MANUAL_doc      8. BOLD8_.  21. fiveAb   34. pri      47. RAWT
10. MANUAL_edt     9. BOOT     22. Rants..  35. QLUDGE.  48. RAWX
11. RUNTIME_CDE   10. boot_T.  23. HELP     36. Qu_Ar    49. RAWX8
                  11. CONEGE.  24. help_t.  37. Qu_Ar_.  50. RAWX6
                  12. CONFIG.  25. Instro.  38. Qu_Ar_.  51. RAWX8
Change The Screen 13. config.  26. instal.  39. Qu_Ed    52. RAWZ
```



```
flp1_Exec v1.00                    1194/1440 sectors
 1. BOOT                 flp1_Exec v1.00      1194/1440 sectors
 2. CONFIG_TASK   pm "Demonstration Command File";pk
 3. DEF_LAYOUT1   pm "Read A Directory Of flp1";pk
 4. DEF_LAYOUT2   r/flp1
 5. DEMO_COMM     pm "Change The Screen Layout";pk
 6. EXEC_HELP     rd flp1_def_layout2
 7. FILE_EXEC_TASK pm "View The Status Of The Medium";pk
 8. ITALIC_FONT   ms;
 9. MANUAL_doc    pm "Display File Names In Mixed Case";pk
10. MANUAL_edt    mm
11. RUNTIME_CDE   pm "And Zap It!";pk
                  z;bz
Change The Screen
                  Press Any Key
```

— that is 39 files at a time. Resizing the window to the maximum screen size increases the potential number of files displayed to 96. No doubt many users will be irritated by the way the QL's built in DIR command, and the add-on commands of other utilities, restrict the number of files displayed simultaneously, and File Executive does a better job here. Some long file names will get truncated when the default window size is used, and if you use the option to increase the number of display columns above the default of three, but the user has the freedom to set whatever number of columns and the window size that best suit the file names used. A defaults file allows the user to save these settings — screen X and Y co-ordinates and dimensions, number of displayed columns, ink and paper colours, Copy speed (in terms of the basic QL methods of copying and the use or otherwise of the Overwrite query), sort facility on/off, and the device name for the default fount file.

One operational problem was noted. When the program was quit, in the prescribed fashion (using the <qu> command), in order to restart it with the default settings, it refused to load; initially, the message from QDOS was "out of memory" (the free space reading was 175Kb) when the command <EXFILE_ EXECT_ ASK> was used, changing to the old-familiar Turbo one of "Task...halted, after line 0..." when the command was expanded to <EXECFLP1_ FILE_ EXECT_ASK>. Yes, I know I should remember the answer to that one! The average user will not be aware of what to do next, though.

Maybe, some day, someone will properly sort out the problems associated with extensions which are "present", but not "correct". Another minor problem was that the demonstration file moved the window and changed the screen fount, and you would have to use both the Fount and F5 commands to get back to the settings (assuming you remembered the position and size figures).

All-in-all, the program is a workmanlike job, operates smoothly and speedily, and has little to its debit. Thoughtful touches, like ensuring that the results of a command are displayed immediately, without the user needing to take further action, show that the writer has given serious thought to the project. The price is reasonable; no doubt there are still some users who do not have any of the well-known utility programs, and they should give this one serious consideration when deciding to make a purchase.

★ POCKET SIZED ★ PC-COMPATIBLE ★ PERSONAL ORGANISER ★

ATARI Portfolio

THE PC IN YOUR POCKET

The new Portfolio from Atari is the world's first pocket-sized electronic organiser that is also a powerful PC compatible computer, with a full QWERTY keyboard and scrolling 80×25 character display.

Portfolio includes everything you would expect from an electronic organiser - an address and telephone book, time manager, diary and sophisticated calculator.

Portfolio also includes several functions that you would not expect - a spreadsheet for your personal budget and expense records, as well as a text processor for typing memos and letters.

And, because Atari's Portfolio is PC compatible, it can communicate with your desktop PC at home, or in the office. You can transfer files from one to the other, to enable you to update your reports and figures with the Portfolio while you are on the move.

FOR ONLY **£199** INC VAT

SPECIFICATION

- ★ ***Processor:*** Intel 80C88 at 4.9152Mhz.
- ★ ***Operating System:*** Compatible with MS-DOS 2.11.
- ★ ***Internal ROM:*** 256K containing BIOS, operating system and applications suite.
- ★ ***RAM:*** 128K with an internal RAM disk, configurable from 8K. Externally expandable to 640K RAM.
- ★ ***Keyboard:*** 63 keys, QWERTY, IBM PC BIOS compatible. Buried numeric pad and function keys. Optional key click.
- ★ ***Character Set:*** Extended IBM ASCII (255 characters).
- ★ ***Mass storage:*** credit card sized memory cards (32K or 64K or 128K RAM).
- ★ ***Display:*** Graphics LCD, supertwist technology, MDA compatible, 40 columns x 8 lines, 240 x 64 pixels (with the option to window a full 80 x 25 character display). Keyboard controlled contrast.
- ★ ***Peripherals:*** 60 pin expansion BUS to take serial and parallel ports and memory expansion units.
- ★ ***Size:*** 8" x 4" x 1" (200mm x 105mm x 29mm).
- ★ ***Weight:*** 495 grammes (with batteries).
- ★ ***Applications:*** calendar and diary, address and phone book, Lotus 1-2-3 compatible spreadsheet, text processor, communications software.

COMING SOON!

In addition to the excellent software built-in to the Portfolio Free of Charge (see right), other software and peripheral products, such as the sophisticated Pocket Finance package and serial/centronics interfaces are available. And it doesn't stop there. Many manufacturers have recognised the potential of the Portfolio and have already started to design new peripherals and software. Products currently under development include: Serial interface with built in mini modem, Apple Macintosh interface, business, utility and programming software plus a range of adventure and battle strategy games. For further free details on the Portfolio range, fill in the coupon below and return it to Silica Systems now.

5 BUILT-IN APPLICATIONS TO GET YOU ORGANISED!

To get you started, Portfolio comes with a suite of five useful functions built-in, all accessible from a simple menu display.

TIME MANAGER

More than just an electronic diary, the Portfolio Time Manager enables you to plan your appointments via a comprehensive calendar and diary. It can even be programmed to sound an audible alarm at specific times to remind you of important appointments.

CALCULATOR

Your Portfolio will be invaluable in the office or at home as a powerful pocket calculator. It has a full range of functions, including factorial, power and root calculations, all with multi display formats and memories.

ADDRESS BOOK & DIALLER

Portfolio has a complete address book facility that allows you to store hundreds of addresses and phone numbers. And, at the touch of a button, you can retrieve any one of them, or search for a specific grouping, such as "all Italian restaurants". And, when you are ready to book your table, hold your Portfolio to your telephone mouthpiece and use its special built-in tone dialler to dial the number for you.

TEXT PROCESSOR

The Portfolio's built-in text processor program includes word wrap, line and column count, string search, in fact most of the functions you would find in a word processor. It handles printer and word processor control codes and allows easy transfer of files between Portfolio and your desktop PC.

SPREADSHEET

For real calculating power, Portfolio has a Lotus 1-2-3 compatible spreadsheet built-in. It has 127 columns x 225 rows and reads/writes Lotus V1.0 and V2.01 files, so you can transfer data to and from Lotus 1-2-3 on your desktop PC. The Portfolio's 256K ROM includes MS-DOS and PC BIOS compatible systems software.

MEMORY CARDS

Portfolio can store and retrieve data and programs from its own RAM, or from small credit card size memory cards, that slot into its built-in card drive. The cards are available in three sizes, 32K, 64K and 128K, so you can carry a library of data in your pocket. The card drive also accepts ROM cards, which can contain commercial or custom software.

POWER SUPPLY

Portfolio is powered by three AA batteries which will run for up to six weeks with normal use, or from the mains using an adaptor. All the peripherals take their power from the Portfolio, so no extra batteries or adaptors are required. A 'battery-low' warning and memory back-up ensure that information is not lost when the batteries are changed.

INTERFACES & PERIPHERALS

Portfolio can communicate with other computers and supports a growing range of peripherals via a built-in 60 pin bus connector. Peripherals available include serial and parallel interfaces and memory expanders (to 640K). You can also add a card drive to your desktop PC, to enable it to read/write to Portfolio's cards.

SILICA SYSTEMS OFFER YOU

FREE OVERNIGHT COURIER DELIVERY: On all hardware orders shipped in the UK.
TECHNICAL SUPPORT HELPLINE: Team of Atari technical experts at your service.
PRICE MATCH: We normally match competitors on a "Same product - Same price" basis.
ESTABLISHED 12 YEARS: Proven track record in professional computer sales.
£13M TURNOVER (with 60 staff): Solid and reliable with maintained growth.
BUSINESS/EDUCATION/GOVERNMENT: Volume discounts available for large orders.
SHOWROOMS: Demonstration and training facilities at our London & Sidcup branches.
THE FULL STOCK RANGE: All of your Atari PC requirements from one supplier.
FREE CATALOGUES: Will be mailed to you with offers and software/peripheral details.
PAYMENT: By cash, cheque and all major credit cards.
CREDIT PAYMENT TERMS: Silica are licensed credit brokers - details on request.

Before you decide when to buy your new Atari Portfolio, we suggest you think very carefully about WHERE you buy it. Consider what it will be like a few months after buying your Portfolio, when you may require additional peripherals or software, or help and advice with your new purchase. And, will the company you buy from contact you with details of new Portfolio products? At Silica Systems, we ensure that you will have nothing to worry about. Silica Systems is a new division of Silica Shop, the UK's leading Atari specialists. This new division has been established to provide a service to the more serious home user, as well as to business and education purchasers. Silica have been established for over 12 years, and have an annual turnover of £13 million. With our unrivalled experience and expertise, we can now claim to meet our customers requirements with an understanding which is second to none. But don't just take our word for it. Complete and return the coupon now, for our latest Free literature and begin to experience the "Silica Systems Service".

SILICA SYSTEMS

MAIL ORDER:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-309 1111**
Order Lines Open:	Mon-Sat 9.00am-6.00pm	No Late Night Opening	Fax No: 081-308 0608
LONDON SHOP:	52 Tottenham Court Road, London, W1P 0BA		**Tel: 071-580 4000**
Opening Hours:	Mon-Sat 9.30am-6.00pm	Late Night: Thursday until 8pm	Fax No: 071-323 4737
SIDCUP SHOP:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-302 8811**
Opening Hours:	Mon-Sat 9.00am-5.30pm	Late Night: Friday until 7pm	Fax No: 081-309 0017
BUSINESS/EDUCATION:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-308 0888**
Order Lines Open:	Mon-Fri 9.00am-6.00pm	Closed on Saturdays	Fax No: 081-308 0608

To: Silica Shop Ltd., Dept QLW 02/91-38, 1-4 The Mews, Hatherley Road, Sidcup, Kent DA14 4DX

PLEASE SEND INFORMATION ON PORTFOLIO

Mr/Mrs/Ms: .... Initials: .... Surname: ....

Address: ....

....

.... Postcode: ....

Tel (Home): .... Tel (Work): ....

Company Name: ....

Which computer(s), if any, do you own? ....

E&OE - Advertised prices and specifications may change - please return the coupon for the latest information.

File Transfer: QL TO PC

Robin Stevenson joins the dots.

The transfer of text between the QL and a PC is a recurring need for many users, and there are even commercial programs to assist in this. The suggestion detailed here is a DIY system, to transfer text without having to leave your QL program such as *Quill*. This simplifies the process, and removes the need for temporary _lis files. As far as the program is concerned, it is printing the document. However, instead of the lead being plugged in to the printer, it will be attached to the serial socket of a PC.

There are three elements required: 1. A suitable lead. 2. A suitable printer driver. 3. MS-DOS commands to get the PC to receive the incoming data.

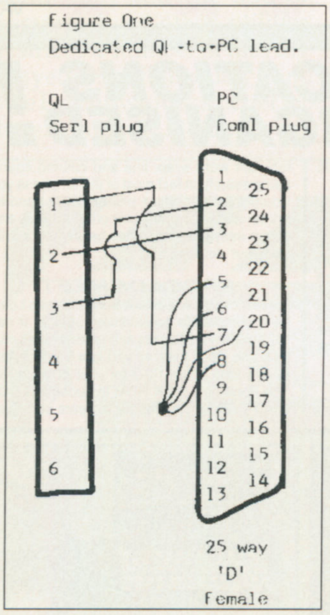


Figure One
Dedicated QL-to-PC lead.

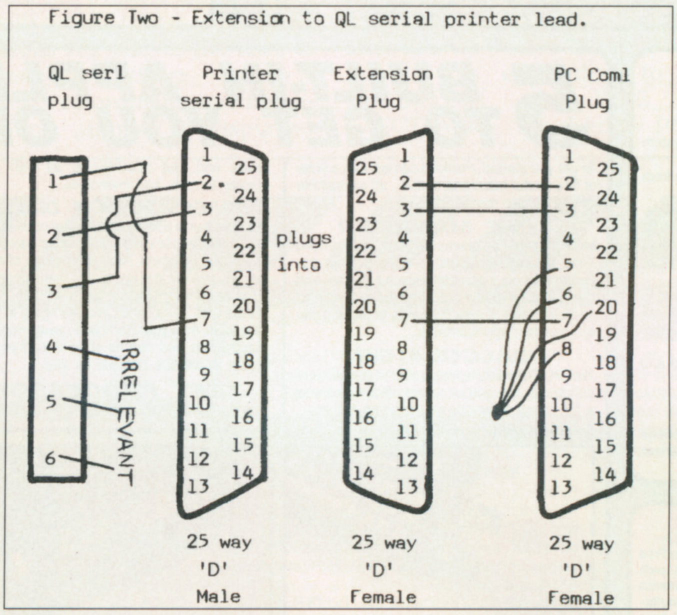


Figure Two - Extension to QL serial printer lead.

Figure Three
Psion printer driver, for
WordStar transfer format.

```
NAME          : QLtoPC
PORT          : serl
BAUD RATE     : 1200
PARITY        : none
CONT. FORMS   : yes
END OF LINE   : CR,LF
PREAMBLE      : none
POSTAMBLE     : SUB
BOLD ON       : STX
BOLD OFF      : STX
UNDER ON      : DC3
UNDER OFF     : DC3
SUBS. ON      : SYN
SUBS. OFF     : SYN
SUPER. ON     : DC4
SUPER. OFF    : DC4
TRAN 1        : '£,27,156,28
```

To start with the lead, getting the right connections is a manageable DIY job, provided you have access to a lead for the QL serial socket. Many users already have a serl-to-25 pin serial printer lead. If so, the necessary adaptor can be made as a short extension to that. The only wires that need to go all the way through are the signal ground (pin 1 on the QL; pin 7 on the PC or printer plug), and the data lines, (pins 2 and 3 in both plugs). The data lines need to cross over, so that the QL's pin 2 goes to the PC's pin 3, and vice versa. In most printer leads, this cross-over will already be present, so the extension just carries the three wires directly across to their matching pins.

In addition, you need to convince the PC that something is about to happen. To do this, pins 5, 6, 8 and 20 all need to be joined together. The simplest way to do this is to solder short sections of wire to each of these pins, and then solder the other ends together in one big, ungainly lump. Do this on a short extension to your printer lead, so that you can still use the original for your printer. The extension can also reverse the gender of the lead. A PC will require a female end to the lead, whereas most printers expect a male one. **Figure One** shows how you would make up a single, dedicated lead. **Figure Two** shows our hypothetical printer lead, plus the extension needed to make it suitable.

The Psion printer driver incorporates features which make it very useful for transfers. In particular the ability to specify any characters for bold, underline etc. makes it possible to convert your embellishments to *WordStar* format, during the transfer. Any word-processor which allows importing of WordStar format text (such as *LocoScript PC*) will be able to make sense of these. It may be best to have no left hand, top or bottom margins, and no headers or footers. These can then be set by the receiving program.

Figure three shows a suggested printer driver, to produce WordStar compatible text. If you don't use the print enhancements any other word-processor should be able to use the result, if it can use or import standard Ascii files. The really vital settings are the baud rate, which must match the PC, and the Postamble, which must send the Control-Z value (Ascii 26, or SUB), to denote the end of the file. If you put this printer driver on a separate cartridge from Quill, you need only insert it into MDVl_ when you wish to transfer. The rest of the time, your normal printer driver will be used.

Finally there are the PC commands. To set up the PC's serial port, you need to enter

MODE COM1:1200,n,8,2

which is an MS-DOS external command. It configures serial port 1 with appropriate baud rate, parity, data bits and stop bits. Then, for every file you wish to transfer, before sending the print instruction, you need to enter

COPY COMl FILENAME.TXT

where filename.txt is your own choice of name, including any appropriate path, and extension. A useful variant on this, while testing is

PY COMl CON

which will send it to the PC's screen instead of to disk.

Exactly the same technique should be usable from any other program. The same printer driver can be used to send Archive and Abacus reports, and provided you can set the baud rate, and send the Control-Z end-of-file marker, any other program that can print out could use the same technique.

If you need to transfer QL files directly, rather than by printing them, you can do this by using the SuperBASIC command

BAUD 1200

After entering the MS-DOS COPY command, you can initiate the transfer from the QL with the line

COPY_N MDV2_FILENAME_EXT TO SERlZ

JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 – 4100 Duisburg 11 – W-Germany
Tel./Fax 0203/501274

QL-Emulator for ATARI ST

Emulator (Hardware, QJUMP's Drivers (including WIN). AtariDOS and more...) **£163**
EPROM Cartridge for ST (makes machine autoBOOT) **£29.50**
Floppy Adaptor to connect QL-Disk-drives to the ST **£8.90**
Other ATARI adaptors, Hardisks etc, at very low prices!

Lower Prices!

QL-Hardware

ABC-Keyboard Interface (for IBM PC/XT keyboards) **£41**
High Quality Mouse for QIMI or ATARI ST **£26**

Software

QD – NOW VERSION !!!
Many additional features! Please write for information about all new features, price and upgrades from Version I or II.

QMENU
Many QD-Users will know it: the Menu Extension. Now with more facilities, even real subdirectories are treated in a new file-select window. Programming instructions and examples for SuperBASIC and machine code. **£10.50**

DIAMONDS
New Game. This is definitely a 'just one more go' game to keep you up at night, a kind of action-puzzle. **£10.50**

QSHANG
QL version of the well-known game. Very nice graphics, 1 or 2 player game and many m ore. **£15**

EASYPTR
Create your own Pointer Environment Menus and sprites and use them in your own SuperBASIC or machine-code programs. Many users waited for this package, as creating Menus is not very easy – until now. Put all the items you need on screen – that's it! Supports all the PE facilities, even split sub-windows and also uses the Menu Extension! Many examples. **£39.00**

THE PAINTER
An excellent graphics program for MODE 4 & 8. Uses Pointer Environment, so that it may be controlled by QIMI or SuperQBoard Mouse. Works fine with QL Emulator. **£29.50**

QLQ £20 Thing & EPROM Manager £18 **Firebirds £10.50** **Brain Smasher! £12.00**
QSUP £26 **Arcanoid II £10.50** **Ion Gold & Doppel Ion £10.50**

Please add £2.10 for postage and package. All programs except Arcanoid, Firebirds & Ions need memory expansion. All prices excl. V.A.T. E&OE

TF Services

ASTRACOM Modem

Intelligent buffered modem with text status messages. Hayes protocol, parallel printer port (can be used as 6k serial to parallel printer buffer). 240vac or 9-12vdc. Autodial/answer and many programmable registers incl printer logging of RX and/or TX data. BT approved. Can be used with any computer.

NEW V22/22BIS WITH V2.03 FIRMWARE NEW

AC1000D (V21/23 v1.5a)	£175
AC1000T (V21/23/22 v2.03 + tone dlng)...	£275
AC1000Q (V21/23/22/22bis V2.03) (tone) ..	£299
V22/V22bis upgrades	ring for prices

QL REPAIRS

QLs tested with Thorn EMI test rig
and ROM software
(MDV hardware extra if required)

£25 – 6 month guarantee

London STD codes

London 01- dialling code is changing to 071- or 081- in May 1990. Use our look-up tables and text (export file) conversion program to save HOURS of work.

Disks contain 5 text lists in database import formats (quote delimited, etc) & a program for text (ie export) file conversion. Also provided:

IBM PC	: Files for major databases/sprdshts
QL	: Psion Archive .dbf file

3.5/5.25 disk or microdrive – £12

QuaLsoft Terminal

Viewdata/VT52/VT100 QL TERMINAL EMULATOR

'Een deluxe communicatieprogramma van Kualsoft'

Multitasking for electronic mail, PRESTEL etc, CET download. Phone directories for ALL (yes ALL) modems for the QL, autodial (where poss) and logon, two-way FILE TRANSFER to ATARI ST, IBM PC, Psion Organiser (via comms link), XMODEM, real time clock/timer, buffered logs to file/printer. transmit files, editable command line, Swedish/Norwgn/German options, translates, editor etc.

SOLVES ALL PACKAGED MODEM SOFTWARE PROBLEMS

Unbuffered modems usable with Miracle Modaptor

Software (3.5" or mdv) & manual £30

QL SPARES

Faulty QL boards (excl plug in ICs)..... **£9**

Keyboard membrane...	**£7**	Circuit diags...	**£2**
68008 cpu...........	**£8**	1377 PAL.........	**£3**
JM rom set.........	**£12**	JS rom set.....	**£25**
8302 ULA...........	**£10**	8301 ULA.......	**£10**
8049.............	**£7.50**	MDV ULA	**£12**

Other components (sockets etc)... plse PHONE

COMPUTER CLEANER

Much improved mains filters. 40-80 db cut & 3 spike filters. 390 joule cut (3 & 4 way)

Certified X/Y 240ac capacitors

2-way (5a adaptor)......................	£14
3-way (5a adaptor)......................	£18
4-way (13a trailing socket)............	£24

FILE TRANSFER

Programs to transfer text AND programs error free (XMODEM) between computers

Available for IBM PC and compatibles, ATARI ST and Sinclair QL. Connects to Psion organiser via Psion comms link

QuaLsoft program (per m/c).....	£7.50
Serial lead (name 2 computers)...	£10
Complete package for 2 computers.	£25

All prices include VAT and postage in the UK **MAIL ORDER ONLY**

VISA

12 Bouverie Place, London W2 1RB (tel: 071-724 9053)
SCROLLING BBS FOR ORDERING/QL HELP – 071-706 2379 (All speeds to V22bis)
Fax: 071-706 2379 Telex: 265451 (ref: 72:MAG90645) Prestel: 017249053

AXXON COMPUTER SUPPLIES

POSTAL ORDER TO :
AXXON COMPUTER SUPPLIES, (QL)
23 SUNRAY AVENUE,
SURBITON,
SURREY KT5 9RA.

ALL PRICES INCLUDE VAT AND 1st CLASS DELIVERY

3½" DISKS 100% Certified, Error Free, DSDD

10..........................	£4.60
20..........................	£9.00
50	£21.50
100	£40.00

5¼" DISKS, 100% Certified, Error Free, 80 tracks.

10	£3.50
20	£6.50
50	£15.50
100	£30.00

DISK BOXES (Lockable)

3½" 40 CAPACITY	£6.95
80 CAPACITY	£8.95
5¼" 50 CAPACITY	£6.50
100 CAPACITY	£8.50

DISK DRIVE HEAD CLEANER

3¼" Deluxe Cleaning Kit	£3.95
5¼" Deluxe Cleaning Kit	£3.95

RIBBONS, Highest Quality Guaranteed	1off	3off
STAR LC-10	£3.95	£3.50
CITIZEN 120 D	£3.95	£3.50
EPSON FX80, MX80, LX80, FX100	£P.O.A	£P.O.A

NEW!! Re-ink (Re-Inks Fabric Ribbons) £12.95

☎ **Call - 081 - 335 3180** VISA Access

Ptices are for mail order only. We reserve the right to alter specifications and prices without notice. Quantity orders please ring for prices. Government, Educational Authority order welcome.

☆ sinclair QL ☆

Sinclair QL COMPUTERS & PRINTERS
QLs FROM £125 PRINTERS FROM £130

The expandable system for small businesses, beginners, and experts.

QLs COMPLETE JM FULLY TESTED AND WITH 6 MONTHS WARRANTY

TV lead, QL Software 2.35, Quill – Word Processor, Abacus – Spread Sheet, Archive – Records, Easel – Business Graphics **£125**

Customers buying one of the QL Computers are given one year's free membership to QUANTA

As above with JS Rom **£150**

(Help, Newsletters, & 400+ Library Programs – most free).
QUANTA MEMBERS can obtain a £5 discount
"The JM QL can run all Programs available for the QL system."

★ SPECIAL OFFERS ★

NEW PHILLIPS COLOUR MONITORS

14in high resolution enhanced graphics, 85 chars. RGS input. Complete with tilt & swivel stand, and QL lead ready to plug-in and go. RRP £379.00

£220 inc VAT

UNIVERSAL DISK DRIVE

1mb, 3.5in cased, complete with built-in power supply, mains switch & 13amp plug, external dip switches adapt drive for QL, PC, Atari, Amiga, etc, comes with full instruction book, plastic cover, and free DS/DD disk

£75 inc. VAT QL lead £10
TWO DRIVES £130 lead £15

UNCASED DRIVES

NEC FD1036A 1mb 3.5in 1/3 height
£49 inc VAT
Leads for QL and other Micros
£10

PC 102 KEYBOARDS FOR **QL £25**
Interface for above £75

MANNESMAN TALLY MT81 EPSON, PC and QL COMPATIBLE **£130**
130 CPS DRAFT, 24 CPS NLQ, Tractor feed. Sendfor printouts & QL details

Interchangeable plug-in interface modules for Star NL-10 printer. Serial, Parallel/Centronics, IBM/PC, or Commodore each £20

Prices include VAT. Terms CWO, Access or Visa. Minimum order £10.
Carriage £8.00 for printers & QL (overseas £20.00) other items £3.00 (overseas £6.00)

EEC LTD

18-21 MISBOURNE HOUSE, CHILTERN HILL, CHALFONT ST PETER, BUCKS. SL9 8UE
Tel: 0753 888866 Fax: 0753 887149

2x Expanded JS QLs, 2 x 3·5" twin drives, 2 x amber monitors, colour monitor, mouse, mainly graphics software. All good order. £500 total. Plus free JM QL. Nick Stratton (071) 272 5239 (evenings)

Qfile programs: MS-QLink v. 1.5 £12.00, DISCOPY v. 2.06 £7.50, £17.00 together. Please state 3·5"/720KB or 5.25"/360KB. Cheques payable to T.F. Leal. Address: Qfile, Apartado 2110, P-1103 Lisboa Codex, Portugal.

CVLS 92 Firtree Est., Thurgoland, Sheffield S30 7BG. 3.5" DSDD Disk 37p with labels P+P 7p per disk, max 24.00. 100 Disks 24.00 inc. P+P. Printer ribbons, disk boxes, micemats etc. Phone Sheffield 0742 884183.

QL ROMS AND SPARES

ROMS, MGE (Spanish, MGF (French), MGG (German), JS (English), £20 each. JM (English), £10

MDVS. Complete QL MDV Subunit £20, head and chassis £10, Microswitch £2, Motor £6. Rubber & Plastic Roller £2. MDV2 Stand-off spacer + srew £2. ULA £5. Tr 4,5,6,7 set £2. MDV pcb inc, ULA £11. Prices include UK postage. Send SAE for price list of all QL spares.

Joe Atkinson, 36 Ranelagh Road, Ealing, London W5 5RJ (Mail Order Only).

QL REAL TIME DIGITIZER

Digitize moving pictures in real time. Transforms QL into a video printer or 3-D image digitizer via a video camera.

* Digitizes in .02 secs.
* 256x256 resolution in 8 levels
* Full, 1/4 or 4x 1/4 screen modes
* Screen update - 5 or 2.7 frame/sec
* 1:1 grey level 9-pin printer dump.
* Interfaces to QL ROM PORT
* Runs on unexpanded QL and multitasks.
* Cost: £123.00 incl. State mdv or flp.

**CL SYSTEMS, 403 CHAPTER ROAD, DOLLIS HILL
LONDON NW2 5NG TEL: 081 459 1351**

TD DESK is the definitive User-Front-End package for all QDOS compatible computers, the standard **SINCLAIR QL,** the powerful **Thor XVI** & the ST with **QL-EMULATOR.** Based on the **QRAM** design with twice the power of **QPAC II!** The powerful QJUMP extended Pointer Environment in included. The menus are: **DESK, DEVICES, FILES, JOBS, KEYBOARD, PROGRAMS, MY PROGS, TOOLS** and **OPTIONS.** There are five desktop accessories included: **TCLOCKS, TDEMOS, TANIMATOR, TINDEX** and **TSNAP.** Software and 34 page manual is available in English or German for Sfr.95.--(disc) or Sfr 103.--(mdv) including P&P. Free updates. Send enquiries & orders with cheques to :

**COWO Electronic,
Münsterstrasse 4, CH-6210 Sursee**

C.G.H. SERVICES

Cwm Gwen Hall, Pencader, Dyfed, Cymru, SA39 9HA (Tel. 0559 384574)

Products	Type	Memory	flp —	mdv
Anel Pum Quat	Text Adventure	128K	£8.00	£8.00
Blag2	Text Adventure	256K	£8.00	£8.00
D-Day Mk II (Special Ed)	Wargame	256K	£13.00	£18.00
Double Block 1-2 Player	Tetris Game	128K	£9.00	£9.00
Dreamlands	Text Adventure	256K	£8.00	—
From the Tower of Valagon	Text Adventure	128K	£8.00	£8.00
Grey Wolf	U-Boat Simulator	256K	£9.00	£12.00
Here We Go	Text Adventure	128K	£10.00	£10.00
MacSporran's Lament	Illustrated Adventure	128K	£8.00	£8.00
Open Golf	Golf Simulator	512K	£9.00	—
Orbiting	Astronomical Sim	128K	£8.00	£8.00
Personal Finance Manager	Home Accounts	128K	£8.00	£8.00
Polytext	Multi column quill output	128K	£16.00	£16.00
Quiz Master	Quiz Game	128K	£10.00	£10.00
Return to Eden	Illustrated Role-Play Adv	256K	£15.00	—
Starplod	Illustrated Adv	128K	£8.00	£8.00
The Gee Gee System	Horserace Prediction	128K	£10.00	£8.00
Uncle Loonie's Legacy	Illustrated Puzzle Adv	128K	£8.00	£10.00
Voyage of the Beano	Illustrated Adv	256K	£10.00	—
Voyage of the Beano	Text Adventure	256K	—	£12.00
Wreck Dive	Graphical Adv	128K	£8.00	£8.00

Please add 10% to all orders by post to cover p&p and other charges. We regret that we cannot deal with orders by Access or other cards. All monies must be in Sterling.

Also available: QL Technical Review 1-4. Subscriptions to QL Technical review nos 5-8 £6.00. Adventurers' Forum 1-9, subscriptions to QL Leisure Review nos 1-4 £6. The Bible – 9 discs – text files £30. We also distribute a wide range of Public Domain and Shareware titles. Please send an s.a.e. for details. We would like to hear from any programmers wishing to have their programs published.

CLASSIFIED

If you would like to place a Classified Advertisement in the next issue of Sinclair/QL World, address it to:

**Jason Newman,
Sinclair/QL World,
Panini House,
116/120 Goswell Road,
London EC1**

on a plain piece of paper. Please also print your name, address, and daytime telephone number.

The cost of 1-30 words inclusive is £15. Prepayment is not necessary as you will be invoiced for the sum of £15 following publication. Settlement of the invoice will be required within 30 days of receipt.

INSTANT

HARDWARE

Care Electronics
0923 672102
Peripherals and accessories.

Thor International
010 4533 930305
010 4533 937544

Digital Precision
081-527 5493
Discs, interfaces, drives, keyboards, RAM expansions.

EEC Ltd
0753 888866
QLs, peripherals, accessories

Miracle Systems
0904 423986
Trump Card interface, modems, hard disc drive.

Strong Computer Systems
0267 231246

Sector Software
0772 454328
Peripherals.

S.P.E.M
Visa Aosta, 10154 Torino, Italy
QL disk interfaces, ram, eprom holders.

TF Services
081 724 9053
Repairs, spares, peripherals.

TK Computerware
0303 812801
Peripherals.

SERVICES

Adman Services
0952 255895

Joe Atkinson
35 Ranelagh Road, London W5 5RJ
Roms and spares

Suredata
081-902 5218
Repairs.

QL World (reader services)
Subscriptions
091 510 8787 (Lazerhold)
Microdrive Exchange
0772 454328 (Sector)
Recent back issues : write to editor

Quanta
User Group, support.
Hon. secretary, Phil Borman.
0933 460690

Xchange User Group
Freepost
Beckenham
Kent BR3 2BR

SOFTWARE

Athene Consultants
0705 511439

ARK Distribution
0983 79496
Archivist, Master Spy and other business and utility software.

Call Us What You Like
55 Greenan, Shaws Road, Belfast BT11 8LX. No telephone. Superspy.

Care Electronics
0923 672102
Tony Tebby software. Miracle hardware, peripherals, utilities.

CGH Services
055934 574
Adventures. QLAF Compware
0270 582301
Task Swopper, Mega Toolbox, Expert, System Shell, Speedscreen and others.

Digital Precision
081-527 5493
Turbo Basic compiler, Desktop Publisher, Editor, Eye-Q Graphics. Sprite generator, Super Forth, Better Basic, Professional and Super Astrologer, Media Manager, Supercharge, Blocklands, Driodzone, Aracadia, Reversi, Backgammon, Master Bridge Tutor, C-M Emulator, C compiler, Lightning speed enhancer.

DJW Software
0256 881701
Home Banker

Eidersoft
089283 2552
Games utilities, Ice. Pyramide.

ImaQLate Software
42 Albion Street, Broadstairs, Kent CT10 1NE.
Biograph.

Jochen Merz Software
010 49 203501274
QL/Atari Emulators, utility and games software.

QL Supersoft
0256 475992
Home Budget, Superdraw.

Rob Roy Software
94 Teignmouth Road, Clevedon, Avon BS21 6DR.
Rob Roy and Palantir.

Sector Software
0772 454328
Flashback and other utilities.

SD Microsystems
0462 422897
Business software including General Ledger. Small Traders' Pack/Invoicer and Stock Accounting.

Software 87
33 Savernake Road, London NW3 2JU
Text 87 and updates.

Talent Software
0303 813883
Utilities, games.

TK Computerware
0303 812801
Add-ons, spares, software.

WD Software
0534 81392

RETURN THE COUPON FOR FREE COLOUR BROCHURES!

NEW! - AMIGA PACK

Commodore A500 Flight Of Fantasy

£399 INC VAT

A500 BATMAN PACK

The Commodore A500 Batman Pack must surely rank as one of the most popular computer packs ever! The pack features the **Commodore Amiga 500 computer** with **mouse controller** and **TV modulator**, plus four top software titles. The software includes: '**Batman The Movie**' - Rid Gotham City of the cunning joker, in Ocean's top selling title based on the blockbuster Batman film; **New Zealand Story** - high quality conversion of the leading arcade game; **Interceptor** - Dogfight with two F-16's in this leading flight simulator; **Deluxe Paint II** - top quality Amiga graphics package which set the standard for others to follow. Return the coupon for further details.

PACK INCLUDES:

A500 Computer & Mouse	£399.99
A520 TV Modulator	£24.99
Batman The Movie	£24.95
New Zealand Story	£24.95
Interceptor	£24.95
Deluxe Paint II	£49.95
TOTAL RRP:	£549.78
Less Pack Saving:	£150.78
PACK PRICE:	**£399.00**

£399 INC VAT

AMIGA 2000

For the more serious or professional applications user, Commodore have a selection of systems based around the expandable Amiga 2000, at prices from £1295+VAT. The A2000 features a full 1Mb RAM (expandable to 9Mb), 9 system expansion slots, plus IBM compatibility with the use of PC-XT or PC-AT bridgeboards. Complete and return the coupon, putting a tick in the A2000 box, for details of A2000 computer systems.

£1295

+VAT= £1489.25

FLIGHT OF FANTASY

Flight of Fantasy is the very latest Amiga 500 pack from Commodore, featuring BRAND NEW software releases, to make this the most spectacular A500 pack ever! The pack features the Amiga 500 computer with mouse controller and TV modulator, as well as four top software titles. These include the following:

DELUXE PAINT II:

The high quality graphics program that set the standard for other Amiga art packages. Deluxe Paint II includes powerful, easy to use tools that bring out the artist in you. Create masterpieces, presentations, 3D perspectives or just doodle.

ESCAPE / ROBOT MONSTERS:

Here's something completely different - a science fiction story with comic book style graphics. Our heroes Jake and Duke are on the Planet X rescuing Humans who have been captured by the Robot Monsters and forced to create an evil Robot Army to DESTROY EARTH! Jake and Duke fight their way through hordes of evil Robots to help the Humans escape.

RAINBOW ISLANDS:

Slip on your magic shoes, practise throwing a rainbow and you're ready to go island hopping. From the Island of Doh to Monster Island, you will encounter Doh himself, stinging insects, lethal combat machines, mechanical assailants, the formidable beings of legend and folklore. Finally enter the world of darkness and its inhabitants.

F29 RETALIATOR:

The ultimate in flight simulation with a choice of two aircraft and four battle environments with dozens of different tactical missions. Aerial combat, strategic bombings, interactive ground based battles, seagoing carriers ... the list of features is endless. Real time cockpit displays, including 'true radar' enhance the realistic feel of this stunning simulation.

PACK INCLUDES:

A500 Computer & Mouse	£399.99
A520 TV Modulator	£24.99
Deluxe Paint II	£49.95
Escape/Robot Monsters	£19.99
Rainbow Islands	£24.95
F29 Retaliator	£24.95
TOTAL RRP:	£544.82
Less Pack Saving:	£145.82
PACK PRICE:	**£399.00**

£399 INC VAT

FOR FURTHER DETAILS OF THE AMIGA RANGE, COMPLETE THE COUPON AND RETURN IT TO SILICA SHOP

THE UK's No1 AMIGA SPECIALISTS

SILICA SHOP OFFER YOU

FREE OVERNIGHT COURIER DELIVERY: On all hardware orders shipped in the UK.
TECHNICAL SUPPORT HELPLINE: Team of Amiga technical experts at your service.
PRICE MATCH: We normally match competitors on a ''Same product - Same price'' basis.
ESTABLISHED 12 YEARS: Proven track record in professional computer sales.
£13M TURNOVER (with 60 staff): Solid and reliable with maintained growth.
BUSINESS/EDUCATION/GOVERNMENT: Volume discounts available for large orders.
SHOWROOMS: Demonstration and training facilities at our London & Sidcup branches.
THE FULL STOCK RANGE: All of your Amiga requirements from one supplier.
FREE CATALOGUES: Will be mailed to you with offers and software/peripheral details.
PAYMENT: By cash, cheque and all major credit cards.
CREDIT PAYMENT TERMS: Silica are licensed credit brokers - return coupon for details.

Before you decide when to buy your new Amiga computer, we suggest you think very carefully about WHERE you buy it. Consider what it will be like a few months after buying your Amiga, when you may require additional peripherals or software, or help and advice with your new purchase. And, will the company you buy from contact you with details of new products? At Silica Shop, we ensure that you will have nothing to worry about. Silica have been established for over 12 years, and have an annual turnover of £13 million. With our unrivalled experience and expertise, we can now claim to meet our customers requirements with an understanding which is second to none. But don't just take our word for it. Complete and return the coupon now for our latest Free literature and begin to experience the ''Silica Shop Service''.

RETURN THE COUPON NOW FOR
FREE BROCHURES

SILICA SHOP

MAIL ORDER:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-309 1111**
Order Lines Open: Mon-Sat 9.00am-6.00pm		No Late Night Opening	Fax No: 081-308 0608
LONDON SHOP:	52 Tottenham Court Road, London, W1P 0BA		**Tel: 071-580 4000**
Opening Hours: Mon-Sat 9.30am-6.00pm		Late Night: Thursday until 8pm	Fax No: 071-323 4737
SIDCUP SHOP:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-302 8811**
Opening Hours: Mon-Sat 9.00am-5.30pm		Late Night: Friday until 7pm	Fax No: 081-309 0017
BUSINESS/EDUCATION:	1-4 The Mews, Hatherley Rd, Sidcup, Kent, DA14 4DX		**Tel: 081-308 0888**
Order Lines Open: Mon-Fri 9.00am-6.00pm		Closed on Saturdays	Fax No: 081-308 0608

To: Silica Shop Ltd., Dept QLW 02/91-32, 1-4 The Mews, Hatherley Road, Sidcup, Kent DA14 4DX

PLEASE SEND INFORMATION ON THE AMIGA

Mr/Mrs/Ms: .... Initials: .... Surname: ....

Address: ....

....

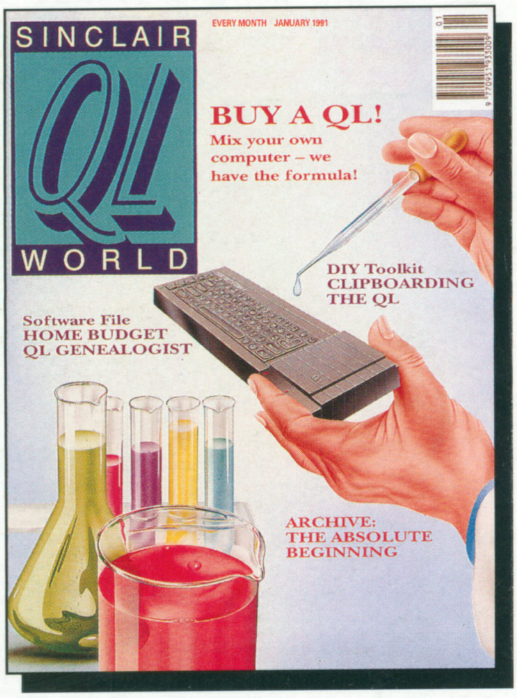
....

Postcode: .... Tel: ....

Which computer(s), if any, do you own? .... A2000 □

E&OE - Advertised prices and specifications may change - please return the coupon for the latest information.

SUBSCRIBE TODAY



For Convenience

To Avoid Disappointment

And Save Money

Direct to your door at **NO EXTRA COST*** – we pay the post!

Never again worry about your local newsagent selling out.

Avoid any price increases for the next 12 issues by subscribing.

Simply fill in the coupon below and send it with your remittance to:
**M.S.M. Subs. Dept., Lazahold Ltd., P.O. Box 10, Roper Street,
Pallion Ind. Estate, Sunderland SR4 6SN**.

The first issue of a new subscription to be delivered will be one or two issues after the one you placed your order in.

*Overseas prices include postage

Please send me one year's subscription to **Sinclair QL World**. I enclose my cheque/money order for £.................... made payable to MCPC LIMITED or debit my Access/Visa card.

Card No. □□□□□□□□□□□□□□□□

Expiry date □□□□□□

Name ....
Address ....
....
Postcode .... (Please enter postcode to ensure prompt delivery)
Signed ....Date.......................... SQLW0291

U.K.: £21.00, EUROPE: £32.00;
REST OF THE WORLD: £38.00
AIRMAIL RATES AVAILABLE ON REQUEST