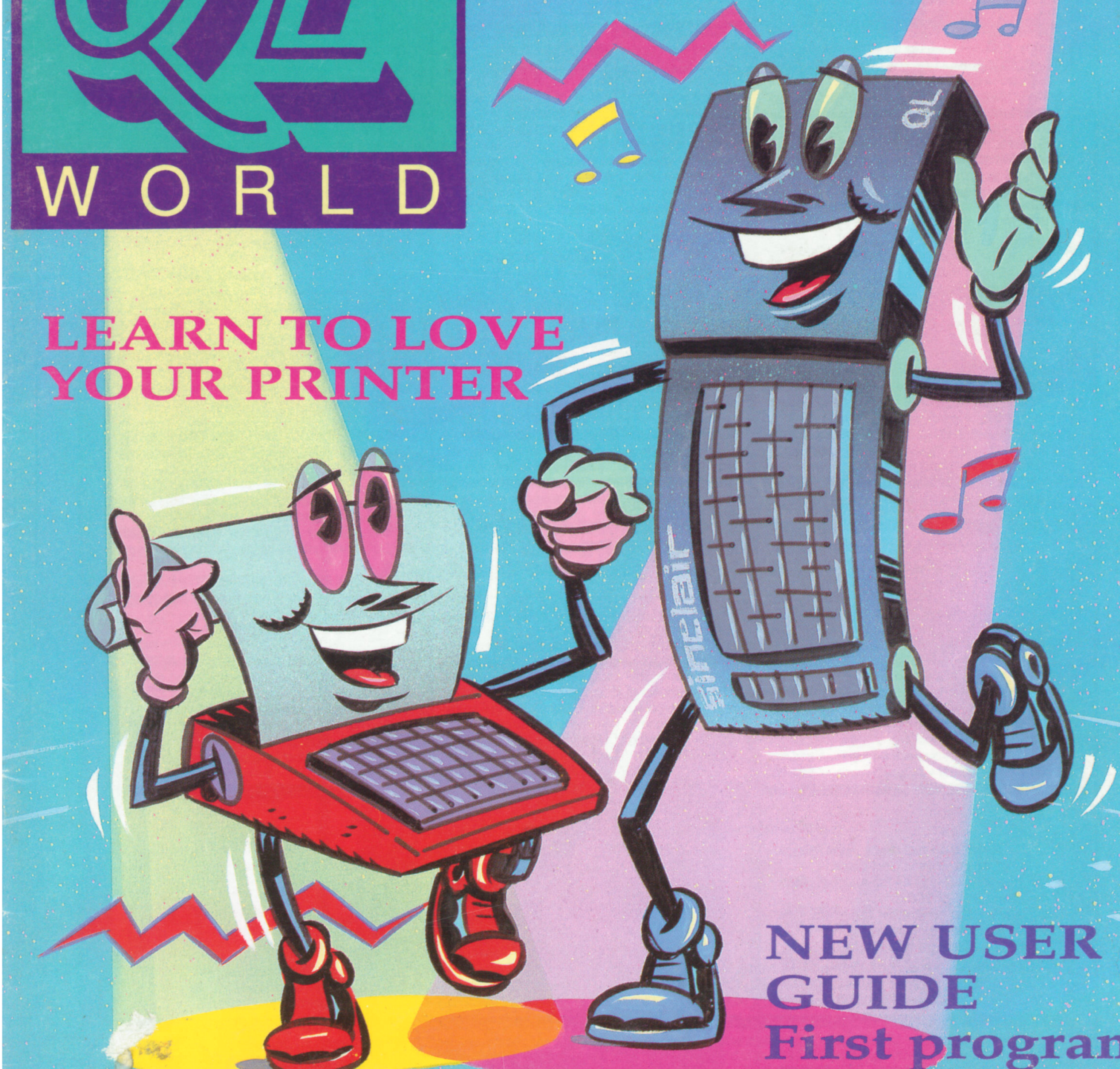
EVERY MONTH APRIL 1991 £1.75

SINCLAIR QL WORLD

DIY TOOLKIT
SET resident
functions

LEARN TO LOVE
YOUR PRINTER



NEW USER
GUIDE
First program
this month

DJC

Dilwyn Jones Computing

41 Bro Emrys, Tal-y-Bont,
Bangor, Gwynedd LL57 3YT
Tel: Bangor (0248) 354023

VISION MIXER PLUS ***NEW!!!***
Joe Haftke, the author of Home Budget, had completely reworked the orginal Vision Mixer. Now you are not limited to the amount of screens you can store in memory, because this version keeps the screens on disc and fetches them one at a time just before they are due to be displayed. Up to 22 screens per 720k floppy disk — use up to 4 drives. Vision Mixer Plus works from Hard Disc too. New facilities include hard copy of screen sequences, mixed MODE 4 and MODE 8 screens, automatic loading of a sequence with a special filename (V–AUTO–seq), new effects and generally much easier to use. All this and it still only needs 256k of memory to run. Vision Mixer 1 users — see below for upgrade details

PICTUREMASTER ***NEW!!!***
A new utility from Joe Haftke to make screens for use with Vision Mixer Plus (or Vision Mixer 1 if you must!). It allows you to make multi-coloured MODE 4 or MODE 8 screens with text captions, including coloured strips, boxes, shadowing and "raised block" effects. A number of predefined screen picture routines are included, together with colour pallette charts showing the numbers of all the QL stipple colours. But the best part is that you can get the source code (BASIC) so that you can study the techniques used and add your own routines and generally experiment with and add to the program — a tinkerer's delight and a very easy program to use.

VISION MIXER ***by Dilwyn Jones***
The original Vision Mixer screen display, effects and advertising software. Great value at only ten pounds. Reviewed in QL World Jan. 1991. "A stunning compendium of more that 100 different video effects . . . will in my view stand comparison with any commercial product I have seen." (John Shaw, QL World review January 1991).

VISION MIXER PLUS (disc only)	**£22.50**
PICTUREMASTER (disc only)	**£15.00**
Both together — save 20%!	**£30.00**
VISION MIXER 1 (disc only)	**£10.00**
Upgrade Vision Mixer 1 to Vision Mixer Plus (quote serial number)	**£14.50**
PICTUREMASTER + upgrade	**£24.00**

HOME BUDGET ***by Joe Haftke £20.00***
Personal income tax calculator (UK only) plus domestic bills and accounts budgeting system. Does Capital Gains Tax cost indexation calculations too. It's coming up to the time of year when the tax calculator may prove useful!

QUICK POSTERS ***by Dilwyn Jones £10.00***
Makes text posters quickly and effortlessly with this program. Large text, centering, NLQ, reconfigurable printer driver, etc. NB Check printer compatibility with us before ordering.

BASIC REPORTER ***by Dilwyn Jones £10.00***
BASIC programming aid. Lists variable names, procedure and function names, extensions, line numbers, addresses and so on. Helps you debug BASIC programs and spot typing mistakes etc.

COCKTAILS WAITER ***by Imre Dominik £10.00***
Cocktail drinks recipes database. Over 400 recipes supplied with the program. Great for parties, or for when you have guests or friends around! Browse through the database, find drinks you can make with a given ingredient, add to the database and many more features.

EXTRA RECIPE SETS, 2 AVAILABLE NOW

MIX 2 — around 300 extra recipes	**£5.00**
MIX3 — includes non-alcoholic	**£5.00**

SUPER DISC LABELLER ***£10.00***
Print labels for your discs showing filenames on the disc. Show file sizes, type and dataspace, dates, sort file names, select using wildcards (e.g.ony Quill -doc files) or manually, edit the final label text, preview to the screen and even make a backup copy of a disc with the sorted directory. Reconfigurable printer driver and label sizes — use our tractor fed disc labels (see below).

QL GENEALOGIST ***by Chris Boutal £19.50***
A best seller in its own lifetime! Record and print your family tree with this comprehensive genealogy database. See the review in QL World Jan. 1991 for further details — John Shaw described it as "guaranteed to make even the most fastidious and exacting Family Historian go wild with delight" Send an SAE for further details of this superb program. Compatible with Thor and the ST QL emulator.

WINBACK ***by Norman Dunbar £25.00***
Hard Disc backup utility, specifically intended for Miracle Systems hard disc and compatibles. Specify directories, backup dates and listings if required. Require Toolkit 2 to run.

POLYTEXT ***by Nick Ward £16.00***
Multi column NLQ text output from Quill! Rearrange the output from Quill, retaining bold, subscript, superscript and underlining plus italic and draft/NLQ via translates. Justify. Use elite, condensed, expanded and double height printing, footers, even small graphics. Up to 7 columns of output. SAE for sample printout.

DISCOVER ***by Dave Walker £20.00***
QDOS to PC disc format transfer. 256k, disc only. SAE for details.

MULTI DISCOVER ***by Dave Walker £30.00***
As Discover, but also transfers to BBC DFS/ADFS, CPM (many formats) and to Unix CPIO format. 256k, disc only.

TEXTIDY ***by Dave Walker £15.00***
Text file conversion utility. Convert Quill — doc files to plain text, to DOS Quill format, to Wordstar (PC) and vice versa. Useful for preparing text files for Discover to transfer to the PC for use on MSDOS word processors. Also converts Archive screen files to DOS format for transfer to PC Archive. Available on disc only.

TASKMASTER ***by Peter Jefferies £25.00***
Multitasking front end utility, calculator, notepad, printer buffer etc.

FILES 2 ***by Peter Jefferies £12.00***
Enhanced version of Taskmaster files utility — add to Taskmaster or use by itself

SPELLBOUND ***by Peter Jefferies £30.00***
30,000 word dictionary, check spelling as you type

SPELLBOUND SPECIAL EDITION ***£50.00***
New version of Spellbound, larger dictionary, retrospective spell-check.

UPGRADE SPELLBOUND TO SPELLBOUND S.E.
Return spellbound master plus £30.00.
(NB Spellbound S.E. is on disc only)

FLASHBACK ***by Peter Jefferies £25.00***
Excellent fast, versatile database.

FLASHBACK SPECIAL EDITION ***£40.00***
Enhanced version of Flashback — new commands, report generator, mailmerge, label printing etc.

SUNDRY COMPUTER SUPPLIES

(Please note: some prices have changed since the last advert — check carefully)

3.5" DSDD unbranded discs, with labels	£0.50 each	3.5" SONY BRANDED DISCS	£1.00 each
20 or more unbranded discs	£0.45 each	Box of 10 Sony BRANDED discs	£9.50
5.25" unbranded DSDD 80 track discs	£0.35 each	New Sinclair microdrive cartridges	£2.50 each
Disc labels, roll of 100	£2.00	Labels for microdrive cartridges	£2.00 /100
Disc labels, 100 on printer feed roll	£2.50	Address labels, on printer feed backing	£2.00 /100
40x3.5" storage box (lockable)	£5.00	80x3.5" disc storage box (lockable)	£7.00
10x3.5" disc storage box (Ryford)	£1.20	Monitor stands, up top 14", tilt and swivel	£15.50

Printer ribbons — we can obtain most types. If your type is not shown, ask and we may be able to order it for you. LC24-10, LC10, Citizen 120D, Brother HR5M1009/M1109, Centronics GLP/GLP2, Epson FX/RX/MX80, PW1080/Kaga Taxan KP810, Juki 6100, Shinwa CP80 and many more normally held in stock, prices from £4.00 each. Ask about others not shown.

TERMS. Postage — UK, add £2.50 postage and packing, except to software only order, which are post free. EC countries — please add £3.50 postage and packing, except to software only orders, to which you should add £1.00 per item for postage. Other countries: add 10% to order value by surface mail, 20% for airmail. Please make payment, in Sterling only please (your bank can tell you how to send Sterling from abroad), payable to DILWYN JONES COMPUTING. Goods remain the property of Dilwyn Jones Computing until paid for in full. Send an SAE for further details of the program or programs you are interested in and a full price list.

SINCLAIR QL WORLD

Editor
Helen Armstrong

Production Controller
Jayne Penfold

Designer
Jeff Gurney

Advertising Manager
Jason Newman

Magazine Services
Yvonne Taylor

Advertising Production
Michelle Evans

Group Advertising Manager
Jean Dorza

Group Editor
John Taylor

Deputy Managing Director
Ray Lewis

Managing Director
Peter Welham

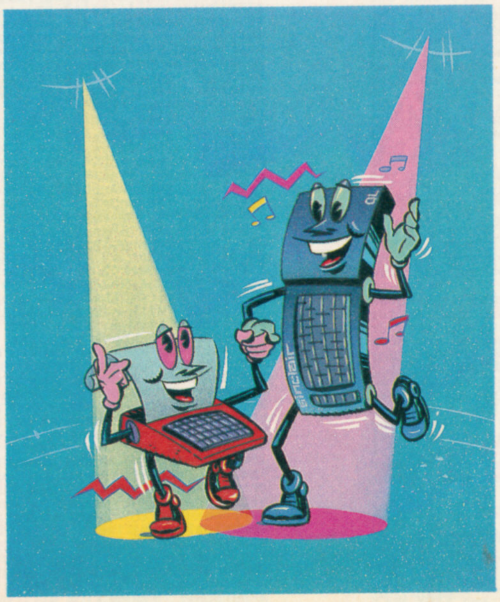
Sinclair QL World
Panini House
116-120 Goswell Road
London EC1V 7QD
Telephone 071-490 7161
ISSN 026806X

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 IPC Market force, King's Reach Tower, Stamford Street, London SE1 9LS.
Subscription information from: MSM Subscription Dept, Lazahold Ltd., PO Box 10, Roper St, Pallion Ind. Est., Sunderland SR4 4SN.
UK: £21.00, Europe: £32.00, Rest of the World: £38.00.

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.

CONTENTS

APRIL 1991



NEXT MONTH

ARCHED

A simple all-purpose line editor that runs within Archive

HARDWARE REVIEW

The Hewlett Packard DeskJet printer — a useful alternative to laser printing

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 Quill 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 DR-DOS v5.0
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 DOS (**Conqueror** works with all versions of DOS).

If you do not have legal access to a copy of DOS, you need to buy DOS too (DR-DOS or MS-DOS)- but we sell the complete DR-DOS (with Viewmax, Shell, Cache and all system utilities) at c 1/2 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 * DITS * 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
Whether or not you can program, **Toolkit III** is of great use!

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**. And **QFlick**'s data is organised so it is easy to program from SuperBASIC, even for tyros!

PERFECT POINTER TOOLS

This excellent program gives you an on-screen pointer (arrow) environment of incredible smoothness, and 6 utilities with it. To explore the world of QPtr, Things, Hotkeys, Window Manager....

QKICK MULTITASKING SYSTEM

A pull-down menu controlled multi-tasking front-end, ideal for running in the background and giving you notepads, file-handlers, quick backup, clock, diary, mini-database, calculator etc 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 maintaining full compatibility and normal disk control...

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 DR-DOS v5.0
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 DOS (any) (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 (data in either direction) between QL and 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!

MICROBRIDGE

Never be short of a four for Bridge again. Superb bidding tutor included, based on random hands dealt with lightning speed. Manual a masterpiece. Understands and obeys ACOL and much more.

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. 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.

MONITOR

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

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), foreign currency cheque (add 10% conversion charge), cash, direct funds transfer (notify us in writing, 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.
* 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. PERFECTION is not an upgrade to EDITOR, but SE EDITOR owners are allowed to claim a £5 reduction when ordering PERFECTION.
* 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. ST/QL Emulator owners will benefit from a c2.7x speed increase on our software.
* All trademarks used or referred to in this ad are acknowledged.
* 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.
* We supply unused cartridges @£3.50 each, £30/10 or £55/20.
* Digital Precision supplies quality hardware too - write/phone.
* 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.

PRICE LIST

PC CONQUEROR WITH DR-DOS V5.0	169.95	eT
PERFECTION PLUS WITH SPELL CHECKER	119.95	dT
THE SOLUTION WITH DR-DOS V5.0	119.95	eT
PC CONQUEROR	89.95	eT
PROFESSIONAL PUBLISHER	89.95	cT
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

LIMITED PERIOD: SPECIAL DEALS ON OUR PROGRAMS
****
* 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
DR-DOS (NOT BEING OUR PROGRAM) MUST NOT BE COUNTED

USER VERDICT

	Poor	Moderate	Good	Excellent
Installation				▲
Ease of use				▲
MS-DOS compatibility				▲
Use of PC's features				▲
Functionality				▲
Documentation				▲
Tutorial		Yes		
Value for money				▲

■ PC USER VERDICT This is the first time I've given any product an all 'excellent' verdict, which speaks for itself. ● PRODUCT DR-DOS 5

DR DOS 5.0

Digital Research
WE MAKE COMPUTERS WORK

OPERATING SYSTEM
for PCs and PC Conqueror

*We can now supply the amazing new **DR-DOS v5.0**, normally sold for £119.95+VAT+P&P = £144, for **£80 all inclusive** (when bought with Conqueror/Solution,or by MSDOS users (who can at the same time send in the old Conq/Sol disk for a free upgrade) - else £90)! This best DOS has 600+ page manuals ViewMAX,Cache, online Help,screen Editor, command-line History, passwords, FileLINK +MemoryMAX, Ramdisk, 71 DOS files(1.4Mb), top compatibility & DR technical support. Our **v5.0** is the **22/1/1991 NEWEST**, updated release. The PC press have raved about it and PC Plus (see left) gave it seven (7!) excellent grades. So get **DR-DOS v5.0** NOW!

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

Z 88

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 and Dual 3.5" Disk Dirves	£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
QDOS Companion	£9

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
Media Manager Special Edn	£50.00
Super Media Manager	£30.00
Sidewinder Deluxe	£20.00
Conqueror	£90.00
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 Workbench	£25.00
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
GST Macro Assembler	£38.00

LEISURE

Chess	£19.00
Super Croupier	£13.00
Flight Simulator	£19.00
Matchpoint	£15.00
Bridge Player 2	£15.00
Prof Astrologer Astronomer	£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	£18.00
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)	£26.00	QL Dustcover	£6.00

PERIPHERALS

Keyboard 90 Interface	£75.00
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:

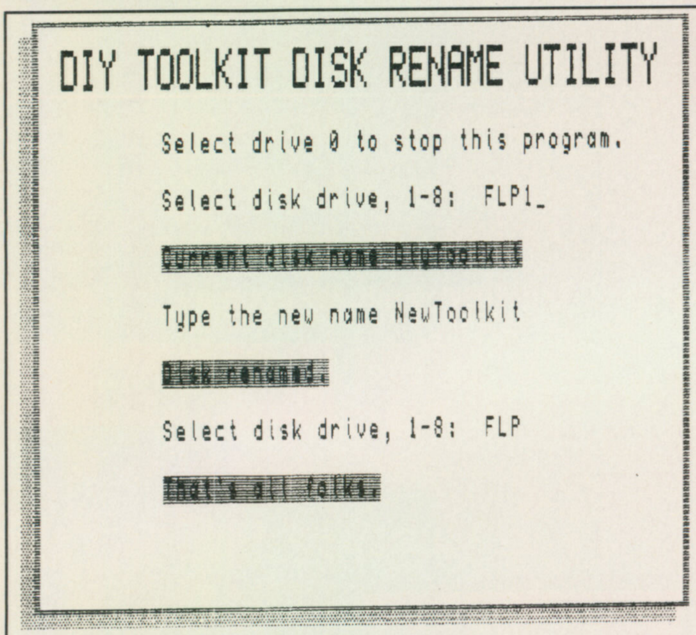
**TK COMPUTERWARE, STONE STREET,
NORTH STANFORD, ASHFORD, KENT TN25 6DF**

TELEX
966676 PMFAB G

VISA · Access · MasterCard · AMERICAN EXPRESS · EUROCARD

FAX
0303 812892

QL SCENE



```
DIY TOOLKIT DISK RENAME UTILITY

        Select drive 0 to stop this program.

        Select disk drive, 1-8:  FLP1_

        Current disk name DiyToolkit

        Type the new name NewToolkit

        Disk renamed.

        Select disk drive, 1-8:  FLP

        That's all folks.
```

The DIY Toolkit budget software disk collection has expanded with new programs and updates for QL, Thor and compatible systems. Phil Spink and upgraded utility programs in Volumes H, J and Q, while Simon Goodwin has added new disk tools to Volume D.

VIEW DISK is an elaboration of CHECKDISK, (*QL World* December 1990). It quickly checks the entire surface of a Qdos disk displaying tracks as concentric circles as it goes along. Errors on either side are colour-coded and reported in a window alongside the disk view.

DISK RENAME lets you changes any floppy disk's name without re-formatting the disk. This utility displays the old name for any FLP drive before prompting for the replacement name, and makes it easy to rename several disks at one sitting.

CHLIST (*QL World* November 1990) identifies all the open channels on a system, showing the name of each channel and its task, plus the Qdos ID and channel definition address. The new version in Volume Q shows SuperBasic channel numbers, like 0 and 3, as well.

DEVLISTS, in Volume H, has been upgraded to show the names of Qdos devices as well as linkage addresses and vectors to open, close and manipulate channels with TRAP 3. This utility helps resolve toolkit or device-name clashes.

TASKNAME shows the name of each task as you select its cursor. The latest version takes the non-standard job headers of *Minerva's* extra interpreters in its stride, naming them as well as normal QL and Thor tasks. Volume J also includes the unique *Taskforce* 128K multi-tasker, plus a utility to rename compiled tasks and another that adds names to anonymous tasks like Psion packages.

The revised Volumes are available from CGH Services from 10th March 1991. The full range of DIY TOolkit programs now runs to 15 Volumes, totalling well over 200 files, including the new *Qlipboard* (Volume S) and this month's SET extension, Volume U.

Volumes cost £3 each, plus a processing charge of £4 per *order*. The total price includes disks and delivery; tape orders must come with one formatted microdrive cartridge for each volume required. Call **(0559) 384574**, or post orders or enquiries to **DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA**

Printer Contest

Printer giants Star Micronics are inviting schools in the UK, Germany and France to enter an annual competition to promote environmental awareness. As well as winning computing equipment for their schools, winning teams will go to Japan this summer on a ten day cultural exchange visit.

The competition is open to full time students aged 16 to 18, and aims to contribute to school studies, A levels, CPVE and TVEI in subjects touching on the environment.

Teams will tackle projects in five areas: Energy, global warming, today's 'throwaway' culture, world use of resources and the ageing of society. Competition regions in the UK cover Scotland, Northern Ireland and Northern England, the Midlands, Wales and the South West, the South East, and lastly Greater London and the areas within the M25.

Star is issuing information packs to schools. Packs and competition forms can be requested from **The World and Our Future, Byron House, Wallingford Road, Uxbridge, Middx UB8 2RW.**

Tax Experts Call

EMSoft of Boston, who have produced and published *Tax-I-QL*, an 105K-Abacus template for the preparation of US Individual Income Tax Returns for a number of years, are now looking for programmers to adapt the template for other nations' income tax systems. They are suggesting that interested programmers "familiar with their country's tax laws" to contact them. They add that "there would be no charge for the rights to use the name Taz-I-QL. Our sole purpose is to increase the utility of the QL Thor computer", and add that they can offer programming tips to maximise memory saving and loading speed.

EMSoft are at PO Box 8763, Boston, MA 02114-8763, USA.

Archive Plus for Dos

DMT Associates and James McCosh, in association with Transform Ltd. are shortly to release a new version of *Archive* for use with DOS. A Unix version is expected in the summer of 1991, together with a DOS network version.

The new program is call *Archivet* and, while it is said to be fully compatible with Psion Archive, it is described as having extra features including ten index files per database file, shell to dos, shell to dos and run application, integer numbers, approximately three times the speed of Archive, match command to locate on second field of index, cut paste and copy in the editor, and 35 database and/or index files available to open at any time.

Older versions of Archive are available on MS-DOS through the Psion Xchange suite or through a PC emulator program such as Digital Precision's Conqueror

Archivet will be available from March 1 priced £299 plus VAT. Enquiries to **Colin Hughes, Transform Ltd., 7c Station Approach, Hayes, Kent BR2 7EQ.**

Tel. 081 462 4666.

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.

Power

In *QL World January 1991* there is an item by Bryan Davies listing attributes that would improve the QL.

One of these is a facility to detect and positively respond to a power failure. I am quite surprised that this is not the norm, because in the late 60s, I worked on Nixdorf machines – ttl, not an ic in sight! – which had a call in the MacroProgram to an SR that, when traced, was found to check the condition of a gate fed from the low volt ac in the power supply. This SR had a conditional branch, and if there were no volts on the gate, ie, no power the branch operated and caused all files to close, and all memories to be saved to a static ram – dynamic had yet to be invented.

When power was restored, the bootstrap found a reference to the saved data, so a <CLEAR> put the operator back to where the power went off, with no losses.

Of course, with the advent of the ic, and the dynamic ram, the rather large static ram (lots of little iron dust rings with wires through them) became relatively to expensive to produce.

But all is not lost, for the same can be done with a few K of dynamic ram available with a battery back -up, but the power supplies would need to have a better capacity, and the current of a single ac supply to get the + and - 12V supplies directly on the mother board would have to be changed. Thinking about it, this is something that should have been done on the *THOR*.

Now for a request, how about a booklet like the Archive and Abacus/Easel ones (thank you muchly) that tells thicky geriatrics like myself how to use *QDOS?* A sort of User's Guide that Sir Clive, in his wisdom, didn't think to leave us.

Alan M Levett
Glastonbury
Somerset

Editor's notebook

The New User Guide picks up momentum this month by rounding-off of the fundamentals of SuperBasic programming. Next month we start on the fancy waistcoats, but once you have the hang of the contents of parts 1 and 2, programming will never again look like a seven-foot high brick wall.

Becher's Brook, the Triple and the obstacle course follow shortly.

We have a shopping list of sorts of the year. I have enough reports in the bag now to run another Printer Report, but would like to hear from other readers who have bought new printers or solved their relationships with their old ones in the last couple of years. Please write for details.

QL World would also like to hear from any user who has become proficient in the use of the QL/Atari emulator, or the QL/Amiga emulator software, or is using any QL/other machine linkup for pleasure or profit.

Back to printers again: I would like to compile a register of eager printer-users who would be ready to assist other printer-users through the pages of QL World. There be a few bob in it if any enquiries crop up. No addresses will be passed on. Drop me a line if interested.

Manual

I would like to thank you for an excellent magazine. As a matter of fact I think we can thank you to a great extent for the current wellbeing of the QL market. Keep up the good work.

For those of you who regularly curse the QL User Guide: you are sitting pretty. To be sure, it could be more logically compiled, and I myself have often and freely cursed it, but that was before I helped a friend of mine setting up a Tulip SX PC clone. Now, I call that a bad manual. As a matter of fact, I found the QL User Guide reasonably good once I learned to ignore the beginners' guide. The keywords and concepts sections are much more productive.

Finally, for those QL users that are still using microdrives, I suggest that upgrading to disk drives should become a matter of some priority. I recently bought a second hand CST interface and twin 3.5in drives, together with 512Kb of expansion ram, and the improvement in performance has to be experienced to be believed.

Can anyone recommend any good books for the QL, especially on assembly language programming and Qdos?

Kari Ossurarson
Reykjavik
Iceland

Conversion

I am glad that *Sinclair QL World* is continuing. I for one will support it. There are many regular features of the magazine which interest me, including the Microdrive Exchange facilities. I have 3.5 inch disks, so I shall be able to make use of the Microdrive Exchange when it arrives. I would anticipate that the conversion is not as easy as all that – particularly for the machine coded programs – if you aim to offer them as 'foolproof'. I have certainly had difficulty with some in doing the conversion, and also with some of the ones in the Quanta library.

One series I particularly would like to see continue is Simon Goodwin's DIY Toolkit. They are the only articles I know which are, effectively, tutorials in machine code for the QL. Such articles are rare for other computers as well.

I hope that you are able to continue the good work.

Peter Tyler
Ormskirk
Lancs

Editor's reply: You will have noticed that we have, indeed, started to supply MDX programs on disk. You will also have noticed (and wondered why), that, having started to do so, we haven't been advertising the MDX in recent issues. This is because we have had a diversity of behind-the-scenes problems with the Exchange, which I am determined to see resolved one way or another *before we attempt to extend the new service. All the programs advertised in the new-style MDX are still available from Sector Software at their current address (39 Wray Crescent, Ulnes Walton, Leyland, Lancs) at the prices listed.*

Some of the disk conversions were done by the authors, and some by Sector. We have not found any problems at QL World with the disk versions so far.

Numeric

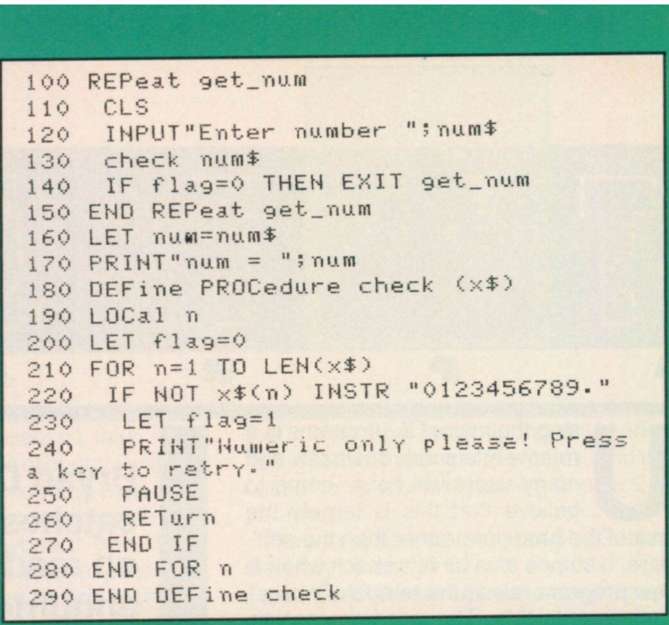
In his *Technical Helpline* column in the February QL World, Howard Clase advises Mark Smith that in order to prevent a crash when a user inputs nononumeric data into a numeric variable, he must resort to machine code or to 35 lines of Basic.

I don't think so! As the accompanying listing shows, it is possible to do this in 20 lines, eight of which (100 to 170) are for the purposes of demonstration.

The key to the problem is to accept the input into a string (num$ in line 110) and to test it in a PROCedure (180 to 290) which rejects it, with a message, if any character is non-numeric. The essence of the program is line 220. It will deal with decimal fractions, but not with numbers input in exponential notation. Using the QL's unique coercion facility, the checked string is relocated in a numeric variable (num) at line 160, ready for further processing.

Brian Storey
Whitley Bay
Tyne and Wear

See the listing at the top right hand corner.



```
100 REPeat get_num
110  CLS
120  INPUT"Enter number ";num$
130  check num$
140  IF flag=0 THEN EXIT get_num
150 END REPeat get_num
160 LET num=num$
170 PRINT"num = ";num
180 DEFine PROCedure check (x$)
190 LOCal n
200 LET flag=0
210 FOR n=1 TO LEN(x$)
220  IF NOT x$(n) INSTR "0123456789."
230   LET flag=1
240   PRINT"Numeric only please! Press
a key to retry."
250   PAUSE
260   RETurn
270  END IF
280 END FOR n
290 END DEFine check
```

Tree

I read with interest the letter from J M Grant in your September issue. Like him, I bought a copy of Family Tree from Microdrive Exchange and immediately sent off for a back copy of the magazine with the original article in it. Unfortunately, I ended up with a July/August 1985 copy of *QL World*, only to find that the article was in the now unobtainable *QL User* of the same date. I would be very grateful if you could send me a copy of the original article.

As a subscriber to QL World for nearly five years, I have found it to be a great help to me as a slowly self-taught user of the QL. Most of my work is on *Quill*, which for me as a self-employed translator has the very useful feature of counting words (we charge by the word, and you can image the nuisance of counting 20,000 by hand).

John Broadhurst
Willenhall
W Midlands

Editor's reply: unfortunately I have had my hands rather full recently and have not been able to process the letters we have received about Family Tree as I write, but I have copies of the original article for people who want it. Please send a large sae.

Space

I am writing to say how impressed I am with Mike Lloyd's *Space Marauders* game, published in the *Super Basic* series (December '90/January '91). As Mike indicated the game is quite fast and playable even in its SuperBasic form.

As suggest by Mike I decided to compile the program using *Supercharge* and, as expected, the increase in speed was most impressive – almost too fast!

However, in order to achieve a compiled version of the program a few things needed adjsting – I though that it would be useful to document these so that others may benefit:

Supercharge objected to "random' RESTOREs, ie line numbers must be explicit rather than calculated. I replaced the offending line (704) with the following code (having first renumbered lines 706-776 to 726-796 inclusive):

```
703 rest = 800+RND(1 TO 12)
704 SELect ON rest
705 = 801 : RESTORE 801
706 = 802 : RESTORE 802
    etc. to . .
716 = 812 : RESTORE 812
717 END SELect
```

Although the Basic version functioned OK, when I attempted to run the first compiled version I encountered a 'Window not open' message against line 115 when using channel two. I have had this problem before and the solution is to replace the WINDOW command, for any channels greater than 1, with an OPEN#n, con_ . . . . command. This may be a problem with my particular set up (JS rom, Trump Card 897K, ICE rom() or, more likely, my own boot procedures/initial selection menus. Having deleted lines 105 and 120 I amended lines 110 and 115 as follows:

```
110 WINDOW#1,488,200,32,16
    Paper#1,0 : Border#1,1,4
115 OPEN#2,con_488x200a32x
    16 : PAPER#2,0: BORDER#2,1,4
```

To avoid compile warnings on lines 930 and 1025, I replaced the NEXT statements with the Superbasic standard END FOR statements.

To enable a more 'controlled' exit from the program, via ESC, (essential for compiled programs) I added/amended the following lines to allow ENTER to continue or ESC to exit whenever text is scrolling in the 'red planet':

```
135 hi = 100: noise 0: esc = 0
282 IF es : STOP
367 IF esc : STOP
620 AT#2,1,23 : PRINT#2,
    "ENTER/ESC"
637 IF CODE (INKEY$(2)) = 27
    : esc = 1 : EXIT show
```

I trust that the above will be of interest and thanks again for an interesting game and magazine - keep up the good work.

Ken Law
Southbourne
Bournemouth

Art

Having read your inestimable magazine since its inception, I now have a list of wants from you!

First, I would like you to run the QL Art competition again – the one where people must write programs no longer than 50 lines long that produce impressive screen displays, I really enjoyed some of the last winners' listings, as I typed them all in – 50 lines take no time at all. I thought that the best was TREE, because alteration of some of the program parameters made the mountains and the trees *very weird.*

Secondly, I would like to request anyone who has a Q-Talk unit, and is bored with it, to get in touch with me. I have a desire to hear my QL say something other than Bleep, even if is a fuzzy-wraparound-random-bleep. If anyone has one, please will they contact me daytime on 081 842 5855 or evenings on 0844 291186, and I will pay any reasonable price.

Ron Peach
Haddenham
Nr. Aylesbury

Editor's comment: This is another letter from the 'lost in March 1990' batch, but fortunately Ron gave us his phone number and we were able to contact him to update the letter. We had another go at the 'short program' competition early in 1990, but the time wasn't right either for us or for the readers. We need another sponsor. Any offers from established suppliers will be looked at seriously.

T ROUBLE

A PROBLE

Bryan Davies catches up on readers' enquiries.

Using the major QL programs is a relatively leisurely business, and many users will have come to believe that this is largely the fault of the hardware rather than the software. It comes as a bit of a shock when a new program raises the tempo of operations appreciably. The word-processing program *Perfection* is remarkably swift in most respects, and makes it clear that the 7.5 MHz central processor in the QL is adequate to provide good performance *if program code is sufficiently well-written.* As yet, the version of the program used in my system is still a beta-test one, and lacks some major features such as the spell-checker and the graphics-import function, but the full version should be in the hands of users by the time these comments are printed. What particularly impresses me is the way Perfection on the QL compares favourably with well-known wp programs on the PC; in contrast to *Quill,* which looks very slow, Perfection imposes no serious restraint on the user. The performance on the ST Atari with QL emulator must really be something to behold (check **Jochen Merz Software** for details of the emulator board).

It was good to see Sector Software once again, at the All Formats Computer Fair in London on February 2nd. This supplier has not been much in evidence for a while, and the reason given for this is that a concentrated programming session has been necessary to complete a program for pairing STD telephone codes with geographical locations. STD codes are the UK telephone codes assigned to localities, such as the 071 and 081 used for London. There are around 6,000 of these on Sector's list, all but a few of them being paired with the corresponding localities. Versions of the program are offered for QL, PC and Amiga.

EEC Ltd. are selling an interface to enable standard PC keyboards to be used with the QL. They also offer two versions of an Epson PC keyboard. The interface appears to plug in neatly on the QL circuit board, and a DIN connector is located adjacent to one of the ports at the rear of the casing, so that the new keyboard can be plugged in and out without opening-up the QL. The QL keyboard has never upset me enough for me to want to get rid of it, but a fast typist would feel more comfortable with the PC type; in addition, it would be much easier to use PC programs under the *Conqueror* MS-DOS emulator with a pukka PC keyboard. Remembering the equivalent QL keyings for PC program commands is not an easy task.

Although many programs stay unchanged from the time they are introduced, others are subject to updates, albeit at irregular intervals. Digital Precision point out that their *Turbo* compiler is now up to version 3.24. The current program disk contains a collection of files having different version numbers; there is a detailed explanation in the instructions of how to select the appropriate versions of certain files, to suit the 'age' of the compiled program being updated.

There is also reference in the instructions to compatibility of the Minerva rom with Turbo; the general message is that (hopefully) final changes to certain parts of Turbo will be made if/when the Minerva 'goalposts' stop shifting! *Media Manager S.E.* has been updated, but no major change is reported.

The Transfer Utility is a handy program for making copies of media where some changes need to be made *within* the files contained, as, for example, when a program on microdrive has to be converted for use on floppy disk. This program allows the user to specify up to 32 substitutions of as many as 64 characters each, to take place during the copying procedure. The obvious substitution would be 'flp' for 'mdv', but there is no reason for the exchanges to be limited to device names – you could use the function for Search and Replace operations on text in documents.

An unusual addition to the DP range is DR-DOS version 5.0, the alternative operating system for PC users who can't get on with the standard MS-DOS (or PC-DOS on actual IBM machines). This will be of interest to QL users who have the Conqueror MS-DOS emulator. Among the several plus-points of DR-DOS is the ability to make considerably more of the basic 640 KB of user space available than MS-DOS can.

Readers' letters

Asking for comments on *Minerva* laid me open to receiving letters that would be, in part at least, over my head. One such came from **PH Tanner,** who is a disgruntled 'owner but not user' of Minerva V1.82. In plain language, he can't use this rom, because it conflicts with software he uses regularly, specifically *QFlash* and the combination of SpeedScreen and *MasterSpy.* Tanner writes in the Forth language, and has a problem with Qdos 'inverse trigonometrical functions'; Minerva appeared to offer a way around this apparent bug. He was not happy (as of early January) with the response he had got from **QView** about the conflicts.

Details of some more conflicts were supplied by **C Grogan.** He has been using versions 1.65 and 1.82 of Minerva, and thinks some of his problems occur only with the latter version. The SDUMP screen dump command incorporated in some toolkits (eg with the Trump Card) no longer works, either from the typed SDUMP command or via a preset hot-key, although the Qram dump does work. In addition to the problem with the View screens, use of the Arc command of *The Painter* now causes that program to 'disappear from memory', although the QL doesn't crash. The Circle and Ellipse commands do not draw anything visible on the screen, and the Turn Shape command produces an immediate crash (these problems did occur with V1.65 Minerva too). These comments apply to the 3.02 version of The Painter; with the 4.00 version, Arc is replaced by Bezier Curves, which work, as do the Circle, Turn Shape and Ellipse commands, but View Screens still causes trouble, and Load Patterns can produce 'visually-spectacular crashes', with the V1.82 Minerva.

3D Precision is another program in trouble; the small cursor is not visible, but one can draw with it. With *Professional Publisher,* problems occur in use of the Edit Page and High Definition (when loading text) options. A plus-point is that the 'fat founts' can be used at any CSIZE setting without the distortion he had experienced with the JM rom. The automatic offering of the name 'flp2_test_task' when Charge is entered, with the Turbo compiler, is no more and 'Parser_task' is offered instead. The Codegen_task compiler function does not start automatically, and the EX command has to be used for it. 3D Designer simply crashes when loaded.

There are programs which do work for Grogan, however. He names *text^{87}, Editor, TechniQL* and *QLiberator; Conqueror* will work if a POKE is keyed-in beforehand. He is obviously not entirely dissat-

SHOOTER

M S O L V E D

isfied with Minerva, as he quotes some advantages for it, but he wonders what other problems are lurking, waiting to cause trouble for him and other users.

A complaint from reader **Hans-Peter Thun** some months ago about a faulty disk drive supplied by **Miracle Systems** apparently didn't get him any solution, but he moved from Germany to England and got some advice, which resulted in his experimenting with disk drive jumper settings and connections. The end-result of this was one ruined QL, and the purchase of a second-hand PC/XT! This might have meant the end of his life as a QL user, but that was not to be. He has since yielded to requests from his daughter to provide her with a fully-operational QL, and now has three varieties of QL in his stable.

File Transfer

It would take a whole article to go over the points Hans-Peter made, and there's not enough room here for more than this one. He had read my article on file-transfer between QL and PC versions of Archive, and he needed to transfer some Archive databases from his QL to the XT. He successfully transferred both files and routines, converting the latter where necessary to run in the PC-Four version of Archive. As he rightly says, Archive is not the most common database program on the PC, so what about transfer to other programs? Amongst other conversions, he has rewritten his QL Archive routines for use with *dBase III* on the PC, and has exported QL *Easel* graphs – using *DiscOver* – to Harvard Graphics, then imported them into Microsoft *Word* (a word-processing program with graphics features) and printed them from the PC. It is clear that some of these operations may not be justifiable from an efficiency point-of-view, but they *are* interesting, and provide added flexibility in one's computing activities.

After transferring files for a year or so now, I no longer regard QL-PC discourse as any big thing, but my operations are usually in the strictly-limited area of Ascii text files. It is true to say you can have far more trouble moving text files between certain PC WP programs than between QL and PC; try making sense of 'text' files from the IBM WP program *DisplayWrite,* if you don't mind being driven up the wall a few times. To my chagrin, my betterhalf occasionally requests assistance with DisplayWrite, and it doesn't take more than about ten seconds of contact with the program for me to be hopping mad.

P J Wilson was one of the people who ordered the PS/2 keyboard from **Keyboard Products,** at the time when that supplier was said to be having technical problems which were delaying shipment of keyboards. He cancelled the order after trying a few times to get assurances on delivery, but had not received a refund of the money paid out in July, as of late-January (despite being advised in November that a refund would be made 'in a few days'). The suppliers assured me some months ago that the problem had been fixed and keyboards were being shipped, but **L M Furminger** says he has been trying from September last year to early January this, to get his money refunded, following failure to deliver a PS/2-style keyboard. He has now threatened to take legal action against the company. **Michael Jackson** says this company delivered his keyboard 'five months after cashing my cheque', but it proved to be faulty. He is awaiting response, after returning the keyboard. The main fault was random generation of characters; what appeared on-screen was not what had been typed. Jackson had advised the company that he was intending to write a review of the keyboard for Quanta or *QL Technical Review,* so it is all the more surprising that his complaint has not yet been dealt with. He asks whether any other *QL World* readers have bought the PS/2 keyboard (current version, not as sold by Schön) and used it successfully.

A satisfied purchaser is **Martin Neave,** who ordered an Epson GQ-5000 laser printer from **SCS Computer Supplies,** following my comments on a GQ bought from that supplier. My experience of delivery was not exactly good, but that was the fault of the delivery company rather than SCS, and Neave reports that he was very pleased with the prompt delivery of his printer, and the responsiveness of SCS.

As was my experience, he found that he was getting prints from the GQ (using the QL with The Editor and Professional Publisher) within a very short time of receiving it. It is straightforward to set up, and simple to use. My own GQ is now four months old and working well; apart from the habit of feeding several sheets of paper together when I put new paper on top of scrap, there is very little to complain about. It does cost money, though. It so happened that most of my work during this period has involved making many prints, and the cost of toner alone is running at over £10 per month.

The print problem experienced by **Gerard Delaney** when using text[87] and the Kaga-Taxan DMP printer seems to have disappeared, without a reason for it becoming obvious. No-one else has written to report a similar problem, so it looks as though that may be the end of the matter. It is interesting, though, that a copy of text[87] sent to me by Delaney did not work at all on my system, although he says he has had no problems with his disk drives, and a copy made from my master disk onto the 'bad' disk he sent me also works fine on his system. So, does one of us have a faulty disk drive? We both say 'no', and are none the wiser as to how the undoubted corruption of the program file occurred.

Delaney wonders if a corrupted program file will run at all, and the answer to that is simply that corruption can take a great number of forms, of which many may not prevent the program loading and working apparently normally. When in doubt, replace your working copy by another one, taken from the master; always write-protect master disks, to make it unlikely they will become corrupted. **Peter Tomlin** also uses the Kaga-Taxan printer, and he offered a SuperBasic program for checking the printer interface; this will be sent on to Delaney. The SB program comes from the manual supplied with the printer.

A complaint from **D S Graham** about non-reply by **Digital Precision** to his request for a replacement program disk may well have been sorted out by now, as the letter was written in December last, but D P will be asked to comment upon the complaint. Although a reader was kind enough to supply an address for **Buzzz Software,** that didn't get **John Easlea** anywhere in his quest for help on the *Home Finance* program. The reply he got was 'haven't dealt with the QL product for years'. Easlea has written to **John Hadley,** who supplied the information on Buzzz, to ask if he can provide any further help.

A lost Miracle hard disk is **Michael Cronsten's** problem; he ordered one from **T K Computerware** in May 1990, and says he had not received it as of January this year. T K told him it was despatched in May-June and must have got lost in the post; they were making a claim for it to the post office. It certainly seems an unduly long time for a claim to be sorted out. As the unit was paid for by credit card, Cronsten should certainly try to reclaim the expenditure from his credit card company.

Mike Jackman has been trying for some months to get a problem with using **TechniKit** (from T K Computerware) sorted out. The 'plot' module works correctly on the QL (but not on his Thor) with a Penman plotter, but he wants to produce drawings for use in articles by making use of the 'screen dump' module, and he says the module doesn't work correctly. On the positive side, he praises a scanner bought from T K. He did have problems getting it to work at first; the software provided didn't work with his Epson RX80F/T printer, but modifications to the software were provided and the scanner has been working successfully for about a year. Check when ordering that the software supports your printer.

Jackman notes that a letter in the December issue of the Quanta group newsletter gave yet-another method of adding fields to an Archive database. The approach suggested is to Export the file from Archive, Import it into Abacus, use Amend to change fields, or add new ones, then Export the file and Import it back into Archive.

A copy of the instructions for the Silicon Express disk drive interface has been sent to **Nigel Ford. J Roy Goodall** has become the most-regular correspondent – he seems to try everything available, and have his fair share of problems. He says he has given up on *TechniQL,* and is now using *Cadette* from **BestMalt;** this does all he requires of it, and prints without problem. His catalogue of difficulties gets larger and larger. No response has been received to his letters to **MBS Data Efficiency, Frequency Precision, Transform Ltd.** and **Sector Software;** he also wrote to another J R Goodall (a *QL World* reader who wrote to us regarding the transfer of data between QL and Psion Organiser) but his letter was returned marked 'gone away'. On top of all this, his PS/2 keyboard was returned to him by **Keyboard Products** but 'still does not work'.

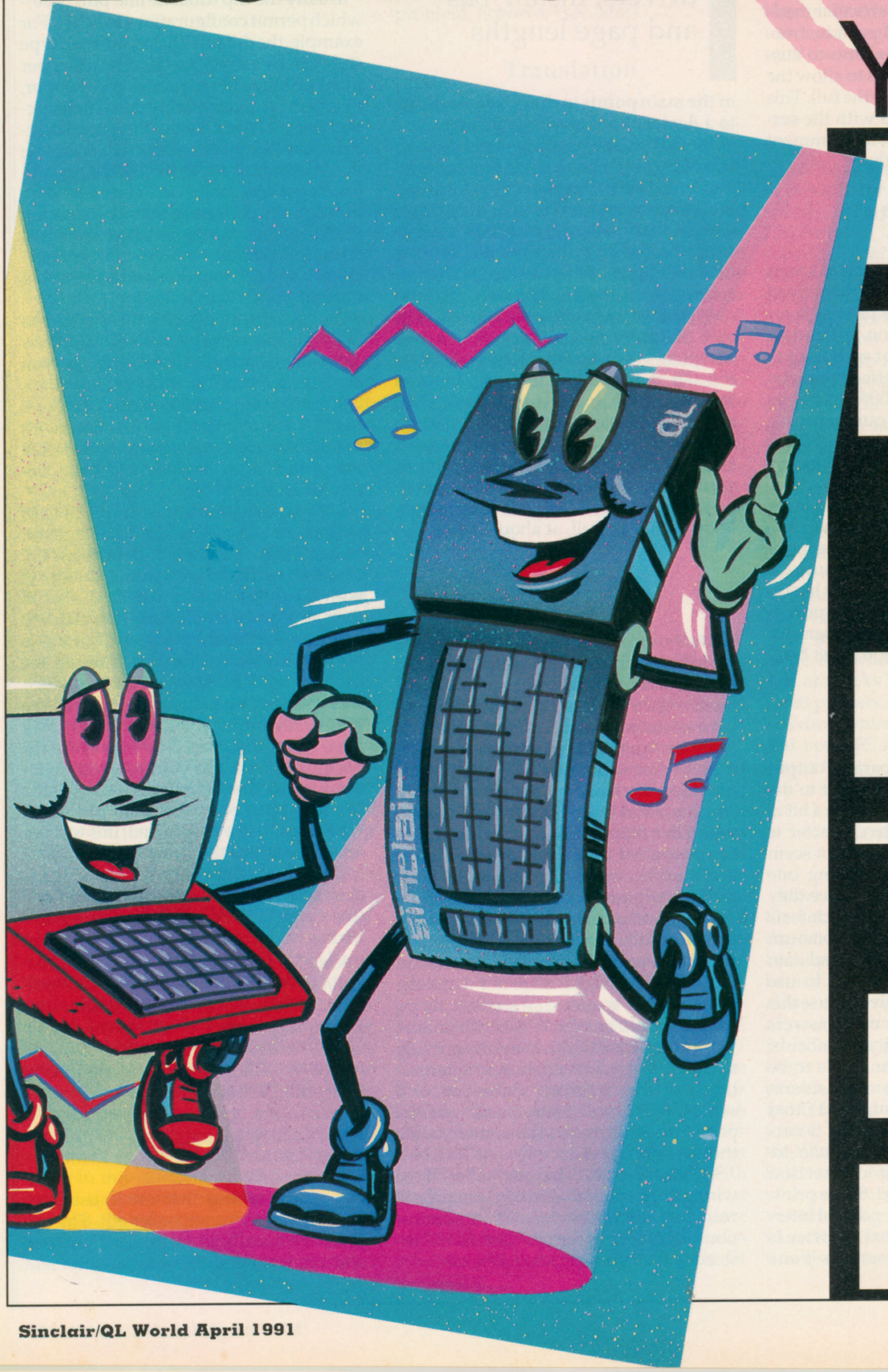
The previous mentions of his problems lead to a call from a neighbour in Belize, who also is a QL enthusiast and QL World subscriber; they have been able to help each other in various ways. His latest wish is for a fax card for the QL. I haven't heard of one, but it does seem a useful thing to have, particularly for those users who already connect their QLs to bulletin boards. On this subject, **TF Services** is the obvious supplier to talk to about connecting your QL on-line to bulletin board services (and also about QL repairs).

For readers who haven't the slightest idea of what a fax card is, it is now possible to have a facsimile transmission feature within your computer (leastways, it is if you have a PC). This function enables 'pages' of information to be passed between computers which can be many miles apart, without the need for a 'reading machine' to scan a document at the originating end, and a similar device to print it at the receiving end. The information is transmitted as a computer file; that is, you can create a document in your wp program, pass it to the fax program, and have it sent to a remote location; at the remote location, the file is received and stored, to be displayed on the screen as and when desired, and printed out (on an ordinary printer) if necessary. All that is required is an additional circuit board and some software. Has anyone developed such a device for the QL?

Sector Software has been the target of several letters of complaint during the past few months. As noted elsewhere, they explained that heavy involvement in a programming project had been the reason for the lack of response, but that is hardly justification for not dealing with QL users' complaints and it is to be hoped that 'normal service will be resumed shortly'. Sector has previously been praised by many customers. **S Long** ordered *FlashBack S E* and some cartridges late in October, and received the cartridges and the instructions for FlashBack, but no program cartridge. The usual kind of story is told – several 'phone calls, but no satisfactory answers, as of early January. He has had trouble when calling, with the fax machine answering instead of the normal 'phone on occasions. The problem may relate to the transfer of sales of this program from Sector to **DJC Computing,** and/or the return of Sector to their old address (in Ulnes Walton).

Just for Starters

Learn to Love Your PRINTER



LOVE YOUR PRINTER

In the first of two articles about living in peace with your printer, Bryan davies looks at printer ports, dip switches, printer drivers, Install_bas and page lengths

There doesn't seem to be much argument among computer people, that the one subject which generates more questions than all others is printers. This article actually started life some time ago, and 'missed the cut', so to speak, but there are still enough QL users having problems getting the desired output from their printers to make the points made here as valid now as they were then. There are two main groups of question associated with printing: how to get particular models of printer to respond to print instructions from the computer, and how to 'customise' the driving programs to allow the printer features to be used to the full. This article is mainly concerned with the second category, but here is some comment on the first one to start with.

Printerports

The basic QL has only two external ports for connecting to printers – those marked SER1 and SER2. So there is provision for only serial printer connection, where data is sent in a stream of 'bits', at speeds up to 9600 baud. This gives an effective transmission rate of around 1000 characters per second. Some printers cannot handle this rate of data input, and a lower one has to be set, but that should not be necessary with the type of printer sold these days. Since printing speed is usually no more than about 150 characters per second, the printer needs to be able to buffer the input to some extent. When large documents are being printed, a buffer is a very handy device to have; it can be in the printer itself, or set up in the QL (some programs, such as *Taskmaster*, allow you to set buffers for this purpose).

Parallel

Typical office PCs have a parallel output port, which allows data to be sent to the printer a byte at a time, rather than a bit at a time. That is, a complete character is sent. In general use, there does not seem to be any great merit in selecting one method rather than the other, since they both work satisfactorily, and the printer is too slow a device to work at the maximum rate at which the computer can send data anyway. It is, however, simpler to use parallel transmission, mainly because this does not demand that the user answers questions (what is the parity, number of data bits etc?). The decision tends to be made for you by hardware considerations. Printers are mostly supplied fitted with a parallel interface, making it necessary to purchase a separate *parallel-to-serial converter module* if connection through a serial port is used. Some printers have both a serial and a parallel interface fitted, but the additional interface is normally an extra-cost fitment; **this is one of the main points to check when talking to a dealer about buying a printer.**

One advantage of using serial connection shows up when linking two computers to one printer; far fewer wires are used for the serial connection than the parallel one, so that making a switch is a much simpler job. The printer parallel connector has 36 pins, whereas the PCC serial connector on the QL has only six (and only three of them are actually used). The cost of cables and switch to link two computers to one printer for parallel data transfer is typically about £50, but you can pay much more if you don't shop around. You can quite easily make up both switch and cables for serial transfer, at a cost of around £5 to £15. However, you will most likely have to buy a parallel-serial converter (Miracle sell the best-known one) as well, at about £30.

Interface

The choice only arises when the QL has an interface fitted with a parallel port, such as the old Medic or Sandy SuperQ interfaces. When the QL is to be connected permanently to a serial device other than a printer – a modem, for instance – you may be forced into getting this type of interface. The same applies if you want to run, say, a dot-matrix printer and a daisy-wheel one from a single QL, when it may be convenient to use one in serial mode and the other in parallel (although I don't think many people can be successfully driving two printers at the same time – certainly not if they wish to carry on using the computer for another job). Note that the Ser1 and Ser2 ports are not wired the same way; it is sometimes stated that a printer should be tried on Ser2 if it doesn't work on Ser1, but Ser1 is the port normally used. To allow for variation in interpretation of the 'Standard' definition of a serial (RS-232) port, Sinclair arranged one port to be, effectively, a transmitter, while the other is a receiver.

Ser2 can be made to be just like Ser1 if the wiring to the connector fitted to it is altered; read the section 'Concepts – Communications RS-232-C' in the *QL User Guide*. Instead of using pins 1, 3 and 4 (as on Ser1), you have to use 1, 2 and 5 to make the same connection from Ser2 to the printer. If you have a QL with a D-type of connector for the ports, the equivalent wiring for Ser1 is also to pins 1, 3 and 4.

Dip switches

Setting up the printer can be tedious. Printers often have sets of small switches – usually the dip (dual in-line pin) type – which permit configurations to be set. For example, the default (that is, normal) type style can be set as Pica (10 characters per inch) or Elite (12 characters per inch) or, maybe, near-letter quality (also 10 characters per inch, but better-formed characters than Pica). The character set can be the USA standard, or the UK set, or others. Paper size can be set. You have to go through the process of setting/checking these switches, as instructed in the manual with the printer, before you connect printer and QL together and start trying to print.

The settings made with these switches can – in many cases – be over-ridden by software commands from the QL, so you are not committed to using only Pica, USA characters, quarto paper, etc., just because you have set these functions with the dip switches, but you do need to tell the printer what its basic, start-up configuration should be.

One frequent problem, is the printing of various other symbols when the English £ (money) sign is intended. If there is a UK character set in the printer, this can be set with the dip switches, and the £ should then appear on the paper when that key has been used on the keyboard, but this is not the only way to obtain it. You can, for instance, set the USA character set, which will very likely produce a # sign from the (QL) £ key initially, but the printer may have more than one character code for producing the £ and you can send a conversion instruction from the QL to tell the printer to use one of its built-in £ signs when the QL £ key is pressed; this will not affect the use of the # key, which should still produce the printed # sign, which is desirable in some technical applications (the # sign is often used to indicate numbering, and it indicates pounds weight in the USA). The translation sequence 96,6 (both codes being decimal) should work on a variety of Epson-compatible dmp printers; 96 is the code produced by the QL when the £ key is hit, and 6 is the code recognised by the printer for producing the £ sign from its built-in set.

Page length

A switch setting that can cause later trouble is that for paper length. The options are usually 11 inches or 12 inches, and the setting at the time of shipment depends upon where the printer is intended to be used. US standard paper is 11 inches long, whereas A4 is 11$^2/_3$ inches. The 11 inch setting is suitable for both cut sheet and continuous A4 paper, with my Kaga-Taxan (Canon) printer. Getting the printer to move continuous paper so that the top of the next page moves up to the correct position each time is an art in itself, and is almost certain to be a matter to be dealt with in the QL software, rather than with the dip switches.

Printer drivers

Setting up printer drivers: for key-combinations joined by dashes, such as CTRL-C, hold down the first key and tap the second one.

This is the area where the main trouble seems to occur. Programs which provide printed output have some form of 'printer driver' routine, to format the text produced with the program into a 'shape' recognisable by the printer. Even such basic things as the codes used for characters can differ between keyboard and printer. No matter what the program, the screen never has a WYSIWYG presentation – the printed output can look very different from what you see on screen. It amazes me that programs are referred to as having 'what you see is what you get' presentation, when they clearly make no more than token efforts to show text as it will appear from the printer. *Quill* is better than many word-processing programs in this respect; it does, at least, indicate areas where text is 'non-standard', such as making bold characters white instead of green, showing underlining on the screen, and using different character shapes and positions for superscript and subscript. Typical PCs seem incapable of showing these features adequately and the PC version of Quill has to use non-obvious representations of these features, such as highlighting, and the same colours for different features (superscript, subscript and italics look the same on my PC). Be happy that the QL and Quill are so good.

Text codes

The nearest to true WYSIWYG I know of is the *text^{87}* presentation, where characters can be different styles, sizes and colours, and there are no 'surplus', non-printing characters displayed on-screen. The 'far end' of the spectrum here is the typical text editor program, which is designed primarily for program writing, and makes no attempt to alter screen characters to match what will appear on the printer. The program in this category which is most spoken of is *The Editor*, there are many good reasons for using this program, but the need to generate – and display on the screen – a set of codes for almost all printer typeface functions can put users off.

One good point in the later versions of this program is that these special codes do not count as columns on the screen (when in Document mode), which avoids the problem present in Quill of having line-justification upset when the codes are removed at print time. In the extreme, printed lines from Quill can look a centimetre or so shorter than they should, because of the gaps left by codes which have been removed; something can be done about this problem, however (see later).

Translation

Try to envisAGE what is required between computer and printer. Effectively, they are two devices speaking rather different languages, and a translator has to convert the output from the computer into 'printer speak'. Whatever the program, printer, or computer, the translation process required for printing is essentially the same. The translator is called a printer driver, and is a program in its own right. A driver with capability to cause all functions on a dot matrix printer to operate can be quite large – for example, over 40 KB for *The Editor*. The fact that the Printer_dat file for Quill is only a hundred or so bytes long is only part of the story, since the Quill program file itself contains most of the driver; many printers will produce reasonable output without the Printer_dat file being present.

Embedded

What is being converted is the codes produced by the text program. Examples may best explain what is required. When you press F4 then B, in Quill, the screen shows following thecharacters in white, as a representation of darker than normal characters. A typical printer has two ways of making characters dark – emphasised print, or double-struck print. Each character can be printed with its lines slightly wider than usual, or printed twice in the normal fashion.

There may be little to choose between the appearance of the two styles, but some other features are incompatible with one or other of these two. For instance, emphasised print may work only with Pica (10-pitch), making it necessary to use double-strike with Elite (12-pitch). I find double-strike most flexible, and the code the Kaga/Canon printer expects to see coming from the computer to switch on this feature is ESC-G. The printer driver has to look through your document, at print time, for the Quill codes for F4-B, and replace them by ESC-G each time they appear. With Quill, you don't need to bother about what the codes for F4-B and such are, as they are 'embedded' in the program, but any codes other than the four produced using the F4 key have to be specified by you and, in this respect, Quill and Editor are similar.

At this point, it is perhaps desirable to refer to specific printer drivers, and that for Quill is the one you meet first. It is likely you will be at least partly equipped to deal with other drivers once you have mastered setting Quill up. So as not to make the job unduly difficult, it is desirable to have the 2.35 version of Quill, which has better printer driver features than the earlier versions had. The following points assume the 2.35 version is being used, although the procedure is similar with the earlier ones.

Install_bas

This SuperBasic program configures Quill to produce the kind of printed output you specify. Run the program by keying-in LRUN mdv1_Install_bas (or flp if you have converted it for disk use). You will normally accept the first option offered, for a standard serial printer; if you wish to use the driver over a network, to a parallel printer, or to any other non-standard device, choose the second option. Apart from specifying a different device name, the procedure for setting up the driver is the same, whichever option is chosen, but the program treats the versions slightly differently.

When you choose the standard port option, it is no use trying to persuade the program to accept n2_Ser1 as the port to be used, because it will not recognise it. The standard port has to be called Ser1 or Ser2; using the non-standard/parallel option, the port can be called n2_Ser1, n1_Ser2, Par1, Par2, etc.

The program displays a list of printer types for which drivers are supplied; if you are lucky enough to have one of the listed printers, all you need to do is use the up/down cursor keys to select the one, and press F5 to install the stored printer driver for it.

Compared codes

More likely, your printer will not be on the list, and you have to decide which of the listed types is most similar to yours. For 'Epson-compatible' printers, the best bet is EPSONFX; this should, at least, enable you to print, using the basic functions of your printer. Should you have no idea which driver is most suitable to use as a basis, select each one in turn, open your printer manual, and compare the codes used for the functions, by Quill and your printer. Make notes for each, and decide at the end which needs least modification. Select by F1 for COPY, not F2 for EDIT, as you need to produce a new, modified version of the stored one, not to change it. (As always, you should not be doing any of this if you haven't made a backup copy of the Install_dat and Printer_dat files first).

The first item on the list of entries then displayed is Driver name, and you enter some recognisable abbreviation for your printer here. Whatever name you choose, it will not appear on the final file – that will always be Printer_dat. If you have a utility program with a 'View' facility (such as *Ice*) you can look at Printer_dat and see the name and function table entries for the current driver. The Port entry is ser1 or similar – no spaces. Baud rate should be set to the highest rate your printer supports; 9600 is usual. It is worth trying a lower setting if subsequent printing reveals missed blocks of data.

Parity has to be checked in the printer manual also; look under a heading such as 'Data Transfer' or 'Communications'. If no information is given on this, start with the setting NONE, and change it only if there seems to be some data transmission problem. These last two entries do not appear when setting up a driver for a non-standard port, which is perhaps reasonable for a parallel port connection, but I don't why they are missed out for a network connection through a serial port.

Page lengths

Lines/page is fairly obvious, and 66 is the default setting, both in the printer driver and within Quill (see Page size on the Design screen). An A4 page is actually long enough for 70 lines (at the standard six lines/inch), and you may want to use this setting, if you have reason to squeeze as many lines as possible onto a page. Note that your printer may not allow you to position the paper so that the first print line is at the top edge of the paper.

There is not much room for manoeuvre with cut sheets, but you can set continuous paper so that the top edge is immediately above the print position, and get an extra few lines. Beware doing this on the first sheet though, as the paper is liable to get mangled if you do. Check the Upper margin setting on the Design screen if a big gap is being left at the top of the page; the default setting for the Upper margin is six (lines), which usually means the printer upspaces 1 inch before printing. Add this to the 'dead space' – the gap between the print head and the top edge of the paper when printing is started – and you can lose about 10 lines.

Be clear

Depending on paper and printer, an Upper margin of 0-2 should be usable. It is sometimes desirable to let the printer upspace for 1 or 2 lines, as it may have a tendency to print some of the first lines too close together when the margin is set to 0, especially if you are in the habit of winding continuous stationery backwards before starting printing.

It is very important to get clear in your mind just what you are specifying by 'Lines/page', wherever you come across the term. Programmers are not consistent in their use of the term, and instructions sometimes fail to make the different uses evident. Telling the printer there are 70 lines per page, through the printer driver, does not commit you to having 70 lines of text printed on every page, but it does allow the printer to make full use of the paper, if required to do so.

Finding errors

When you enter Page size with the Design command, working in Quill, you are stating how many lines of text you want on the printout (including any Header or Footer lines specified), and this value can be set to give a printout which is within the printer's range, and looks acceptable. If you use continuous stationery, and have difficulty persuading the printer to move to the top of the next page, or to print what you put on the bottom of the screen page correctly, you have two places (maybe more) to look for the error – the dip switch(es) inside the printer may be set to the wrong paper length, or the combination of upper margin, lower margin, header and footer settings may not add up to the Page size set on the Design screen.

The standard number of characters/line is 80, based upon the Pica fount, which gives 10 characters per inch as used in dmp (dot matrix) printers. That leaves 1$^1/_4$ inches total for left and right margins. As with lines/page, there are some people (certainly myself) who stuff more onto a page than secretarial school would approve of. Apart from this, some also use Elite fount, which gives 12 characters per inch.

Mixed sizes

In some circumstances, it may be essential to use condensed print and reduce the margins to a minimal size to get all the required text onto a page, such as with tables of values. 80 characters are not enough to cover these requirements, and you can set the value to whatever your printer will accept. This setting is actually specifying the width of paper the printer will use, as well as the number of characters. Condensed print gives about 17 characters/inch. If the specified characters/line value is 80, only 80 characters can be printed on a line but, if the value is set to whatever maximum number the printer can cope with, then that number will be printed. Typically, a printer can manage 132-134 condensed characters on a line.

This brings up a problem which you will meet if you mix character sizes in a document; generally, you will want the text to fit in between the same margins, regardless of character size and you need to devise a set of values – for left and right margins, and Characters/line – that achieve this. As far as the printer driver is concerned, it doesn't matter too much what its setting are, because the settings made in Quill when a document is prepared will override the driver settings. As an extreme example, you can set the driver to 10 Characters/line and 10 Lines/page, but a document with margins 60 apart and with 50 lines of text will be printed as it appears on the screen.

There can be some limitations on this. With the values given, you might find an indented paragraph first line becoming 'outdented', and ending short of the right margin. My own feeling is that you might as well set the driver to the maximum values the printer can cope with – say 134 Characters/line and 70 Lines/page – regardless of what character size you normally use. This avoids the need for having two printer drivers just to cope with documents typed in 10-pitch and 12-pitch (it will be explained later how the different character sizes can be "called" during typing of the document).

A wrinkle

The Continuous Forms setting deserves some consideration. The obvious answer to insert is 'No' for cut sheets of paper, and 'Yes' for continuous stationery. One drawback I find to answering 'Yes' for continuous paper is that you can't (neatly) stop printing when you notice that something is coming out wrong. Therefore, I still insert 'No' even though continuous paper is loaded most of the time. It does mean having to hit a key at the end of each page, to get the next one printed, but it allows you to 'vet' pages one by one, and call the print off when a mistake is spotted. What if there are many pages to print and you know there are no mistakes? Simply find yourself a heavy weight, about the size of the ENTER key (I use a small magnet, although it would be preferable to use something non-magnetic, so close to a microdrive). Wait till the first 'Press Enter' message appears, at the end of page 1, then put the weight on the ENTER key to hold it down. That will ensure that printing is continuous from there on, but you need to be around when it is nearly finished, because you will get an endless stream of blank pages after the final page of text has been printed unless that weight is removed in time! The End of line code defaults to CR,LF, and should be left at that unless your printer automatically line feeds with just the CR (carrier/carriage return) code.

That is the basic section of the printer driver. From there on comes the section in which you can 'make your mark'. All the following entries can be set to give printer functions which you require. These will be dealt with in Part 2 of this article.

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

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

MIRACLE SYSTEMS

We are exhibiting at the

CONTINENTAL MICROFAIR

St. Joriscollege,
Rostenlaan 296,
EINDHOVEN,
NETHERLANDS

Saturday 6th April, 1991

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 ______________________________
____
____

SUPER BASIC

Make the most of your multi-tasking programs with the concluding part of Mike Lloyd's investigation into boot files.

Boot files are those essential collections of commands which ease the chore of loading useful programs such as *Quill, Desktop Publisher, Flashback* and so on. Due to the comparative size of most quality software packages and the capacity of a single microdrive cartridge, early boot files had the simple task of loading the single program contained on the microdrive. With the widespread introduction of floppy disks which are faster, safer and hold roughly seven times as much as a microdrive cartridge, boot files have had to adapt to cope with a changing demand.

In the February issue of *QL World* the rudiments of the various program-loading commands were covered from a non-technical viewpoint in order that as many QL users as possible could begin to modify boot files to suit their own needs. This month's article examines the way in which a boot file for a disk full of Psion programs might be written. Again, the emphasis is on simplicity so that non-programmers can benefit from modifying their own boot programs.

Before starting to play with boot files, one worthwhile hint is to remember to copy any existing boot file to a safe place so that, if your replacement fails to work for any reason, you can copy the original back. To take a copy of a boot file, type in the command:

COPY FLP1_BOOT TO FLP1_OLDBOOT

A second hint is to make sure that before testing any boot file it is itself saved to disk or microdrive. Many is the time that I have developed a boot file which eventually deletes itself, only to find that I had forgotten to save it before testing it, losing all the work that went into it. A final thought for the more advanced dabblers: do not forget to test a boot file following a full reset and a 'warm' reset of the computer before declaring it error-free.

The boot program developed for this month's article assumes that all of the Psion programs are stored on a single floppy disk. It works whether or not you are lucky enough to have Qram or Qpac2 loaded on your machine. It will work with a purely microdrive-based system, but the advantages are fewer due to the need to switch microdrives to load different Psion programs. Of course, it is perfectly possible to take this boot file as a model for one which loads all sorts of executable programs other than those bundled with the QL.

From the user's perspective, the example boot program produces a menu from which any Psion utility can be selected. On exiting from the Psion program, the menu re-appears and another selection can be made.

Without Qram or Qpac2 fitted and extra memory on board, only one Psion program at a time can be loaded, but this ought to suit most people's working arrangements.

The reason that the menu can re-appear is that a SuperBasic program – the contents of a boot file form a simple SuperBasic program – can be retained in memory even when executable programs, such as *Quill* and *Archive,* are loaded. Indeed, larger QL systems can have three or four

LISTING 1: EXAMPLE BOOT FILE FOR PSION DISKETTE

```
                                                      (Prepare Screen)
100 REPeat LOOP
110   WINDOW 512, 256, 0, 0
120   MODE 4: CSIZE 1, 0
130   PAPER 4, 0: CLS
140   BLOCK 256, 200, 127, 20, 7
150   BLOCK 252, 198, 128, 21, 0
160   PAPER 0: INK 7: UNDER 1
                                                           (Draw Menu)
170   AT 3, 20: PRINT "PSION PROGRAMS DISK MENU"
180   INK 4: UNDER 0
190   AT  5, 20: PRINT "A.   QUILL"
200   AT  6, 20: PRINT "B.   ARCHIVE"
210   AT  7, 20: PRINT "C.   ABACUS"
220   AT  8, 20: PRINT "D.   EASEL"
230   AT  9, 20: PRINT "E.   SUPERBASIC"
240   AT 10, 20: PRINT "F.   FANFOLD PAPER"
250   AT 11, 20: PRINT "G.   CUT SHEET PAPER"
                                                      (Read keypress)
260   LET KEY$ = INKEY$(-1)
                                                      (Act on keypress)
270   IF KEY$ == "A": EXEC_W FLP1_QUILL
280   IF KEY$ == "B": EXEC_W FLP1_ARCHIVE
290   IF KEY$ == "C": EXEC_W FLPL1_ABACUS
300   IF KEY$ == "D": EXEC_W FLP1_EASEL
310   IF KEY$ == "E": EXIT LOOP
320   IF KEY$ == "F"
330     DELETE FLP1_PRINTER_DAT
340     COPY FLP1_FANFOLD_DAT TO FLP1_PRINTER_DAT
350   ENDIF
360   IF KEY$ == "G"
370     DELETE FLP1_PRINTER_DAT
380     COPY FLP1_CUTSHEET_DAT TO FLP1_PRINTER_DAT
390   ENDIF
400 END REPeat LOOP
                                                      (Clear screen)
410 PAPER 0: CLS
```

of the Psion programs and a boot program loaded simultaneously. While the executable program is running, the SuperBasic program is dormant: it receives no keypresses, does not amend the screen and its commands are not ignored. When the executable program is finished, the computer returns to the SuperBasic program and resumes carrying out its commands.

Knowing this, the design for a boot file for a disk full of executable programs can quickly be sketched in as a loop in which there is a menu-drawing segment, a pause to pick up a keypress from the user, and a section which controls what happens according to which key has been pressed. There is an enormous benefit in keeping boot files short and simple: concise boot files load quickly, enabling you to reach the important, working programs with the least delay.

Listing one shows the complete boot program, emphasising its simple structure. Within the loop, the default window (Window #1) is re-sized to fill the entire screen and cleared down to a stippled green and black colour which is easier on the eye than solid, vibrant green. Mode 4 has been selected to make the menu text more readable, because for my money the double-width characters of Mode 8 are too large and clumsy to be easily read.

QL users still plugged into a television set rather than a dedicated monitor are catered for by widening the gaps between the characters using the CSIZE 1,0 command. Their needs are also recognised by the positioning of the menu text so that it is well within the restricted screen area visible on tvs.

Onto this blank screen the program next places a black rectangle outlined in white which will be used to display the menu options. The outline is merely the outer edge of a solid white block which is slightly larger than the black block which eclipses it. Suitable foreground and background colours are chosen for the text and the UNDER option is set to 'on' before the menu heading is printed.

While the menu heading is white on black and underlined, the options themselves are in green ink and not underlined, as ordained by Line 180. Instead of some fancy loop containing a READ statement taking information from DATA lines, the program has been kept extremely simple by using straightforward AT and PRINT commands for each option. It is in the nature of boot files that they are frequently amended to include or exclude menu options, so simplicity is usually welcome.

Each menu option begins with a letter. To select an option the user has to press the appropriate letter. The INKEY$(-1) function waits until a key is pressed and then stores the keypress in the KEY$ variable. The following few lines repeatedly test KEY$ in order to take the relevant action, if any. Note the use of double equals signs to indicate that both upper and lower case keypresses are accepted as being valid. If an incorrect keypress is encountered the loop simply cycles again.

Most of the menu options cause an executable program to be loaded and run. On exiting such a program, the loop is repeated, drawing the menu once more and waiting for a further keypress. If the 'SuperBasic' option is chosen, the loop is exited and the screen is cleared before the program terminates. When the boot program has been fully tested, an extra line containing the NEW command could be appended to the end of the program so that when the "SuperBasic" option is chosen the QL's memory is cleared out ready for a new SuperBasic program.

In addition to the mundane task of launching other programs, boot files often include useful extras which are either implemented automatically before the menu is drawn, or are offered as menu options. A good example of the latter type is included in **Listing one** at options F and G.

Ram disk

An annoying feature of Quill and its stablemates is that there is no easy way to change features in the printer driver while Quill is running. I often need to print out quick drafts of articles on continuous fanfold stationery before committing the final article to higher-quality cut sheets printed in near letter quality (nlq). The Psion printer driver can be configured to meet one requirement or the other, but not both together. Therefore, I have developed two printer driver files, one for draft quality printouts using continuous stationery and the other for nlq printouts on single-sheet stationery.

Options F and G of the boot menu swap between these two printer drivers, allowing me to print in exactly the style I want. The preliminaries required to introduce this system are a bit long-winded. Firstly, load the Psion printer driver editing program, an excessively long and over-complex SuperBasic program which seems to take an age to sort itself out. Create or modify a driver, install it as the 'current driver' and save all the drivers. Then, stop the program (using CTRL-SPACE as you would for any other SuperBasic program) and copy the file called 'printer dat' to another file which should be given a meaningful name, such as 'fanfold_dat' or 'nlq_dat'. If there is another driver you would like to develop, run the program again. You will notice that there will be a very long, unavoidable wait before the first menu options are displayed again.

Now that you have a set of files each containing exactly one printer driver, it is a simple matter of remembering to copy the one you want to use into the file called 'printer_dat' before printing anything. The 'printer_dat' file is the one which the Psion programs look for each time the 'print' option is selected.

If you are using Qram or Qpac2 it is easy to CTRL-C out of the Psion program, select a new printer driver from the menu, and then CTRL-C back to the program to use the printer. Readers without either of these utilities will have to make up their minds before calling up Quill, or whatever, because they can only return to the menu by exiting permanently from an executable program. Utilities do not have to be selected from a menu. It is often useful to precede drawing the menu with a number of commands which load background programs and features. Digital Precision's *Lightning* is a perfect example of this. If you have Lightning, you will definitely want to use it to speed up the Psion programs, and so an additional line can be added to the beginning of the boot program which reads:

```
50 LRESPR flp1_lng_text_ext
```

If you are a spreadsheet user, it is worthwhile to add Lightning's maths extensions to speed up calculations.

Before anyone complains that their SuperBasic does not acknowledge the LRESPR command, let me make it clear that LRESPR is provided by *Super Toolkit 2* and by some disk interfaces. If LRESPR is not available to you, you will have to use the RESPR, LBYTES and CALL sequence of commands detailed in the February article.

Another example of an incredibly useful background facility is the ramdisk contained in the Qram suite of programs. A ramdisk behaves exactly like an extremely fast diskdrive, except that as soon as the power is removed from the QL all its contents are lost. Ramdisks can be formatted in a second or two to any size commensurate with the available ram on the QL. Files can be saved to the ramdisk, loaded back again, modified, and so on.

When the Psion programs are being used, a ramdisk is extremely handy for storing printer drivers. If data is being swapped between programs, for instance when data is being extracted from *Archive* to be graphed in *Easel,* a ramdisk is the perfect medium for holding the temporary data file. With additional memory, a QL can easily hold Archive, Easel and a substantial amount of data on a ramdisk.

Regular Quill users will know that the program becomes extremely temperamental when documents of more than about 2,500 words are being worked on, especially if many amendments are being made. When Quill is loaded onto an otherwise empty and expanded QL its appetite for large documents is enormous, but the resulting indigestion is best avoided.

One answer is to format a large ramdisk which reduces available ram to around 150K before loading Quill. The wordprocessor should be configured to create its temporary working file on the ramdisk (use the config_bas program supplied with Quill to achieve this), so that the best of all worlds is achieved. Quill will handle very large documents with high speed because all file activity is kept within the computer's ram, but the problems normally associated with very large documents are avoided. Of course, an even better answer is to use one of the more recent word processors on the QL market, such as *Editor, text*^{87} or the new *Perfection* program from Digital Precision, all of which cope admirably with novella-sized documents.

To establish a ramdisk, a boot program should have a line such as this before the main menu loop starts:

```
70 LRESPR flp1_ramprt: FORMAT
ram1_200
```

Other contenders for the pre-menu installation treatment are the disk-based Super Toolkit 2 commands, Simon Goodwin's *Turbo Toolkit,* the Qram file manager, Qpac2, and so on.

Before concluding this review of boot programs, it is worth mentioning the menuing facilities offered by the recently-released Qpac2 program from Tony Tebby's QJump software house. Qpac2 is primarily a memory-manager, extending the functionality of Qdos in much the same direction as Tebby's earlier Qram suite. However, Qpac2 also includes the Hotkey 2 environment which is excellent for developing powerful menus to control all manner of multi-tasking activity.

The essence of Hotkey 2 in its 'menu mode' is that a single keypress can set off a SuperBasic command, or even a small program, which launches an executable program. The keypress can be almost anything the user wishes, although to avoid confusion with valid keypresses used within programs it is sensible to stick to combinations which include the CTRL or ALT key. It is relatively easy to remember that CTRL-Q means 'run Quill' and CTRL-E means 'run Easel', but CTRL-A cannot be used to refer to both Archive and Abacus. As more and more hotkeys are introduced they become less easy to remember, but with Qpac no feats of human memory are required at all.

Instead, Qpac allows users to create 'buttons' with which to activate programs. A button is a small window containing a program's name. Button windows can be collected in a 'button frame' and displayed only when required. An arrow on the screen is moved using the cursor keys to light up the required button and then the Enter key is pressed. This has the effect of loading and running the program associated with that button.

Buttons have many other uses. They can replace the menu options for changing the Psion printer drivers so that the drivers can be changed without the Quill document ever leaving the screen. Buttons can also supplement the menus of other programs. For example, the awkward menus used in *Desktop Publisher* and to a lesser extent in *Professional Publisher* have always irritated me. Now, the boot file which loads Pro Pub also loads Lightning to speed up screen handling and loads parts of Qpac2 to manage a full button frame containing the menu options which I particularly find useful. The result is that Freddy Vachha is happy because I don't moan at him, and I am getting the full performance from Pro Pub without the limitations imposed by the program's native menu options.

Be they ever so humble, or be they the sophisticated products of the Qpac2 system, boot files are of great value in the QL world. It is perhaps significant that a large proportion of the programs forwarded for consideration for publishing by this magazine deal with boot files of one sort or another. It is also significant that most of them are far too personalised to be of interest to other readers. The main benefit of having boot programs is that they can tune commercial products to meet personal requirements. Use the boot programs provided with software by all means, but rather than complain that commercial programs as provided are not exactly configured to your system or to your needs, take control of your software and change your boot programs to bring your programs into line.

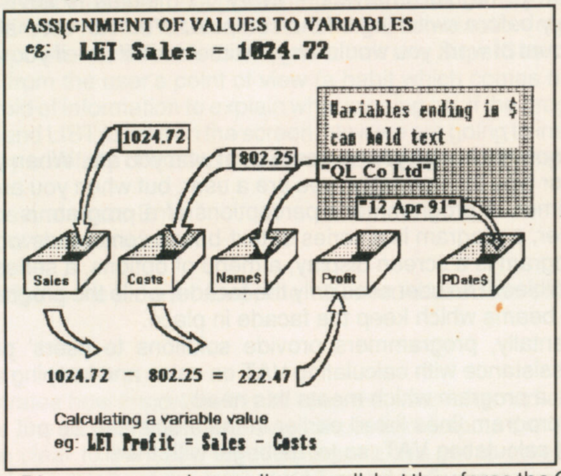
SINCLAIR QL WORLD

THE NEW USER GUIDE

SECTION TWO

The second instalment of our New User Guide for the Sinclair QL. This month, Mike Lloyd parallels the contents of Chapter 2 of the original User Guide, but expands the scope to include what he describes as 'absolutely everything you need to know about computer programming'.

Readers of last month's introductory part of the New User Guide to the QL will now know how direct commands are issued to the QL – and what happens when the command is not recognised for any reason. They will also know about variables – the named memory locations which are used to store single pieces of numeric and textual information in the computer's memory. **Figure one** is a reminder of some of the features of variables, based on the analogy used last month of shoe boxes holding pieces of information.


ASSIGNMENT OF VALUES TO VARIABLES
eg: LET Sales = 1024.72
Variables ending in $ can hold text
1024.72
802.25
"QL Co Ltd"
"12 Apr 91"
Sales
Costs
Name$
Profit
PDate$
1024.72 - 802.25 = 222.47
Calculating a variable value
eg: LET Profit = Sales - Costs

What is a program?

Direct commands are all very well, but they force the QL to act as little more than a clever calculator. The value of a computer arises from its ability to store sequences of commands and then to carry them out automatically. A sequence of commands is called a program. Incredibly, there are only three fundamental things to learn about programs: all the rest is window-dressing.

Direct commands and commands in a program are very closely related. A direct command becomes a part of a program if it is given a *line number.* Line numbers can be any whole, positive number less than 32676. The upper limit is not restricting in any way as even the largest programs rarely exceed 3,000 lines. Commands in programs are often known as *statements.*

Type in the following short program of two lines:

```
100 LET SALES = 240
110 PRINT SALES * 0.15
```

Instead of carrying out the commands the instant that the Enter key is pressed, the QL stores them in its working memory. For the programmer's benefit they are also listed on the screen, using Window #2. The program is executed only when a direct command is given telling the computer to proceed. This command is RUN, so type it in now, remembering not to give this direct command a line number.

Three things may happen. The most likely is that the program instantaneously prints the value of 16% VAT on sales of £240. Alternatively, there may be a typing error in your program which produces an error message. Thirdly, if you have accidentally prefaced the RUN command with a line number it will be tagged onto the end of the program.

Having committed our program to the QL's memory several things can be done with it – and indeed should be done with it – in order to make the most of the computer's programmability. It would be nice, for instance, to be able to edit program lines, not only to correct errors but also to amend the way the program works. The work required to produce a 200-line program makes it uneconomic to type it in each time it is needed. Therefore, programs must be capable of being saved permanently and recalled when needed.

Editing

The command to edit a program line is EDIT. The keyword must be followed by the line number of the line you wish to edit. For example, to edit the second line of the program listed above, type:

```
EDIT 110
```

On pressing the Enter key the line will be copied to the command window where it can be edited. When Enter is pressed again the amended line is replaced in the program listing.

It is of course possible to change the line number while editing the line. You may wish to save typing, for example, by taking an existing line, editing it slightly, and re-inserting it into the program with a new line number. Line numbers in a program must be unique: if you enter a second line numbered 110 it will overwrite the existing program line with that number.

Line numbers

Line numbers are used to sort lines into sequential order, so giving a program line a new number can also have the effect of changing its position in the program. Line numbers do not need to be sequential. Traditionally, programmers begin programs with line numbers incremented by 10. Then, when they are revising the program or adding to it, new program lines using the intervening numbers can be added. The program which will be developed during this article will make use of this approach.

The important thing to grasp is that programs are never written like letters – starting at the beginning and working through to the end. Programs tend to develop like a painting, starting with perhaps the bare outlines and then adding more and more detail until the work is complete.

Save and Load

When a great deal of care and time has been expended on a program, it is important to be able to save and recall it. This is done using the commands SAVE and LOAD. To save a program, type in the direct command SAVE followed by a valid filename. Filenames comprise a drive designation and an identifying name, and the typical 'mdv1_program' or 'flp1_test' constructions will be familiar to all QL users. Having saved a program, it can be reloaded into the Q L's working memory by typing in the direct command LOAD followed by the program's filename.

You are saving a snapshot of your program as it was at the time it was saved. If it is amended after being saved, the amendments will not be reflected in the program file. Always save a program under construction at regular intervals – every 15 minutes is advisable – during development and immediately before switching off the computer. You can estimate the desired frequency of saves by the amount of work you would lose, in time and effort – if you were to crash or switch off without saving.

What is a programmer

It is time now to change your perception of who you are. When you enter a direct command to the computer or type in information you are a user, but when you are constructing a program you are a programmer. A programmer's perceptions of a program is very different from a user's. To a programmer, a program is a series of text-based commands which form a logical structure. To a user, a program is a screen display, a menu of options, a series of requests for information, and a set of facilities. The user sees only the facade, while the programmer sees the backstage props, ropes and beams which keep the facade in place.

Fundamentally, programmers provide solutions to users' problems. Assuming that a user requires assistance with calculating VAT on sales and keeping a tally of how much VAT is owed, let us write a program which meets this need.

The two program lines listed earlier in the article can be put to good use in the solution to the problem of calculating VAT, so let us begin with them:

```
100 LET SALES = 240
110 PRINT SALES * 0.15
```

A fundamental drawback is that the program is only of any use when articles valued at £240 (or perhaps £2.40) are sold. The answer is to replace the LET statement with one which obtains input from the user. Having already discovered that the keywords for editing, saving and loading are EDIT, SAVE and LOAD, it is easy to guess that the keyword to obtain input is INPUT.

INPUT

To form a valid command, the keyword INPUT must be followed by a variable name so that the computer knows exactly where to store whatever is being typed in. The variable can be for numeric information or for text. If the variable is for numbers only, typing text will cause an error message to appear. For the moment, assume that users can be trusted to type what the QL is expecting to read (although programmers will quickly learn that trusting users to do anything predictable is foolhardy). Use the EDIT command to alter Line 100 so that the program reads:

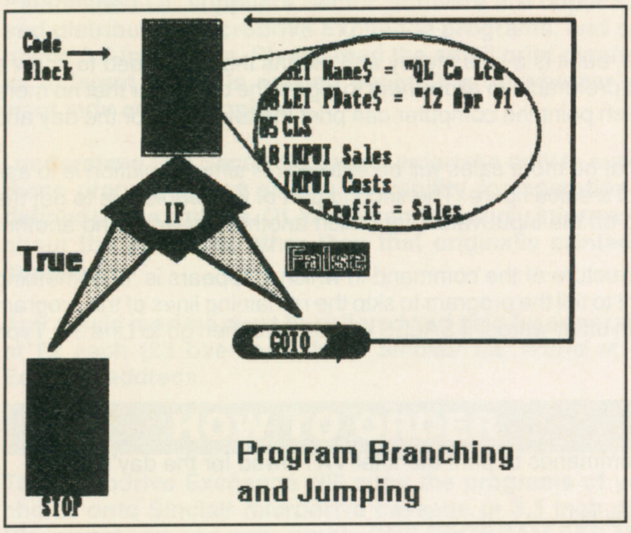
```
100 INPUT SALES
110 PRINT SALES * 0.15
```

Now when the program is run (by typing the direct command RUN) a flashing cursor appears in Window #1. Type an amount and press Enter and the computer should write beneath it the appropriate VAT value. If not, examine your program to work out why it has failed.

The program is of limited use because it only copes with a single calculation. The user will be selling things all day and will not want to keep typing in RUN to make the program work. After each calculation, the user will wish the computer to be ready for the next.

Repeating commands

The simplest way of achieving this aim is to type in more INPUT and PRINT commands, but this is wasteful and impractical. The best answer is to find some way of directing the computer to carry out the original pair of commands repeatedly. This can be done by telling the computer to go to the first line in the program. Once again the Basic language has a very obvious keyword and statement syntax which provides this facility. Add this command to the end of your program:



```
LET Name$ = "QL Co Ltd
LET PDate$ = "12 Apr 91"
05 CLS
10 INPUT Sales
  INPUT Costs
  Profit = Sales
```

Program Branching and Jumping

```
120 GOTO 100
```

GOTO

Run the program again and as soon as the first calculation is complete the cursor in Window #1 is flashing again for another input. As many inputs as you like can be entered. Having made the program loop eternally, making it stop has new become an important issue. Simply hold down the CTRL key and press the SPACE bar – on the QL the CTRL-SPACE combination equates to the BREAK key on many other computers. Its effect is to bring to a halt any SuperBasic program.

Looking again at the program from the user's point of view (a habit which comes hard to many programmers) the screen is devoid of information to explain what is going on. If the program cannot be seen, type in the direct command LIST to list it on the screen. Now we are going to insert program lines in between existing lines by carefully choosing line numbers. Add the following lines:

```
90 PRINT "Type in a sales value and press Enter"
105 PRINT "The VAT on the above amount is. . ."
```

So that the new Line 90 is included in the program loop, line 120 should be edited so that it reads:

```
120 GOTO 90
```

The revised program demonstrates how important it is to keep the user informed about exactly what the screen display means. The program obtains sales values and outputs the VAT, and the screen display makes this quite clear to the user, but the program has yet to meet the original requirements in full. It does not keep a tally of the total of VAT owed to Customs and Excise.

Adding variables

Some logical thought is needed here to work out how the computer is going to cope with this task. At the beginning of the day, the amount owed in VAT is nothing, because there have been no sales. With each sale, the total amount of VAT can be described as the previous VAT total plus the VAT associated with the most recent sale. These two values can be represented by the variables TOTALVAT and VAT respectively. At the beginning of the day, TOTALVAT is zero, so add this to the program:

```
80 LET TOTALVAT = 0
```

With each sale, VAT must now be calculated and stored, so add the line:

```
108 LET VAT = SALES * 0.15
```

It is worthwhile simplifying the next line in the program to read:

```
110 PRINT VAT
```

This makes the program easier to understand and it saves having to do a calculation twice.
The value of VAT must now be added to TOTALVAT, hence the addition of the line:

```
115 LET TOTALVAT = TOTALVAT = VAT
```

Many novice programmers have a little difficulty understanding how this sort of command can make sense. What it means is: fetch the values held in the variable locations TOTALVAT and VAT, add them together, and store the result in the variable location called TOTALVAT. As only one value can be stored in a variable, the new TOTALVAT value overwrites the old one.

Run the program again to ensure that it is working. When you have entered a few sales values, press the CTRL-SPACE combination to complete the program and find out what TOTALVAT is worth by typing in the direct command:

```
PRINT TOTALVAT
```

Users could do this for themselves, but it is a little messy and means that they need to know a little about programming. It would be preferable to allow them to signal the computer that no more sales are going to be entered, at which point the computer can print the total VAT for the day and stop processing.

Firstly, how can the users signal that no more sales will be entered? A simple solution is to ask them to enter a zero value instead of a sales figure. The second part of the problem is to get the computer to make a decision based on the input value, for which another keyword and another command are required.

IF and THEN

The keyword is IF and the basic structure of the command in which it appears is 'IF something is true THEN do something'. We want to tell the program to skip the remaining lines of the program if a sales value of 0 is encountered; in other words, if SALES equals zero then go to Line 30. Type in:

```
102 IF SALES = 0 THEN GOTO 130
```

At Line 130 we can now include commands to print the total VAT owed for the day's sales:

```
130 PRINT "The total VAT for the day is:"
140 PRINT TOTALVAT
```

Run the program, enter a few sales values and then enter a sales value of zero. If the program does not behave exactly as expected, review the lines you have entered and the lines given in the text above to see where it is going wrong. To help you, **Figure two** is a listing of the complete program using the rather odd line numbers which have been used as the program has developed. For the fastidious, properly spaced program line numbers can be restored by typing in the direct command RENUM, short for 'renumber'. The computer is clever enough to update the GOTO statements as the line numbers change. Save the program with a direct command such as:

```
SAVE mdv1_vatcalc
```

```
LISTING 1

 80 LET TOTALVAT = 0
 90 PRINT "Type in a sales value and press Enter" <
100 INPUT SALES
102 IF SALES = 0 THEN GOTO 130        ==>
105 PRINT "The VAT on the above amount is..."
108 LET VAT = SALES * 0.15
110 PRINT VAT
115 LET TOTALVAT = TOTALVAT + VAT
120 GOTO 90                           ==>
130 PRINT "The total VAT for today is:" <
140 PRINT TOTALVAT

The arrowed lines indicate the way the computer is
forced to move through the program lines - they are
not part of the program itself.
```

Branching

When computers are told to go to a specific line number rather than to continue with the next line in the program, the process is called branching. Branching can be forwards or backwards, as the two GOTO commands used in the program demonstrate. When branching occurs only if a condition is found to be true, as in the IF command, then it is called conditional branching. When the branching occurs no matter what the circumstances, as is the case in Line 120, then it is described as unconditional branching.

There is nothing more to program logic than this. Computers perform a series of statements in a pre-determined sequence until they are required to branch, either conditionally or unconditionally. Of course there are many more commands to learn in order to do different things, and there are other branching commands than IF and GOTO, but these extra commands simply add to the richness of the SuperBasic language: they do not provide any alternative method of executing programs.

It is an absolute truth that all SuperBasic programs, and indeed all conventional computer programs, can be re-written so that they consist only of straightforward commands, such as PRINT and LET, and the two branching statements IF and GOTO. Of course, programs are faster, more efficient and more elegant if use is made of the other commands and structures available in the language, so the New User's Guide to SuperBasic therefore continues next month.

MICRODRIVE EXCHANGE

NOW WITH DISKS

B = SuperBasic; A+O = assembler and object code; M+B = machine code and Basic loader; A+B+O = assembler and Basic loader and object code; S = supercharged; L = QLiberated; f1 = monitor mode; f2 = TV mode.

Established QL suppliers Sector Software are duplicating and distributing Microdrive Exchange programs, and supplying 3.5 inch disks. Please read the small print carefully, as different action is needed depending on whether you want mdv or disk copies.

I understand that Sector Software undertake only to supply these programs, and accept no liability for operation as defined by the authors. QL World cannot supply information about the programs other than that originally printed or supplied.

Any article reprints must be ordered and paid for separately at £2 each (£3 overseas) from *Sinclair QL World* at the Editorial address.

HOW TO ORDER

The Microdrive Exchange will copy the programs of your choice onto Sinclair microdrive cassette or 3.5 inch disk. Please note we can only supply ONE PROGRAM PER MDV OR DISK.

Orders for programs on DISK can only be supplied on the MDX's own disks, priced at £1, one disk per program ordered. Please do NOT send disks. Please state any specific density requirement for your set-up. Microcassettes cost £2 each.

Please send your orders to Sector Software, 39, Wray Crescent, Ulnes Watton, Leyland, Lancs. PR5 3NA. Tel. 0772 454328. Access/Visa accepted.

ORDER FORM

Value of programs ordered	**£................**
I require one microcassette/one disk for each program ordered (delete as applicable)	
No. of 3.5in. disks required @ £1 each	**£................**
No. of mdvs required @ £2.60 each	**£................**
Post and packing	**£1.00**
15% VAT	**£................**
TOTAL	**£................**

PROGRAM IDENTITY NUMBERS:
/ / / / / / / / / / /

NAME ....

ADDRESS (BLOCK CAPITALS PLEASE) ....

....

....

.... **Postcode** ....

ACCESS/VISA NUMBER IF APPLICABLE:

□□□□ □□□□ □□□□ □□□□

Please copy on to mdv or disk the programs which I have indicated. I enclose a cheque/PO/Access or Visa number to the total of £.. payable to Sector Software.

THE PROGRAMS

1. LEAGUE SECRETARY by C.B. Storey (B) £3
You enter the match results and this program updates the league tables. Suitable for any sporting league organised on the lines of the Barclays Football League. Mdv only at present.

2. THE DOUBLE by P.G. Ives (Bf2) £4
A large football strategy game. You manage a team through four divisions, buying and selling, boosting morale through the league and F.A. Cup season.

3. SUPERBREAKOUT by R. Davison (M+B) £2
Fast m/c version of the classic bat, ball and wall game. Optional double bats and/or balls.

4. SPACE PODS by Simon Quinn (M+B) £3
Your lone ship must protect six energy pods against the aliens. Machine code. *QL World,* December 1987.

5. GRAPHIC WRITER by S.M. Walker (B) £2
A graphic design program which can save your pictures as SuperBasic commands for use in other programs. *QL World,* December 1987.

6. ZAPMAN by L. Miles (M+B) £3
Fast-action m/c version of the Pacman genre. Variable skill levels and maze formats.

7. SPACE INVADERS by Paul McKinnon (M) £3
Very fast, challenging version of the classic, with ugly aliens and protective shields.

8. SPELLED by Timo Salmi (T) £3
A complete spelling checker for Quill_list files, 7,500 words automatically expandable. Requires two cartridges and 512K expansion.

9. ADVENTURE PLAYTIME by A. Pemberton (B) MDV only £3
An extensive adventure where you must complete tasks for the inhabitants of a strange land. Coded messages and hints included.

10. YAHTZEE/GIRO by Jason Price/Henry Wrighton (B) £3
The popular dice game with instructions and graphics, for one or more players. *QL World* November 1987. Also space-disaster rescue game, in zero-gravity. *QL World* December 1989.

11. LOCK AND KEY by Henry Wrightson (B) £3
Unlock the secrets of eight screens of platforms, ladders, keys, poison weeds and mixed fruit — against the clock. *QL World* February, 1990.

12. RADAR by Nigel Ford (B) £3
You are control, monitoring the skies, checking aircraft, scrambling jets to intercept UFO's and shooting down enemy aircraft.

13. TAKTIX by Nigel Ford (B) £3
Six or more can play the computer in a fierce game of European conquest. Put aside at least an hour. *QL World* July, 1988.

14. BRIDGE by Peter Etheridge (B) £4
Excellent version, including accurate bidding, automatic or manual card play, replay hands, save and loads more.

15. CONQUEST by Andrew Pritchard (T) £4
"Superb graphics, with lots of original ideas. The best strategy game I've reviewed for *QL World.*"

16. MOLECULAR GRAPHICS by Mark Knight (M,B) £4
Molecular structures of any compound can be saved, reloaded, drawn and rotated on screen. "Excellent, one of the best educational programs on the QL." Disk or 2 cartridges and 512K needed.

17. SPEEDMIND by William Henderson (B) £3
A mastermind-style game played with coloured pages. You have 12 attempts at breaking the code against the clock. *QL World,* January 1988.

18. STELLARIS by David Carmona (Bf1) £4
Real-time space adventure against the computer, including economic simulations, lunar landing and superb graphics. *QL World,* June 1987.

19. BUSINESS GAME by David Smith (B) £4
A business simulator for any number of players, human or computer. The winner is the one who makes the most money! Networking advice from author. See *QL World,* April 1989.

DIY TOOLKIT

Simon Goodwin describes a new SuperBasic command that lets you roll your own resident functions, without limit. SET functions are useful, fast and accessible to compiled or interpreted Basic. The code appears next month. The code appears next month.

The new QL command SET creates resident functions – SuperBasic names with values that survive CLEAR, LOAD or NEW. You choose the names: they can be as long, short, arcane or expressive as you like, in any language that can be typed as Ascii code.

The programs here are original, but SET was first dreamt up by reader Luca Pivato, and his colleagues Fulvio and Valerio, at Thor developers Human Interfaces in Rome.

SET allows up to 18,000 new system pointers or data values of any type, up to 32K each, accessible to SuperBasic and compiled tasks. Strings do not have to be dimensioned. The theoretical limit of about 18,000 names is determined by the size of the Name List, which must be addressed with 16 bit offsets from the Name Table.

SET helps to make programs more meaningful and lets you store persistently useful values, like the address of RESPR'd space. I have extended Luca's concept to provide direct access to major 'pointers' in the SuperBasic system – default addresses, system tables, and the like. These SET functions update automatically as Qdos shuffles tasks and memory.

The name you want to SET must initially be unset; it must not already identify a SuperBasic value or routine in the program. The QL reports BAD PARAMETER if you try to SET a name that already has a meaning. It says ERROR IN EXPRESSION if the value you supply is unset or does not suit the type of the name.

As usual for SuperBasic, the first time you enter the name you fix the way it will be displayed thereafter. It is consistent to use CAPITALS to match other resident function names.

Names make program editing easier, and therefore more reliable and powerful. If you use a name you can search and replace it, without getting confused by other instances of the same value that mean something quite different.

The QL imposes various odd schemes on us – integer colours 0-7, system variable offsets and the like. I wish it could remember that YELLOW% is 6, rather than force me to learn its codes. Names take up more physical space on disk and on the screen, but they are concisely stored inside SuperBasic and you can use thousands at once with no loss of speed.

```
SET ENTER$ TO CHR$(10)
SET MARGIN% TO 8
SET TRUE TO 1
SET FALSE TO NOT TRUE
SET GREEN% TO 4
SET BACKGROUND% TO GREEN%
SET SD_XSIZE% TO HEX("1C")
SET MAXINT% TO 32767
SET MININT% TO -32768
SET ASPECT_RATIO TO 16/9
SET OFF TO 0
SET PIby2 TO PI/2
SET Ln2 TO LN(2)
SET e TO EXP(1)
SET HELP_DEVICE$ TO "FLP3_"
SET MY_SERVER$ TO "N2_"
SET DIGIT$ TO "0123456789"
SET WHITE% TO 7
SET BLANCHE% TO WHITE%
SET PIPE_ID TO CHANNEL_ID(#3)
SET SYS_VARS TO 163840
SET SCREEN_STATUS TO SYS_VARS+51
SET START_TIME TO DATE
SET SCREEN_RAM TO 131072
SET WIDTH% TO 124
SET BV_LINUM TO HEX("68")
SET PRINTER$ TO "SER"
SET SPACE$ TO CHR$(32)
SET SPACES$ TO FILL$(SPACE$,80)
SET STRING_TYPE% TO 1
SET INTEGER_TYPE% TO 3
SET NULL$ TO ""
SET # HEAP_START TO 4
SET # RAM_TOP TO 32
SET # KEY_QUEUE TO 76
SET # TASK_INDEX TO 100
SET # DEVICE_LIST TO 68
SET # DRIVE1_LINKAGE TO 256
SET # ICE_BUFFER TO 228
SET # DATA_DEFAULT TO 176
SET # SER2_BUFFER TO HEX("9C")
SET # QJUMP_THINGS TO HEX("B8")
SET # THORXVI_SHARED_RAM TO 192
SET # TRANSLATE_TABLE TO 326
SET # MESSAGE_TABLE TO 330

TABLE - Sample SET values

DIY Toolkit, QL World, April 1991.
```

2

Meaningful names make programs easier to read and check. FOR K=2 to GS could mean anything, while FOR DAY=1 TO CYAN% is clearly a mistaken substitution. You probably mean FOR DAY=MONDAY TO FRIDAY. If you use CAPITALS for new names, and type programs in lower case, the helpful QL will show up mis-typed names as you enter lines.

If you can't type as fast as you make up the program, you may be thinking at the wrong time, or need *DIY Toolkit* files on disk! If you want complete words or phrases to appear at one lunge, try Toolkit 2 ALTKEYs or Quanta's FKEY20. Humans read whole words, not individual characters. Apt words are easier to recognise and distinguish than arbitrary numbers.

Even Sinclair thought the computer should remember PI; likewise VER$ is a resident constant, although its value varies between machines. Useful addresses of fonts or data may be 'constant' for each session, but dynamically re-calculated and SET by your BOOT program.

The IBM PC reserves a small amount of memory for 'environment variables' – text values, set by MSDOS commands, and read by successive programs or 'batch' routines. These often hold configuration details. Programs that check them can be told to use different devices with no need to patch the program code.

Commercial QL programs often POKE the name of their default device into 'spare memory', where it may interact with other software, or system changes. It is much better to SET PATH$ TO "FLP1_" in the BOOT program, and read PATH$ thereafter.

Current QL disks use sectors numbered from 1 to 9, but it seems likely that more sectors will be allowed by higher capacity disks on Qdos or super-clones. If you want your program to survive the upgrade, it's better to use a name, like MAX_SECTOR%, rather than try to recall the significance of every '9' in the program the day you need to recover a big new disk and your program expects the old sort!

Even if your programs are more prosaic, you may become glad that they all OPEN PRINTER$, a SET constant, rather than literal names like "SER1EHC" or "PARCF_24K", when you boot up on another system – or upgrade your printer, interface, translation or buffering software.

SET can be used to share character fonts:

```
x=RESPR (2+9*256)
LBYTES "FLP1_FONT",x
SET_FONT x
SET MY_FONT TO x
```

SET_FONT is a *Turbo Toolkit* function, equivalent to *Speedscreen's* _FOUNT, OR CHAR_USE IN *Toolkit 2.* The example creates a resident constant called MY_FONT, which holds the address of the font memory. Subsequent programs can share the same font by including SET_FONT MY_FONT, or something similar, when they prepare a display window.

To change the shared font, at any time later, just re-load the shared memory, with LBYTES FLP1_NEW_FONT,MY_FONT. This changes the font in all current tasks that used MY_FONT to prepare their windows.

The provisional *User Guide* given out by Sinclair Research at the 1984 QL launch suggested that 'if an identifier starts with an ampersand '&' then it is assumed to be a system identifier', but this feature never reached production.

I have added a similar feature to SuperBasic. In this case the name is marked with a # when you SET it; ampersands don't pass the syntax check. SET # creates an interesting class of Resident function.

SET # takes a name and associates it with a particular pointer in the System Variables. To read the pointer later, just refer to the name. Rather than SET SV_BASIC TO 16 and PRINT PEEK_L(SYS_VARS+ SV_BASIC), just SET #BASIC TO 16 and PRINT BASIC.

The function finds the value of SV.BASIC, the system's pointer to the SuperBASIC task. Similarly, SET # TASKS TO 16 makes TASKS point at the start of the area used by other tasks. Thereafter, PRINT TASKS-BASIC gives the size of the memory area allocated to Basic.

There are dozens of suitable pointers in the Qdos and Argos system variables. **Table one** lists some possibilities.

Whether interpreted or compiled, SuperBasic is fastest and most efficient if you give it the 'expected' type of data, so it does not have to coerce what you supply into the type expected.

The type-checking in this version of SET is not strictly necessary, but it makes programs that use SET values much more efficient than might otherwise be the case. Literal constant values, of any type, are handled quickly by SuperBasic, and I do not want people to abandon the use of mnemonic functions for reasons of speed. Even fast programs should be readable and configurable.

Access to SET values involves an extra name in the program, but this is handled quickly because SuperBasic stores the Name Table index of the function, with no need to search to find its code address. The code inside the functions has been pared to the minimum; a 'short cut' in the code removes the need for redundant checks if the command is called from a compiled task. The rom spends more time checking for space than it does finding the value, whether SET or LET.

SET could be implemented to store all values as strings – perhaps ignoring the last character of the name – and programs would still work, but many slow and unnecessary coercions would be required. This would be appropriate for slow interpreted systems like MSDOS environment variables, but it would slow SuperBasic needlessly. Saving time and space at the expense of a little code, SET associates string, integer and floating-point values with corresponding 'types' of unset name, distinguished by their last character.

TRUE and FALSE are best defined as floating-point constants, because IF expects a floating-point value. If 1IE600 THEN CLS clears the screen – it does not report an over-flow error. CH_XSIZE% and BV_LINUM% are Window and Basic variable offsets, for use with older DIY functions like CHBASE, CHAN_W% and BPEEK_W%.

More accurate

PIby2, Ln2 and 'e' do not contribute much to readability, but are evaluated faster than their computed equivalents; this helps most if they appear in loops, when the computer might otherwise have to work out the expression time after time. Computed values are often more accurate than literal ones, because SuperBasic works to an internal precision of nine digits, but does not display the last two. To see this, enter

```
PRINT P1, P1-3, P1-3.1
```

The example values of SCREEN_RAM and SYS_VARS suit Sinclair QL Roms, but may vary on other systems. KEY_QUEUE returns the address of the current key queue, or zero, and changes when you switch the active cursor. PEEK or POKE SCREEN_STATUS to check for CTRL-F5, pause or re-start screen output.

PEEK_L(TASK_INDEX) reads the address of the current tasks from the system's task table. Similar SET # examples find other tables. If you have trouble remembering the numbers used to indicate parts of the screen to PAN, SCROLL and CLS, why not SET them?

Tyros may be slow to see the point of resident constants. They have learnt the literal value of all the flags, trap keys, magic numbers, Ascii and Qdos codes in their systems. But after a while you 'know' several sets, and sometimes get them confused. Either you invest effort in learning the literal values, and regularly update your knowledge, or you start giving names to the things: names which suit your perspective, and the way you want to think about systems in future.

Grizzled hackers may be glad to standardise on ENTER$, rather than CHR$(10) or CHR$(13), or both, in either order. SET your own standards, and stick to them.

Once upto a time we had to assign constants at the start of every program that used them – though we could merge them from a file. I ended up with lists like GREEN%=4 : WHITE%=7 at the top of many listings, or raw values later.

SET lets you teach new names to the QL or Thor, without adding to your programs. The names reside in the machine, like other Toolkit functions, so they survive CLEAR, NEW, LOAD and LRUN; tasks can find them, too.

Once a name has been SET you cannot use it as a program variable, DEF or loop name. This is a general rule which stems from the design of the Name List and the scanner that encodes each line. Each SuperBasic name has a distinct type; you cannot have an array, variable and procedure all with the same name, as you can in SAM Basic.

You must avoid SETting names that your system already uses. SET OFF TO 0 allows FLASH OFF : UNDER OFF and suchlike, but SET ON TO 1 gives BAD LINE, because ON is a preset QL keyword and already has defined meanings.

Sadly, accented letters cannot be included in SET names, as Jan Jones did not allow them to appear in SuperBasic identifiers. I have often thought that this is a pity. I'd love to use graphic characters in names, writing DEF PROC (right-facing PacMan) or DEF FN (Clockface).

SET means that individual programs might become incomplete without their constants, but the same is true of all Resident Toolkit functions. This can easily be fixed by adding SET statements or substituting LETs in a lone program. Dilwyn Jones' *Basic Reporter* utility can help a lot, quickly naming all the unset identifiers in any program. My keyword code may be distributed freely in non-Toolkit-related packages that need it.

The 'user heap' routines buried inside SET show how Qdos can organise an arbitrary collection of memory areas on extra lumps of memory at any time.

Thie assembler code appearing next month expands and creates records inside the user heap. The user heap is set up inside one or more linked blocks of memory taken from the Common heap, or pool of free memory near the bottom of the QL memory map.

It might have been simpler to allocate each SET value its own space in the Common heap, but that would probably not work well. Among other 'objects' the heap already contains drive and channel definitions, FILL buffers (a temporary property of channels), ALCHP space and ram disks.

As SET functions proliferate I do not want the rest of the system to slow down, or memory to become split into small chunks. Such problems would be much more likely if each SET command created or released a distinct entry on the Common

Heap.

User Heaps are a good way to keep lots of bits together, especially when you consider the alternatives: continuous moves to take up slack space, changing pointers and copying as extra space is allocated, or stacks of pointers to scattered gaps.

A user heap is a linked list of variable-length records. It keeps all but one pointer inside the heap itself, so it needs no external tables. It uses 32 bit pointers and 32 bit sizes, allowing lumps several thousand times bigger than the maximum QL ram. This is perhaps over-generous but preferable to 16 bit limits, which restrict lumps to 32K.

There's no point releasing a space that is not big enough to hold its own details, two long words, so the allocation unit is eight bytes. Ask for anything from one to eight, you get the full eight. Ask for nine and you get 16.

This rounding is slightly wasteful, but has the advantage over a heap of bytes that you can be sure of a word boundary at the start of each lump. 68000 code and data words are expected to be stored that way.

The Common Heap records the task that owns each lump, making 16 bytes of header. Thus the minimum Common heap allocation is 24 bytes.

Overhead value

All you need to start a new User heap is a long word address pointing to the start – a sentinel. Initially make it 0, indicating the end of the list. SET END_OF_LIST TO 0, if you like; some programmers like to exploit the deliberate link between logical FALSE and the zero terminator.

When the user heap is first needed, allocate some common heap space with MT.ALCHP, and release it into the user heap with MM.LNKFR. Pass the address of your sentinel and LNKFR will set up the user heap header, pointing the sentinel at its second long word.

User Heaps use relative pointers to link areas of free space. The entire space is available to the routine calling MM.ALLOC, but you need to remember the size if you want to release a space to MM.LNKFR later. Qdos helps by storing the size of the space in the first long word; my routines leave it there. This may seem a waste of four bytes per used entry, but it makes deallocation and debugging much easier.

The second long word is the 'relative pointer'. This holds the offset to an address in bytes, which may be positive or negative, rather than the address itself. This is convenient if all or part of the list is liable to move, as often happens inside SuperBasic. To find the target address of a relative pointer, add the address of the pointer to the long word value at that address. Thus the length of the first free entry in the heap is at:

PEEK_L(SENTINEL)+SENTINEL-4

Assuming you SET SENTINEL TO x+12 after loading the SET code to address X. The subsequent long word contains the offset of the next pointer – with the space free in the previous long word – or 0 at the end.

To grab some space, call MM.ALLOC, which rounds up the size in D1 and tries to find the space, searching all the free bits and returning ERR.OK or ERR.OM; do not rely on the Z flag corresponding to DO on return.

User Heaps are an important concept in Qdos, but they are not well documented. Pennell's *Qdos Companion* hardly mentions them at all. As far as I know this is the first worked example of their use. My main references were section 8.4.2 of Adrian Dickens' *QL Advanced User Guide*, largely written by Tony Tebby, and *Fundamental Algorithms* by Donald Knuth, section 2.5.

All SuperBasic variable values are held in a User Heap, so scalar integers and decimals both take up 8 bytes of memory, the minimum. *Qliberator* employs User Heaps extensively, but *Turbo* and *Supercharge* tasks do not use them at all, because they are compiled to work reliably within the bounds of the task dataspace; buffers are 'phased' in and out as programs run, keeping two dynamic areas at either side of a single pool of free space.

User Heaps have one failing, which is more of a problem for Basic than it is for SET. They tend to grow as space is allocated and released. This matters little when lumps are small – Knuth's *Art of Computer Programming* suggests that physical memory fragmentation is usually manageable until the size of individual allocations approaches ten per cent of the available space. This rule of thumb explains why SuperBasic grows alarmingly when you manipulate big slices or arrays. Any busy heap needs a margin of slack space, or it clogs.

Qdos tasks need large continuous blocks of memory in which to run. As the common heap grows from low QL addresses after the system variables, towards the tasks and RESPR code at the top, it limits the space available for task code and data. Volume H of DIY Toolkit explores the Common Heap.

Volume B includes a short program called FORGET which gets rid of resident procedures or functions. This is easy on current Thor systems, due to a bug. Just DEF a dummy function in Thor Basic, with the same name as the unwanted extension; both will vanish as soon as you type NEW or load a different program. QL users can get the same effect by changing the type Word in the Name Table from 2304 (Resident Function) to 1, 2 or 3, depending on the data-type.

Logical link

Such tricks excise the name from the Name Table, but they do not de-allocate the memory used to store the code. This overhead is tiny in the case of SET routines – a couple of values plus a JSR instruction, for each constant.

The new DIY Volume U includes a User Heap Explorer that can locate all the parts of a user heap, showing size, address and the smaller allocations inside. It also identifies intermediate heap blocks that split the user heap.

It is important to FORGET all the resident constant names before you deallocate SET space, or SuperBasic may try to call code in memory that has been re-located to some quite different purpose. POKE_L SENTINEL,0 resets the User Heap.

You can set the minimum size of each Common Heap lump with POKE_W SENTINEL-2,SIZE%. Values from 16 to 32760 bytes are recommended – the default is 100B bytes (1K when the common heap header is added), which gives room for 60

```
100 REMark Sinclair QL World HEX LOADER v 3
110 REMark by Marcus Jeffery & Simon N Goodwin
120 :
150 CLS: RESTORE : READ space: start=ALCHP(space)
160 PRINT "Loading Hex..." : HEX_LOAD start
170 INPUT "Save to file...";f$
180 SBYTES f$,start,byte : STOP
190 :
200 DEFine FuNction DECIMAL(x)
210 RETurn CODE(h$(x))-48-7*(h$(x)>"9")
220 END DEFine DECIMAL
230 :
240 DEFine PROCedure HEX_LOAD(start)
290 byte = 0 : checksum = 0
300 REPeat load_hex_digits
310    READ h$
320    IF h$="*" : EXIT load_hex_digits
330    IF LEN(h$) MOD 2
340       PRINT"Odd number of hex digits in: ";h$
350       STOP
360    END IF
370    FOR b = 1 TO LEN(h$) STEP 2
380       hb = DECIMAL(b) : lb = DECIMAL(b+1)
390       IF hb<0 OR hb>15 OR lb<0 OR lb>15
400          PRINT"Illegal hex digit in: ";h$ : STOP
420       END IF
430       POKE start+byte,16*hb+lb
440       checksum = checksum + 16*hb + lb
450       byte = byte + 1
460    END FOR b
470 END REPeat load_hex_digits
480 READ check
490 IF check <> checksum
500    PRINT"Checksum incorrect. Recheck data.":STOP
520 END IF
530 PRINT"Checksum correct, data entered at: ";start
560 END DEFine HEX_LOAD
570 :
580 REMark Space requirements for the machine code
590 DATA 450
600 :
```

```
610 REMark Machine code data
620 DATA "43FA01B234780110","4ED203F000000000"
630 DATA "7202616A205F3390","E80078034E757206"
640 DATA "615C205F36107000","4E41243030002202"
650 DATA "670C323C08205341","D4826AFAE28A3381"
660 DATA "E8002382E8027802","4E7572066130205F"
670 DATA "3398E8002390E802","78024E75205F2050"
680 DATA "7203D25008810000","6114E24C53442449"
690 DATA "3598E800548A51CC","FFF878014E752801"
700 DATA "4A2E00546B063278","011A4E91226E0058"
710 DATA "93C42D4900587000","4E7549EB0010BBCC"
720 DATA "6620740072007002","4E4128481A33E801"
730 DATA "6710080500076702","7A001233E8006700"
740 DATA "000670F14E753E33","E8026FF648C7E78F"
750 DATA "DEAE0018508B0205","0003180566027803"
760 DATA "4884D84491C890C4","41E8011830504E90"
770 DATA "66D24E407210BA3C","0001670ABA3C0002"
780 DATA "670A720C60065C81","D271E800280141FA"
790 DATA "FEFC347800D84E92","4A80674C307AFEEC"
800 DATA "2204B28862027200","D288740070184E41"
810 DATA "4A80667A74109282","43FAFED2347800DA"
820 DATA "4E9222046OC870F1","60640831000OE801"
830 DATA "66F445FAFECA6028","45FAFEB6BA3C0003"
840 DATA "671A45FAFEE66014","226E0058BA3C0001"
850 DATA "65D862E445FAFEE6","04440009E24C5D44"
860 DATA "22182988780430FC","4EB920CA39BC0900"
870 DATA "78005305660E45E8","000A20CA54887410"
880 DATA "928220C130F1E800","548951CCFFF8027C"
890 DATA "D8FF4E750001FEE4","0353455400000000"
900 DATA "0000",'*',41255
```

new resident numeric constants or up to 40 strings. If a string is too big to fit the default lump, SET calls for 100B bytes plus the space required.

It's only worth increasing the value, by patching the world at offset 10 in the code, if you intend to SET scores of names or long strings. It may be better to allocate a few K at the start, rather than risk fragmentation a long time later when the original buffer overflows.

SET creates resident entries in the SuperBasic Name Table, the source of name data for all subsequent Basic tasks. SET constants offer an efficient way of communicating a mix of data items from Basic to other tasks, but SET would be even more useful if communication could go both ways.

In May's *QL World,* the assembly code for SET will appear, and I shall develop this routine, making it possible for tasks to set changed values to existing SET functions. Thus any number of SuperBasic tasks can share values of any type, efficiently and dynamically. The extra code involves more User Heaping, and further useful applications.

The new **DIY Toolkit Volume U** contains the assembler and binary code for SET, plus scores of useful constants and vectors, the *User Heap Explorer* and associated routines.

To obtain DIY files, or a catalogue, ring Richard Alexander on (0559) 384574, or write to **DIY Toolkit, Cwm Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA**

DIY Toolkit illustrates Qdos programming from concepts to testing , with short, interesting examples that fill the gaps in existing QL Toolkits. Suggestions for future keywords or topics are welcome at the QL World editorial address.

HARDWARE UPDATE: QL - Keyboard - 90

Mike Lloyd follows up his Februatry 1991 report.

Jeurgen Falkenberg's new keyboard interface was reviewed in pre-production form in the February edition of *QL World*. Since the review went to the publishers, further information about this impressive product has been released.

QL-Keyboard-90 is a card of microchips which plugs into your QL either directly onto the motherboard or through the rom port. Attached to the card is a short lead ending in a DIN socket which is compatible with the DIN plugs placed on the end of PC keyboard leads. By purchasing Falkenberg's keyboard utility hardware, you are then free to purchase your own choice of pc keyboard (available from around £40), or perhaps to borrow one from a pc in your possession. PC keyboards are generally of much better quality than those available for the QL and suit both the native QL software environment and the MS-DOS emulators available from Digital Precision.

Finer

February's review concentrated on the fitting of the card inside the QL and the practicalities of the straightforward use of an add-on keyboard. Because full documentation was not available at the time I was unaware of some of the finer points of Falkenberg's programming for the keyboard interface.

The essence of matching a new keyboard to an old computer is called 'key-mapping' – the need to make best use of the new keyboard layout to provide the facilities associated with the old keyboard. The traditional QL has a 65-key layout with five function keys and no separate numeric keypad. Dedicated delete and 'break' keys are also missing, as is a hardware-driven 'escape' key.

PC keyboards come in two main flavours: the 84-key AT layout which has its function keys banked vertically to the left of the qwerty keys and the new and infinitely more popular 102-key PS/2 layout which places twelve function keys in a row along the top of the keyboard. Far from the additional keys making life easier for the programmer mapping them to the QL's smaller keyboard, the extra keys add problems of their own. For example, what is to be done with the PC's 'Print Screen' key, which performs a character dump of the screen to the printer? The difficulties are exacerbated not only by having to consider the 25 cursor-related functions listed in the character set topic of the *User Guide's* Concepts section, but by having to take heed of the sometimes different cursor facilities commonly used in various QL software products.

Falkenberg has tried to provide a logical function for each of the extra keys on the PC keyboard, although 'Print Screen' sadly does not produce a pixel-perfect screen dump to the printer. Instead it performs the arguably more useful job of switching between active Qdos tasks in exactly the manner of pressing CTRL-C. The CTRL-F5 key combination which pauses a scrolling screen is supplemented on the pc keyboard by the Scroll Lock key. However, the closely-related Pause key performs not a pause but a break – it is exactly the equivalent of pressing CTRL-SPACE on the normal QL keyboard.

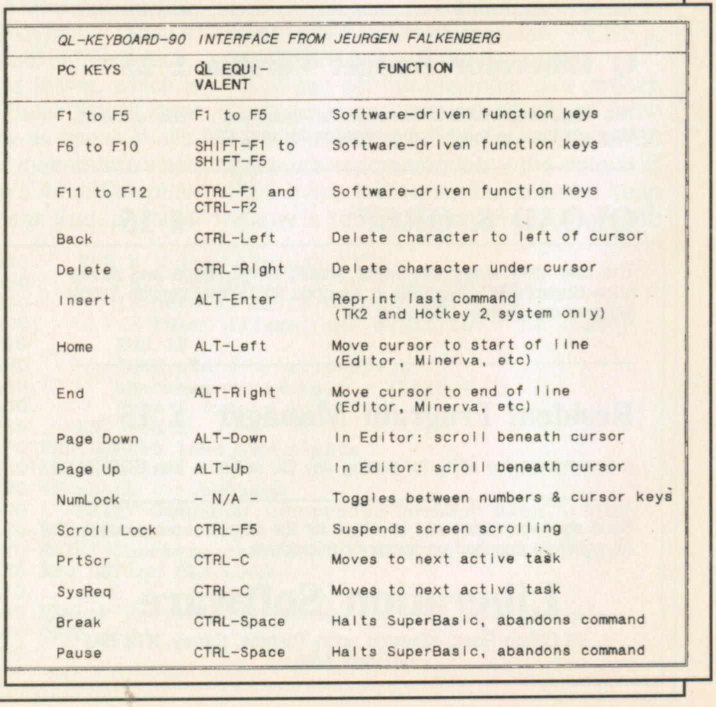
Knowledge

Another bright spot is the implementation of the famous Ctrl-Alt-Del key combination. On the QL this combination does exactly what it does on a pc: it resets the computer. Falkenberg's knowledge of the full range of QL software is revealed by the use to which he puts the PC's Insert key: it forces the re-display of the last SuperBasic command entered, provided that the QL has *Super Toolkit 2* or Tony Tebby's Hotkey 2 environment loaded.

In line with most DOS packages, the pc keyboard's Home and End keys move the cursor to the start and end of the current line respectively. This exactly emulates the Alt-Left and Alt-Right functions planned for QL but which were not implemented in Qdos or in Quill. Minerva owners and users of *The Editor* are able to use Alt-Left and Alt-Right in this way, and so for them the Home and End functions are of value.

The other keyboard functions are listed in **Figure one.** The documentation for the interface reveals more evidence of attention to detail. For example, the auto-repeat timing can still be set using POKEs to the appropriate Qdos system addresses. The scrap of ram present on the Intel cpu chip has partly been used to add a lengthy 55-character type-ahead buffer to supplement the seven-key buffer on the QL itself.

The UK distributor for the Falkenberg keyboard interface reports promising sales since its release. For those who missed the February review, information can be obtained from **TK Computerware, North Stanford, Ashford, Kent on 0303-81-2801.**



QL-KEYBOARD-90 INTERFACE FROM JEURGEN FALKENBERG

PC KEYS	QL EQUI-VALENT	FUNCTION
F1 to F5	F1 to F5	Software-driven function keys
F6 to F10	SHIFT-F1 to SHIFT-F5	Software-driven function keys
F11 to F12	CTRL-F1 and CTRL-F2	Software-driven function keys
Back	CTRL-Left	Delete character to left of cursor
Delete	CTRL-Right	Delete character under cursor
Insert	ALT-Enter	Reprint last command (TK2 and Hotkey 2, system only)
Home	ALT-Left	Move cursor to start of line (Editor, Minerva, etc)
End	ALT-Right	Move cursor to end of line (Editor, Minerva, etc)
Page Down	ALT-Down	In Editor: scroll beneath cursor
Page Up	ALT-Up	In Editor: scroll beneath cursor
NumLock	- N/A -	Toggles between numbers & cursor keys
Scroll Lock	CTRL-F5	Suspends screen scrolling
PrtScr	CTRL-C	Moves to next active task
SysReq	CTRL-C	Moves to next active task
Break	CTRL-Space	Halts SuperBasic, abandons command
Pause	CTRL-Space	Halts SuperBasic, abandons command

QL SCENE

Twin peeks for CGH

Two new publications have arrived from CGH Services.

QL Technical Review issue 5 is now out. Features include retrospective reviews of *Turbo* and *Turbo Toolkit* by Rich Mellor (which Simon Goodwin has already reported to us to be mistaken in some respects, but not unhelpful), *Flashback*, by Richard Alexander, and *The Painter* and *Clip Art* from Progs Van Auwera, by Keith Reader.

Rich also contributes a set of potted assessments of Dilwyn Jones' list, and Stephen Bedford writes part 2 of a guide to *Super Toolkit 2*.

The 'Newsette' page covers stop-press items including a summary of contents for QL World December 1990, and there is an errata sheet for issue 5 and updates for the previous issue.

We quote in full one item of general interest:

"Dennis Briggs tells me that, contrary to certain advice printed earlier in QLTR, it is not possible to erase mdvs with a magnet or with an ordinary bulk tape eraser. To erase a tape it must be subjected to a strong high frequency field. (Info on this from Ablex and Maxell). I usually find that putting an important file on an mdv is quite enough for the mdv to lose whatever is on there!".

This paragraph contains a number of important truths, although Dennis and Richard are talking about complete erasure for re-use as any source of electro/magnetic activity can potentially corrupt and degrade the contents of a disk or mdv.

QLTR5 has 36 pages A4, and costs £1.50.

Also new from CGH is a one-off publication, *The QL Survivor's Source Book.* This is an A4 stapled one-side-of-the-paper-only booklet with a coloured card cover, carrying an international listing of all the software and hardware suppliers known to have been supplying QL materials in recent historical times, other than those which are definitely known to have ceased trading. All the QL user groups, clubs and periodical publications are also listed. A further useful listing would have been non-periodical publications (manuals, books etc), still available, and their sources, but this is perhaps beyond the scope of a guide which is basically a source of addresses.

Each company is listed in alphabetical order, with full address, phone number, and a brief list of its products. No further advice is given apart from the occasional health warning, but space is left – typically 1 by 3 inches – for users to add their own notes.

The Survivor's Guide costs £2 for a stout number of pages containing every name you have ever heard of and many that you probably haven't. CGH are doing a final update as we write, but it should be ready for sale as you read.

Both publications are available from **CGH Services, Cwn Gwen Hall, Pencader, Dyfed, Cymru SA39 9HA. Tel. 0559 384574.** Prices as quoted, plus 10% post and packing for postal orders.

Richard Alexander of CGH would also like to hear from anyone who has a copy of *Oracle Fortunata* by Brian Richardson. CGH now have permission to re-publish this program, but are unable to obtain a copy.

STD Index

Sector Software have now completed the update of *STD Index* for the QL. This is a 'much uprated' version of the program that has been on sale for the last two years. Counties have been added to '5,600 of the 6,000+' entries. The program costs £14 but is available for £5 as an upgrade for existing owners of STD Index returning their original program. The program requires an expanded QL and is available on 3.5 and 5.25in disk only.

An international version of STD Index – new with this release – contains dialling codes world-wide and costs £9. All prices are subject to an additional £2 for post and packing.

Some QL programs will now be available on 5.25in disk as well as 3.5in., including Microdrive Exchange programs and Sector's own house software. A new version of *Ferret* for hard disks is now out, priced £12, or £5 plus the original cartridge for current owners wishing to upgrade.

Keyboard membranes are available at time of writing ex-stock for £8 plus £2 post and packing.

Orders and enquiries to **Sector Software, 39 Wray Crescent, Ulnes Walton, Leyland, Lancs. PR5 3NH. Tel. 0772 454328/432414.**

Quanta in Loughton

The Essex group of Quanta are holding a computer workshop in Loughton on Saturday 18 May 1991.

Organised by Bob Gingell, the Workshop hopes to attract a number of traders, although no names are available yet. There will also be software demonstrations and Library facilities.

The meeting venue will be the Debden Community Association, Loughton Hall, Rectory Lane, Loughton, Essex, from 10am. The group has booked four rooms, with options on others, and will have a number of machines up and running. The site has parking, refreshments and the use of a licensed bar.

For more information, write to **Bob Gingell, 22 Paley Gardens, Loughton, Essex LI10 2AN, or phone 081 508 8370 at social hours.**

Fair Sailing

The All Formats Fair on Saturday 2 February attracted a healthy amount of activity around the reports a *QL World* reader.

Freddy Vachha of Digital Precision had collected his usual crowd of admirers, EEC looked busy, Miracle had two people who appeared fully occupied, Syd Humphries of Quanta didn't look at all bored. The surprise was David Batty, spread across two stands. He said that his 'disappearance' over the past couple of months had been due to his move back to Wray Crescent and intensive programming on his STD Codes package (which is available on the QL, PC and Amiga). It is interesting that these frequent fairs don't seem to be overstating the excitement; there are as many people attending now as when they started.

DBQL

Tom Ashcroft continues, with part two of his simple relational database.

Last month's article presented the core procedures for a flat file database and included simple methods of searching for information which just involve looking at each record in turn until a match is found with a search key. This is reasonably fast in a short file, even in Superbasic, but when a file contains hundreds or even thousands of records it takes too long to be a practicable method.

However, if the records of a file are stored not in random order but in alphabetical order, there is a much faster way of looking up a record, called a binary search. In fact, the binary search is the method we ourselves use instinctively when looking up an item in an alphabetical list. The computer binary search is given a search key and compares it not with the first record in the file but with a record near the middle of the file. If this is, say, above the search key in the alphabet (closer to z) the wanted record must lie in the half of the file below that position and so, with a single comparison, half of the file has been eliminated from the search. The program then picks a record in the middle of the lower half of the file and repeats the process, so that with each comparison, the search is narrowed to half of the remaining records.

Quick search

In a file of 1000 records, even if the wanted record were the last one examined, a total of only 10 comparisons would be needed. In the same time, a simple paging search would have eliminated 10 records, with only 990 more to go!

There are two ways of using alphabetical files in a database. The simplest is just to arrange the records of a file in alphabetical order of one of the fields – an ordered file. The most familiar example is the Telephone Directory, which has fields for initials, surname, address and phone number and is ordered on the surname field. The disadvantage is that a file can only be ordered on one field at a time and if a search is required on another field the records have to be resorted – a process which takes time.

```
4000 REMark ******************* INDEX FILE
4010 DEFine PROCedure in
4020 DIM ndx$(1,nn,2,chars):active =1
4030 IF NOT pac THEN
4040     CLS:FOR j=1 TO fields:PRINT j,a$(0,j,3 TO)
4050     INPUT"Enter field number: ";fldno$:IF fldno$="" THEN
PRINT"Terminating":RETurn
4060     fldno=fldno$: PRINT"Creating index file "; a$(0,
fldno,3 TO LEN(a$(0,fldno)))&"_ind"
4070 END IF
4080 a$(0,fldno,1)="I"
4090 FOR k=1 TO nn:ndx$(active,k,1)=a$(k,fldno):
ndx$(active,k,2)=k
4100 DIM comp$(2,chars)
4110 FOR item=2 TO nn
4120 in_sort
4130 END FOR item
4140 indname$=a$(0,fldno,3 TO LEN(a$(0,fldno)))&"_ind"
4150 DELETE "mdv1_"&indname$
4160 OPEN_NEW#5,"mdv1_"& indname$
4170 FOR k=1 TO nn:PRINT#5,ndx$(active,k,1): PRINT#5,
ndx$(active,k,2)
4180 CLOSE#5
4190 IF NOT pac THEN dbsave
4195 lload
4200 END DEFine in
4205 REMark ***************** IN_SORT
4210 DEFine PROCedure in_sort
4220     p=item
4230     comp$(1)=ndx$(active,p,1):comp$(2)=ndx$(active,p,2)
4240     ndx$(active,0,1)=comp$(1):ndx$(active,0,2)=comp$(2)
4250     REPeat compare
4260       IF comp$(1)>=ndx$(active,p-1,1) THEN EXIT compare
4270       ndx$(active,p,1)=ndx$ (active,p-1,1):
ndx$(active,p,2) = ndx$(active,p-1,2)
4280       p=p-1
4290     END REPeat compare
4300     ndx$(active,p,1)=comp$(1):ndx$(active,p,2)=comp$(2)
4310 END DEFine in_sort
```

Listing 13

The second way of using alphabetical files is to make up an index file of the required field. The index file works in exactly the same way as the index of a book. It is a separate array with a record corresponding to each record in the main file. It has only two fields, one to hold the contents of the field to be searched, sorted into alphabetical order, and one to hold the corresponding number of the record in the main array. When the surname, say, has been located in the index file by means of a binary search the record number in the main file is read off and the record can be displayed. If necessary, a separate index can be held for each field in the database and there is no need to sort the database at all. Let's say we have a database to hold names and addresses of mail order customers. A printout of part of an index file for city might look like this:

City	Record no. in main file
Cardiff	72
Doncaster	40
Manchester	3
Sheffield	98
Sheffield	186
Sheffield	54
Swindon	65

Records with the same city are found grouped together.

To create an index file, we would need to set up an array with the same number of records and field length as the main file and copy into it the 'city' field and record number of each record in the main file. The array now needs to be sorted on the 'city' field. There are various different algorithms for sorting lists alphabetically, some of which are more efficient than others, but it is beyond the scope of this article to go into these in detail. For our purposes the insertion sort routine described on pages 96-98 of the Sinclair *QL User Guide* is ideal as it allows new records to be added to the file at a later date. It will be divided into two procedures **(listing thirteen)** so that different parts can be called from different sources.

Implementing index files in DBQL requires several new procedures and substantial changes in some existing procedures. The new procedures will be numbered from line 4000 onwards. In modifying old procedures, the original line numbers have been adhered to as far as possible, and new lines mostly inserted between them. In theory, an index might be needed for every field of a database and a multidimensional array will be used eventually to store these, but each index will be created separately and saved on microdrive.

Index file

The procedure IN is intended to create an index file for a database already in memory. It begins by dimensioning the index string array, ndx$, as 1 group of nn records, each consisting of two fields, chars characters in length. Active is a variable normally storing the number of the active index when several fields in a database are indexed, but only one index is active at present. Pac is a variable which will be explained later. Its value is usually 0 and so 4040 prints a list of field names and numbers and 4050 requests the number of the field to be indexed. Keying ENTER here will abort the whole procedure.

Line 4080 flags the chosen fieldname by inserting a letter I in place of the leading blank character in a$(0). The flag will be used in future when loading the index from microdrive. The chosen field and record number of each record in a$ are copied into ndx$ by 4090. Comp$ in 4100 is an array used by the sort procedure IN_SORT, which is called by the loop in 4110 to 4130, for each record of ndx$ after the first. IN_SORT itself is closely modelled on the program in the Sinclair QL Guide. After sorting, ndx$ is saved in a microdrive file with a name made up of the fieldname and the extension '_ind', after deleting any previous file of the same name. Finally, 4190 resaves the database with its added flag, using a modified save procedure.

Before the new index file can be used it must be loaded into the definitive index array, in which it may make up only one dimension if the database has multiple index files. The procedure LLOAD has been modified by the addition of lines 2260 – 2270 **(listing fourteen).**

Line 2261 scans the field in a$(0) and increments the variable, indices, whenever a flagged field is encountered, to find out how many index fields are held for the

```
2100 REMark ******************** OPEN
2110 DEFine PROCedure op
2120 CLS:INPUT\"Enter file name: ";f$
2130 lload
2140 CLS:AT 10,7:PRINT"The file '"&f$&"' is open":PRINT\ TO
13;nn;" records"
2145 IF indices THEN PRINT\ TO 13;indices;" index file(s)"
2150 END DEFine op
2160 REMark ******************** LLOAD
2170 DEFine PROCedure lload
2180 OPEN_IN #6,"mdv1_"&f$
2190 INPUT#6,nn:INPUT#6,fields:INPUT#6,chars:INPUT#6,deleted
2200 DIM a$(nn+10,fields,chars)
2210 FOR record=0 TO nn
2220 FOR fld=0 TO fields
2230 INPUT#6,a$(record,fld)
2240 END FOR fld
2250 END FOR record
2254 CLOSE #6:oldfile=1
2260 IF pac THEN RETurn
2261 indices=0:FOR j=1 TO fields:IF a$(0,j,1)="I" THEN
indices=indices+1:a$(0,j,2)=indices
2262 IF NOT indices THEN RETurn
2263 DIM ndx$(indices,nn+10,2,chars)
2264 FOR j=1 TO fields
2265   IF a$(0,j,1)="I" THEN
2266     OPEN_IN #6, "mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind"
2267     FOR k=1 TO nn:INPUT#6, ndx$(a$(0,j,2),k,1):INPUT#6,
ndx$(a$(0,j,2),k,2)
2268     CLOSE#6
2269   END IF
2270 END FOR j
2271 END DEFine lload
```

Listing 14

```
4340 REMark *********** BINARY SEARCH
4350 DEFine PROCedure lo
4360 CLS:INPUT\"Please enter item to look for: ";s$
4370 FOR j=1 TO fields:IF a$(0,j,1)="I" THEN PRINT
a$(0,j,2),a$(0,j,3 TO)
4380 PRINT\"Current field for search is ";active\"Press ENTER
to confirm"\"or enter another number:  ";:INPUT q$:IF q$<>""
THEN active=q$
4400 FOR sfld= 1 TO fields:IF a$(0,sfld,1 TO 2)="I"&active
THEN EXIT sfld
4410 z=bin_s(nn,s$,ndx$)
4420 c=ndx$(active,z,2)
4430 IF a$(c,sfld,1 TO LEN(s$))<>s$ THEN PRINT\"NOT FOUND ":
RETurn
4440 updown s$,ndx$,z
4450 END DEFine lo
4460 REMark *********** FN BIN SEARCH
4470 DEFine FuNction bin_s(n,x$,y$)
4480 LOCal s
4490 power=INT(LN(n-1)/LN(2))
4500 s=2^power
4510 FOR k=power-1 TO 0 STEP -1
4520 s=s+(2^k)*(x$>y$(active,s,1,1 TO LEN(x$)))-(2^k)*
(x$<y$(active,s,1,1 TO LEN(x$)))
4530 IF s>n THEN s=n
4540 IF s<1 THEN s=1
4550 END FOR k
4560 RETurn s
4570 END DEFine bin_s
```

Listing 15

```
4580 REMark ******************* UPDOWN
4590 DEFine PROCedure updown(x$,y$,s)
4600 REPeat upward
4610 s=s-1
4620 IF s=0 THEN EXIT upward
4630 c=y$(active,s,2)
4640 IF a$(c,sfld,1 TO LEN(x$))<>x$ THEN EXIT upward
4650 END REPeat upward
4660 REPeat downward
4670 s=s+1
4680 c=y$(active,s,2)
```

Listing 16

database. At the same time it inserts a second flag as the character following I in the field name. This holds the number of the index and will be used to set the value of the variable, active when the index file is used. Line 2262 aborts the procedure if there are no index files. Line 2263 sets up the index array, ndx$, with as many dimensions as there are index files and the loop in 2264 to 2270 again scans the field names in a$(0) and when a leading I is found loads the index file with the appropriate name into the dimension of ndx$ indicated by the second character in a$(0,j). The index files are now ready for use in a binary search **(listing fifteen).**

The search is carried out by a new LO (lookup). This first requests the item to search for, which is stored as s$. After displaying the list of indexed fields (4370), numbered according to the flag in a$(0,j,2), the user is asked to indicate which index is to be active. The only other information the program now needs is the number of the field to be searched and line 4400 uses a loop to find which field in a$(0) starts with an I followed by a number equal to active. When a match is found the loop exits and the counter variable, sfld, is left holding the appropriate value.

Logarithms

The actual search is carried out by the defined function bin_s. This is called by 4410 which passes the variables nn, s and ndx$ (see the Sinclair Guide pp 86 to 88). The function first uses logarithms to find the highest power of 2 which is less than nn. The search position, s, is set equal to the next lower power of 2, and the loop in 4510 to 4550 is commenced. Line 4520 uses logic functions (see Sinclair User Guide pp 54-56) to increase or decrease the value of s according to whether the field in the active dimension of ndx$ at position s is greater or less than a$. If the field in ndx$ is equal to s$ neither of the logic statements is true and the value of s is unchanged.

Lines 4530 and 4540 prevent the value of s being moved beyond the limits of a$. The final value of s is returned as a value for z in 4410 and 4420 translates this into a value for c which 4430 uses to check the appropriate field in a$ against s$. If they do not match the procedure aborts with a screen message "NOT FOUND", otherwise the search moves on to the next stage – UPDOWN, called by 4440 **(listing sixteen).**

A match in 4410 means that the program has found one matching record in ndx$ but there may well be more. If so, these will be grouped in adjacent records in ndx$(active,nn,1), which was alphabetically sorted. UPDOWN simply pages upward through ndx$ until it reaches a record which no longer matches. It then pages downwards through ndx$, record by record, printing the corresponding record

```
4690 IF a$(c,sfld,1 TO LEN(x$))<>x$ THEN EXIT downward
4700 b_s_display
4710 IF s=nn THEN EXIT downward
4720 END REPeat downward
4730 END DEFine updown
4740  REMark ********** B S DISPLAY
4750 DEFine PROCedure b_s_display
4760 PRINT"#"!c!!:FOR j=1 TO fields:PRINT a$(c,j)&" ";:END
FOR j:PRINT
4770 END DEFine b_s_display
```

Listing 16

```
1980 DEFine PROCedure ssave
1981 IF NOT resave THEN RETurn
1982 dbsave
1984 IF indices THEN indexsave
1985 resave=0
1986 END DEFine ssave
1987 :
1988 DEFine PROCedure dbsave
1990 IF oldfile THEN COPY "mdv1_"&f$ TO "mdv1_"&f$&"_temp":
DELETE "mdv1_"&f$
2000 OPEN_NEW #6,"mdv1_"&f$
2010 PRINT#6, nn-(deleted*(pac=1)): PRINT#6,fields:PRINT#6,
chars: PRINT#6,deleted*(1-pac)
2020 FOR record=0 TO nn
2030 IF a$(record,0)="D" AND pac=1 THEN NEXT record
2040 FOR fld=0 TO fields
2050 PRINT#6,a$(record,fld)
2060 END FOR fld:END FOR record
2070 CLOSE#6:DELETE "mdv1_"&f$&"_temp"
2071 END DEFine dbsave
2072 :
2073 DEFine PROCedure indexsave
2074 FOR j=1 TO fields
2075   IF a$(0,j,1)="I"THEN
2076     COPY "mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind" TO
"mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind_temp"
2077     DELETE "mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind"
2083     OPEN_NEW#6, "mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind"
2084     FOR k=0 TO nn:PRINT#6, ndx$(a$(0,j,2),k,1):PRINT#6,
ndx$(a$(0,j,2),k,2)
2085     CLOSE#6
2086     DELETE "mdv1_"&a$(0,j,3 TO LEN(a$(0,j)))&"_ind_temp"
2087     END IF
2088   END FOR j
2090 END DEFine indexsave
```

Listing 17

```
4775 REMark ************** UPDATE_INDEX
4780 DEFine PROCedure update_index
4790 DIM comp$(2,chars)
4800 FOR j=1 TO fields
4810   IF a$(0,j,1)="I" THEN
4820     active=a$(0,j,2)
4830     ndx$(active,nn,1)=a$(nn,j): ndx$(active,nn,2)=nn:
item=nn
4840     in_sort
4850   END IF
4860 END FOR j
4870 END DEFine update_index
```

Listing 18

```
4880 REMark ************************ PACK
4890 DEFine PROCedure pa
4900 pac=1
4910 dbsave
4920 lload
4930 CLS:PRINT\"Rebuilding the index files.  Please wait."
4940 FOR j=1 TO fields
4950   IF a$(0,j,1)="I" THEN
4960     fldno=j
4970     PRINT\a$(0,j,3 TO LEN(a$(0,j)))&" index"
4980     in
4990   END IF
5000 END FOR j
5010 pac=0
5020 lload
5030 END DEFine pa
```

Listing 19

```
1800 DEFine PROCedure am
1810 CLS:CLS#0
1820 CLS#4:PRINT#4,"                                        ENTER  leaves item
unchanged "
1830 PRINT#4,"                                  To change an item, type new
version"
1835 DIM hold$(fields,chars)
1840 resave=1:reenter=0
1850 FOR fld=1 TO fields
1860    PRINT TO 9-(LEN(a$(0,fld))-3);a$(0,fld,3 TO LEN
(a$(0,fld))); TO 11;":   "&a$(c,fld)
1870    INPUT q$:IF q$="" THEN :hold$(fld)=a$(c,fld):ELSE:
hold$(fld)=q$:reenter=reenter+(a$(0,fld,1)="I"):END IF
1880 END FOR fld
1881 IF reenter THEN
1882    a$(c,0)="D":deleted=deleted+1:nn=nn+1
1883    IF nn=DIMN(a$) THEN PRINT"Saving file to microdrive.
Please wait.":ssave:lload
1884    FOR j=1 TO fields:a$(nn,j)=hold$(j)
1885    update_index
1886 ELSE
1887    FOR j=1 TO fields:a$(c,j)=hold$(j)
1888 END IF
1889 IF reenter THEN c=nn
1890 display:menu
1900 END DEFine am
```

Lisintg 20

```
1020 DEFine PROCedure start
1025 MODE 4
1030 OPEN #3,ser1
1040 OPEN #4,con_480x30a10x0
1050 WINDOW 480,205,10,30
1055 WINDOW#0,480,20,10,236
1060 resave=0:oldfile=0:nemo=1:chan=1:deleted=0:c=1:pac=0:
indices=0: active=1
1070 PAPER 0:PAPER #0,0:PAPER#4,0:INK 7:INK
#0,7:INK#4,4:CLS:CLS#0
1080 menu
1090 END DEFine start
```

Lisintg 21

in a$ to screen, until it again reaches a record which no longer matches and then terminates. The result is that all matching records are printed to screen in alphabetical order. A new procedure B_S_DISPLAY is used to print the fields of a$ in line across the screen, in order to make best use of the screen space.

An important point is that s$ is only compared with the field content up to the length of s$, so that, for example, entering A as a search key would find all records whose 'city' field, say, began with A, whatever the rest of the field might be. Similarly, entering 'Ab' would find 'Aberdeen' and 'Abingdon' but not 'Arbroath'.

The presence of index files requires changes in the procedures for saving, entering, amending or deleting records. Saving is in **listing seventeen.**

Procedure SSAVE now has two sections: DBSAVE which saves the database as before after making a temporary copy on microdrive, and INDEXSAVE which does the same for each index file in turn, taking each filename from the corresponding fieldname in a$(0). The function of the variable, pac, is explained below.

When a new record is entered, the index files must be updated and two new lines have been added to the procedure EN.

```
1251 a$(nn,0)=""
1252 update_index
```

Line 1251 places two blank characters in field 0 of the new record and 1252 calls a new procedure UPDATE_INDEX. This checks each field of a$(0) in turn and, if a leading "I" is present, places an entry in the appropriate dimension of ndx$ and calls IN_SORT to move it to its correct alphabetical position. This is repeated with each new entry.

Deleting a record is a little more complicated. If it were done each time the file is saved, it would involve deleting an item from each index file and closing up the resulting gap each time which would be tedious if several deletions were needed. It is better to flag records for deletion as before but to save the actual deleting for a convenient time when a good number of records have been flagged. The index files can then be completely deleted and recreated afresh. These operations are carried out by a new procedure PA (pack) beginning at 4880 **(listing nineteen).**

Dropping of flagged records at resave now only occurs if a second flag pac is set and so PA begins by resaving with pac set, to eliminate all the deleted records. The fields of a$(0) are then scanned and when a leading "I" is found the procedure IN is called to create a new index which is saved to microdrive. Finally, pac is set to 0 and the database is reloaded with its new index files. This method of deleting means that if the data base is not to be packed immediately, the value of deleted must be saved and reloaded with the file, and 2010 and 2190 have been amended accordingly. If pac is set, 2010 automatically adjusts the value of nn and sets deleted to 0, using logic statements to make the changes:

```
2010 PRINT#,nn=(deleted*(pac=1)):
PRINT#6,fields:      PRINT#6,chars:
PRINT#6,deleted*(1-pac)
2190 INPUT#6,nn: INPUT#6,fields:
INPUT#6,chars: INPUT#6,deleted
```

Amending a record is also more complex with index files in place **(listing twenty).**

Changing a non-indexed field is no problem but changing an indexed field would mean complex manipulations of the index file and it is easier to delete the original record and add the amended version to the end of the database as a new record. The original record and its index entries are still present, of course, until the file is packed, but it could be made invisible by making DISPLAY skip any record with the "D" flag set.

In the changed procedure AM, a temporary array, hold$, is dimensioned to store one record. As the fields are displayed one by one, pressing ENTER causes the original field to be copied into hold$, while any new amended field is sent directly to hold$. If the changed field happens to be an indexed field, a flag, reenter, is set. If the end of the record is reached with reenter not set, the fields of hold$ are copied back into a$(c) as the amended version.

If reenter has been set, hold$ is copied as a new entry at the end of the database and the index files are updated. The original version of the amended record remains unchanged, but flagged for deletion. Line 1883 makes room for the new entry if a$ happens to be completely filled, in the same way as 1260. The original record will be deleted the next time the file is packed.

Finally, the procedure START has been modified to initialise new variables and to increase screen space **(listing twenty-one).** The program now uses mode 4 and the window sizes have been changed to give 80 characters per line. Program instructions are printed in channel #4 while channel #0 has been reduced to two lines only. The menu should be changed to include the new procedures INdex and LOokup.

Next month the program will be further developed to provide a SELECT function and procedures for validation of input will be added.

PSION SOLUTIONS

Howard Clase explains how the Psion printer driver translates can be supplemented with codes sent from within the text, using the QL foreign character set.

One of the questions frequently asked about the Psion printer driver is whether it is possible to increase the number of 'translates' beyond 10, and so far the answer is always 'no'! However, many printer control codes can be sent directly from within the text using the second (or foreign) character set. Our QL is frequently called upon to produce special characters for Finnish, French, German and chemistry, and we find that there is no problem if the 10 translates are used in conjunction with this trick. (Russian did require some modification to *Quill.*)

The QL, like most other computers, communicates with printers using the Ascii code, which consists of the 128 numbers from 0 to 127 (as 7 bit binary numbers). The first 32 numbers and the last one (127) are used as control codes, while the other 95 represent printable characters. The QL uses more or less the same codes for its screen display, but there is no direct connection between the character as it appears on screen and on the printer. When, for example, you type 'PRINT#n, "A";' the computer sends the number 65 to the appropriate channel (printer or screen according to "n"), not a description of a letter "A". The printer or screen firmware then looks up 65 in a table in its memory to find out how to display an "A".

Two ways

From Basic you can send a character code to a channel in two ways ('PRINT#n, "A";" or 'PRINT#n,CHR$(65);'); but if you want to send a *control* code from Basic you can normally only use the second form. 'PRINT#n,CHR$(7);' will sound the beeper on a printer, but prints nothing on paper or screen.

CHR$() is also available in *Archive* and *Abacus,* but not in Quill. The required printer control codes are generated by the printer driver, which can be adapted to your printer's requirements using INSTALL_BAS. This gives you two ways of sending control codes:

1. The hidden 'embedded' codes, which are visible on the screen only in their effect on the display. These are usually set up to produce the corresponding effect on the printer output, but this is not necessary: you could, for example, have the underline feature on the screen produce italics on the printer.

2. The 'translate' options, which enable any single code to be intercepted on its way to the printer and forwarded as a string of up to nine codes of your choice. This means that you can use a single screen character to represent another on the printer (eg £ for #) or as a 'special' character to send a complex set of instructions to the printer. Many word processors on other machines use special codes to send all their control messages to the printer.

However, there is a third way of sending control codes to the printer, which is not documented in the manual, and that is to use the foreign characters with codes higher than 127. This arises because the QL, like most other modern computers and printers, uses all eight bits of its bytes, rather than only the last seven as in the original Ascii code, giving 256 possible codes (0 to 255). However there is no standard use agreed for the 128 codes from 128 to 255. The QL uses the first 63 of them (128 to 191) for its international and miscellaneous characters on the screen, and will send the corresponding codes out through the serial or parallel ports, but these are not recog-

SINGLE CHARACTERS•

Character	Keying	Effective code (mnemonic)	Effect on printer
û	CTRL & ;	27 (ESC)	Escape
ë	CTRL & /	15 (SI)	Select compressed
ï	CTRL & 2	18 (DC2)	Cancel compressed
ç	CTRL & SHIFT & 9	08 (BS)	Backspace
ü	CTRL & ˙	07 (BEL)	Sound beeper

TWO CHARACTER "ESCAPE" CODES

Characters	Effective code	Effect on printer
ûM	CHR$(27);"M"	Select elite type
ûP	CHR$(27);"P"	Select pica type
û4	CHR$(27);"4"	Select Italic mode
û5	CHR$(27);"5"	Cancel Italic mode
û8	CHR$(27);"8"	Disable {paper-out
û9	CHR$(27);"9"	Enable {sensor
ûT	CHR$(27);"T"	Cancel super/subscript
û0	CHR$(27);"0"	1/8 in line spacing

THREE CHARACTER "ESCAPE" CODES

	Characters	Effective code	Effect on printer
	ûS0	CHR$(27);"S0"	Select superscript
	ûS1	CHR$(27);"S1"	Select subscript
	û-1	CHR$(27);"-1"	Select underline
	û-0	CHR$(27);"-0"	Cancel underline
*	û3n	CHR$(27);"3";n	n/216 in line spacing
*	ùln	CHR$(27);"l";n	set L margin n spaces
*	ûQn	CHR$(27);"Q";n	set R margin n spaces

* the "n" in these lines refers to the numerical code of a character, not the character itself.

nised as the same characters by printers.

Epson printers in their normal state interpret the 32 Ascii codes from 128 to 160 as though they were the 32 codes from 0 to 31. (The most significant bit of the binary number is effectively ignored). This makes it possible to use the first 32 characters of the second character set to send printer control codes to the printer from the QL either from Basic or from within Quill, in the latter case without using up translates in the printer driver.

Before I explain how to do this let's have a look at the printer control codes themselves. In the early days of computing 32 codes was enough to control the primitive teletype printer terminals, and many of the codes give a direct instruction, — 10 = new line, 8 = backspace, etc. However, modern printers like my Epson FX85 can accept nearly 100 instructions.

The key

The key to this expansion is code 27 or ESCape, which suspends the 'normal' role of the following code or codes and gives them another meaning. For example, code 77 normally causes an "M" to be printed, but the sequence 27 77 (ESC "M") switches the printer to elite pitch. In this example the printer doesn't expect any more control information after the 77 ("M") and treats the next character normally, but longer sequences are sometimes involved.

Sometimes a third code is used as a flag or to provide other information, for example 27 87 49 (ESC "W1") selects expanded mode while 27 87 48 (ESC "W0") turns it off again, and 77 74 n (ESC "J";n) causes the paper to advance n/216th of an inch. In the second example n stands for the number of 1/216ths of an inch to advance: it is not the character "n". If you use "n" the number 110 would be sent, moving the paper about 1/2 inch.

In printer manuals, which in my experience are rarely a model of clarity, the following methods may be used to represent codes. It is important to know which applies in each case:

1. The corresponding printed character in quotes, eg "A" for 65, "8" for 56, etc. These are found in the Concepts section of the *QL User Guide* manual (pp 5-9 in the 12/84 edition).

2. The number itself, usually in decimal, but sometimes in hexadecimal or octal. This is often expressed as CHR$(n), since this is the normal way to send it to the printer from Basic. When a number is given it is important to make sure that it is the value of the number that is intended and not the character, which will be given in quotes: 8 = 8, but "8" = 56.

3. The Ascii mnemonics for the 32 control codes, eg ESC, NUL, LF, CR. These pop up in the Psion program "INSTALL_BAS" for writing printer drivers. They are listed in order in the Information section of the QL manual (p.5 in the 12/84 edition), but you have to count to work out which corresponds to which code: NUL = 0, SOH = 1, etc.

The code – character – Ascii mnemonic correspondences will probably also be listed somewhere in your printer manual.

Since 'foreign' characters are not available in normal type and might present *QL World's* printers with some problems, they are represented by their keying enclosed in angle brackets in the text, eg <CTRL&;> for hold down CTRL and press ";". This causes a "u" with a circumflex accent (^) to appear on the screen. (Do not type the <> characters.

This is the most important one in the present context since its QL code is 155, and 155 minus 128 is 27 or ESCape. If you PRINT this character to an Epson printer it ignores the first bit, which is the same as subtracting 128 from the value, and has the same effect as sending an ESCape, so, for example, PRINTing "<CTRL&;>" followed by "M" (or "<CTRL&;>M") selects elite pitch.

This means that you can easily send most of the control codes from within Quill as embedded codes. There are a couple of other foreign characters that are directly useful, and these are given in the table, but most of the useful control codes use ESC followed by normal characters, so there is not much strange keying to learn. The table contains my selection of the most useful codes for my FX85. You may have to experiment with different printers or to produce other effects, but the examples cover most of the types of combinations that occur.

If you are using this method from within Basic you must enclose the characters in quotes, but from within Quill this is not required. (If you do the quotes will be printed, but the printer will still respond to the instructions). You can send multiple instructions simply by stringing them together (without spaces).

Drawback

The main drawback of using the foreign characters as printer control codes in this way is that it upsets right justification, since they appear on the screen, but not in the printed line. So I generally use them for commands that I can put in the blank lines between paragraphs, such as changing printer typeface, or to send the ignore paper-out sensor command when printing single sheets. However, you can use a translate to define a single character as a double space and use this to replace enough single spaces in the line to get the margins right again, or use left justification.

The hidden 'typeface' codes are best used for changes that you want within lines, and the 'translates' best reserved for just that – one for one translations of characters, eg £ for #, or to get some of the foreign characters to come out on printer as on screen. The big advantage of the method is that you don't have to leave Quill and mess about with INSTALL_BAS to make use of it: the foreign character codes are available all the time.

Another problem is that it is not possible to use the foreign characters in this way when the printer expects a numerical value, the number of characters to set a margin to for example. Unfortunately in this instance the printer reads all eight bits of the code, so "<CTRL&;>" would have its full value of 155. It is not possible to send the numbers from 0 to 31 from within Quill without using translates, but beyond that just use the corresponding character, eg space for 32, ! for 33 etc.

An example of the use of foreign character codes that I often use to save space is two-column printing from Quill. It puts two pages of compressed text onto a single page of paper. First change the right margin of your _doc to 65 and you can increase the page size to 80 lines for 11in paper or 85 for A4. Avoid headers, but footers are OK.

Then type <CTRL&;>8<CTRL&;>0<CTRL&/> at the head of each odd page of your document. This will switch off the paper_out sensor, and select 1/8th in spacing and compressed print. Or you can set up a special printer driver with this as the preamble, and begin each even numbered page with <CTRL&;>1>. (ESC "1" 62 – the last ">" contributes Ascii value of 62 here and unlike the others should actually be typed in). To avoid justification problems it is best to put the codes on a separate line, filling it with tabs or spaces if you are in the middle of a paragraph and don't want the first line indented.

Use a printer driver set for separate sheets and print the document two pages at a time. The first page will print down the left half of the sheet. Then wind the page back to the top and print page two down the right-hand side, and so on with 3&4, 5&6 etc. (If your printer driver is set up for continuous forms you will have to repeat the <CTRL&;>8<CTRL&;>0<CTRL&/> at the head of every page followed by <CTRL&;>1> on the even ones and print one page at a time).

You can change the print style from within *Abacus* by inserting columns of the appropriate printer control code characters, eg if you precede a block with a column of <CTRL&/>s and follow it with a column of <CTRL&2>s (no quotes needed – make each column one character wide) the block will be printed in compressed type while the cells outside the block will be normal size. (This will not change the margins.)

I have not attempted to give an exhaustive list, since the details will vary with the model of printer, so get out your printer manual and experiment. Once you get the hang of it you can play your printer like a virtuoso!

EXCLUSIVE

SINCLAIR QL WORLD

For one month only, subscribers to **SINCLAIR QL WORLD** will be sent a free Mensa IQ test, to be completed at home.

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.

Please send me one year's subscription to **Sinclair QL World and my Free Mensa IQ Test**. 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............................ SQLW0491

U.K.: £21.00, EUROPE: £32.00;
REST OF THE WORLD: £38.00
OVERSEAS RATES INCLUDE AIRMAIL

Pass this first test and you will be invited to take a supervised test. Pass again and you become a fully-fledged Mensa Member.

Archive Answers

In the first of an occasional column, Robin Stevenson tacklets trouble-spots. This month: altering file structures.

A glaring omission from the built in features of the *Archive* database program is a way of changing the structure of an existing data file, I can still remember searching long and hard for the way to correct that silly typing mistake in a field name. Only with great reluctance did I accept that Psion had made no provision for doing so.

Typing mistakes may stare back accusingly, but you can grit your teeth and live with them, or if you have not entered the data yet, start again with a new file. But what about when you want to add another field? And removing redundant fields will save file space. Database needs, like everything in this world, are in a constant state of flux. There must be a way of adapting your data file, to meet the new circumstances.

Recent suggestions in this magazine involve exporting the data, tinkering with the resulting file, and then re-importing it back to Archive format. But this is a cumbersome, slow and messy process, particularly if a large database is involved. The solution offered here may not be fast, but it is flexible and powerful. It works entirely from within Archive;' gives all options on screen; and does the tricky bits by itself, making it simple to use.

The underlying idea is simple enough. You create a new file with the corrected or modified structure, and copy the data across. And if the new file ends up with the same name as the original, the illusion of having 'modified' it is complete.

The first complication is the need to edit the existing structure, instead of starting again with the normal create command. To do this the field names of the file in question (we shall call it 'oldfile') are all stored as data records in a second file — a structure file. This has just two fields, to hold a name and number for each of oldfiles's fields. This structure file can then be manipulated in any way required, just like any other Archive data file. When the changes to the field names have been made the file is used to create a new file, with the new structure.

It is in the transferring of information from the old to the new file that there is greatest scope for complication and error. For example, if you wish to add an extra field, 'firstname$', between 'title$' and 'surname$', a different transfer process will be needed than if you are changing a field name from 'sirname$' to 'surname$. In the first case existing data is matched from the old to the new by its name. But in the second case date is transferred between two fields with different names, so to match this data the position of the field is used. Data from old-field(1) is transferred to new-field(1), regardless of its name.

This is a fundamental aspect of changing the structure of a file. If you change the names, the transfer is by order. If you change the order, the transfer is by name.

```
ARCHIVE ANSWERS Listing for MODSTRUC_PRG

proc CopyStrucFile;Oldfile$,Strucfile$
  local COUNT        : rem NEWFILE$ WILL HAVE LOGICAL NAME 'STRUC'
  print : print "Copying structure of "+Oldfile$
  use Oldfile$      : rem OLDFILE$ IS LOGICAL NAME OF AN OPEN FILE
  create Strucfile$ logical "struc"                   : rem CREATE FILE
    FieldNumber
    FieldName$
    endcreate
  let COUNT=0: while COUNT<numfld(Oldfile$)    : rem EACH FIELD OF
    let struc.FIELDNUMBER=COUNT+1               : rem 'OLDFILE$' IS
    let struc.FIELDNAME$=fieldn(COUNT,Oldfile$)  : rem   A RECORD
    append "struc"                              : rem OF 'NEWFILE$'
    let COUNT=COUNT+1: endwhile
  print
  endproc
------------------------------------------------------------------
proc update2      : rem TRAPS A BUG IN ARCHIVE 2.00 UPDATE COMMAND
  local P: let P=recnum()       : rem STORE CURRENT RECORD NUMBER
  update : position P     : rem UPDATE, AND RESTORE RECORD POINTER
  endproc
------------------------------------------------------------------
proc CenPrint;Line,Text$    : rem TIDY WAY OF DISPLAYING A MESSAGE
  local COL: let COL=32-(len(Text$)/2)
  print at Line,0;chr(27)+"A"; tab COL;Text$;
  endproc
------------------------------------------------------------------
proc EditNames             : rem EDIT OF FIELD NAMES, IN STRUC FILE
  local K$,P
  CenPrint;6,"MODIFY FIELD NAMES AND DATA TYPES"
  let K$="": while K$<>"F"
    sprint :CenPrint;11,"[N]ext [B]ack [E]dit [F]inish"
    let K$=upper(getkey()): print
    if K$="N": next : endif
    if K$="B": back : endif
    if K$="E":CenPrint;11,"Enter/Modify field name"
      sinput FieldName$
      update2: endif
    endwhile
 endproc
------------------------------------------------------------------
```

If you need to do both to a file you must run the program twice, doing one of each. In the program there are different file editing options, depending on whether you elect to change the names or the order of the fields. The correct transfer sequence is used automatically when you have finished editing. This means that you have to decide which kind of alteration you wish to make before you start.

We shall now look the various elements of the program in turn. Type in each of the procedures using the Archive editor. If you know enough about Archive to test each of the procedures separately, this will be much better than typing it all in one go. In either case, save the program every time before trying it, and use test data files. Only when you are sure it is working correctly should you submit your valuable data to it. And even then make sure you have a backup copy.

CopyStrucFile copies the structure of the existing data file e (logical file name Oldfile$) into a new structure file (physical name Strucfile$). This is a simple process, because Archive gives ready access to field names via the FIELDN() procedure. A loop appends each oldfile field name to the structure file.

Manipulate

After the two short supporting procedures, there are the procedures for manipulating the structure file. *EditNames* allows changes to the text of the field names. It is also possible to change the type of the field by this means, simply by removing or adding the '$' sign. The transfer program will carry out any necessary type conversion. Note that apart from type conversion, you will not affect the contents of the fields by editing the names. All that changes is the field name.

Renumber keeps FieldNumber tidy, and is used by the next procedure, *EditStruc,* with only allows fields to be deleted or inserted. Any added fields will be blank when the data is transferred, and data in deleted fields will be lost. If you delete a field from one place, and add it (using the same name) in another, the data will be correctly transferred. In this way, a field can be moved to a different place in the file structure. The *NewField* procedure adds a field to the file, either above or below the current file, using FieldNumber to place it correctly.

From here on the eventual program will work without user intervention. The first problem is to create a file from the data held in the structure file.

The only way to solve this is to have a procedure that writes the required, specialised procedure there are then. The resulting program can then be merged and run. *CreateFrom* redirects output to a program file, and LPRINTs the proc and create lines. A loop writes each field name in turn, and the endcreate and endproc

```
proc Renumber              : rem RENUMBERS FIELDNUMBER, IN SEQUENCE
  local P,Q: let P=recnum(): last
  let Q=count()
  while Q>0
    let FieldNumber=Q: update : position Q-1
    back : let Q=Q-1
    endwhile : position P
  endproc
------------------------------------------------------------------------
proc EditStruc                : rem EDIT OF STRUCTURE, IN STRUC FILE
  local K$,V$       : rem A FIELD POSITION CAN BE MOVED BY DELETING
  CenPrint;6,"ADD or DELETE FIELDS." : rem IT FROM ONE PLACE, AND
  order FieldNumber;a .              : rem  ADDING IT  TO ANOTHER
  let K$="": while K$<>"F": sprint
    CenPrint;11,"[N]ext [B]ack [D]elete [F]inish"
    CenPrint;12,"Insert [A]bove or [U]nder this field"
    let K$=upper(getkey()): print
    if K$="N": next : endif
    if K$="B": back : endif
    if instr("AU",K$):NewField;"",K$: endif
    if K$="D": delete : sprint : endif
    if instr("AUD",K$):Renumber: endif
    endwhile
  endproc
------------------------------------------------------------------------
proc NewField;Name$,Above$   : rem INSERT NEW FIELD TO STRUC FILE
  local NUM: let NUM=FieldNumber
  if Above$="A" and recnum()>0     : rem IF [A]BOVE CURRENT RECORD
    back : let NUM=NUM+FieldNumber: endif
  if Above$="U": next : let NUM=NUM+FieldNumber : rem IF [B]ELOW
    if eof(): let NUM=NUM+2: endif
    endif :CenPrint;12,""
  let FieldName$=Name$
  let FieldNumber=NUM/2: append
  CenPrint;11,"Enter field name"
  sinput FieldName$:update2
  endproc
------------------------------------------------------------------------
proc CreateFrom;Strucfile$,Newfile$,Log$
  let TEMPFILE$="TEMP_PRG"          : rem USE RAMDRIVE IF AVAILABLE
  use Strucfile$: first : print
  print "Creating "+Newfile$+" from structure file";
  spoolon TEMPFILE$ export       : rem GENERATF TEMPORARY _PRG FILE
  lprint "proc Temp"
  lprint "create '"+Newfile$+"' logical '"+Log$+"'"
  while not eof()
    lprint FIELONAME$
    next : endwhile
  lprint "endcreate": lprint "endproc"
  spooloff : merge TEMPFILE$:Temp    : rem MERGE AND RUN TEMP FILE
  kill TEMPFILE$                     : rem       AND THEN DELETE IT
  print
  endproc
------------------------------------------------------------------------
proc AppendFrom;Source$,Dest$,ByName
  local COUNT,MAX
  let TEMPFILE$="TEMP_PRG"          : rem USE RAMDRIVE IF AVAILABLE
  use Dest$: last : let MAX=numfld()
  use Source$: first
  print "Appending records from "+Source$+" to "+Dest$;
  spoolon TEMPFILE$ export          : rem CREATE TEMPORARY _PRG FILE
  lprint "proc Temp"
  let COUNT=0: while COUNT<MAX      : rem AN ENTRY FOR EACH FIELD
    lprint "let "+fieldn(COUNT,Dest$)+"=";
    if ByName:AppByName;Source$,Dest$,COUNT
      else :AppByOrder;Source$,Dest$,COUNT: endif
    let COUNT=COUNT+1: endwhile
```

lines finish the procedure off. If you test this program, you will find afterwards that the new procedure is still there, called TEMP.

There are four procedures to transfer the data from the old to the new file, starting with *AppendFrom*. This uses a technique similar to CreateFrom, in that it too builds up a temporary procedure which actually does the work. Here each field of the new file has to be assigned values drawn from the old file. As explained earlier, the way they are obtained depends on whether we are transferring by name or by order. The parameter ByName, (which is 1 or 0, true or false) dictates which process is used. For each field in the new file, the second half of the statement is written by either AppByOrder or AppByName.

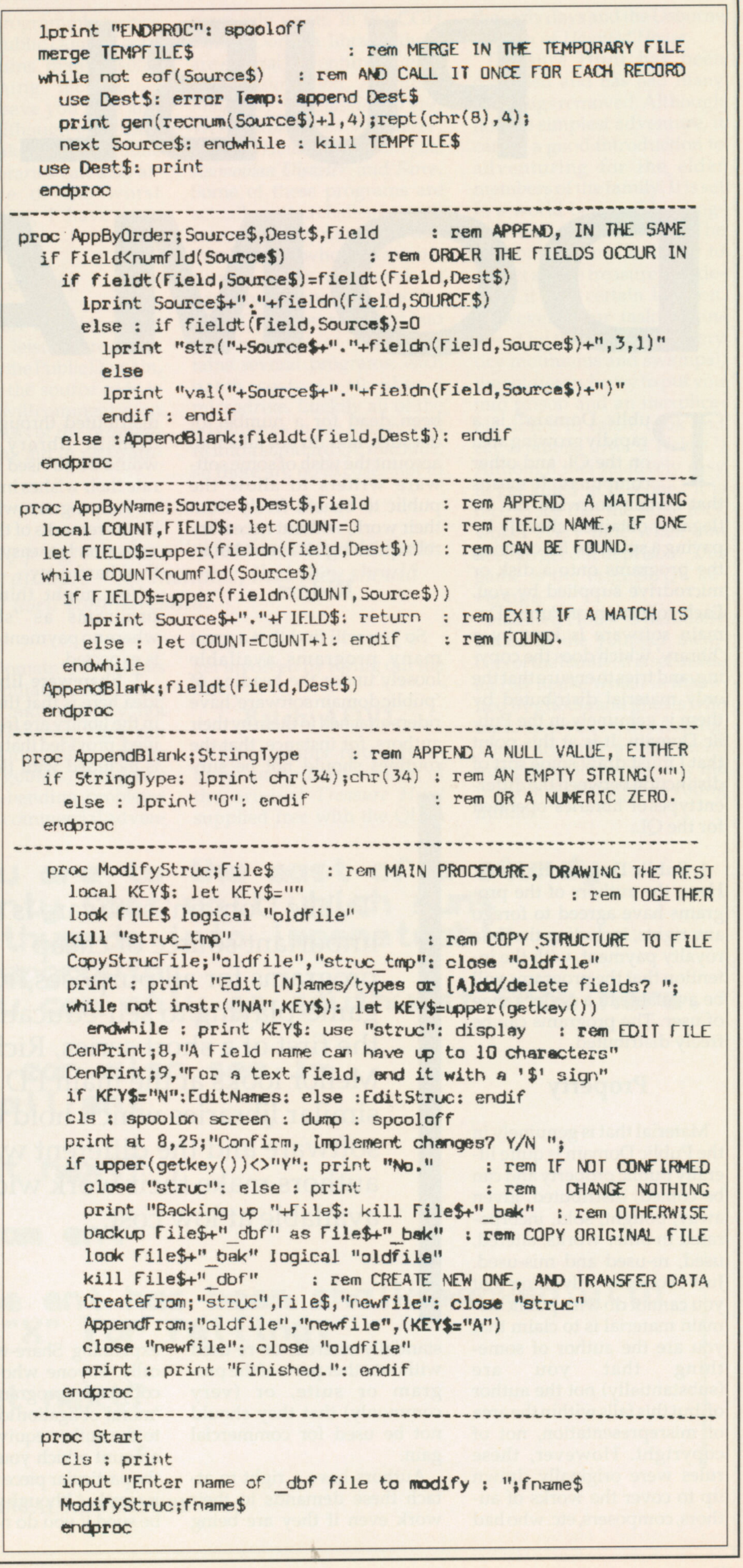
AppByOrder has to cope with four possibilities: A straightforward transfer (which is the same process, even if the names are different); type conversions from number to string; and from string to number; and where there is no corresponding field. This last should not occur unless you are transferring to a file not produced by this program.

AppByName has a rather different approach. For each newfile field, it scans down the list of oldfile fields, looking for an exact match. If it finds it, it can write the relevant line and RETURN. If there is no match, the new field is set to empty ("" for strings, Zero for numbers), for which the *AppendBlank* procedure is used.

Once the transfer procedure is complete it is merged into the program, over-writing any TEMP proc already in memory. A loop then steps through each record in oldfile, using the temporary procedure to transfer the data, which is then APPENDed to newfile. Note: if the original file is ORDERed, this will become the physical order for newfile's records. To preserve the original order of entry oldfile must have been RESET first.

All that remains is to provide a shell, to draw all these parts together into a single program. *ModifyStruc* does this by linking up the other procedures. It assumes the file you wish to modify is not open. In order for newfile to re-use the previous name, oldfile is copied (using BACKUP) to a file with the same name, but a '_bak' extension. When you have finished, you will find both a normal _dbf file (containing the new structure), and the _bak file (containing the original data). If you don't copy this _bak file it will be over-written if you modify the new file again.

The final procedure is called *Start*, and after obtaining the name of the file to modify, it simply launches ModifyStruc. Any previously saved program with a procedure called Start can be loaded and run automatically by typing RUN "<filename>" from the Archive prompt. To run it again simply enter START at the prompt.



```
  lprint "ENDPROC": spooloff
  merge TEMPFILE$               : rem MERGE IN THE TEMPORARY FILE
  while not eof(Source$)   : rem AND CALL IT ONCE FOR EACH RECORD
    use Dest$: error Temp: append Dest$
    print gen(recnum(Source$)+1,4);rept(chr(8),4);
    next Source$: endwhile : kill TEMPFILE$
  use Dest$: print
  endproc
------------------------------------------------------------------
proc AppByOrder;Source$,Dest$,Field      : rem APPEND, IN THE SAME
  if Field<numfld(Source$)         : rem ORDER THE FIELDS OCCUR IN
    if fieldt(Field,Source$)=fieldt(Field,Dest$)
      lprint Source$+"."+fieldn(Field,SOURCE$)
      else : if fieldt(Field,Source$)=0
        lprint "str("+Source$+"."+fieldn(Field,Source$)+",3,1)"
        else
        lprint "val("+Source$+"."+fieldn(Field,Source$)+")"
        endif : endif
    else :AppendBlank;fieldt(Field,Dest$): endif
  endproc
------------------------------------------------------------------
proc AppByName;Source$,Dest$,Field        : rem APPEND  A MATCHING
  local COUNT,FIELD$: let COUNT=0         : rem FIELD NAME, IF ONE
  let FIELD$=upper(fieldn(Field,Dest$))   : rem CAN BE FOUND.
  while COUNT<numfld(Source$)
    if FIELD$=upper(fieldn(COUNT,Source$))
      lprint Source$+"."+FIELD$: return   : rem EXIT IF A MATCH IS
      else : let COUNT=COUNT+1: endif     : rem FOUND.
    endwhile
  AppendBlank;fieldt(Field,Dest$)
  endproc
------------------------------------------------------------------
proc AppendBlank;StringType          : rem APPEND A NULL VALUE, EITHER
  if StringType: lprint chr(34);chr(34) : rem AN EMPTY STRING("")
    else : lprint "0": endif            : rem OR A NUMERIC ZERO.
  endproc
------------------------------------------------------------------
  proc ModifyStruc;File$      : rem MAIN PROCEDURE, DRAWING THE REST
    local KEY$: let KEY$=""                          : rem TOGETHER
    look FILE$ logical "oldfile"
    kill "struc_tmp"                       : rem COPY STRUCTURE TO FILE
    CopyStrucFile;"oldfile","struc_tmp": close "oldfile"
    print : print "Edit [N]ames/types or [A]dd/delete fields? ";
    while not instr("NA",KEY$): let KEY$=upper(getkey())
      endwhile : print KEY$: use "struc": display   : rem EDIT FILE
    CenPrint;8,"A Field name can have up to 10 characters"
    CenPrint;9,"For a text field, end it with a '$' sign"
    if KEY$="N":EditNames: else :EditStruc: endif
    cls : spoolon screen : dump : spooloff
    print at 8,25;"Confirm, Implement changes? Y/N ";
    if upper(getkey())<>"Y": print "No."      : rem IF NOT CONFIRMED
      close "struc": else : print             : rem   CHANGE NOTHING
      print "Backing up "+File$: kill File$+"_bak"  : rem OTHERWISE
      backup File$+"_dbf" as File$+"_bak"  : rem COPY ORIGINAL FILE
      look File$+"_bak" logical "oldfile"
      kill File$+"_dbf"      : rem CREATE NEW ONE, AND TRANSFER DATA
      CreateFrom;"struc",File$,"newfile": close "struc"
      AppendFrom;"oldfile","newfile",(KEY$="A")
      close "newfile": close "oldfile"
      print : print "Finished.": endif
    endproc
------------------------------------------------------------------
  proc Start
    cls : print
    input "Enter name of _dbf file to modify : ";fname$
    ModifyStruc;fname$
    endproc
```

PUBLIC DOMAIN

'Public Domain' software is an important source of cheap programs for all purposes from game-playing to self-education. In the first of a short series, Rich Mellor looks at the main PD and similar libraries which hold QL software and the different ways authors make their work widely available at low cost.

"Public Domain" is a rapidly growing area on the QL and other computers. It means that certain software can be (legally) obtained cheaply by paying a small fee for copying the programs onto a disk or microdrive supplied by you. Each collection of Public Domain software is kept by a 'library' which does the copying, and tries to ensure that the only material distributed by them is genuinely in the Public Domain. It is at this point that I must draw some sort of distinction between the different types of 'libraries' operated for the QL.

1. Public Domain libraries. Here, the authors of the programs have agreed to forego any rights, including that of a royalty payment, with the intention that the programs will be available to a wider range of user. The programs can be freely distributed.

Property

Material that is genuinely in the Public Domain is quite literally public property and can be copied, distributed, given away, bought, sold, incorporated in other works, altered, used, re-used and mis-used. Just about the only thing that you cannot do with public domain material is to claim that you are the author of something that you are (substantially) not the author of, but this falls within the area of misrepresentation, not of copyright. However, these rules were originally drawn up to cover the works of authors, composers, etc. who had been dead for a number of decades, and do not take into account the wish of some software writers to allow the public to benefit freely from their work, without necessary relinquishing all control of it.

Riders

So you will find that a great many programs available loosely under the heading of 'public domain software' have riders attached to them by their authors, for instance, that the routines should not be substantially altered, or re-used without other parts of the program or suite, or (very commonly) that they should not be used for commercial gain.

Authors have a right to attach these demands to their work even if they are being distributed through a public domain library. Authors would be advised to incorporate their wishes into the body of the program where possible, as remarks of this kind are a fraction less easy to 'mislay' than hard copy.

You might think of such programs as 'shareware' where no payment of any kind is expected.

2. Shareware libraries. The idea here is that the programs in the library are freely distributed, provided that a document is included with the program explaining Share-ware. Typically, anyone who receives a copy of the program is under a 'moral' obligation to send a sum to the author equivalent to an amount which you think that the particular piece of software is worth. Although you will not be sued if you do not send the author any money, if you don't then the author may withdraw the program from its Shareware status and write no more – pirates are their own worst enemies.

3. Member's libraries (eg the Quanta library). This type of library has an odd status. Although programs can be distributed free of charge, as with Public Domain software, it is illegal to distribute copies of such programs to anyone who does not belong to the organisation involved.

It is very difficult to maintain any sort of library because of the complexities of the copyright status of software. On top of this, it must always be noted that the actual copyright in any software involved remains vested in the author unless it has been expressly assigned to another party, such as a publisher. This means that any toolkits, routines or programs cannot be used in another program (whether it is commercial, public domain or shareware) without the copyright holder's (provable, if you wish to be safe) consent. In many cases the documents giving instructions for a program will include a note from the author allowing routines or parts of the program to be used in other programs so long as they are not to be sold commercially. On top of this, it must also be noted that any commercial toolkits, programs or routines, whether or not they have appeared as a listing in any magazines, also remain the copyright of the copyright holder and cannot be used in any Public Domain software without express permission. In the past many software authors (although not often in the QL world) have fallen foul of this rule.

European

QL World hears that in some European countries it is accepted in law that once a program has been published in a magazine, it can be freely distributed in the public domain. This is not the case in the UK. Frankly we cannot see the difference between taking somebody's poem, article, design or novel and circulating it without recompensing them, and doing the same thing with somebody's program, but the exact laws governing copyright are quite shambolic and differ considerably not only from country to country but from 'medium to medium' within one country's laws.

Varied

As for the range of software available in the Public Domain scene, this remains very varied, with many small routines which have never appeared anywhere else on the market, either because the author thought that their appeal was limited, or preferred to make the programs available to a wide audience at low cost. Indeed included in CGH's Public Domain library are a few programs which were originally sold on the open market, but whose authors have now decided to distribute freely.

Of course the actual quality of the programs can vary quite a lot, but with Public Domain the old maxim that you only get what you pay for does not always apply. For instance there are at least eight adventures in the CGH library which range from the very simple, with few locations, to one very complex adventure (*Fantasia Adventure*) with well over 100 locations which (so far as I am aware) at the time of writing no one has solved!

Code

However, with many of the programs in the library, the source code is supplied so that users can examine it or alter it as they wish for their own purposes. This can be an ideal way to get into programming, since much QL Public Domain software contains many useful programming techniques which can save you time and memory (although great care must be taken, since Public Domain libraries can also attract some of the worst programming I have ever seen).

Leisure

In the future I shall look at a few of the leisure programs available in the Public Domain, and where the source code is available. I will comment upon the usefulness of the program as a teaching aid in programming.

> *"Many programs have 'riders' attached to them by the authors – for instance, that they should not be used commercially."*

Public Domain adventures have a reputation for being simplistic and certainly not difficult. However, they can be a useful introduction to the world of goblins, dungeons and mind-bending problems which the commercial adventures rely upon. In the CGH Public Domain library, there are several adventures: *Fantasia, Treasure Hunt, Adventure Playtime, Ye-Classical Type, Werewolves & Wanderer, Haunted House, Aftermath of the Asimovian Disaster*, and *Farce*. Some of these programs are actually quite a challenge, with two of them written by Alan Pemberton, who is fast becoming one of the best adventure writers for the QL.

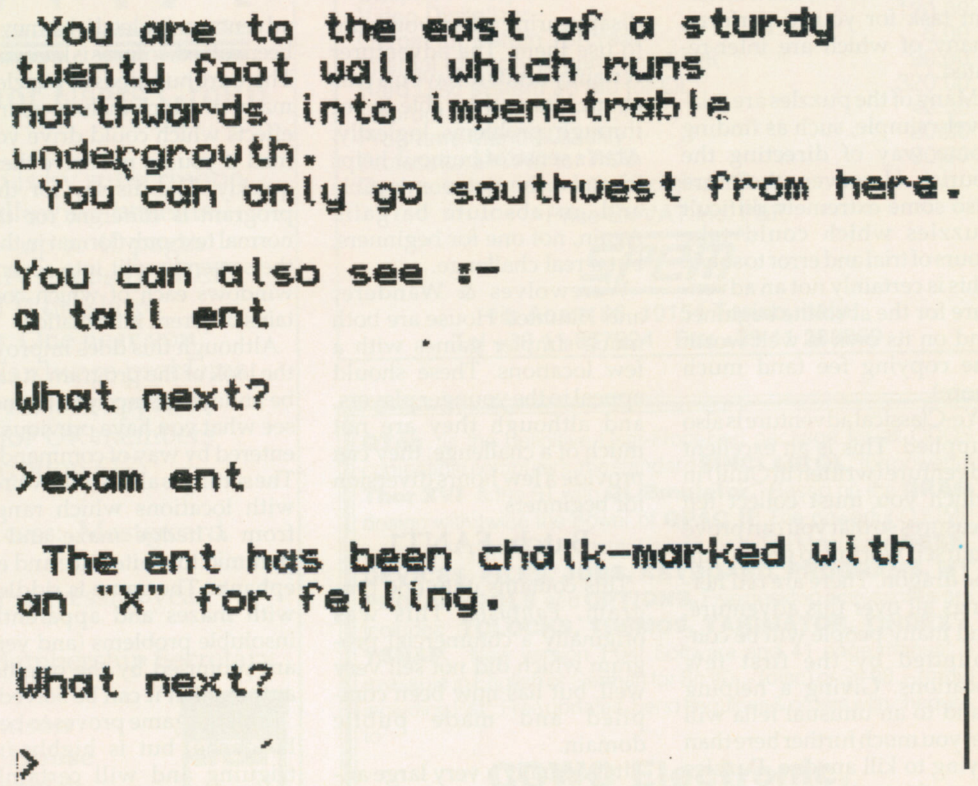
The CGH library is split into 'batches', each of which contains several programs; each batch costs £1.10 plus a microdrive. Luckily all of the adventures in the library will fit into a standard QL machine, so there are no excuses not to get involved by trying out the fun.

Batch: ADV1

This contains two adventures: an updated version of the original *Treasure Hunt* supplied free with the QL in the early days and the Usborne version of *Haunted House.*

Treasure Hunt has been compiled and has had many of its bugs removed. Although not the simplest adventure, it can be a good introduction to adventuring for the elder members of the family. It is set in a world besieged by dragons, orcs and soldiers. The idea of the game is simply to collect all the treasure and deposit it at a certain location. However, your task is hindered by the fantastic scenery (icy mountains and swamps), the soldiers who try to put you into prison and an inexplicable desire to catch, cook and eat a rabbit. *(Not a Hobbit?)* Although it is relatively easy to get killed in this one, it is nice to see that the same trick works every time when you need to get out of prison! This game is the original QL adventure.

Haunted House is definitely aimed at the younger members of the family. It is set inside an old house which is inhabited by ghouls and ghosts who try to stop you from escaping. There are only a very small number of locations, but the program can challenge young minds.

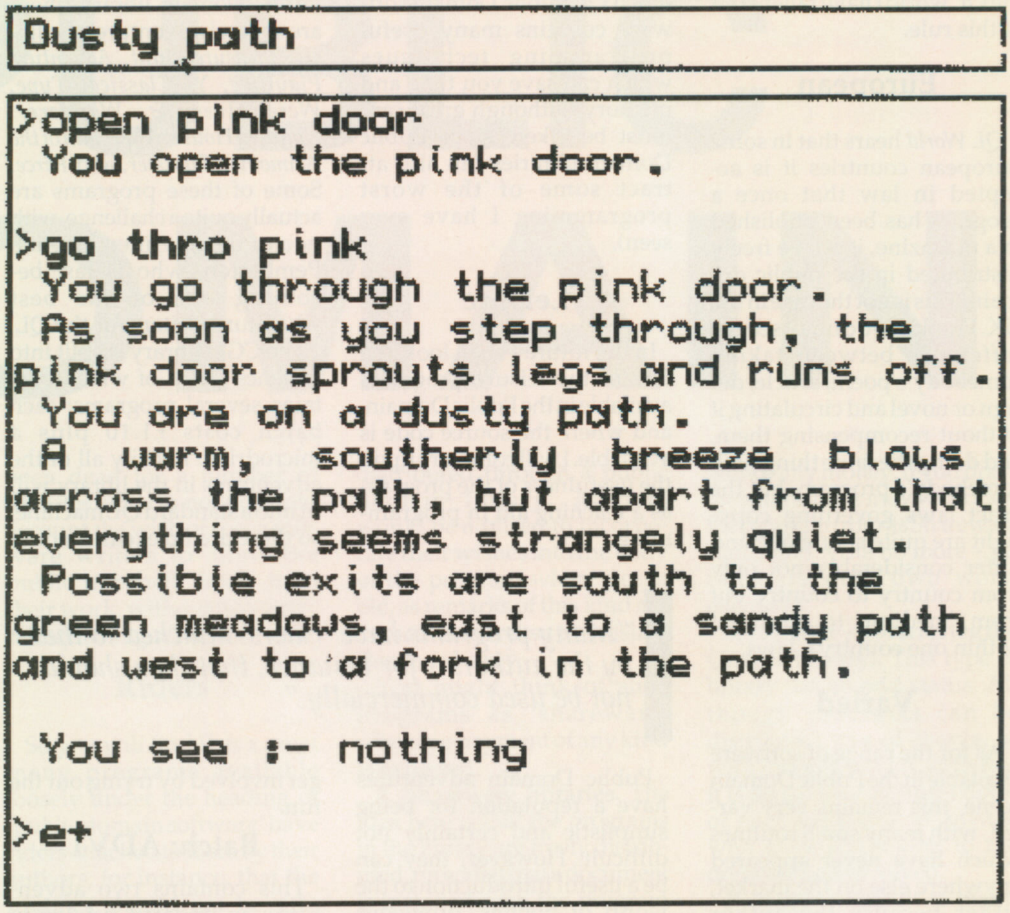


```
 You are to the east of a sturdy
twenty foot wall which runs
northwards into impenetrable
undergrowth.
 You can only go southwest from here.

You can also see :-
a tall ent

What next?

>exam ent

 The ent has been chalk-marked with
an "X" for felling.

What next?

>
```

YE CLASSICAL ADVENTURE

Batch: ADV2

This is probably the best value for money in the whole library. It contains Alan Pemberton's *Adventure Playtime* and *Ye-Classical,* plus a version of Haunted House translated from the French and *Werewolves & Wanderer*.



```
Dusty path

>open pink door
 You open the pink door.

>go thro pink
 You go through the pink door.
 As soon as you step through, the
pink door sprouts legs and runs off.
 You are on a dusty path.
 A warm, southerly breeze blows
across the path, but apart from that
everything seems strangely quiet.
 Possible exits are south to the
green meadows, east to a sandy path
and west to a fork in the path.

 You see :- nothing

>e+
```

Adventure Playtime

Adventure Playtime is only available as public domain on disk due to copyright problems (the microdrive version is available from the Microdrive Exchange). This version has been compiled and places you in the safety of your own home. Unable to get out of the front door and bored to tears, you read the note and learn that you are supposed to be out helping people solve their problems. While wondering what to do, you munch on the thoughtfully provided pizza, and inspiration soon hits you.

Once 'outside', you enter a world where even the most hardened adventurer would feel at home. There is a maze (and a maze bypass), lost tourists, librarians with large chests, magic doors, prisons and even a public convenience. Each character (yes, even the six foot tall rabbit) has a different task for you to perform, many of which are inter-related.

Many of the puzzles are relatively simple, such as finding some way of directing the tourist. However, there are also some extremely difficult puzzles which could take hours of trial and error to solve. This is certainly not an adventure for the absolute beginner and on its own is well worth the copying fee (and much more).

Ye-Classical adventure is also supplied. This is an excellent adventure (written in Quill) in which you must collect ten treasures so that you can prove yourself worthy to go and kill the dragon. There are red herrings all over this adventure, and many people will be confounded by the first few locations. Giving a helping hand to an unusual fella will get you much further here than trying to kill anyone. Puzzles range from simple anagrams to keys which have a habit of disappearing when you want to use them. The adventurer certainly needs to have his wits about him and be able to see through problems logically. Alan's sense of humour helps to make the game enjoyable and an absolute bargain. Again, not one for beginners, but a real challenge.

Werewolves & Wanderer and Haunted House are both much simpler games with a few locations. These should appeal to the younger players, and although they are not much of a challenge, they can provide a few hours diversion for beginners.

Batch: FANT1

This contains only one program, *Fantasia.* This was originally a commercial program which did not sell very well, but has now been compiled and made public domain.

It is actually a very large adventure which proves a real challenge to the most seasoned adventurer (indeed it has never been solved, so far as is known!). There are hundreds of puzzles, many of which include sound effects which could drive you mad if you do not solve them quickly. The display for this program is different for the normal text-only format in that the screen is split into several windows each of which contains different information.

Although this does improve the look of the program, it can be annoying since you cannot see what you have previously entered by way of commands. The setting is also a little weird, with locations which range from a hedge maze and a pyramid to white mice and elephants. The game is riddled with mazes and apparently insoluble problems (and yet I am promised by the original authors that it can be solved).

In all the game proves to be a headache, but is highly intriguing and will certainly provide more than a few days entertainment. I can highly recommend it, although it is really only for well-seasoned adventure players.

Batch: ASIM1

This contains only one program, *Aftermath of the Asimovian Adventure.* Batch ASIM2 is also available, which contains a few other non-essential loading screens. This is a rather simplistic adventure and is really suited for beginners. It may not contain many puzzles or locations, but it is certainly a good introduction to adventuring.

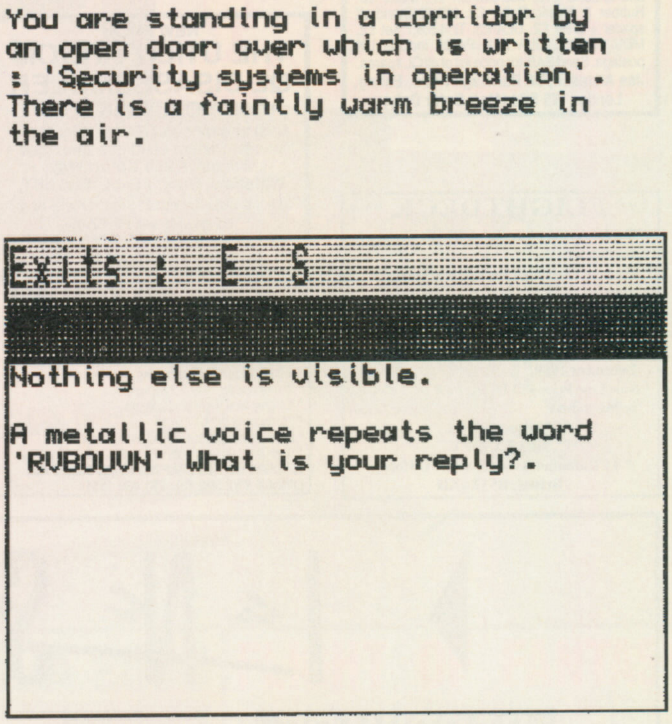
There is only one shareware adventure in the CGH library: *Farce Adventure.* This is set on a desert island, from which you must escape. There are several puzzles, but unfortunately the one with the raft has me completely stumped, since this is apparently a means to getting around the island to new locations away from the initial few. The game is text only and written on Quill. It is certainly not for beginners, but can prove quite a challenge, despite the lack of multiple locations. It may provide a few hours entertainment, but I am sorry to say that it is too frustrating to be addictive.

Source Code

The source code for Werewolves & Wanderer and Fantasia is available on batch ADV SOURC1. Neither of these programs is an excellent example of programming on the QL; indeed, Fantasia is next to impossible to follow, and certainly could not be adapted to another setting. However, the source for Fantasia does provide one useful feature of having access to the Basic program — it can be inspected for clues to solving the adventure!

Batch ADV SOURC2 contains the source code for Adventure Playtime (only available on disk) and Usborne's Haunted House. Again, Haunted House is not very good programming, since it is a direct copy from another computer (with a few minor alterations to get it to run on the QL). On the other hand, Adventure Playtime shows some excellent programming techniques and proves a very useful example of how to program an adventure in Basic.

Next month I shall look at the different Public Domain utilities available to help adventure programmers. If you have any details of any other public domain libraries then please contact me, so that I can look into the range of software which they provide.



```
You are standing in a corridor by
an open door over which is written
: 'Security systems in operation'.
There is a faintly warm breeze in
the air.

Exits : E S

Nothing else is visible.

A metallic voice repeats the word
'RVBOUVN' What is your reply?.
```

Fantasia
Adventure

INFORMATION
CGH Services, Cwm Gwen Hall,
Pencader, Dyfed, Cymru SA39 9HA. Tel
0559-384574.
£1.00 copying free for each microdrive-sized 'batch'.
QUANTA (membership secretary):
David Johnson, The Corner House,
Loxley, Warwick CV35 9JT.
£14.00 per annum membership, then
copying or programs free.

MICRO ADS

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).**

*** FLIGHTDECK ***

* The Jet Airliner Simulator the QL has been waiting for !!!
* Machine Code high resolution shaded graphics
* Accurate Database of 25 major UK airports and over 200 navigation beacons
* As Reviewed in "Sinclair QL World", December 1989
* New Low Price £15.00 inc P&P on 3.5" Disk or Microdrive

Cheque / PO to:
DELTASOFT
11 Dumaine Avenue, Stoke Gifford
Bristol, BS12 6XH

For Sale QL + Centronics + Joystick + fifty microdrive cartridges, £150. Call M.Kemp 0452 413774 evenings and weekends.

NEW FROM
THE UTILITY ROOM
QL DISASSEMBLER
ONLY £10

First in a range of S/W at affordable prices. Please send one formatted Microdrive to **5 Upton Way, Walshaw, Bury, Lancs, BL8 3BT.** Disks unavailable at the moment and up-grades will be £2.00

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/recorder
* 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**

C.G.H. SERVICES

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

Program	Type	Memory	flp —	mdv
Anelpum Quat	Text Adventure	128K	£8.00	£8.00
Assault and Battery	Shoot 'em up	128K	£10.00	£10.00
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	£8.00	£8.00
MacSporran's Lament	Illustrated Adventure	128K	£8.00	£8.00
Open Golf	Golf Simulator	384K	£10.00	—
Orbiting Stars	Astronomical Sim	128K	£8.00	£8.00
Personal Finance Manager	Home Accounts	128K	£8.00	£8.00
Polytext	Multi column Quill	128K	£16.00	£16.00
Quick Mandel Brot	Fractals	128K	£8.00	£8.00
Quiz Master	Quiz Game	128K	£8.00	£8.00
Return to Eden	Role-Play Adv	256K	£15.00	—
Sector X	Shoot 'em up	256K	£10.00	£10.00
Speedfreaks	Car Racing Game	128K	£10.00	£10.00
Starplod	Illustrated Adv	128K	£8.00	£8.00
The Gee Gee System	Horserace Prediction	128K	£8.00	£8.00
Uncle Loonie's Legacy	Illustrated Puzzle Adv	128K	£8.00	£8.00
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

QL Survivors Source Book Directory £2.00

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.

Sinclair QL. 768K TC, 3.5" drives+P/S fitted into PC style enclosure, keyboard, Tandata modem, Lx86 printer, monitor, Conqueror, MSDOS, lots more software. £600–ono (0423 500690. Patrick Smart, 34 Norwood Grove, Harrowgate, N. Yorks MG3 2XL. Daytime tel. 0845 577655

Qfile programs: MS-QLink v. 1.5 £12.00, DISCOPY v. 2.06 £7.50, £17.00 together. Please state 3.5"/720KB. Cheques payable to T.F. Leal. Address: Qfile, Apartado 2110, P-1103 Lisboa Codex, Portugal.

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
Via Aosta 16, 10154 Torino, Italy.
+39 11 857924 Fax +39 11280009
QL disk interfaces, ram, eprom holders.
TF Services
071 724 9053
Repairs, spares, peripherals.
TK Computerware
0303 812801
Peripherals.

SERVICES

Adman Services
0952 255895
Joe Atkinson
36 Ranelagh Road, London W5 5RJ
Roms, mdvs and all 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
0559384 574
Adventures. QLAF, DIY Toolkit.
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 Budget, Vision Mixer, QL Genealogist, Textidy, Spellbound and others.
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.
QFILE
Apartado 2110 p-1103 Lisboa Codex, MS-Link and Discopy. Portugal.
QL Supersoft
0256 475992
Spell, Numbers, 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 04/91-32, 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.

PERFECTION IS PROBABLY THE BEST WORD-PROCESSOR EVER CREATED... TRULY WYSIWYG, BOLD APPEARS AS **BOLD**, ITALIC AS *ITALIC*, UNDERLINE UNDERLINE, $^{\text{SUPERSCRIPTS}}$ $_{\text{SUBSCRIPTS}}$ AND MUCH MORE. THIS HAPPENS BOTH ON SCREEN AS WELL AS ON PAPER. PERFECTION IS EXCEPTIONALLY EASY TO USE, TEN TIMES FASTER THAN WHAT YOU'RE USED TO, SUPREMELY POWERFUL - PERFECTION TRANSFORMS THE QL.

PERFECTION COSTS £79.95 ONLY - SEE DIGITAL PRECISION'S 4-PAGE AD...

DIGITAL PRECISION LTD