AN ARGUS SPECIALIST PUBLICATION

JULY 1988 £2.95

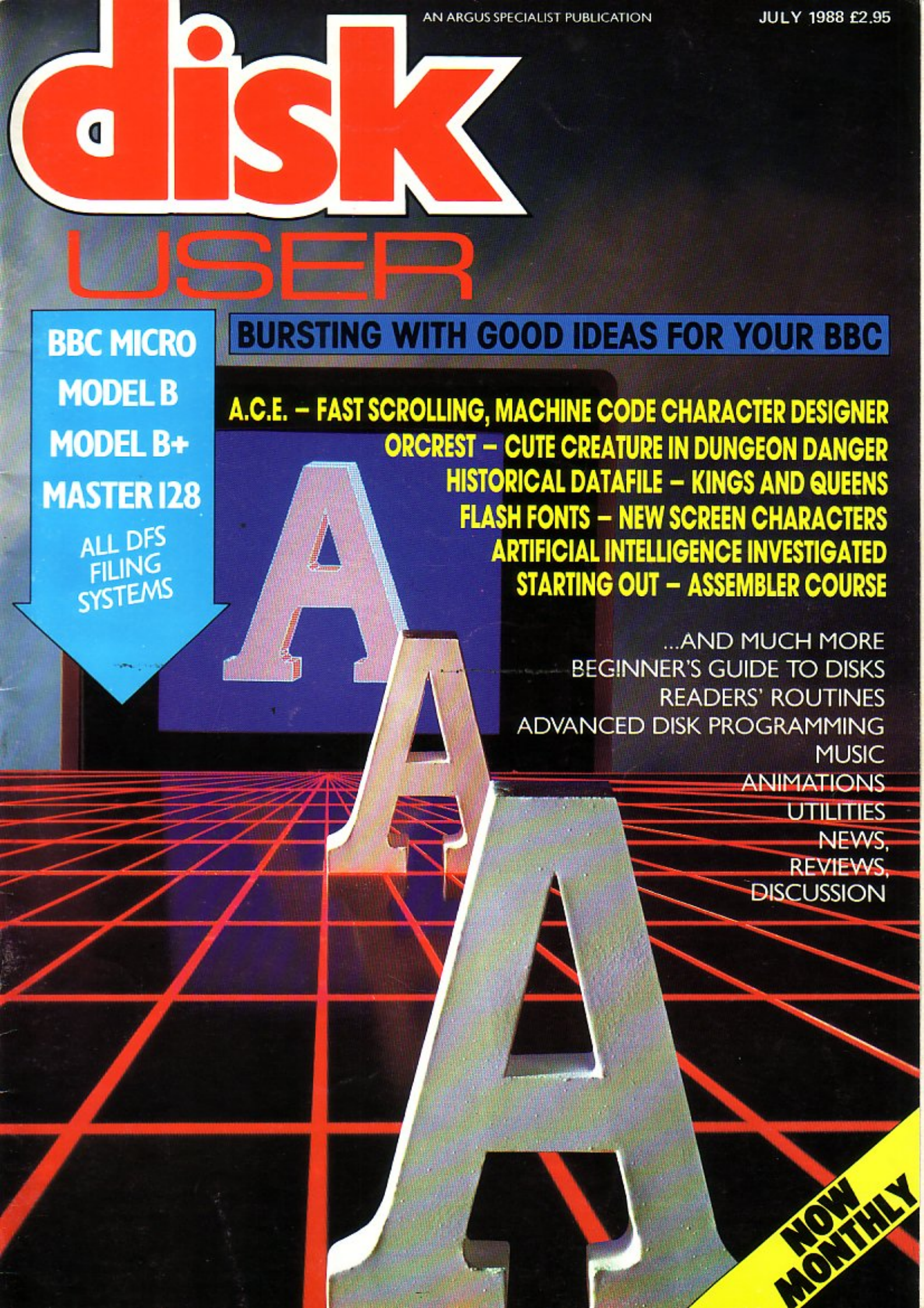
disk USER

BBC MICRO
MODEL B
MODEL B+
MASTER 128
ALL DFS FILING SYSTEMS

BURSTING WITH GOOD IDEAS FOR YOUR BBC

A.C.E. – FAST SCROLLING, MACHINE CODE CHARACTER DESIGNER
ORCREST – CUTE CREATURE IN DUNGEON DANGER
HISTORICAL DATAFILE – KINGS AND QUEENS
FLASH FONTS – NEW SCREEN CHARACTERS
ARTIFICIAL INTELLIGENCE INVESTIGATED
STARTING OUT – ASSEMBLER COURSE

...AND MUCH MORE
BEGINNER'S GUIDE TO DISKS
READERS' ROUTINES
ADVANCED DISK PROGRAMMING
MUSIC
ANIMATIONS
UTILITIES
NEWS,
REVIEWS,
DISCUSSION



NOW MONTHLY

By popular demand

disk USER

is

NOW MONTHLY

For INSTANT, ENTERTAINING and USEFUL software you only need to know about two keys

SHIFT and BREAK

- **Just BOOT for instant magazine action.**
- **Disk User – the unique interactive magazine on disk.**
- **FULL to brimming with BBC BASIC and machine code software**
- **AND graphics, demos and adverts on disk**

PLUS pages full of reviews, hints and tips and tutorial on how to use your disk system effectively.

In the 1987 Magazine Publishing Awards organised by The Publisher, Disk User for the BBC Micro won a second place certificate. The experienced panel of judges praised the "sheer value of the cover-mounted disk that formed part of the new publication's concept".

SUBSCRIBING MADE EASY
JUST TURN TO PAGE 25 . . .

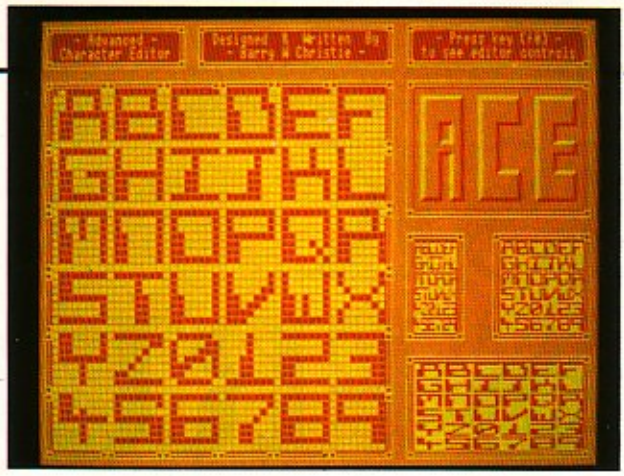
SUBSCRIBE NOW!

We've made it easy to subscribe to Disk User. You can choose from a 6 monthly or 12 monthly subscription. Twelve disks full of quality software PLUS colourful magazine for just £35.40.

	6 months	12 months
UK	£17.70	£35.40
Europe	£21.60	£42.00
Middle East	£21.70	£43.40
Far East	£23.20	£46.40
Rest of the World	£22.00 or USA $32.00	£44.00 or USA $64.00

Airmail rates given on request

disk USER



Number Nine
July 1988
Editor: Andrew Brown
Software Consultant: Matthew Fifield
Group Editor: Mark Webb
Advertisement Manager: Sarah Musgrave
Advertisement Copy Control: Francisca Perez

IN THE MAGAZINE

DISK DOCUMENTATION

ALL MODEL B, B+ and Master Series compatible

ELECTRON COMPATIBLE:

100% Electron compatible:

ORCREST
A.I. Investigated
Flash Fonts
Assembler Course
Memory Lister
Multitasking Music

Disk User is supplied on a 40 track disk format and can be run without conversion on a 40 track drive.
If you have 40/80 switchable drives then make sure the drive is switched to the 40 option.
For 80 track only drive owners, a conversion program is provided – see Disk Instructions
Most files can be copied to and used on ADFS systems

DISK INSTRUCTION

To get the best from your copy of *Disk User*, please carefully read the instructions below. We have made *Disk User* able to run on a very wide range of systems.

CATALOGUE

This month Disk User presents state of the art software. First there is the excellent **Advanced Character Editor.** This utility is the last word in designing characters. It has every feature you could think of and more!

Also on this month's disk is **Orcrest.** This is a very large and very difficult arcade adventure. We wish all you game players out there a lot of luck in completing it. We are assured by the author that it IS possible to finish.

Also starting this month is a series on machine code programming. Dave Stiles gives step by step guidance on how to write a powerful utility.

Disk Instructions

To get the best from your copy of *Disk User*, please carefully read the instructions below. We have made *Disk User* able to run on a very wide range of systems.

All Users

Please make a **Backup copy** and keep the original in a safe place with a Write-Protect tab on. You should use this copy as your working copy, as many of the programs need to write to the disk, and doing this will diminish the usefulness of the original, and may not be possible anyway due to the 31 file limit imposed by many DFSs.

New Users

If you are a new user **Don't Panic!** ,first find out whether you have 40 or 80 track drive(s) attached to your computer (ask someone knowledgeable if you don't know). Then go to your User guide or Welcome Manual and read the chapter on filing systems. In particular find out how to use the *COPY command. Next reread the section above **All Users**, and then go to the appropriate section dealing with your particular filing system and follow the instructions listed there.

Advanced Users

You do not need help to run Disk User, but do refer to the instructions for the filing system you are using, and **Don't forget to make a Back-up copy.**

40 Track Drive Systems

Disk User is supplied on a 40 track disk so will work on any 40 track BBC Micro system (at least, any that we know of!) straight away. Remember to make a working copy before use.

40/80 Switchable Drives

If you have this sort of drive, you can use *Disk User* straight away with the drive switched to the 40 track setting; don't forget to make a copy for normal use. However, you may wish to copy the disk on to 80 track format, in which case, with a single drive, you should follow the instructions for 80 track systems.

With two switchable drives, or one switchable drive set to 40 track and an 80 track drive (or even a 40 track drive and an 80 track drive), you can easily copy *Disk User* on to 80 tracks; put *Disk User* into drive 0 (40 tracks) and a blank formatted 80 track disk into drive 1 (80 tracks) and type:

COPY 0 1*.<RETURN>

Here <RETURN> means hitting the return key. You can set the boot option to drive one by typing:

***DRIVE 1**<RETURN> ***OPT 4 3** <RETURN>

80 Track Drives

Because *Disk User* is supplied as a 40 track disk, 80 track disk drives have to double-step through the disk. Probably the most convenient thing to do is to copy *Disk User* on to 80 track format. This can be done in

Disk User is published monthly on the third Friday of the month preceding cover date.
Advertising enquiries to Disk User, Number One Golden Square, London W1R 3AB.
☎ 01 437 0626.
Editorial enquiries to Disk User, 6C Belgic Square, Off Padholme Road, Peterborough PE1 IXF
☎ 0733 53355.
Contributions should include full source code and instructions file on disk. Payments are extremely competitive.
Distributed by SM Distribution Ltd., 6 Leigham Court Road, Streatham London SW16 2PG.
☎ 01 677 8111.
Telex 261643.
Fax 01 677 0136

Typesetting and artwork by Island Graphics
Cover design and design by the ASP Design Studio
Colour by Croxons
Printed by Chase Web, Devon
Duplication by DisCopy Labs, 20 Osyth Close, Brackmills, Northampton, NN4 0DY ☎ 0604 760261.
ISSN 0953-0606

two ways.

If your filing system allows double-stepping, we recomend using the system's own command. As a general rule, built-in 40-to-80 track converters should be used where available; the documentation for your filing system or utility ROM will give full instructions, and we give suggestions for some better-known systems further on.

Not all filing systems have facilities for double-stepping; Acorn's DFS is one such system. To overcome this, a program called CHANGE is supplied on the *Disk User* disk in a section which can be accessed by 80 track drives.

Using CHANGE

Insert *Disk User* into an 80-track drive (or 40/80 switched to 80-track) and type:
***CHANGE** <RETURN>
The program will prompt you to insert a pre-formatted blank 80 track disk when it is ready to write to it (you will have to swap back and forward between the two disks several times if you are using only one drive). Once this is completed, you can use the newly created 80-track version of *Disk User* and keep the original as the back-up.

Our suggestions on how to use *Disk User* on some popular DFSs now follow.

Master 128

This Acorn DFS has a software double stepping mode for a 80 track drive. Set it with the command
***DRIVE 0 40** <RETURN> and then hit <BREAK>
Disk User will then work without any need for conversion. However this may not allow writing to the disk in 40 track mode; in any case, you should make a working copy, so copy to a 80 track disk.

DFS on Master Compact

The DFS is supplied as an image on some versions of the Master Compact Welcome disk (or is available from Acorn on disk) and this may be used in conjunction with a 5¼ inch 40 track disk drive to run Disk User.Please note that we **cannot** at present supply *Disk User* on a 3½ inch disk (if there is sufficient demand, we may be able to in the future).

Opus DDOS/Challenger 3

If you are using the Opus DDOS disk filing system or Challenger 1.0/DDOS then issue the command
***4080 AUTO** <RETURN>
or
***ENABLE 40/80** <RETURN>
and Disk User will work without any need for conversion.

Challenger 3

If you have the later ROM version Challenger 1.1 then issue the command
***OPT 8,1** <RETURN>
to achieve the same result. Disk User will work effectively from the RAM disk. Use
***COPY 0 4 *.* *CONFIG 4=0 0=4**
***OPT 4 3**
to run from RAM disk

Solidisk DFS

With the Solidisk DFS 2.1 and 2.0 you can set a software double stepping mode for a 80 track drive with the command
***ENABLE 80** <RETURN>
Disk User will then work without any need for conversion.

Watford DFS

The Watford DFSs also have a software double stepping mode for an 80 track drive. Consult your manual for the appropriate FX call or command. Disk User will then work without any need for conversion.

Disk failure

If for any reason your copy of Disk User will not work on your system then please carefully re-read the instructions given above.
If you still experience problems then:
1. If you are a subscriber, return it to:
INFONET LTD, 5 River Park Estate, Berkhamsted, Herts HP4 1HL.
2. If you bought it from a newsagents, return it to:
Disk User Replacements (BBC), Diskopy Labs, 20 Osyth Close, Brack Mills, Northampton NN4 0DY
☎ 0604 760261.

Please use appropriate packaging, cardboard stiffener at least, when returning a disk. Do not send back your copy of the magazine. Only the disk please.

Editorial/Technical Enquiries

You can make telephone enquiries about *Disk User* on Wednesday and Thursday afternoons on 0733 53355 (please ask for *Disk User* Editorial). Enquiries in writing to the following address:
Disk User, 6C Belgic Square, Off Padholme Road, Peterborough PE1 IXF

Disk User JULY '88

All change – 40 track to 80 track convertor.
Files:-
CHANGE – Machine code file.
To use type
*RUN CHANGE <RETURN>
Disk User – Disk magazine title page animation (yes we know it goes in backwards!).
Author: Abbas Files:-
P.RUNDISC – BASIC program
A.DISC – Machine code file
Disk Menu – Easy selection of the software.
Author: Matthew Fifield Files:-
DUMENU – BASIC program
Orcrest – A large and challenging arcade adventure.
Author – Alan Taylor Files:-
X.ORCREST – Compressed data file
ORCLD – BASIC program
EXPAND – BASIC program
Theme Music – Groovey tune to get you in the mood.
Author: Ian Waugh Files:-
LOADER – BASIC program Theme – Data file
A.C.E. – The Advanced Character Editor.
Author: Barry Christie Files:-
EDITOR – BASIC program edcode – Machine code file example – Data file
A.I. Investigated – Your computer tells you what it's faults are.
Author: Frank Botto Files:-
EXPERT2 – BASIC program
Flash Fonts – Half size characters in any mode.
Author: Terry Blunt Files:-
COMPRES – BASIC program
Collectors Item Menu – The letter 'J' gets moving.
Author: Abbas Files:-
COLLECT – BASIC program
I.MENU – Data file J.Alpha – Data file
Assembler Course – Useful machine code routines made easy.
Author: Dave Stiles Files:-
HEXFND1 – BASIC program
Memory Lister – Reader's routine shows what lies in your machines memory.
Author: Kevin Mortimer Files:-
MEMLIST – BASIC Program
Multitasking Music – Listen to sweet music as you program.
Author: D. S. Peckett Files:-
MULTASK – BASIC and assembler
BASIC 1 Hipage – Part of the Superfont suite converted for use under BASIC 1.
Author: Dov Rosner (converted by M.F.) Files:-
HIPAGE2 – BASIC program
Kings and Queens – A datafile for TRACER users.
Author: C.R. Woodings Files:-
I.MONARCH – Data file for use with TRACER

DISK NEWS

Disk Media Roundup

Action, specialist suppliers of computer equipment, have announced the introduction of a new brand of floppy with the offer of *Eleven-for-Ten*. Disk User readers need not get too hot under the collar however, as the offer is only on until June the fourth, and to qualify you have to buy **Ten Boxes** of Maxell RD series Diskettes.

The new disk is claimed to have a tougher and smoother surface, which is designed to reduce wear and increase resistance to dust and dirt. The RD in the product designation appears to be short for *Reliable & Durable* by the way. Typical price for a box of 5.25in disks capable of being formatted to 1.2 Mb is £22.37.

Of slightly more interest is the **Disk Porter** a nifty little device for carrying up to twenty 5.25in disks in a slim, 1.5in thick case. Each disk is visible when in the case, and the lid folds back into a stand when open. Price is £19.50 for a one off, although a discount to £14.49 is offered if goods worth £100 are bought.

Action also offer a free 288 page catalogue of their whole range of computer and office related goods which is available for the price of an 0800 call, nothing.

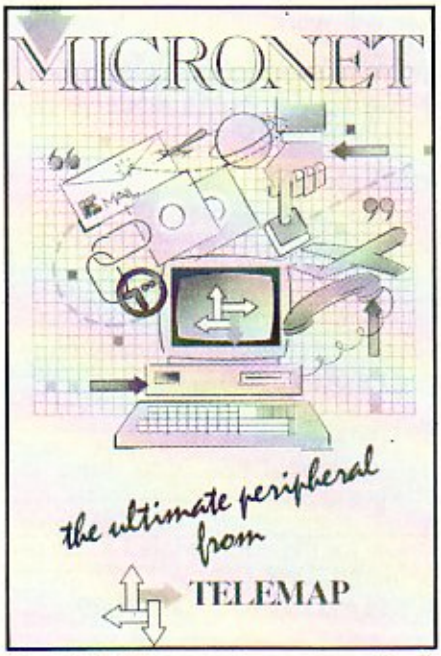
Action Computer Supplies, Abercorn Commercial Centre, Manor Farm Road, Wembley, Middx, HA0 1WL. ☎ 0800 333 333

On the lighter side, Fuji Film (UK) Ltd have produced an informative little booklet which tells the novice disk user all they need to know about how those funny square things work. The booklet, about the size and shape of a 5.25in disk is packed with illustrations, and even includes a glossary at the back. It is available without charge, and would be particularly useful for teachers introducing kids to the subject for the first time.

Fuji Photo Film (UK) Ltd, Fuji Film House, 125 Finchley Road, London NW3 6JH. ☎ 01 586 5900

Arc and Master Sales Boost

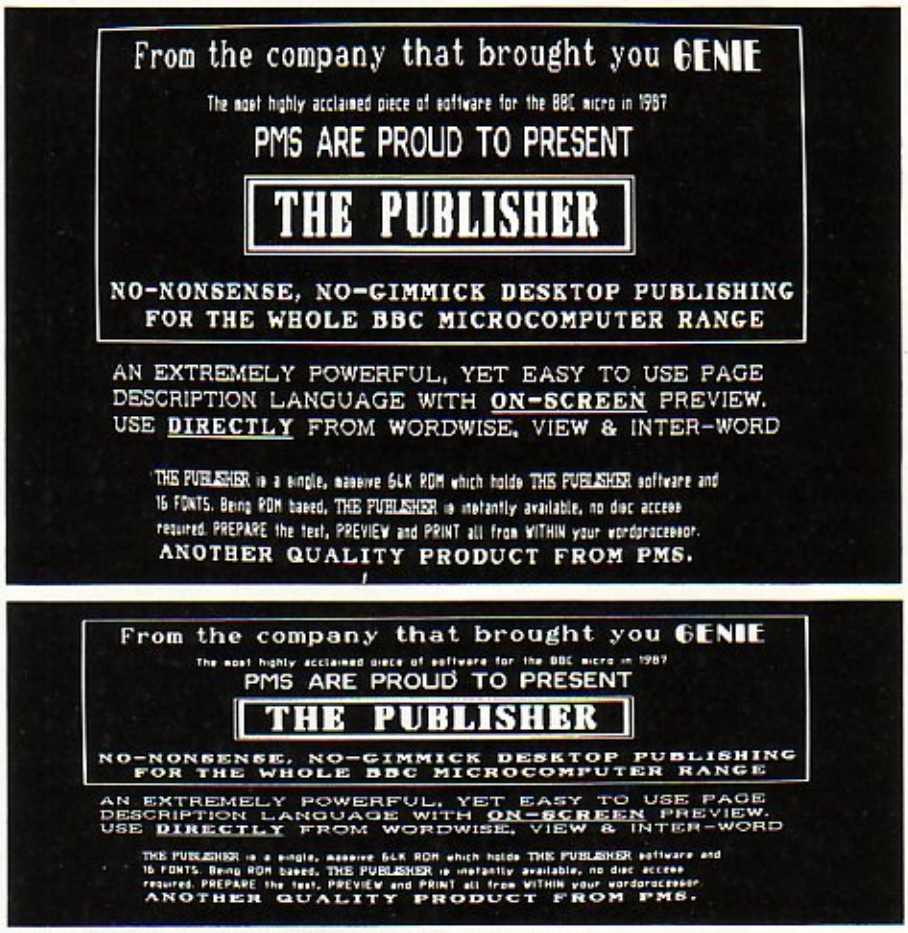
There have been a succession of local authorities placing orders for 32 bit Acorn Archimedes systems in recent months. These include Derbyshire, West Glamorgan and Cambridgeshire. The main reason for choosing the Acorn machine over other competing systems would appear to be the availability of a large pre-existing software base for its predecessors the BBC B, and Master 128 computers. Indeed Acorn expect to sell many tens of thousands of Master series machines this year, so continuing the remarkable longevity of the original concept.



A new look Micronet means an improved online BBC magazine.

Mouse-Based Painting Package

Nidd Valley Micro Products Ltd have produced an upgrade to their popular graphics package *Illustrator*, called *Illustrator Paintbox*. The program can be used independently as a stand-alone mouse operated graph-



•PREVIEW on PMS' newly launched Publisher screen and printer driver.

ics design package. *Paintbox* is a disk based system offering features comparable to high priced ROM packages.

Operation is by means of the mouse and pull down menus. There are features to create up to twenty two shades of colour allowing attractive effects to be created. Routines for dumping to either an Epson or Integrex printer are provided.

Availability is across the whole BBC B to Master Compact range, with only the Electron not supported. Prices range from £14.95 for the software only, to £59.90 for a comprehensive package including *Illustrator* and *Digimouse*.

For more details contact Nidd Valley Micro Products Ltd, Thorp Arch Trading Estate, Wetherby, West Yorks. ☎ 0937 844661

Learner Software

Some interesting new releases for sixth form students this month from the television companies. BBC Soft have brought out *Help Yourself! Maths*, dealing with directed numbers. The software is attractively programmed by Tom de Havas, who programmed the Design Technology series.

Help Yourself is a computer synchronised audio (CSA) package. This format, which combines audio cassette tape and computer software, is uniquely used by BBC Education.

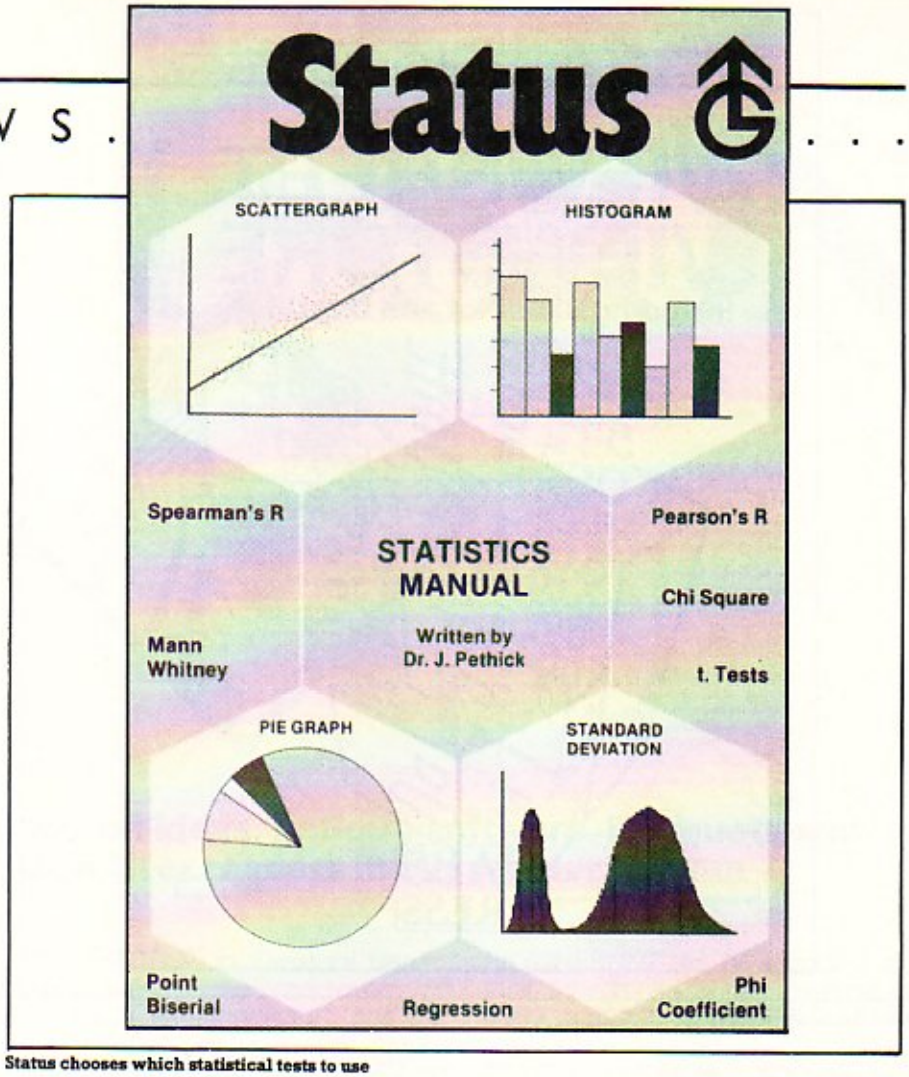
The audio tape is formed from a series of radio programmes and the announcer prompts the student to move on through the interactive computer program. Although it is claimed that it is also possible to continue through the program at your own speed independently of the audio tape, we found this difficult in practise.

BBC Soft, 80 Wood Lane, London W12 0TT.

On the other channel, GSN Software have programmed Status, a three disk statistical package for earth, social and biological sciences. Status accepts data from the keyboard or imported from GSN's KEY database program. Status will present scattergraphs, piecharts, histograms and standard deviation.

The statistical tests available include Mann Whitney, Spearman's R, Chi Square, t. Tests, Regression and Phi Coefficient. Uniquely, Status will choose the appropriate tests to make. The students only have to provide the statistical question and type of measurement involved.

Mercury Educational Products, Mercury Music Co Ltd, PO Box 194, Sevenoaks, Kent, TN15 8TZ.



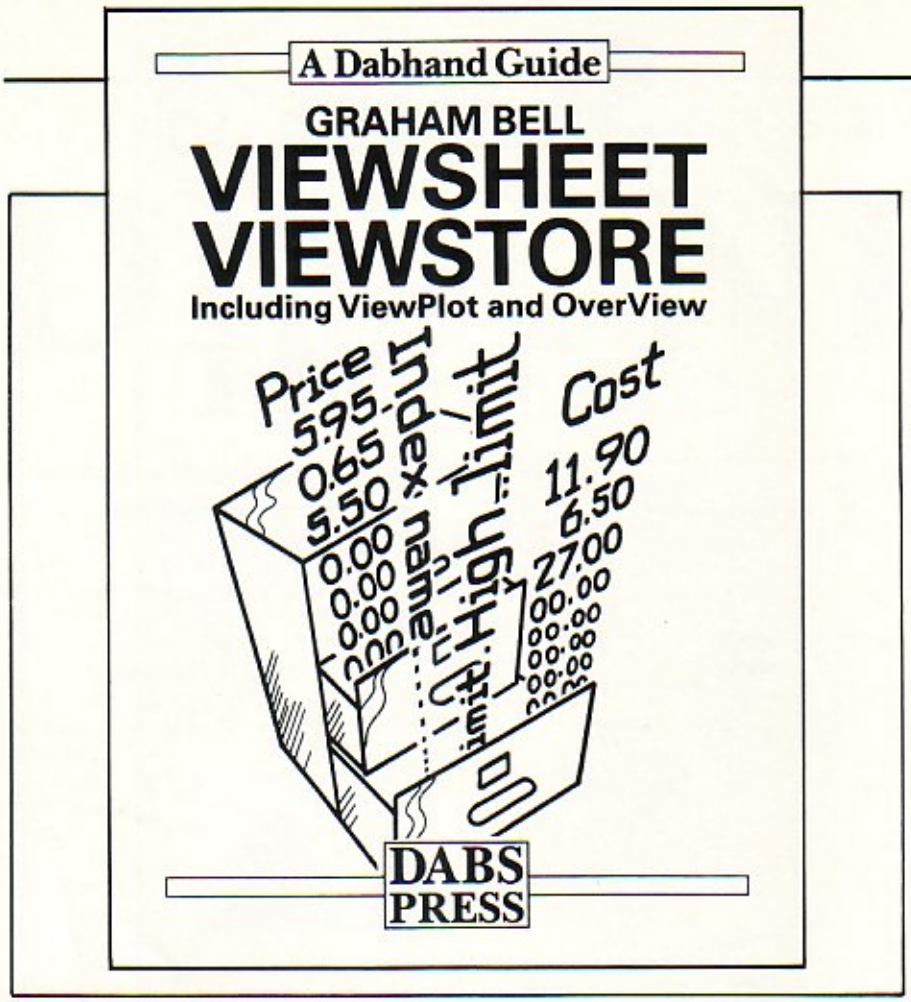
Status chooses which statistical tests to use

View Family Software

Users of Viewsheet and Viewstore will be interested in the new Dabhand Guide to these two popular products. Disk User readers will appreciate the availability of a support disk for the book costing just £7.95. The disk contains example spreadsheet and database files and batch files for automatically configuring the View family programs.

Printer owners can achieve a sideways printout of large spreadsheets. A set of utilities for Viewstore includes JOIN, PURGE and MULTI-MACRO and there are printer drivers for both applications. If you are familiar with Viewsheet and View-

Another kind of disc – Oxford University Press' debut on AIV.



View family users take note

designed to help students prepare for their RSM grades exams. Price around £15.00. For those aiming at Grade 5 Theory there is a practice program called Basic Theory of Music, price £9.00.

NotePerfect, 97 Valence Road, Lewes, BN7 1SJ. ☎ 0273 473135.
Perfect Fourth, 11 Otters Brook, Buckingham, Bucks, MK18 7EB ☎ 0442 76311.
Basic Theory, Ted Kirk, 33 Humber Crescent, Sutton Learch, St.Helens, Merseyside, WA9 4HD.

Anyone interested in making music with the help of their BBC Micro should make sure they get hold of a June copy of A&B Computing. It features reviews of Perfect Fourth, Music 2000, UMI MIDI hardware as well as lots of advice on connecting your BBC Micro to musical equipment of all kinds. It goes beyond the Beeb to talk about suitable keyboards, synthesisers, sequencers and drum machines from Casio, Yamaha, Cheetah, Roland and other household names.

You can order a copy of June A&B Computing by ringing 0442 776661.

store already then the disk and instruction booklet is good value alone.

Dabs Press, 76 Gardner Road, Preswich, Manchester, M25 7HU. ☎ 061 773 2413

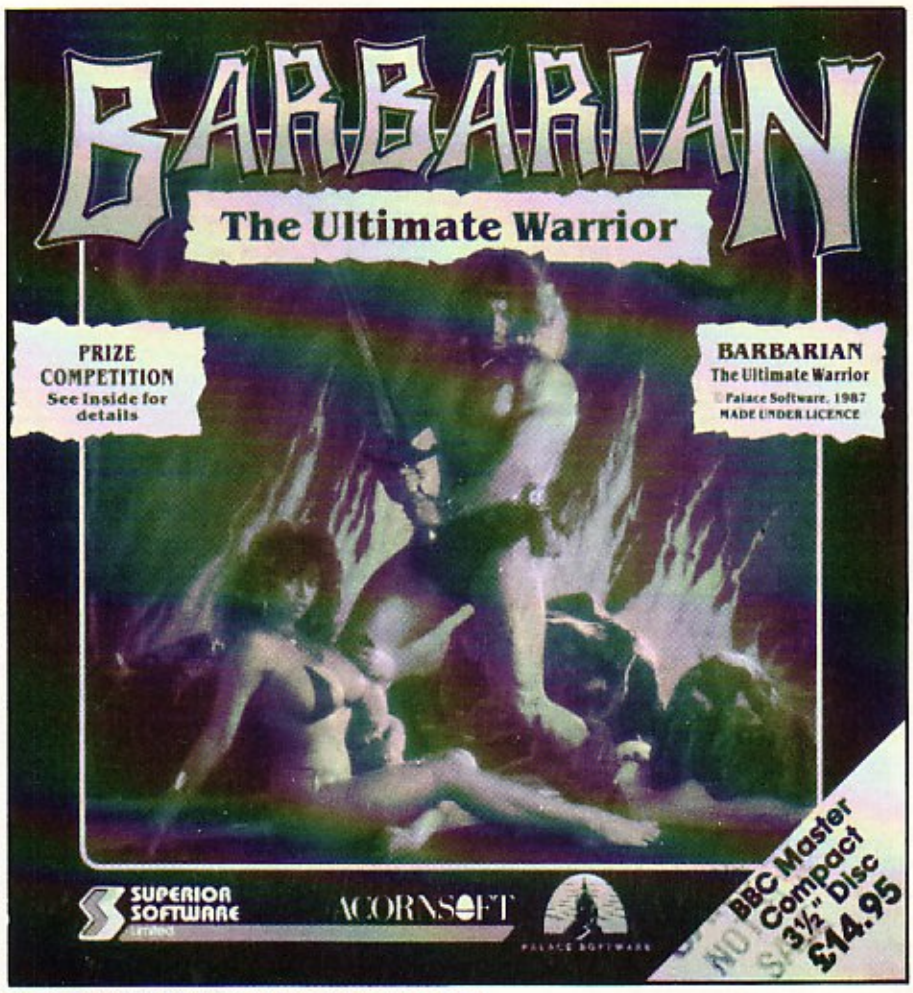
News Roundup

● Acorn lost around three million pounds in its last financial year. Olivetti however seem to be standing by Acorn and intend to use Acorn technology in a RISC based workstation. On the other hand there is disquiet that the Master Compact, after early successes in Italy, has been passed over for Olivetti's own PC1, MSDOS based 3.5" disk drive computer. The PC1 has just arrived in Britain via Dixons.

● The Healthline health education database is now available in disk form, having been "online" (01-9864360 1200/75 viewdata) now for some years. The information is aimed at Human Biology GCSE and Personal and Social Education. Prices start at £12.95.

Healthdata, 21 Vicars Close, London E9 7HT.

● Budding musicians may like to improve their technique with the help of the BBC and some new software. NotePerfect, priced at £9.95, provides over 200 different Scales and Studies on one disk. Perfect Fourth Software is a series



The controversial Barbarian. No "cover-up" in Disk User.

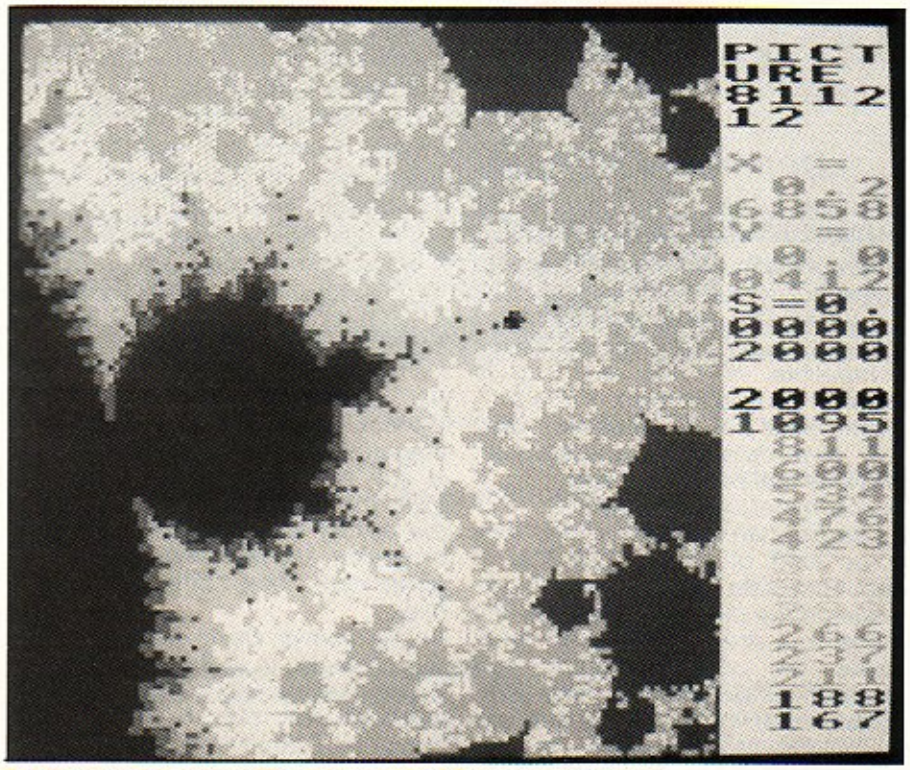
DISCUSSION

Generous donations, add-on ideas, antique software. Bouquets and brickbats from Disk User readers in our regular column

Three Cheers for Our Readers

I would like to thank Disk User readers for their generosity. following the publication of a subset of PSYCHEBROT in the April issue I have already sent out over 100 copies of the "careware", and forwarded over £250 to the mental health charity, MIND. Requests for copies are still coming in, and it is still not too late to get one (send in your £2.00 plus donation to me at Codil Language Systems, 33 Buckingham Road, Tring, HP23 4HG. There is no doubt that the concept of careware is creating a lot of interest, and I would like to encourage those who have already got PSYCHEBROT to give copies to their friends, encouraging them to make a donation to MIND. Chris Reynolds.

It's really pleasing, although not surprising to see what a caring bunch all you Disk User readers are. Just keep those donations rolling in. Incidentally, to whet your appetite, Dr Reynolds notes that an Archimedes version of Psychebrot might be on the cards.



Disk User 6 carried a subset of Psychebrot – a Mandelbrot generator program

Addproc Add On

I have been a Disk User reader since issue one, and if I benefitted from only one piece of software from each issue I would still consider it excellent value. I have in the past made alterations to several programs in the belief that they were being improved or enhanced, but these alterations were so simple I felt I would be insulting your readers intelligence were I to submit them. However on reflection your magasine is produced in a manner which assumes limited ability of its readers, and as a result speaks in simple and clear terms. Because of this I believe I have improved the Procedure Library Manager from issue seven. I now submit it to you as a simple readers routine. The enhancement allows the selection of any drive from either page one or two of the

*expanded program Addproc. I believe that while most BBC Micro owners now have disk drives, many have opted for the maximum configuration of twin 40/80 switchable double sided drives. For owners with these kind of drives my routine allows the expansion of the library (irrespective of the use of *Swap) by a factor of four. Adjustment to lesser configurations is not difficult.*
LOAD and LIST Addproc
Amend lines 510 and 520 as shown below, inserting the drive No to indicate the drive on which the program is to be saved (0 in this case)

```
510 PRINTTAB(0,0)CHR$(129);
CHR$(157);CHR$(141);"ADD A
PROCEDURE Drive 0. (0-3)?"
```

Now alter lines 380 and 690 as shown and add the new lines 381,382,691 and 692.

```
380 file$=GET$:B=ASCfile$:IF
B<48 OR B>51 GOTO 390
381 OSCLI("*DRIVE ")+file$
382 CHAIN "ADDPROC"
690 ans$=GET$:B=ASCans$:IF
B<48 OR B>51 GOTO 700
691 OSCLI("*DRIVE ")+ans$
692 CHAIN "ADDPROC"
```

Now save "ADDPROC" back to disk change to the next drive in the logical sequence (0,1,2,3) and list again lines 510 and 520. Copy these lines, changing only the drive number in order to reflect the drive to which you now save this copy of "ADDPROC". When all this is done you will have up to four copies of Addproc, one on each drive, and the individual programs, when run will indicate in the heading which particular drive has been selected. On !BOOT it will of course be zero.
G W Hetherington, Newcastle
Thanks for the listings Mr Hetherington, you must have some fairly enormous programming projects on to require all four sides of a disk drive. Thanks also to Messrs Savvides, and Shaw for suggesting improvements to the Animations Menu, we have now incorporated these into the program. I must point out that we would like our readers to try and make their listings BASIC I compatible. The listing above, for example is not compatible.

BASIC I (again)

Before our BASIC I brethren write in to complain that not absolutely everything in Disk User is BASIC I compatible I would like to explain that we do test everything, its just that the odd one slips through, however we do understand the frustration ensuing when a program does not run through being incompatible. We must therefore say sorry to 'frustrated' of Romford alias Mr C Johnson. For him and other BASIC I users there is a new version of HIPAGE in this issue, together with a listing of those EQU statements, and how to simulate their use in BASIC I.

Problem Corner

This is going to be an occasional series where we print cries for help from our readers. These are problems the editorial staff cannot help with, usually exotic hardware, or rare software. On these occasions we are appealing to knowledgeable readers to share their wisdom with someone in need of assistance. All you have to do is write to Disk User in the normal way, and we will either publish the letters, or forward them to the right person as appropriate.

Superfont Compatibility

I Thought that the Superfont printer program in issue 7 looked very interesting. However my printer an (Acorn) Olivetti JP101 is not Epson compatible. Is there any way in which a revised version of the program could be produced, or could you give details of any program changes that could be made.
Malcolm Matthews, Redhill
As far as I can recall the JP101 was some kind of ink-jet, but that is about the state of the editorial knowledge on the subject. Is there is any reader out there who knows what codes the JP101 accepts, and whether it is sufficiently Epson compatible to drive Superfont, then we would be most grateful for any suggestions to help Mr Matthews.

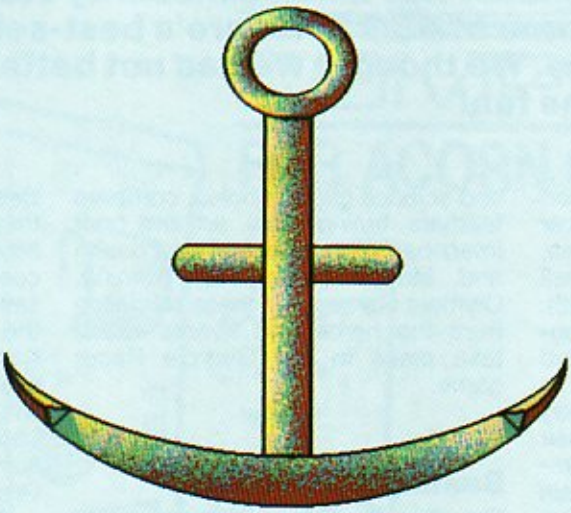
That ole £ sign blues

I have bought a copy of Disk User each issue so far, largely out of curiosity, but also because it represents a good way of getting hold of some useful utilities. I also have few copies of the earlier tape based Model B Computing. One of these issues contained a database program called Casfile. Now transferred to disk it has been invaluable in keeping various lists for me, unfortunately it does not seem to be able to print £ and # signs. I also would like to be able to utilise the function keys more fully, ie using SHIFT/CTRL etc. Thanks for an interesting magasine. Michael J Cowie, London
That's certainly an antique piece of software you have there Mr Cowie, why not take the plunge and start using something new like Tracer in issue eight of this magasine. Seriously though, you do get used to a piece of software, like a comfortable pair of old slippers. I've still got an ancient beast of a CPM wordprocessor that I haul out from time to time when the mood takes me. What you don't tell us is the kind of printer you possess, and in fact this can often have more bearing on the matter than the type of computer. If there is any reader out there with a foolproof way of doing this we would certainly like to hear from them.

Palace of Magic Cheats

I am writing to complain about the April 1988 issue of Disk User, where the complete solution to Palace of Magic was printed. the competition is still open so that your solution gives the cheats a chance of winning the prizes. It is only fair that cheat programs and solutions should be given after the competitions are over.
Antony Leaver, Bexleyheath
Our apologies to all those concerned. A large gentleman with a chainsaw has been sent round to Mr Reeder's house so I don't think he will do it again.

ANCHOR
it NEVER lets go



The complete memory resident programmer's assistant Calculator, Notepad, Monitor, ASCII Table and more Instantly available – just press <CAPS LOCK><SHIFT LOCK> together No more irritating disc swapping. Integrated programming environment. Anchor for BBC Master/B+128, or BBC B/Solidisk SWR. Powerful debugging/memory management facilities. Pop up/Pop down, over your favourite applications. This is it, the program you always promised yourself.

Have you ever wanted a reliable helper, someone to sit by your side night and day. Someone who never complains or goes to the pub for a drink, or asks for money?. Well here is the answer to your dreams. Now you can do all those things that were just too much bother, and still be better organised. Now when the telephone rings in the middle of your latest meisterwerk all it takes is a second to jot a note down, or do a quick calculation, and hey presto you can get back to work straight away.

Fully Documented

Anchor comes complete with a desktop-published, and laser printed manual that will help you get the best out of Anchor's more powerful features. Complete with worked tutorial examples for beginners, the manual is a valuable reference work in itself.

Availability

Anchor is available now in both Master/B+128, and BBC B with Solidisk SWR versions. Please follow instructions at the foot of this page for ordering.

Please send me a copy of Anchor (please tick box of choice).

☐ for the BBC Master 128/B+ 128 (DB92)

☐ for the BBC Model with Solidisk sideways RAM (DB93)

I would like Anchor on:
(please tick box of choce)

☐ 40 track

☐ 80 track

I enclose a cheque/postal order made payable to:
Argus Specialist Publications Ltd for £15.00

Name: .....

Address: .....

.....

.....

..... Postcode: .....

Total enclosed: £.....

Send your order to Reader Services, 9 Hall Road, Maylands Wood Estate, Hemel Hempstead, Herts HP2 7BH. Visa and Access accepted. ☎ 0442 41221.

DISK DATA

Geordie Racer has taken schools by storm and become one of BBC Software's best-selling programs. We thought we had not better miss out on the fun!

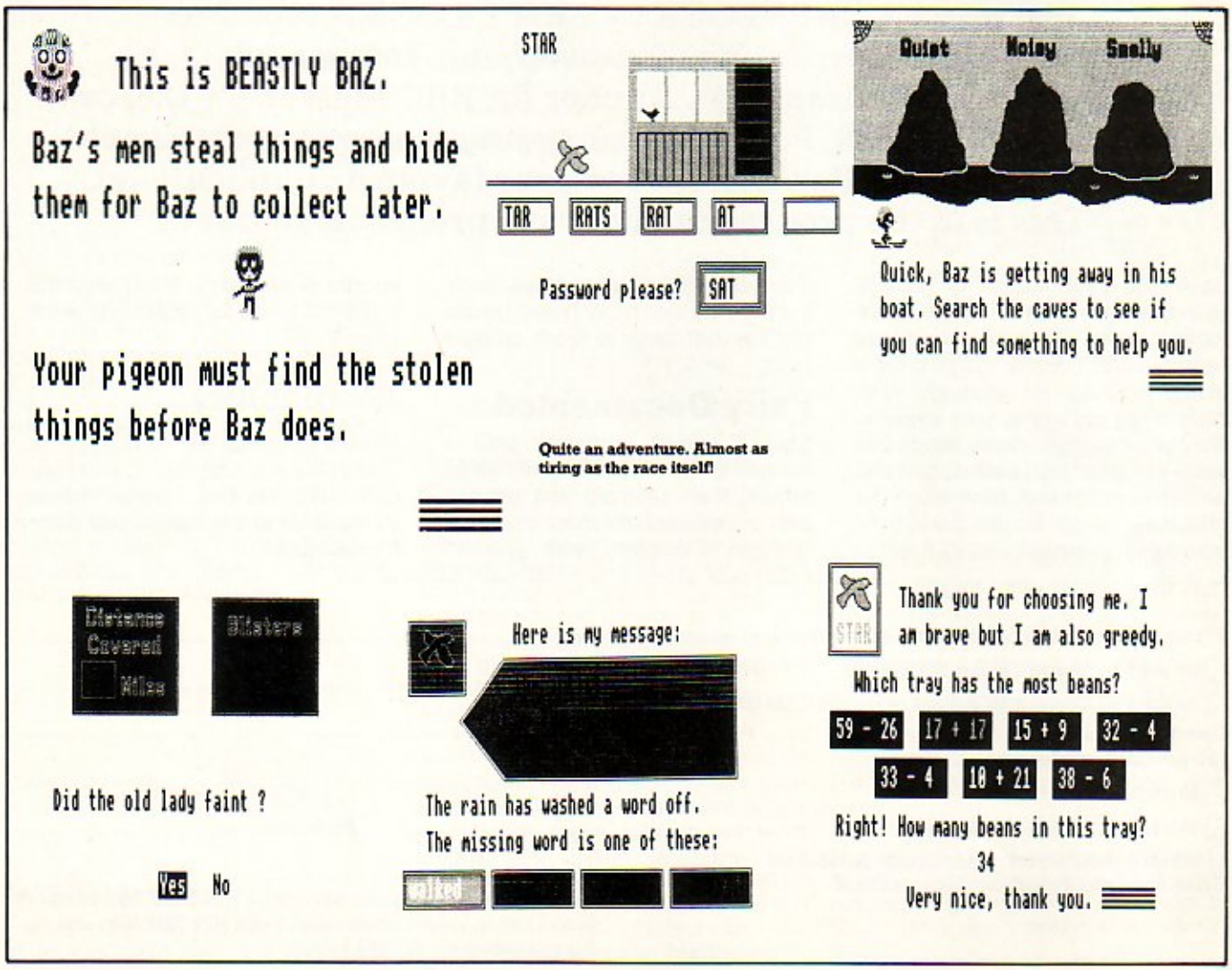
The popular BBC Schools Television series *Look & Read:Geordie Racer* has spawned a computer game, designed by Christopher Russell and programmed by Peter Smith. The game is a sequence of adventures, challenges and puzzles and it's great fun to play.

Any parent could happily play through the entire game with their children. The single disk is accompanied by a booklet whose main purpose is to supply follow-up information and lots of ideas for projects based on the game. These include follow-up projects for maths and science (pigeon holes, compare feathers, timing runs), art and craft (marzipan pigeon recipe) and health and fitness (history of running, Olympic Games). All these follow on from the heros and events which take place in the Geordie Racer game.

Beating Beastly Baz

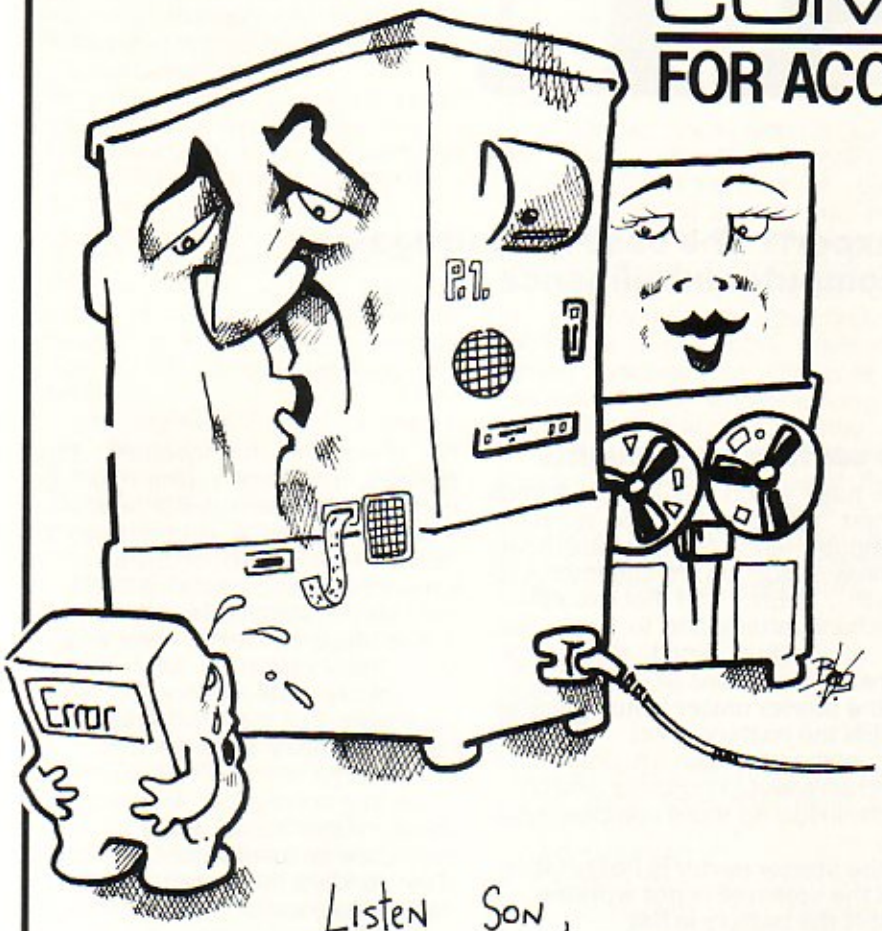
The Geordie Racer software combines arcade quality movement of characters, easy to read text and control via menu options. The puzzles are aimed at young children and they are given plenty of chance to have another go if they don't succeed first time round. There is no save game option but you can enter the sequence of events at any stage from the first menu. The game is split into two parts, firstly finding out what you have to do and getting to know your pigeon, secondly pursuing the treasure and beastly Baz. Great fun.

The only problem is that Geordie Racer shows up the limits of a 100K disk. It's a shame it had to end. Follow up please BBC Soft?



SUBSCRIBE NOW

A&B COMPUTING
FOR ACORN COMPUTERS



12 ISSUES U.K. £18.00
12 ISSUES EUROPE £27.30
12 ISSUES MIDDLE EAST £27.60
12 ISSUES FAR EAST £31.10
12 ISSUES REST OF THE WORLD
£28.30 OR USA $42.00
AIRMAIL RATES GIVEN ON REQUEST

Listen Son,

You may be a 2nd Generation Computer but just remember that compared to a human you've only got the brain-power of a HADDOCK.—

Please begin my subscription(s) to A & B COMPUTING with the ..... issue
I enclose my cheque/postal order for £ ..... made payable to Argus Specialist Publications Ltd
or debit £ ..... from my Access/Barclaycard No. ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐
valid from ..... to .....
NAME (Mr/Mrs/Miss) .....
ADDRESS .....
.....
Postcode .....
Signature .....
Date .....

Send your remittance to:
INFOMET LTD
5 RIVER PARK ESTATE
BERKHAMSTED
HERTS. HP4 1HL

Access

BARCLAYCARD VISA

WELCOME TO AI?

Intelligent expert? The case for and against computer intelligence

Expert systems, as intelligent programs are a bit like flowcharts, you either believe in them or you don't. Yes, it's that ongoing debate about the existence of AI – which is getting a bit boring to be quite frank. With the believers proudly announcing latest advancements, saying things like, 'we've got a robot that can map out a room and sweep the floor', then the sceptics reply, 'well that doesn't really require intelligence' and so on, and so on. It seems to me, the sceptics change the definition of the word intelligence following each and every advancement in AI, whilst believers, in an equally wangling vain, cajole us into regarding intelligence as being something quite specific to suit their particular needs. This seemingly eternal bickering perhaps leads us to regard – and not without good reason I think – the whole AI debate as being one great farce.

Be realistic

To many, the very acronym AI, refers to intelligence in the loosest possible terms. For a current program designated as being intelligent often displays nothing more than a few unremarkable AI features, which all to often underwhelm the user, and are duly taken for granted. Take for example a current wordprocessor which is said to be intelligent, it will merely check spelling as you type, and perhaps highlight errors in real-time. The typical AI wordprocessor may not even offer a correct spelling – is this the sum total of artificial intelligence for the average personal computer user?

To capture our thoughts

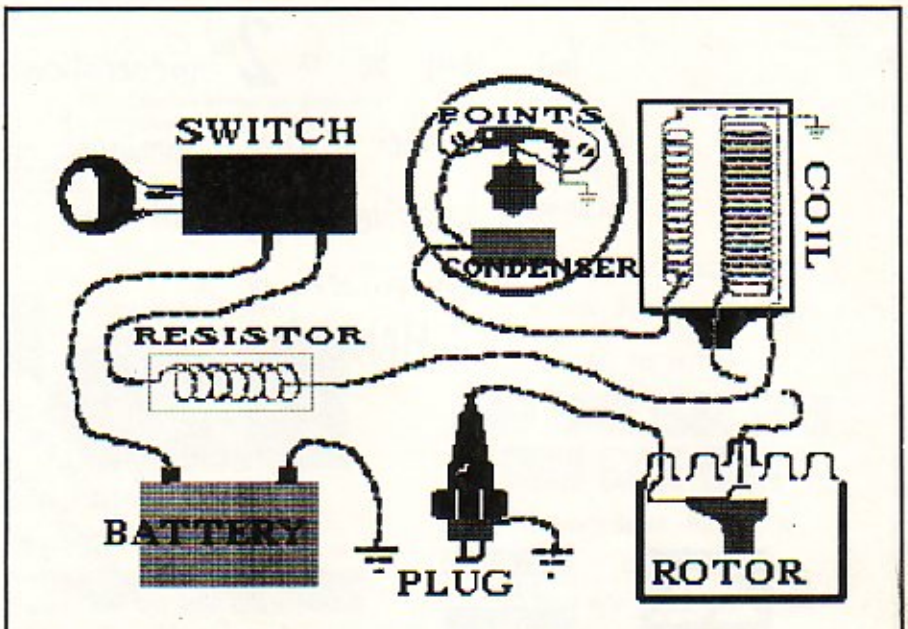
The rules which any kind of human expert will observe are in basic computer terms fundamental, being entirely made up of statements of the IF... THEN... sort. For example, a mechanic attempting to find a fault on a car that won't start, might consider such rules as –

IF the starter motor is not rotating
THEN the battery is flat

Or perhaps more realistic, the mechanic would consider additional facts, involving more complex rules, like –

IF the starter motor is not rotating
OR the solenoid is not working
THEN the battery is flat

No doubt, as the mechanic progresses, many such rules might be considered. Some rules or heuristics – which are often simple rules of thumb – will be wholly based upon experience, and quite difficult to emulate in computer terms, as many a human expert cannot freely explain their methods in a way that is compatible with a computational approach. For the mechanic might not sequentially consider a number of rules, but as if by magic, merely arrive at a conclusion. Arguably, it is this inimitable human ability to seemingly draw an instant conclusion that often renders the expert system as being incomparable with our thought



Expert systems apply "rules" to solve a problem. Fault diagnosis is a common application

processes, and ultimately deemed 'unintelligent'.

If we were to directly emulate the beginnings of the mechanic's thinking in this instance, we would no doubt hastily begin by programming the knowledge base with the appropriate IF... THEN... rules, and cataloguing their 'facts' and 'conclusions' within data statements, giving something like this –

1000 DATA "starter motor not rotating","battery flat" 1010 DATA "starter motor noisy","loose starter bolts" 1020 ....

In turn we might read the data into an appropriate two-dimensional array like Knowledge$(n,n) for example, storing facts in the Knowledge$(1,n) column and conclusions in the Knowledge$(2,n) column. Subsequently, comparing an inputed string with the Knowledge$(1,n) elements, would eventually allow us to print an appropriate conclusion – namely, one of the Knowledge$(2,n) elements.

We may even find it necessary to ask the operator some relevant questions of one kind or another. Though crude, the program would work, but really the inference engine would be nothing more than a simple 'compare' routine incessantly sifting through the array. But perhaps more significant, would be the program's stubborness in terms of drawing a conclusion according to one rule, and one rule only. This is contrary to real life, where one rule will often relate to another, and another, and so on.

Yes, there has to be a better method, and indeed there is – a more flexible and perhaps more intelligent approach, would involve the program considering a number of rules within the rule set, paying particular attention to the manner in which they relate. Consequently the program would draw a more informed and ultimately, more sensible conclusion. 'Backward chaining' represents just one computational technique by which an inference engine may consider the relationship of a number of rules. The concept is best illustrated by two complementary rules:

1) IF battery is flat THEN battery electrolyte level low

2) IF starter motor not rotating THEN battery is flat

Beginning with rule 2, it tells us that if the 'starter motor is not rotating' then the 'battery is flat'. Within a program adopting the previously discussed approach, if 'starter motor not rotating' were an entered fact, then only this rule would be observed. There would be a total disregard for rule 1, the 'fact' of which quite clearly corresponds with the 'conclusion' of rule 2. This shortcoming is adequately negated by backward chaining.

For example, if the rules were backward chained, the program would firstly select rule 2, and then compare its conclusion with the fact of rule 1. If they correspond, as they do in this instance, the conclusion of rule 1 will be assumed correct. So, in response to 'starter motor not rotating', the program would suggest the electrolyte level were too low, and not merely state that the 'battery is flat'.

It almost goes without saying, that such a backward chain can involve a vast number of relative rules, where the search may begin anywhere within the actual 'chain'. Ideally, the search terminates when the conclusion of a particular rule is not the same as any other fact. The technique is not solely devoted to expert systems, as I'm sure the adventure game writers among you might have noticed? It may also have occured that many adventure game principles could lend themselves to expert systems – for example, text compression techniques could play a role, provided of course it were really necessary.

Load EXPERT 2 from your disk at this stage, either from the menu or by typing

LOAD "EXPERT 2"

You should list the program so that it can be studied while reading the following notes (some line references are made).

Having talked about some of the nuts and bolts of a typical expert system, it is only fitting that we should put the discussed methods into some kind of practice. And in the best possible Blue Peter tradition, the basic program – EXPERT1 – is one I did earlier. The system emulates the fault finding procedure with regard to a personal computer. I know what you're thinking: how are you supposed to run it if your computer goes down? Well the sheer meagreness of the program – at this stage – should shatter any suspicion of it being of any practical use, so for the present, just consider it as being a first step.

Commencing with a top-down analysis, the high level operation can be condensed as follows: The array K$(n,n) supports both facts and conclusions – with facts stored in the K$(1,n) column and conclusions in the K$(2,n) column, as previously discussed. 'Faults' are entered in lower case letters and stored in the user$ variable. Then, the user$ is sequentially compared with K$(1,n) elements until the correct rule is located. Consequently, a backward chaining routine is invoked which considers three preceding rules – preceding in terms of their arrangement within both the array and the data statements (LIST 550 – 990).

At a lower level the program can be broken down into two main subroutines. The inference engine (LIST 410 – 460) which searches the K$(n,n) array, and duly compares the inputed user$ with K$(1,n) elements, which in this case are facts. And the backward chaining routine (LIST 480 – 520), which merely compares a located conclusion with the fact of a preceding rule, and as previously mentioned, three such rules will be considered. To process a greater number of rules however, simply alter line 490. Try for example,

490 FOR B% = A% TO 1 STEP -1

This will ensure the consideration of all rules preceding an initially located rule within the inference engine routine. If you have already contemplated entering your own knowledge base within this program, you will no doubt have realised the inherent sequential nature of this particular backward chaining technique. For, any rule set has to be arranged in a specific order, otherwise the backward chaining technique will be nullified. So not only have we got a program which will be poor in terms of ease by which a knowledge base may be entered, but it could also ignore some perhaps relevant rules. Yes this program demonstrates backward chaining possibly in its rawest form, with conclusions of rules only being compared with the facts of rules directly preceding them.

ONE ON ONE

Poking among the entrails; our resident programmer explains the differences between BASIC I and II

Do you have *BBC BASIC version 1* fitted in your machine?. "How can I tell?" I hear you ask. Well to find out is quite easy. All you have to do is press CTRL and BREAK. Then before typing anything else type REPORT <RETURN>. If you are faced with the message

(C)1981 Acorn

then you have *BASIC 1* fitted.

Having *BASIC 1* instead of version 2, 3 or 4 means that some software published in earlier issues of our humble magazine failed to work. The story is a common one to *BASIC 1* users. There you are happily running a program and then, for no apparent reason, the computer reports an error in the program listing. Upon investigation you can find no obvious reason for the error. The explanation is simple enough.

Programmers with *BASIC 2* in their machines use BASIC *keywords* such as **OSCLI** or **OPENUP** which were not included in *BASIC 1*. When the original BASIC comes across these new words it does not recognise them. It cannot even show them in the listing.

For example *BASIC 1* does not have any **EQUATES** for use in the built in machine code assembler. These are keywords such as **EQUB, EQUW, EQUD** and **EQUS.** Don't fret because there are ways of getting around this problem. As shown in the program on this months disk *(BASIC 1 EQUATES)*. Also shown in the program is a *BASIC 1* version of the missing **OSCLI** command. Short for Operating System Command line Interpreter, **OSCLI** is a very useful way of passing commands to the operating system within a BASIC program. So whenever you come across a mysterious line in a program that reads something like:

```
file%=file$
```

the chances are it should read:

```
file%=OPENINfile$
```

or if you find an erroneous line such as:

```
("SAVE "+file$+" 2000+100")
```

using the *BASIC 1* equivalent it should read:

```
PROCOSCLI("SAVE "+file$+"
  2000+100")
```

and work perfectly.

There are other problems as well. Lines such as:

```
INPUTTAB(10,10)"Input age
  ";age%
```

are not allowed in *BASIC 1* but are OK in version 2 onwards. The line would have to be changed to this

```
PRINTTAB(10,10)"Input age
  ";:INPUTage%
```

to work under *BASIC 1*. The best solution to this problem is to buy a *BBC BASIC 2 ROM* and replace the BASIC 1 ROM in your machine. Sadly this is an expensive measure, and involves opening the computer up and throwing away a perfectly good ROM. To help *BASIC 1* users, the software published in Disk User is rigorously tested at all stages of production for *BASIC 1* compatibility. We would also like programmers who have written *BASIC 1* versions of the software we have published to send us details of the alterations. We can then print these for the benefit of other readers who face similar difficulties. This also goes for any readers who have fixed programs themselves and like to share their knowledge with other users of *BASIC 1*.

One such program which did not work under *BASIC 1* was the *HIPAGE* program in issue 7's *Superfont* suite. Included on this months disk is the *BASIC 1* conversion called *HIPAGE2*. Comparing this program with the original will shed some light on how to convert *BASIC 2* programs to work under *BASIC 1*. Other parts of the suite also caused some problems.

Using the information presented in this article you should now with a little bit of detective work and patience, be able to rectify those, and any other problems you have come across.

Barry W. Christie

A.C.E.

This impressively easy to use utility allows you to design your own sprites and font sets. These can then be incorporated into your own BASIC programs for that professional look!

Selecting A.C.E from the Disk User main menu boots the program directly into the main editing screen. Study it carefully, as this is where you will be doing most of the work.

DESIGN

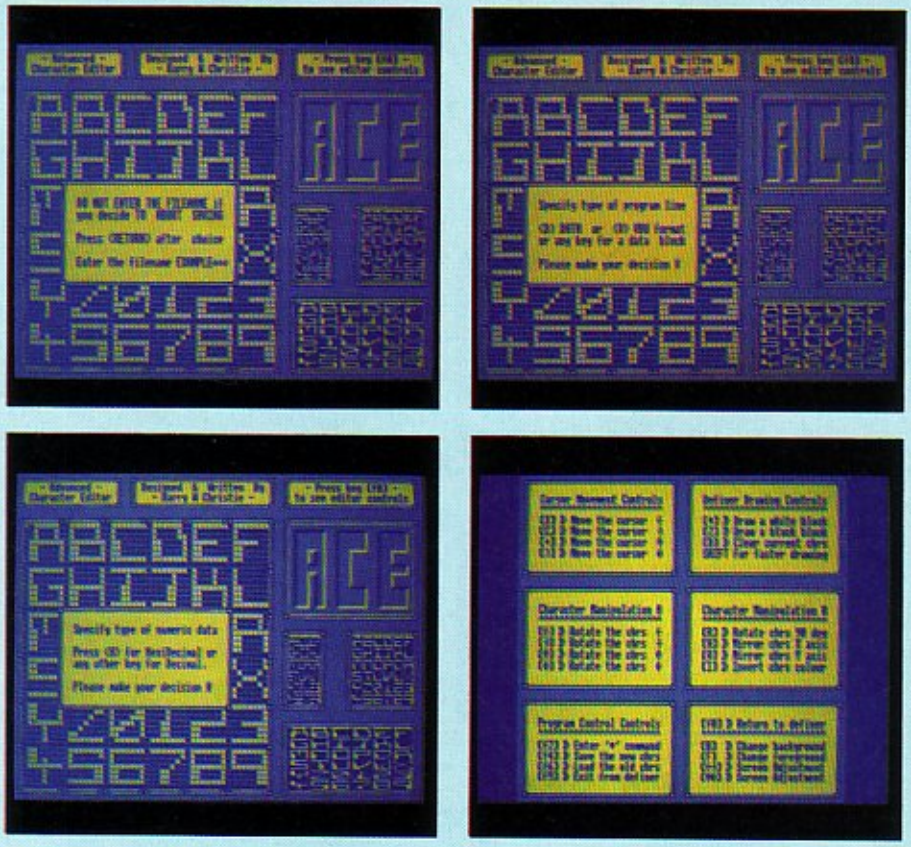
The screen is dominated by a large grid on the left hand side, on which you can define up to 36 characters at once. To the right of this there are three windows which give actual size versions of the characters in screen Modes 0 to 6.

HELP

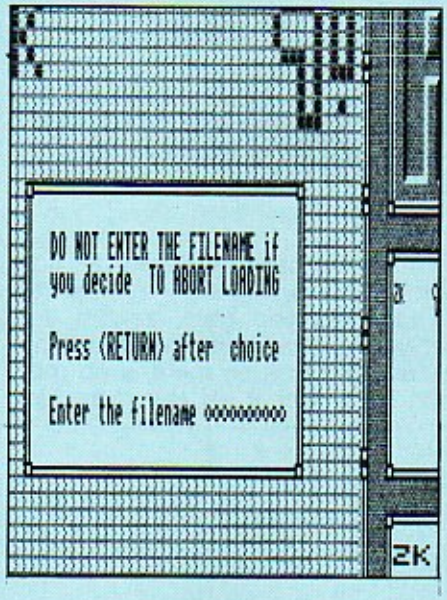
At the top right of the screen is a reminder that pressing <f0> (f refers to the red funcion keys) toggles the help screen with the editor. Do this now and the editor screen is scrolled up to be replaced by the help screen. The <f0> key can be used at any time to switch back and forth between the two screens, but the various control keys are only available from the editor screen.

STRIP

At this point stop and cut out the function key strip from the bottom of the page, and insert it under the perspex cover just above the line of function keys. This now acts as a handy reminder of their definition.



A.C.E. – Advanced Character Editor
- **Preview in all graphics modes**
- **Easy to use**
- **On-line HELP screen**



FILING

If you are looking at the help screen press <f0> to get back to the editor, and then press <f6> to load the sample file provided on disk. A requester window will pop-up. Press

Key f0	Key f1	Key f2	Key f3	Key f4	Key f5	Key f6	Key f7	Key f8	Key f9
Toggle To Editor/Keys		Command Line		Save Chr Definitions		Load Chr Definitions			Quit Using The Editor

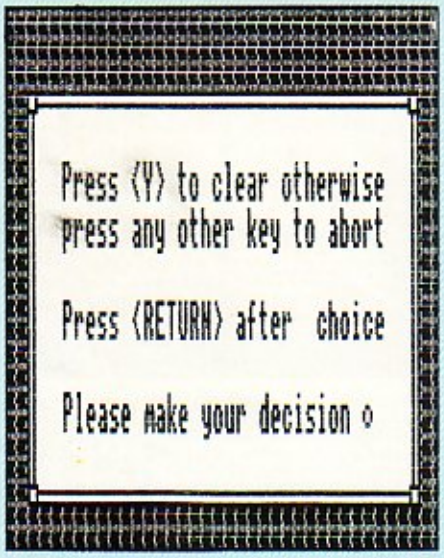
<RETURN> if you wish to abort at this point. The file you wish to load is called

example

Enter this into the requester box and press <RETURN>. When the file has loaded another box will pop-up to confirm this, or report an error if an incorrect filename has been specified, and ask you to press a key.

EDITING

Press <SPACEBAR> for example, and the screen will scroll to reveal the loaded character set. You can now practice using the powerful editor controls to modify the characters in any way. Remember pressing <f0> will scroll up the editor screen and display the controls at any point.



SAVE

When you have mastered the controls you will want to save the characters you have created onto disk. At this point insert a blank formatted disk as there is no room on Disk User 9 for any more files.

Now press <f4> to save the character definitions. The same requester window pops-up as when you loaded in the definitions. This time however, on pressing <RETURN> another window pops-up. This one gives a choice of three options D, V, or any key for a data block. The last will save the data block and return you to the editor.

DATA

Pressing D or V will present yet another window. The options available are H for Hexadecimal, or any other key for Decimal. Choose your option, and press <RETURN>. This completes the saving process. What you do now depends on which options you chose. See below for a brief explanation of how to use saved DATA statements, or VDU codes.

USING YOUR DESIGN

To incorporate the saved VDU or DATA statements into a program of your own just follow these simple instructions:

1 Load in the saved VDU or DATA statements (the filename you chose above).

LOAD "filename"

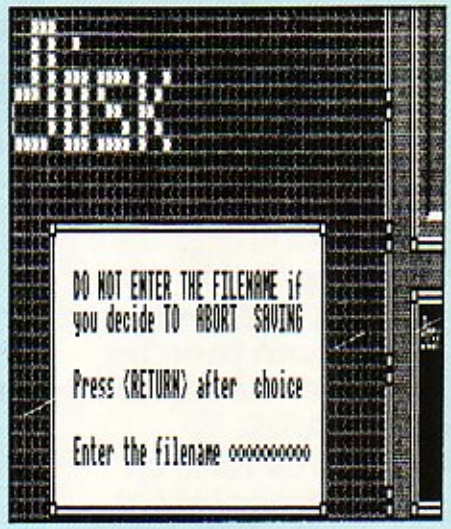
2 When the VDU/DATA statements are loaded you must renumber them to suit your program. You could also put the VDU/DATA statements into a procedure.

For example, if you want the character definitions to start at line 320 of your program you must type

RENUMBER 320

3 Now you have to save the VDU/DATA statements as an ASCII file.

***SPOOL "newfile" LIST *SPOOL**



This will now produce an ASCII file which can be EXEC'd into your main program.

4 You can now load your own program.

LOAD "myprog"

5 The VDU/DATA statements can now be EXECed into your main program.

***EXEC newfile**

If you LIST the program you should have your ACE generated characters in your program.

If you have DATA statements instead of VDU statements you cannot run the program yet. Here is a small routine which will define the characters using the data.

```
10 DEFPROCdefine
20 FOR char=219 TO 255
30 VDU23,char
40 FOR byte=0 TO 7
50 READ def
60 VDU def
70 NEXT byte
80 NEXT char
90 ENDPROC
```

Finally SAVE the finished program.

Further information on combining BASIC program lines appears in *Beginners* this month.

NEXT MONTH

August issue – OUT JULY 15 1988

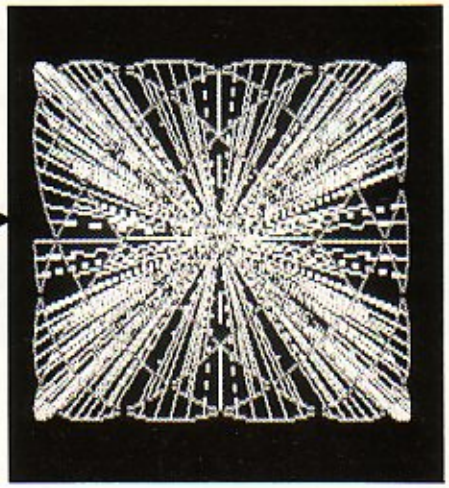
Textshape - 80 column WYSIWYG wordprocessor which uses disk storage to give you LONG documents. Advanced online help system, printer control and page formatting. Search and replace, block copy, move and delete.

Oodles of Doodles - a highly original graphics generation technique

All Assembled - second part of the series

Laser Lettering - showing off on screen

PLUS Advanced and Beginners programming, Readers Routines and an extensive wordprocessor roundup



BEGINNER'S GUIDE TO DISKS

This month a quick recap on how to BOOT a disk and how to combine bits of BASIC programs

Your option

One of the things we look for as beginners is how to start up our own disks automatically with the <SHIFT-BREAK> key press combination which is referred to as *boot*ing the disk. We've received many letters on the subject, typically:

Now I am not a computer wiz kid but a 50 year old trying to keep up with technology. In straightforward language can you tell me how I can get my own disks to "boot up".

Booting a disk is the concern of OPT, a command which seems a mystery to many users who have written to us.

OPT is a *star* command, which means that it is only recognised by the computer if you type it in after a * character. What is more, the command is useless on its own. It's the numbers that come after it that cause the computer to do something. These numbers are called *parameters* because they lay down the parameters by which the computer is going to work henceforth. OPT is followed by two numbers separated either by a comma or a space – either will do.

The most useful of OPT command parameters is 4. *OPT 4,1 causes the !BOOT file to be *LOADed, *OPT 4,2 to be *RUN, and *OPT 4,3 to be *EXEC'd. *OPT 4,0 cancels all other *OPT4s.

!BOOT file

What is a !BOOT file? Well it is the *one* filename which the computer can recognise as having special significance. No other file on your disk means anything to the computer. However when the computer has been told by a *OPT4, it will look at the catalogue of the disk to find a file called !BOOT whenever it detects the <SHIFT-BREAK> key combination. Incidentally you can setup other boot keys but we won't go into that here.

The !BOOT file itself has to be

present or the computer will just ignore your <SHIFT-BREAK>. However if the file is present the computer will carry out the selected action upon the file. The most common action is to EXEC the file (set with *OPT4 3). Each line of a text file is taken from the !BOOT file on the disk and fed back to the computer for a response. The text file might contain a message, for instance "don't press SHIFT and BREAK"! More usually it contains a star command or BASIC command or combination of these. Special control characters can also be inserted. The most commonplace !BOOT file? My guess is CHAIN "MENU", how about your's?

Other options

A command you may be familiar with is *INFO, which is used to display information about the files which you have on disk.

Another way of getting detailed file information on screen is to use the *OPT command. *OPT 1,1 will force the display of full file information whenever the file is accessed. This is useful when programming but not desireable in the middle of a program, so you can disable the information with *OPT 1,0.

Adding program routines

A second program is EXEC'd from the disk onto one already in memory. In such cases the line numbers of each program must be carefully compared to make sure that the new lines appear in the right places. When you EXEC a file, all you are doing is getting the computer to *type* the contents of the file for you. And, just as when you type an existing line number it overwrites an old one, so does the EXEC file.

Glue together

For those who just want to glue together two bits of BASIC what follows is the easiest way. You can must however prepare your BASIC programs in advance, making sure that the PROGRAM part has line numbers below that of the ROUTINE part. That should be easy enough using RENUMBER.

Follow these instructions to install the routine (ROUTINE) into any BASIC program (PROGRAM):

LOAD "PROGRAM"
PRINT ~TOP-2

This will print a hex number on the screen eg 2D00. Use the number which is displayed on the screen as you type in the following line:

***LOAD ROUTINE hexnumber**
OLD SAVE "PROGRAM2"

PROGRAM2 now contains a single BASIC program made up of the line numbers in PROGRAM immediately followed by the line numbers in ROUTINE.

Data

Many beginners have asked how to combine bits of BASIC program and for this reason we have run articles and programs explaining overlay techniques. A core overlay routine appeared in issue 3, and a complete procedure library manager appeared in issue 7. Both are available as back issues. See SERVICES.

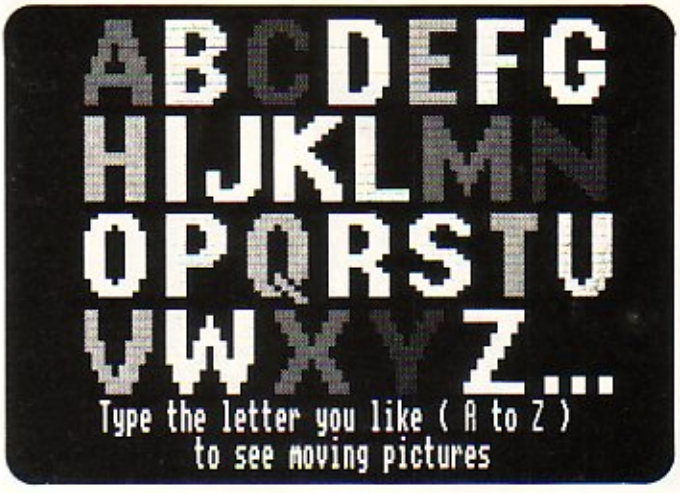
MENU PAGE TWO. OPTION ONE.

Abbas

COLLECTOR'S ITEMS

J is for J...J...J...Juggler
Animate the alphabet with computer artist Abbas. Every month he animates a letter. Just choose this month's letter from the alphabet menu to see the action.

Instructions on how to use the menu for other letters were given in Disk User number seven. Letters A to I are available on back issues (see SERVICES this issue).



SECTOR ZERO

WIN £25 with a star routine plus Acorn trivia and competition winners

Acorn Trivia

1. How many filing systems have been implemented on the 8 bit BBC Micros?
2. Who was the original "voice" in the BBC speech system?
3. At what memory location does screen memory start on the BBC Micro with Mode 2 selected?
4. Which English king was crowned in 1189?

Winners in Print

Graham Sturzaker from Preston has won our Paint the Town competition (Fleet Street Editor) and takes away Fonts'n'Graphics, Disney Graphics and Admin Xtra (all programmed by Clares incidentally).

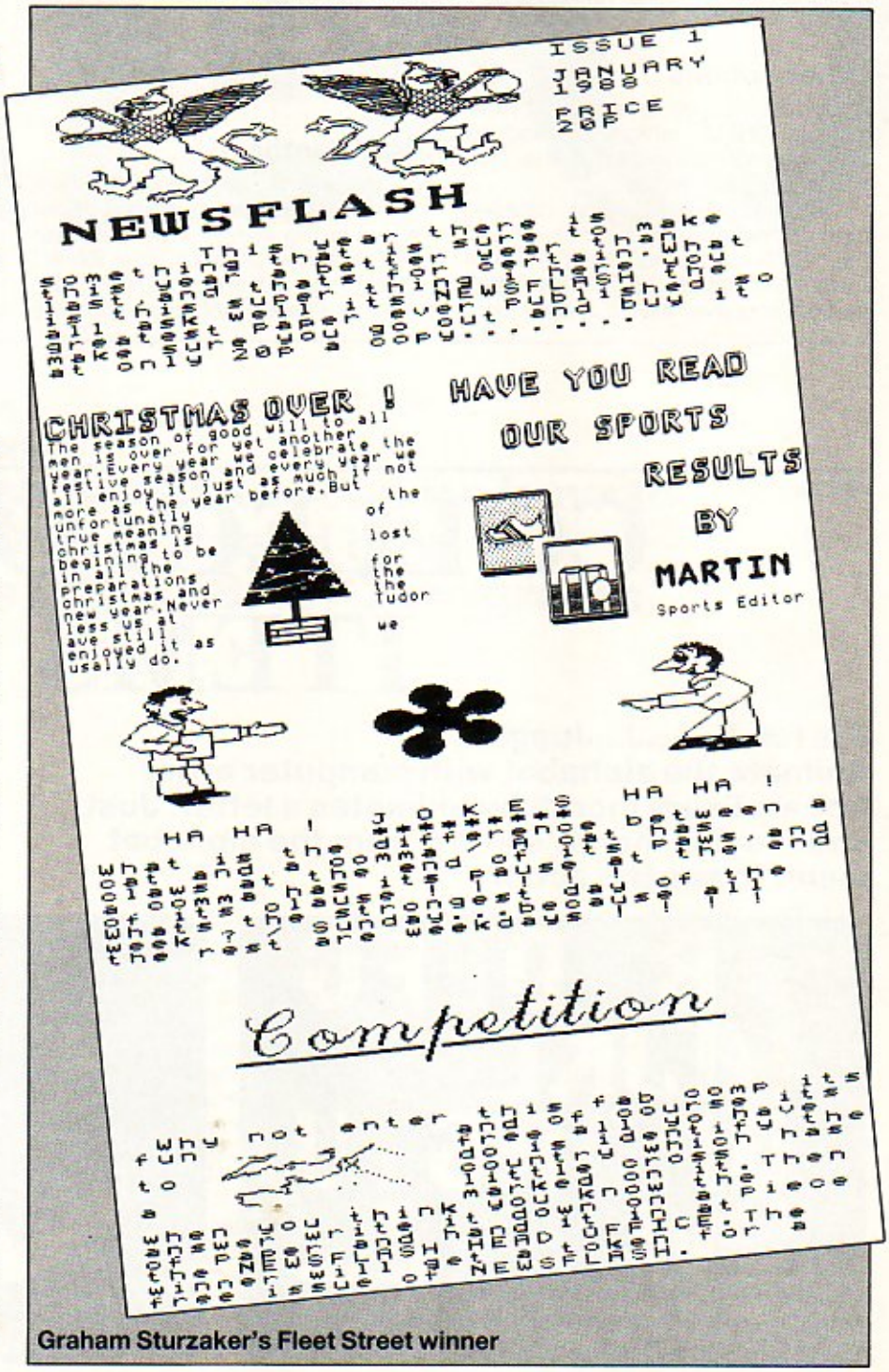
Dorian Goring has won our Paint the Town (Stop Press) competition with an array of artwork created with Stop Press (from AMS), some digitised, some clip art and a variety of fonts. Dorian uses a Master 128, Cumana disk drives, Citizen 120D printer and Phillips colour monitor. The posters are used by Dorian in media studies teaching pupils how photography is involved in newspapers, film-making and television.

Thanks also to our youngest entrant Edward Ferrari (aged 11). We've awarded a special prize for Edward's *BBC Bonanza* – graphics software from the Disk User SERVICES page.

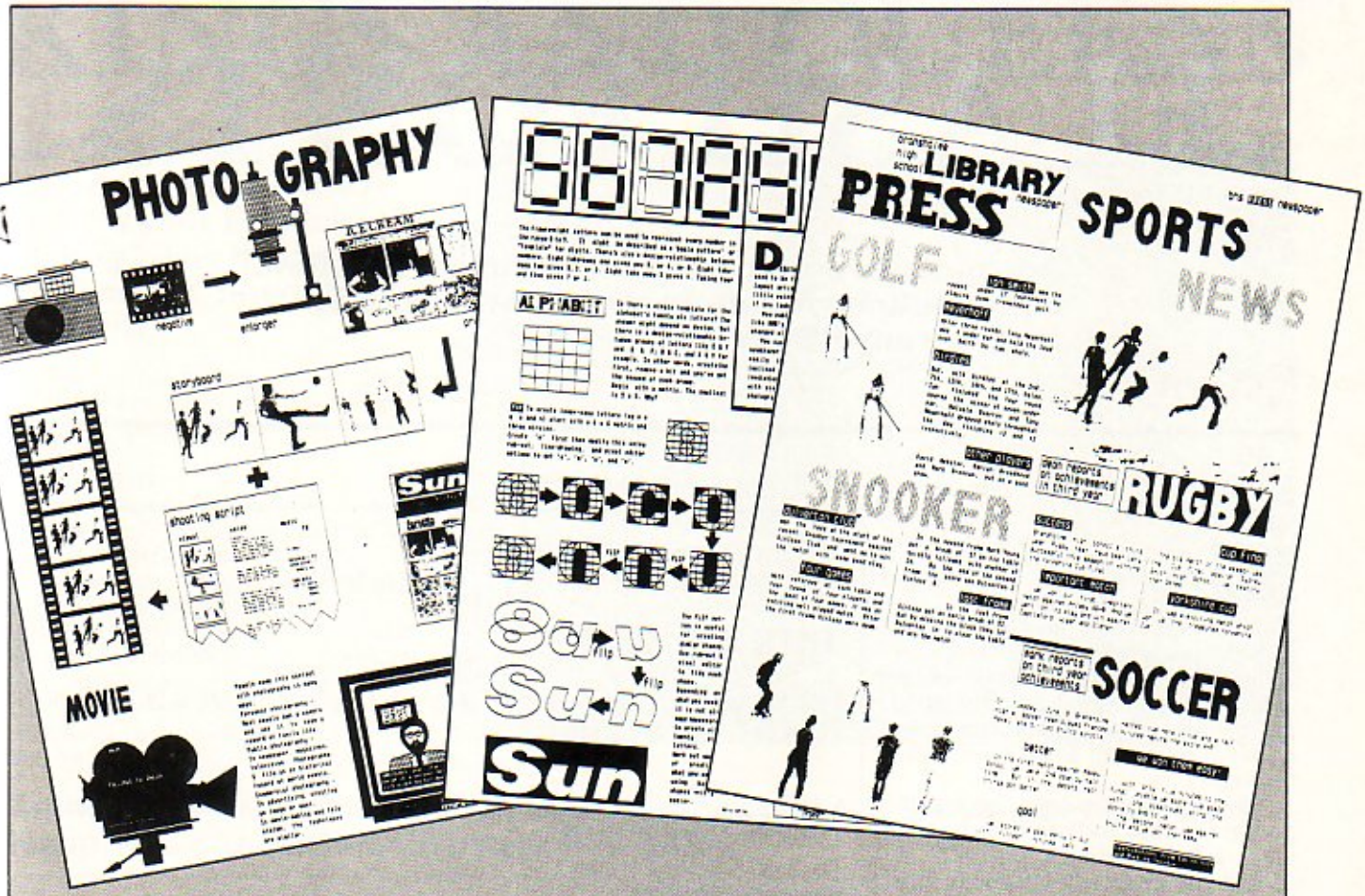
Prizes on their way. If you've created a print masterpiece then send it in to us at Disk User and we'll put it on display in Sector Zero.

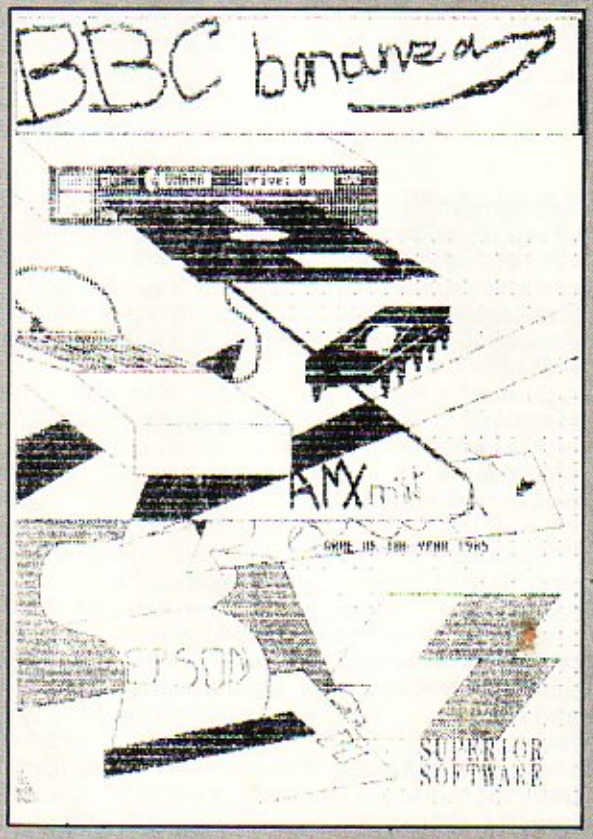
Readers Routines

Your reader routines are beginning to flow in to the Disk User offices. Thanks and keep them coming. Twenty five pounds is still up for grabs if your routine is the star routine of the month published on Disk User.



Graham Sturzaker's Fleet Street winner





Edward Ferrari (aged 11) with BBC Bonanza

Dorian Goring's winning Stop Press creations

£25 WINNER

This month we've continued a run of cash-winning programs by Kevin Mortimer. The first is a memory dump. Just enter the memory address at which you wish to start and the program will display the contents of the memory location. Multiple lines can be displayed by pressing <SPACE>.

The second program is a program which plays a piece of music in the background, letting you carry on with other activities at the same time. Function keys zero and one are used to control the action of the program.

Watch out for more great reader routines next month.

1. Four. ROM filing system, Disk Filing System, Advanced Disk Filing System, Cassette Filing System
2. Kenneth Kendall
3. Hexadecimal 3000
4. Richard 1st (see Kings & Queens this month!)

FLASH FONTS

Advanced programming with efficient machine code routines for alternative screen display fonts on any BBC Micro

The file 'COMPRES' on this month's disk is a utility for the BBC computer series that prints text in a compressed form, effectively doubling the number of characters that can be set in any line of any screen mode (except 7).

The routine uses only *Acorn* supported OS calls and can therefore be used with confidence on 6502 or 65C12 processors, including co-processors. The routine will also work correctly with all shadow screen systems.

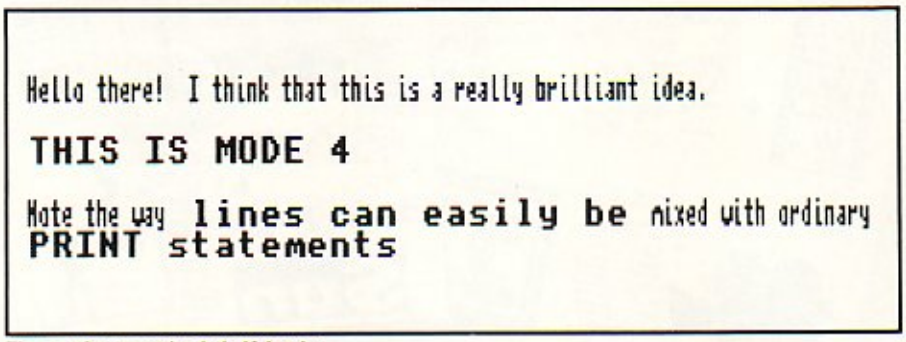
The code occupies only 172 bytes of memory and can be located almost anywhere. It is ideal for generating in a header program of some sort, in say, a game.

The routine works by taking pairs of characters from the string offered, squashing them into character number 255 and then printing it. This is done by using only alternate bits of the original characters. If the string has an odd number of characters then a dummy space character is added to the end.

It would have been possible to produce a more fully filled font, but this would have taken considerably more code and would not print as fast. The appearance is quite pleasant, particularly in Modes 1,4, or 6, and is clearly readable.

Whilst there are several text compressors around that use pre-defined character sets, they all suffer from the disadvantage of the memory space taken by the character set, or alternatively they re-define the whole of the existing set, preventing the user from performing normal printing on the same screen.

This utility suffers neither of these disadvantages and has the added bonus of permitting any printable character, other than number 255 to be included in the text string, regardless of when they were defined.



The new fonts are simply half the size.

Program design

The routine uses the *proper* entry via CALL with the string to be printed passed as a parameter. A check is made on the screen mode, producing a 'Bad mode' error if it is 7. Also tests are made for the number of parameters passed and parameter type. 'Parameters' is reported if either is incorrect.

Below is a breakdown of the program lines.

40	**define memory space for code**
70 – 110	**initialise addresses and entry points**
170 – 200	**test screen mode**
210 – 250	**bad mode error message**
240 – 270	**argument error message**
300 – 320	**test number of parameters**
330 – 350	**test the parameter type**
360 – 510	**find location and length of string in memory**
530 – 650	**main loop for getting pairs of characters**
730 – 800	**define character 255 and print it**
810 – 830	**test for last character and loop back for more or exit**
850 – 1070	**transfer alternate bits to new character 255**
1090 – 1160	**workspace**
5000 – END	**demonstration text**

The routine is designed to use the absolute minimum of memory space. Apart from the code itself, only two bytes of zero page are used.

Of particular note are lines 570 – 600. When assembled this has the code equivalent of 'CPY #&FF'. However, every time the code is called the string length is poked in place of the '&FF' by line 420, turning the line into 'CPY #(string length). As this byte is marked with the label 'endmark' it is also accessible to line 810 (CPY endmark) for testing. This system saves two bytes of memory over more conventional methods.

Note also the error messages share a byte value 0 at line 250. This only saves one byte, but every little helps!

For games etc. I suggest that the assembly routine be put in the 'front page' with line 50 changed to:

50 code=&750

This is in the memory area used by BASIC mainly for the editor. It is hardly likely that anyone will be editing lines partway through playing a game, so the code will be safe.

Where it is guaranteed that the routine will not be used in Mode 7 or with incorrect arguments then lines 170 – 350 may be omitted. This will reduce the code size to only 127 bytes.

KINGS AND QUEENS

Our first datafile for the Tracer database. How good is your English history? The Kings and Queens file can help you out!

To use the Kings and Queens datafile you will need a copy of the Tracer database (see below). Copy the file
I.MONARCH
to your Tracer disk, run Tracer and choose the EDIT option followed by the MONARCH file handle. It's as simple as that. Now you can find out who ruled, when, and for how long in British history. You can sort the Stuarts from the Windsors, the Plantagenets from the pretenders, find out when they were crowned and how long they managed to last the course.

```
MONARCH
-----MONARCH-----  -HOUSE-  CRND  DIED  AGE
****PLANTAGENETS=  PLANTAG  ****  ****  ***
HENRY II           PLANTAG  1154  1189   56
RICHARD I          PLANTAG  1189  1199   42
JOHN               PLANTAG  1199  1216   50
HENRY III          PLANTAG  1216  1272   65
EDWARD I           PLANTAG  1272  1307   68
EDWARD II          PLANTAG  1307  1327   43
EDWARD III         PLANTAG  1327  1377   65
RICHARD II         PLANTAG  1377  1399   34
HENRY IV (LANCS)   PLANTAG  1399  1413   47
HENRY V  (LANCS)   PLANTAG  1413  1422   34
HENRY VI (LANCS)   PLANTAG  1422  1461   49
EDWARD IV (YORK)   PLANTAG  1461  1483   41
EDWARD V  (YORK)   PLANTAG  1483  1483   13
RICHARD III(YORK)  PLANTAG  1483  1485   32
```

Tracer was published on Disk User number eight. It is available as a back issue, see our SERVICES pages this issue.

ORCREST

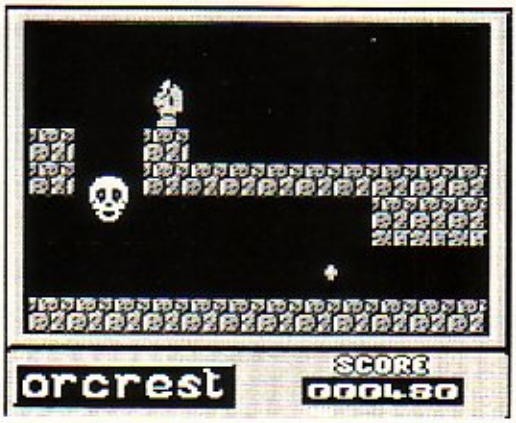
The *Lurking Nasties* are out to get *Orcrest*, but you must avoid them and collect all the crystals; if you can.

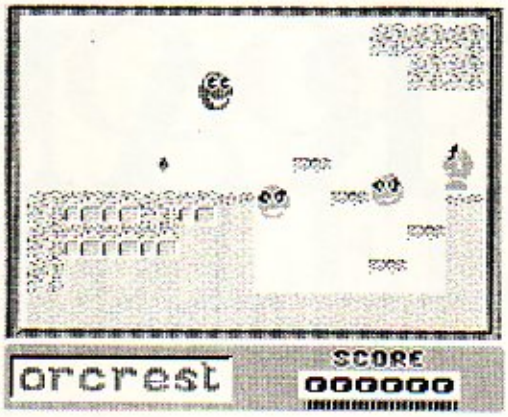
"It's a b*h of a game" said our resident games wizard as he went to cool off his blistered hands in some sulphuric acid.**

"No.." muttered monosyllabic programmer Taylor when begged on bended knee to at least quadruple energy levels.

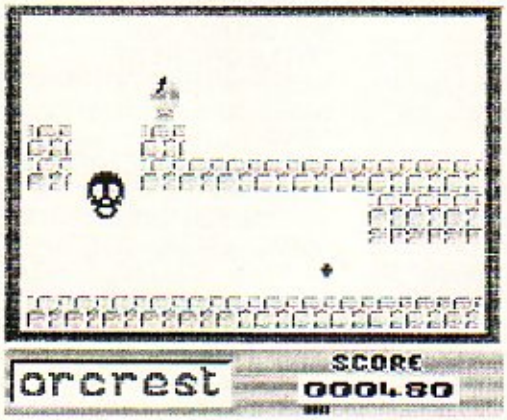
Multi-coloured, multi-room, multi-sprite..... multiple dangers.

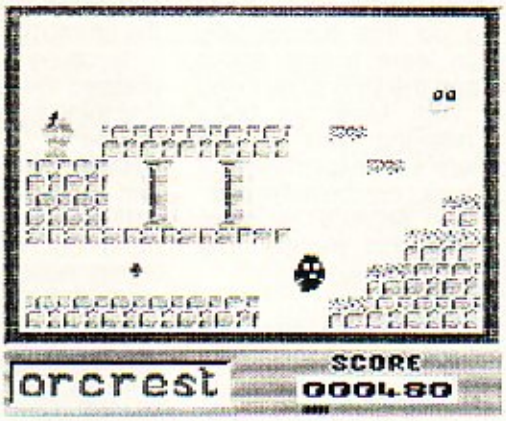
Play this game, and twitching hands, insomnia, and brain meltdown are guaranteed.





One touch from a skull, and you're finished, for good.





COMPETITION

Play Orcrest and WIN a Replay System from Vine Micros

How to Enter

The current price of Replay is £29.90 inclusive but you can get one FREE by playing Orcrest and telling us your high score. Send it in on a postcard to Orcrest, Disk User, 6C Belgic Square, Padholme Road, Peterborough, PE1 1XF by 30th June 1988.

Replay

The Replay System Mk 2 is the latest version of the popular tape to disk transfer and game-playing aid from Vine Micros. The utility is now available for 1770 disk interface users. All the features of the original version have been retained and additional routines added.

Replay has been traditionally used to save screen displays for reloading and printing as well as for "freezing" games. The Mk 2 includes built-in formatting and verification of disks, sound on/off and the ability to make program modifications (including games cheats).

Replay makes transfering programs easy with automatic file creation and double density 18 sector operation with up to 22 files per disk. It can also create coded disks for security backup.

The Rules

The results will be published in the September issue of Disk User. The editor's decision is final. No employees of Argus Specialist Publications or their suppliers are permitted to take part.

Details about Replay are available from Vine Micros, Marshborough, Nr. Sandwich, Kent, CT13 0PG. ☎ 0304 812276.

TRANSFER

Taking your disk files one step further

Disk User programs can be so useful that you'll often want to transfer them to their own disks and use them separately, without title page or menu. To do this successfully you'll need to learn a little about BBC BASIC and the DFS (Disk Filing System). In Disk User we don't believe in referring you to the manuals to here's an explanation of how such a transfer can be achieved.

Let's take the *ORCREST* program as an example this month. The relevant file on Disk User is:

X.ORCREST

Make sure you have a blank data disk ready to receive the file. Insert Disk User and type

***COPY 0 0 ORCREST**

and press the RETURN key. Follow the keypress prompts on the screen until the > prompt returns. Now

***COPY 0 0 EXPAND**

and press the RETURN key. Follow the prompts.

Because ORCREST is a compressed file special to Disk User the first thing to do with your new disk is to type

CHAIN "EXPAND"

and to choose the option for ORCREST. The program will now expand out the separate programs which make up the package (be patient!). First however you must add the !BOOT file and EXEC option to the disk. You do this by typing:

***BUILD !BOOT 1 PAGE=&1900 2**
CHAIN "ORCREST"
(press ESCAPE)

***OPT 4,3**

Give it a title

You can now type

***TITLE ORCREST**

and press the RETURN key. Now type

***CAT**

and press the RETURN key and the screen will display the title *ORCREST* with the files ORCREST, ORCRES1, NUMS, DATA, SDATAA, SDATAB, SDATAC, SDATAD, SDATAE, S.SPTDATA, LOADMEM lined up below along with EXPAND and ORCREST – which you can now delete.

Details are also given on screen when you select ORCREST from the Disk User menu.

Disk User is now monthly. You can order one from any newsagent. Watch out for Disk User on the third Friday of every month. The publication dates for 1988 are:

Nov 1988

Mon	Tue	Wed	Thu	Fri	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

Jul 1988

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Sep 1988

Mon	Tue	Wed	Thu	Fri	Sat	Sun
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2

Aug 1988

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Oct 1988

Mon	Tue	Wed	Thu	Fri	Sat	Sun
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Dear Newsagent

Please reserve me a copy of Disk User for the BBC Micro Monthly.

Name .....

Address .....

.....

.....

.....

Thankyou
Signed .....

STARTING OUT – ASSEMBLER

Develop your machine code programming on Model B, Master or Electron with this tutorial which results in a useful programming tool.

There is much mystification concerning machine code programming. This three part article tries to shed light on the arcane subject by describing how a specific utility program was constructed and how it works.

The first thing which confuses the potential lowlevel programmer is the use of the terms *machine code* and *assembler*. The former is a binary code which the machine uses and is not easily understood by mere humans, whilst the latter is a mnemonic language, similar in some respects to BASIC, which is converted to machine code by an *assembler* program before the program may be run. You may come across the term *compiler* from time to time, but this is generally used in relation to higher level languages.

The program discussed here began life a few years ago when I decided to kill two requirements with one project. I figured it was about time to learn yet another microprocessor language, namely that of the 6502 on my (then) new BBC computer, and at the same time I desparately needed something to find BASIC variables and text for me.

The result was a piece of code taking up just one page of memory at location &D00 (it was then a cassette machine), including the input buffer and all the variable work space except for some necessary page zero locations. After discovering that BREAK overwrote the first location of the program I moved it up to reside at &D01.

Electron compatible

I used the program regularly until recently, when I offered it to an Electron owner only to discover that it did not work on said beast. After poking about in it for a while I decided that it was also time to overcome the annoyance of the program always displaying 25 screen lines, even in 32-line modes. I also decided to take into account the occasional need to find the odd keyword by permitting the first character entered to imply that the following two characters form a hexadecimal code.

In deference to the Electron I also included a keyword search based on the legends on its keys, which bear no relationship whatsoever to the actual BASIC tokens which they produce. The program was built on the BBC model B using BASIC I and tested on the Electron, which of course has BASIC 2.

These modifications naturally took up more room than before, so I decided to provide two versions for the price of one source program; a short one with only a variable/text search implemented (a slightly improved version of the original), and the longer one which also finds keywords and hex codes. The reasoning behind this is that, if you are writing for a cassette system (are there any these days? – Ed), it is difficult to find more than 256 consecutive bytes free below &E00 which will not at some time be overwritten, whilst in a disk system it is usually permissible to utilise the extra space otherwise given to cassette buffers.

The short version of the program takes up one page of memory, which can usually be fitted into &D01 plus a small input buffer which could fit in any odd corner of 16 or more bytes. The longer program takes up about 380 bytes plus the buffer, which may be left at the same location as that of the shorter version or tacked onto the end of the program. Both versions require five or six bytes of zero page memory.

In residence

In the program HEXFIND, procedure PARAMS is inserted between the two assembly passes, as this is needed to store program constants and data (such as the OSWORD

Table of electron keyword codes

This table gives the values used in creating the Electron keyword table ELKTAB. All values are in hexadecimal. The letter key values are given for the upper case version; the keyword section of the program adjusts lower case letters to upper case before use. The entries for – and @ are not original, but were added as convenient padding.

KEY	ASCII VALUE	BASIC TOKEN	BSIC FUNCTION
,	2C	C8	LOAD
–	2D	A4	FN
.	2E	CD	SAVE
/	2F	F1	PRINT
@	40	E1	ENDPROC
A	41	C6	AUTO
B	42	CC	RENUM
C	43	FB	COLOUR
D	44	DF	DRAW
E	45	8B	ELSE
F	46	E3	FOR
G	47	E5	GOTO
H	48	9D	DEG
I	49	E8	INPUT
J	4A	B2	RAD
K	4B	D7	CHAIN
L	4C	C9	LIST
M	4D	EB	MODE
N	4E	ED	NEXT
O	4F	CB	OLD
P	50	F0	PLOT
Q	51	EA	LOCAL
R	52	F9	RUN
S	53	88	STEP
T	54	8C	THEN
U	55	FD	UNTIL
V	56	EF	VDU
W	57	F7	RESTORE
X	58	F2	PROC
Y	59	F5	REPEAT
Z	5A	E0	END

```
The long version can be used to find
hex codes and keywords; the short can't.
To find a hex code enter a dot followed
by two hex characters.

To find a key code enter @ followed by
a single character equal to the keyword
minus 128 decimal (80 hex).

To find an Electron keyword enter colon
followed by the Electron keyword.

All codes must be the first item of the
find string, but may be followed by
ordinary text.

Warning: the code for the short version
fits in 256 bytes (one page); the long
version (finding hex and key codes)
takes up more than 256 bytes.

Include coding for these? (y/n)...
```

Screen 2 – long version

control block) at the end of the program code on the first pass. It could also be added on the second pass without any real harm in this instance, but this is known as bad policy; something might be altered later which would change the situation. Procedure ELKPARAMS is similar in operation to PARAMS, but it is only executed if the long version of the program is required.

The resident BASIC variable P% is used by the assembler as a program counter in order to keep track of the program addresses as it assembles the program. When listing a program during assembly the value of P% is shown at the left hand side of the page in hexadecimal as the address of each byte (or group of bytes) of code; the machine code assigned to that memory address is shown in the next column, followed by the mnemonic code which generated it. Since the constants which the program requires are added in by PARAMS, this procedure is made to increment P% according to the amount of data which is added for each constant.

In HEXFIND I give the resident variable E% the value of the program counter P% immediately after the first pass and when all parameters have been assigned, as P% is then at the maximum value which it will ever attain. E% is used at the end of the completed assembly to show the amount of code which has to be saved to disk or tape. Things like this could be printed out as they occur, but this produces problems when debugging the program, and again comes under the heading of bad policy.

The final program line calls a procedure to display the results of the assembly, such as what memory is used and where. I have used very similar base programs for assembler work several times, and find them simple and convenient to use in this fashion, as procedures may be removed or added as circumstances dictate. For example, INCLUDE–KEYS was added specifically for this program, and does not appear in any other which I have designed; although other procedures have appeared in this position.

The Procedures

Taking the procedures roughly in order of appearance, INCLUDE–KEYS gives a brief outline of the program requirements (for when you've lost this magazine!). It also sets the boolean variable INCLUDE according to whether the long or short version of the program is required; this flag will be used several times during assembly for directing operations.

ADDRESS defines default addresses for the program, with values which I generally use on a disk system. The addresses are then offered for redefinition, so that you may put things where you wish. Note that PAGE% is initialised to the current PAGE value. This must be set up to the value of PAGE used in the target computer environment. It would be nice to set it up automatically, but there is no reliable way of doing so that I could discover.

Procedure VARS sets up the program's working space and input buffer. Two separate areas of memory are used of necessity. Zero page (&00 to &FF) has to be used for indirect addressing by the program in order to perform an efficient search, but is so precious (only &70 to &8F are available to the user when in BASIC) that it is unwise to use it as general parameter space unless you are unable to find room elsewhere, or are not intending to use the search routine with assembler source programs. Any of the data areas may be overwritten by any other program at any time without any ill effect.

VARS uses two functions to aid it in allocating variable space. A simple way to assign the work areas would be to give each variable a specific address above the base value and leave it at that. Then in six months time you decide to alter one of them and forget to adjust the others. The method shown here always gives the correct adjustment automatically.

In the case of FNy, Y% is assigned the base value of zero page memory to be used, starting from Z%. The argument to the function call is the number of bytes to be reserved under a given variable name (eg MSG% reserves two bytes),

```
Enter addresses for code and workspace:
enter new address in hex or press
RETURN to accept offered value...

PAGE address is &1900. New one? &

Code address is &900. New one? &

Work space start is &880. New one? &

Work space end is &8BF. New one? &

Zero page start is &70. New one? &

Assembled to run at PAGE &1900
Code is at &900 to &94E
Work space is at &880 to &8BE
Uses page zero locations &70 to &75

Copy the following to save machine code

 *SAVE lfind 900 94F
```

Screen dump of basic section of program

and Y%, the index to the variable space, is incremented accordingly to point to the next free space. The same applies to FNv and (later) FNm and FNp. In BASIC II facilities are provided which replace this method, but any program designed using them will not work on a BASIC I machine.

The only thing to say about OSCONST apart from the fact that it defines various operating system addresses is that it also gives the address of SCNMODE%. This location contains the operating system's current screen mode number, and is used in determining the number of lines available on the screen.

Parameters

Leaving out for the moment the actual assembly procedure calls, procedure PARAMS uses the aforementioned FNm and FNp to define messages and other constants required by the program. MSGFIND% is the address of a string constant set to one of two messages, depending on whether the keyword and hex codes are included or not. The preliminary CHR$ codes for each message cancel any open windows and then clear the screen, leaving it in its current mode.

PAGE1% is the address of another string, this time of control codes equivalent to the VDU28 window function to set up the window in which listings and editing will take place whilst leaving the top of the screen still displaying the results of the search. The window may be cancelled by typing <CTR-LZ> at the keyboard (ASCII 26). Note that by using **$P%=m$** FNm automatically adds a carriage return character (ASCII 13) to the end of each string.

PARM% points to a block of constants required by OSWORD when it is used to get a text string from the keyboard. The space for it is reserved by FNp, and then the details are filled in by PARM%?0

etc. The first two bytes are the address of the buffer into which the keyboard data is placed. PARM%?2 contains the length of the buffer, obtained from the known address of the end of the work space and the address of the input buffer INBUF%. The last two PARM% parameters contain the ASCII values of the first and last characters which the keyboard routine is allowed to accept; in this case a space and tilde (~).

CONV% addresses a table of 2byte integer values used in converting the BASIC line numbers, which are stored in binary, into decimal. MODES% is a table containing the number of screen lines which each screen mode allows, from mode 0 to mode 7. SCNMODE% is used as an index to this table to extract a value which is then inserted into the third character of the string at PAGE1% to define the bottom line of the screen. Watch this if you try to put this program into ROM; copy the PARM% data into a RAM location first and use it there.

The BASIC outline

Machine code on the BBC and Electron computers is handled within BBC BASIC in what is known as an inline assembler. Whilst this produces a few problems in coding, mainly because of the limited space in which to fit all the nice frills, these are generally outweighed by the ability to easily insert various procedures for telling the user what is going on and ask for comments on it. It is also very convenient for combining machine code and BASIC in the same program when this is needed.

The best approach to assembly programming in this kind of environment is to make the program resemble any other reasonably structured BASIC program, and this first part of the article will concentrate on this aspect by describing the shell around the assembly code.

The program on this month's disk is called HEXFIND and is the BASIC portion only; the assembly code is tacked onto the end, in the dummy procedure ASMBL, and will be included on next month's disk. HEXFIND should be loaded with
LOAD "HEXFIND"
and listed while reading this article.

The search program given here starts with a list of things to do. The first program line contains a VDU statement which cancels any windows which were set up (necessary when testing the program) and clears the screen. The procedure INCLUDE–KEYS then asks the intrepid program operator (you) whether the long or short version of the program is required. The computer uses this information to decide which chunks of code to leave in or out.

The next line calls procedure ADDRESS to get the memory address at which to place the program code and work space, whilst the following lines actually set up the work space and define the operating system address constants such as OSWORD and OSBYTE. In any program it is always better, where possible, to define addresses and data by giving them labels, as this makes the program more readable and saves confusion; and it is easier to modify later.

Assembly programs are usually assembled in two parts. The first pass assembles all the mnemonics into binary machine code and places them in their correct position in memory. At this time the poor assembler could get terribly confused, because it has not yet come across a lot of the addresses which it requires. To avoid a nervous breakdown it merely remembers all these occurences and ploughs on through regardless. The second assembly pass then goes through it all again in the light of this new knowledge and inserts all the addresses and data which it could not find on the first pass.

ELKPARAMS actually only does something if the long version of the code is required. It then sets up a table of equivalents for the Electron keyword codes. The first character represents the keyword on the comma key, the next on the minus key, and so on up to the Z key. Not all characters on the keyboard have keywords associated with them, and there is a gap in the table (as well as a few extra codes) which will be explained later.

ELKPARAMS shows a way of using FNp to set up a long table. The table values are held in DATA statements and extracted in a REPEAT-UNTIL loop after first setting the variable name up with a null function call, FNp(0). The full table size is then recorded with a second call to FNp, this time with n%, a count of the data read in. Note that the DATA is terminated with a null character value of 0.

The final procedure ASMBL is a dummy added to the end of the program so that a trial run of the BASIC section can be carried out to test the first part of the program. When run the screen should look like the screen dump given here; when an address was requested I just pressed RETURN. The next part of this article will look at the reason for all the above; the assembly code itself.

List of addressing variables

E% end address of code segment
F% start address of code segment
P% program counter
V% working pointer for W%
W% start address of general work space
X% End address of general work space
Y% working pointer for Z%
Z% start address of zero page space

List of operating system addresses

osword%	**osword subroutine**
osbyte%	**osbyte subroutine**
osnewl%	**newline subroutine**
oswrch%	**write character**
osasci%	**write character and newline**
scrnmode%	**os current screen mode**

List of general addresses and constants

page%	**value of PAGE in target**
messages%	**start of message block**
msgfnd%	**start of initial message**
page1%	**listing window string**
inbuf%	**start of input buffer**
parm%	**parameter block for input**
conv%	**decimal conversion table**
modes%	**table of screen lengths**
elktab%	**table of Electron keywords**

List of page zero addresses

msg%	**print s/r string pointer**
line%	**line index pointer**
offset%	**input buffer offset value**
lzflag%	**leading zero control flag**
sums%	**address of calculation space**

SERVICES

ORDERING INSTRUCTIONS

Order back issues and subscription from Infonet Ltd. Order software and binders from ASP Reader Services. Please make out separate cheques to speed the progress of your order.

BACK ISSUES

Issues two to five are available in full colour magazine and floppy disk.

Disk User Two:

Ants!
Expert investigator
World Wars
Automatic Disk Catalogue Utilities
Mode 7 machine code
Software Manager
Guide to essential games on disk
Operating with OSWORD
Which RAM disk?

Disk User Four:

SIDE 1:

Introduction to ADFS
Blockade
Collectors Items
Three dimensional graphics
Random Access Tutorial
Music Tutor
Interceptor utilities
Disk repair
Repton screen printer
New Repton 3 screens

SIDE 2:

Full Micronet demo

Disk User Five:

How your disk drive works
Program transfer
Arkenoid/Thrust cheats
Ravenskull screen maps
Magic Wall puzzle
Save Your Bacon
Disk User theme tune
Disk routine library
Overlay techniques
Spock's Logical disk menu
Toolkit character editor

Disk User Six:

Clip art
Psychebrot mandelbrots
Graphics from BASIC
Bobber arcade game
Martian Nim strategy
Pixel Perfect – zoom editor
Schizoscreenia – double screen
Collectors Items
Zoom Lens – enlarge or reduce screen areas
Hot Key – instant access to the OS

Back issues of Disk User cost £2.95 and £0.50 postage and packing. Disk User back issues and subscriptions are sent in special packaging to protect your software.

Send your remittance to:
INFONET LTD
5 River Park Estate
Berkhamsted
Herts HP4 1HL.
Tel: 04427 76661-4 for VISA and ACCESS credit card orders.

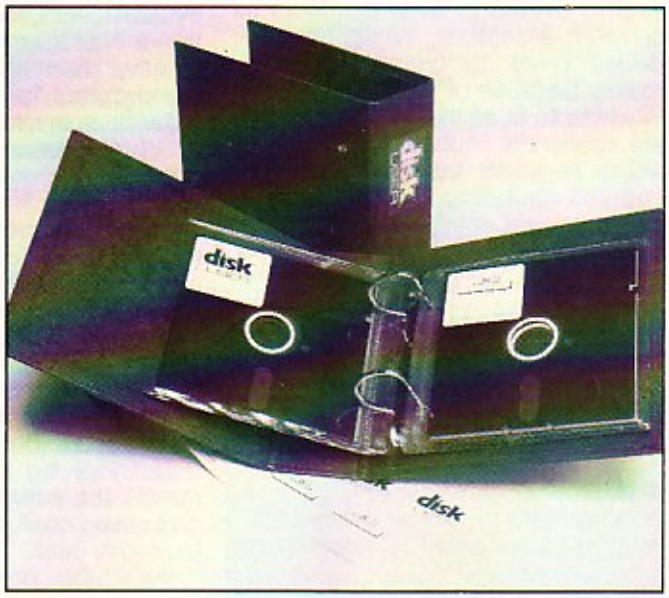
BACK ISSUES

Disk User Seven

System Wadgebury arcade game
Background print utility
Procedure library manager
How to write a hit game (routines)
Superfont printer driver and character generator
Collector's Items
*CLOSE utility
*FORMAT for personalised disk formatting

Disk User Eight

Tracer database
Ecosoft superlife game
Extended disk editor
Autorun for BASIC programs
Hyperdriver NLQ font demo
Easypoke advanced cheating for games players!



SUBSCRIPTIONS

SUBSCRIBE NOW!
We've made it easy to subscribe to Disk User. You can choose from a 6 monthly or 12 monthly subscription. Twelve disks full of quality software PLUS colourful magazine for just £35.40.

	6 months	12 months
UK	£17.70	£35.40
Europe	£21.60	£42.00
Middle East	£21.70	£43.40
Far East	£23.20	£46.40
Rest of the World	£22.00 or USA $32.00	£44.00 or USA $64.00

Airmail rates given on request.

BINDERS

Organise and protect your disks with Disk User disk binders and data disks. Why not keep your Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder contains 10 disk sleeves to organise, and protect, your program disks.
And we've produced special Disk User data disks to match. The Disk User logo immediately identifies your Disk User program disks and there's room to title them and document the relevant magazine details.
Prices are as follows:

Disk User disk binder £4.95, which includes all postage and packaging
Order product number BDAU1
Disk User disk binder including 10 Disk User data disks £9.95, which includes all postage and packaging
Order product number BDAU2

PRODUCT CODE NUMBERS

PROGRAM TITLE	BBC B/B+	MSTR 128	COMPACT	ELECTRON
Global View	DB01	DB16	DB33	DE06
Graphics Pack 1	DB03	DB17	DB37	DE01
Musician	DB06			
Mode 7 Utilities	DB12			
Adventurescape III	DB20	DB28	DB35	
Combat Zone	DB21	DB30	DB32	
Easy Font	DB26	DB39	DB44	
ADFS Menu		DB27	DB31	
Graphics pack 2	DB29	DB34	DB38	DE29
Colour Ikon	DB36			
A&B Bibliography	DB40	DB53		
Bibliography Upgrade	DB55	DB54		
Statistics	DB41	DB42	DB43	
Home Office	DB46		DE46	
Graphics Constr. Set	DB49	DB50	DB51	
Master Only		DB56	DB57	
Getting Into Assembler	DB61	DB62	DB63	
Wordwise Plus	DB64	DB65	DB66	
Sideways RAM	DB67	DB68	DB69	
Pascal	DB70	DB71	DB72	
Adventure Special	DB73	DB74	DB75	
Graphics Special	DB76	DB77	DB78	
Skywatcher	DB79	DB80	DB81	
Online!	DB82			
Disk User One	DU1			
Disk User Three	DU3			

AVAILABILITY AND PRICE

ALL SOFTWARE AVAILABLE ON 80 TRACK (ADFS LARGE) FORMAT. ADFS OWNERS ENQUIRE FOR 40 TRACK AVAILABILITY.

Name	Product	Price
Global View	DB01	£10.00
ADFS	DB16	£10.00
Graphics Pack 1	DB03	£10.00
ADFS	DB17	£10.00
Musician	DB06	£8.00
Mode 7 Utilities	DB12	£8.00
Adventurescape III	DB20	£15.00
ADFS	DB28	£15.00
Combat Zone	DB21	£8.00
ADFS	DB30	£8.00
Easy Font	DB26	£10.00
ADFS	DB39	£10.00
ADFS Menu	DB27	£12.00
Graphics pack 2	DB29	£10.00
ADFS	DB34	£10.00
Colour Ikon	DB36	£8.00
A&B Bibliography	DB40	£15.00
ADFS	DB53	£15.00
Statistics	DB41	£10.00
ADFS	DB42	£10.00
Home Office	DB46	£10.00
Graphics Construction Set	DB49	£10.00
ADFS	DB50	£10.00
Bibliography Upgrade	DB55	£6.00
Master Only ADFS	DB56	£8.00
A&B '87	DB58	£8.00
ADFS	DB59	£8.00
Getting Into Assembler	DB61	£10.00
ADFS	DB62	£10.00
Getting Into Wordwise Plus	DB64	£10.00
ADFS	DB65	£10.00
Getting Into Sideways RAM	DB67	£10.00
ADFS	DB68	£10.00
Getting Into Pascal	DB70	£10.00
ADFS	DB71	£10.00
Adventure Special	DB73	£22.00
ADFS	DB74	£22.00
Graphics Special	DB76	£22.00
ADFS	DB77	£22.00
Skywatcher	DB79	£8.00
ADFS	DB80	£8.00
Online!	DB82	£10.00
Disk User One	DU1	£4.00
Disk User Three	DU3	£4.00

DISK USER NO.1

Available only as a software product, floppy disk and laserset manual. Programs are:

Collectors Items
Barmy Butterflies
Cube of Zoth
Auto Cataloguer
Disk Examiner
Karate Kid

Order product number DU1

DISK USER NO.3

Cholo cheat editor
Auto Save
Tazman
Dual Catalogue
Quick Break
Squash program
Count with Teddy
Quick Copy
Scientific calculator
Collectors Items

Order product number DU3

SOFTWARE

9 Hall Road, Maylands Wood Estate, Hemel Hempstead, Herts HP2 7BH Tel: 0442 41221

CHECK WITH THE TABLES TO SEE IF OUR SOFTWARE IS AVAILABLE FOR YOUR COMPUTER. PLEASE INDICATE YOUR REQUIRED PRODUCT NAME, NUMBER AND TRACK FORMAT IN THE ORDER FORM

Software is only available on the formats recorded in the tables.

All prices include VAT and postage and packing. Please make cheques/postal orders payable to Arqus Specialist Publications Ltd and print your name and address clearly on the back of your cheque. Telephone orders are taken for Visa and Access. All software is sent first class. Prices include postage and packing for European destinations. Please include £1.00 postage and packing (airmail) for other overseas order. Please allow 28 days for delivery.

ORDER FORM

PRODUCT NAME	PRODUCT NUMBER	DISC Please specify 40 or 80 track	QUANTITY	PRICE
			TOTAL	

Name.....

.....

Address.....

.....

.....

Postcode

Signed.....

.....

Date.....

.....

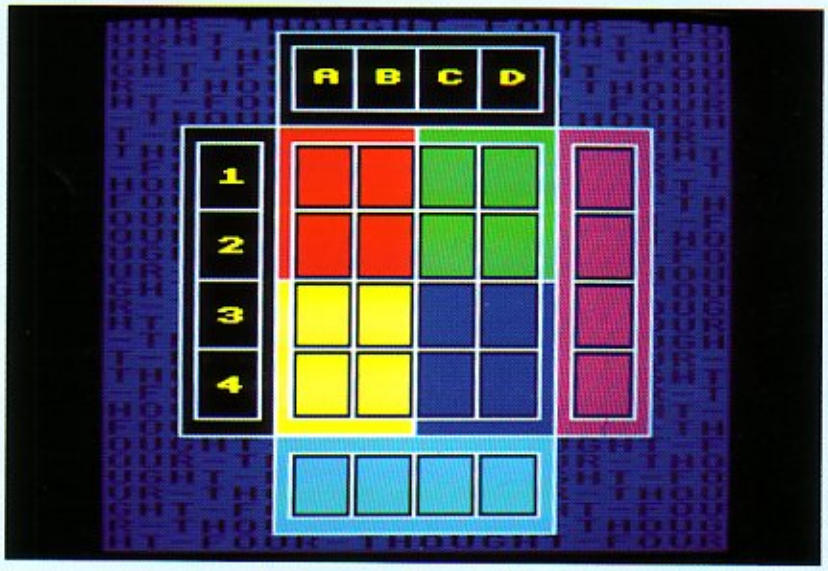
I enclose a cheque/postal order made payable to ASP for £.....

Please complete the form in block capitals and send it with your remittance to Software Sale, Reader Services, 9 Hall Road, Maylands Wood Estate, Hemel Hempstead, Herts HP2 7BH. Tel: 0442 41221. You can direct technical enquiries about software to Phoenix Software on 0733 53355.

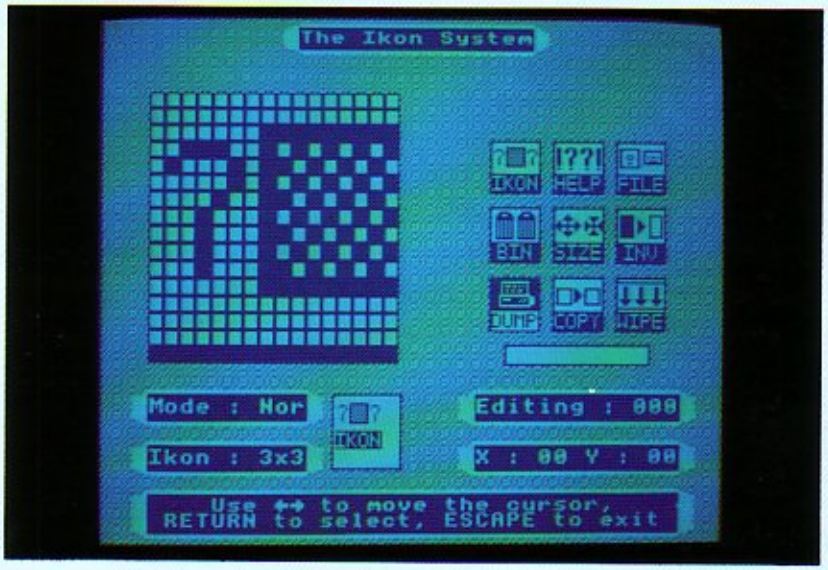
COMING SOON!

TO A SCREEN NEAR YOU

FAST AND FURIOUS – Action all the way with Terry Blunt's humourous gameplay in ORLOND'S MEANIES. Lots of surprises as you search the weirdest caverns in arcade adventuring!



BRAIN BOGGLING – Logic puzzles to boost your IQ, and your ego! Test friends and family.



IKON SYSTEM – Easy to setup professional WIMP environment for your own programs.

Other great programs planned for the coming months include comms, graphics, graphs and charts, computer aided learning, specials on educational software and adventures. Disk User, now showing on your very own computer screen. New programme every month.

REPTON

THRU TIME


PREHISTORIC REPTON


PRESENT DAY REPTON


EGYPTIAN REPTON


VICTORIAN REPTON


FUTURE REPTON

REPTON — THE TIME TRAVELLER

Where did Repton come from? Does he have an Egyptian mummy? Is he the real Jack the Repper? Where is he now? Where is he going to?! We have been inundated with these and many other questions about our lovable hero. Now all is revealed in 40 new screens that vividly reveal Repton's evolution from prehistory to the future.

The PREHISTORIC screens with their caves, mountains and volcanoes, have Repton battling against pterodactyls and dinosaurs to collect edible berries. In EGYPTIAN times, Repton chases around pyramids and sphinxes collecting scrolls and meeting a mummy or two! Amidst the smog and grime of VICTORIAN times, Repton collects gold coins — but can he avoid the police and the gallows? PRESENT DAY Repton has even greater dangers to face as he rushes around the city's jungle of parking meters and skyscrapers, looking for cans of cola. If the gangster's machine gun doesn't get him, the traffic wardens will. Amazingly Repton may make it to the FUTURE. As he zooms through space collecting crystals, the Martians give chase. Will he succeed or disappear forever into the infinite depths of a black hole? Only you can help Repton Thru Time!


THE SCREEN EDITOR

Each copy of Repton Thru Time includes:
• the Repton 3 main program • the Repton 3 editor and • the 40 new game screens.

BBC Micro Cassette.....**£6.95**	Acorn Electron Cassette.....**£6.95**
BBC Micro 5¼" Disc.....**£7.95**	BBC Master Compact 3½" Disc..**£9.95**

(Compatible with the BBC B, B+ and Master Series computers).

The screen pictures show the BBC Micro version of the game.


THE CHARACTER EDITOR

PRIZE COMPETITION

If you complete all 40 screens of Repton Thru Time without using passwords, you can enter our competition. A draw will be made from all the correct entries received to select 25 prize winners. Each will receive an EGYPTIAN REPTON Cuddly Toy and a signed certificate.

Closing date: 30th September, 1988.


REPTON ORIGINAL
Cuddly Toy

SPECIAL OFFER — NEW!

A unique range of Repton Cuddly Toys is now available. Each Cuddly Toy is about 24 inches high and has a realistic face and features corresponding to the particular Repton character. They are only available from Superior Software.

REPTON ORIGINAL **£9.95**
ARCTIC REPTON **£9.95**
TEENAGE REPTON **£9.95**
Postage and packing free.

SUPERIOR SOFTWARE Limited

ACORNSOFT

Dept. RT3, Regent House, Skinner Lane, Leeds LS7 1AX. Telephone: 0532 459453.

Please make all cheques payable to "Superior Software Ltd".

ACCESS VISA

24 HOUR TELEPHONE ANSWERING SERVICE FOR ORDERS

OUR GUARANTEE
- All mail orders are despatched within 24 hours by first-class post.
- Postage and packing is free.
- Faulty cassettes and discs will be replaced immediately. (This does not affect your statutory rights).