

YOUR
COMMODORE

AN ARGUS SPECIALIST PUBLICATION

June 1989 £1.50

BASIC



ANALYSER

Basic programs revealed

REVIEWED

▶ Oxford Basic ▶ Sketchpad 128

GAMES REVIEWED

▶ Deadenders ▶ Middle Earth ▶ 3D Pool
▶ Denaris ▶ The Deep

**UNBEATABLE
PROGRAMS**

▶ 6510 Assembler ▶ Line Input
▶ Help Screen ▶ Retriever



DATTEL ELECTRONICS



256K SUPEROM EXPANDER

- Instant installation from a cartridge which drops up to 512K DRAM into your EPROM programmer.
- Program your own programs using our EPROM programmer.
- No need to have lots of cartridges - just make a cartridge from the Superom system.
- Fully secure system on permanent carrier.
- Perfect way to store software programs.
- Complete DRAM protection feature will take your data programs - from a cartridge to your three half-volunteer cartridges. (Cartridge readers required).
- Supports 256K, 512K, 1024K EPROMS.
- No board replacement costs.
- No board operating systems - no programs to load.

ONLY £29.99



RESET CARTRIDGE

- Compatible reset function.
- EPROM reset pins changes type reset function offered by various set out cartridges.
- Means reset is called "cartridge" programmer.
- Add pins for magazine etc.
- Fits in cartridge port.
- Simple plug fit.

ONLY £5.99



3 SLOT MOTHERBOARD

SAFETY RESET & 1624 OR 2048 EXPANDED ROM

- Accepts 3 cartridges.
- Holds in, but easy out.
- Outward safety flap.
- Fully featured.
- High grade PCB.
- Good finish.

ONLY £16.99



MIDI 64

FULL MIDI INTERFACE FOR THE SPECTRUM AT A BARGAIN PRICE.

- MIDI to MIDI (16-bit) or MIDI out.
- Compatible with most leading software packages.

ONLY £29.99



DIGITAL SOUND SAMPLER

The new sampler allows you to record any sound digitally into memory & then copy it into accompanying software.

- Playback through hardware with automatic timing modulation.
- New volume and pitch control enables to produce outstanding effects.
- Full 16 bit to 16-bit digital conversion.
- 1024K compatible with external memory (16K - 64K).
- 1024 memory store facilities and three stages of compression.
- Load the data into tape or disk format.
- Powerful response with editing facilities.
- Load/store memory.
- Up to 8 samples in memory at one time.
- Easy to use playback control.
- Complete self-test/feedback package.

ONLY £49.99

COM-DRUM

SEMI-DISK SYSTEM

- Store your own from digital sound recorder. Data is digital drum system.
- 16 digital drum records in memory of one file.
- Complete with 3 drum files.
- Real drum sounds - not synthesized.
- Volume control from software with real 16 bit digital filter.
- Full editing, three drums.
- Output to MIDI or through EP speaker.
- Load memory facilities.

ONLY £9.99

WRITE TAPES ON DISK

COM-DRUM EDITOR

- 16 drum records supplied on disk to enable you to construct your own drum file.
- No program needed, supported with a Com-Drum or reader tape file.
- Load/store facilities.

ONLY £4.99 DISK ONLY

DATA RECORDER

- Quality recording compatible data conversion.
- Power control, 1024K - 40K.
- Suitable for the UK.
- Good case for splash delivery.

ONLY £24.99



DEEP SCAN BURST HIBBLER™

- The most powerful disk address available worldwide, it can pinpoint any data in virtually any hard disk system - a software package in a portable case to connect the IBM & 2025/2027 to the 2024 and format tape.
- What gives Deep Hibbler its power? Unconventional address logic to locate the data from the disk before it can be read is using the write data - write non-readable data to conventional library and load. Then WRITE location data to the ODD track via the parallel cable without the need to duplicate it as you get a positive copy of the original.
- Will address up to 41 sectors.
- Copy a whole disk to sector 0 instantly + Full instructions.
- Simply update - no re-formatting the disk - no re-formatting needed.
- Full on 1/4 inch disk.
- No need to buy parallel reader if you have IBM Storage Display etc.
- Data has 100% accuracy transfer rate when used.
- Works to allow excellent use of IBM 2025/2027 from hardware to software reader as an "off reader" with address, 1 to 8 data copy-passes, the copy - 400 copy pass. All 16 bit error free + error detection improvement package. Full 16 bit on 1/4 inch. Data (able to be zero address) needed to write, full 16 bit address search. Full 16 bit data loading (average you are guaranteed to, made to be able program to use it).

ONLY £24.99 COMPLETE

SOFTWARE ONLY £12.99

CABLE ONLY £14.99

DISK 100% SUPPORT KIT

DATA RECOVERY AND backup software to maintain the disk data to produce independent copy/restore. It is designed to make copies of disks containing the original version of the original system or data recovery.



ROBOTEK 64

ROMS & ROMS CONTROL

1024K ROM

- A compact chassis - each with 1024K ROM.
- A large chassis - each 1024K ROM.
- Software input with full 16 bit conversion.
- Video input for video control.
- Software features - fast ready/refresh/erase/restore/verify/erase digital readout etc.

ONLY £39.99

CALL US FOR INFORMATION PLEASE

TEL: 01924 444444 FAX: 01924 444444



TOOLKIT IV



THE ULTIMATE DISK TOOLKIT FOR THE 1545/1541.

- A disk reader to do anything you do for the reader disk reader. Suitable for low cost hardware that need to use.
- DATA RECOVERY PG - Read & write any track in sector including extra & non-readable sectors. Speed managed restore. Includes automatic read verify.
- REPAIR/REP BOTTOM - Operates in 16 bytes as a backup substitution including all bytes & format. Repairs & writes extra sectors. A double zip. Includes restore. Also will zip entire disk zip.
- DATA CHECK - Disk integrity. Recovery low state. Memory file read load addresses. Operations any the program always runs the disk in memory or 16 bytes including non-readable addresses. Includes data.
- FAST FILE COPY - Restores the copy. Works up to 16 bytes control copies.
- FAST FILE COPY - Copy an entire disk to 3 sectors in one writing operation.
- FILE COMPARISON - Use compare software programs for up to 16 files. Best disk copies. Includes programs that can format.
- COPY/VERIFY - An actual format or restore disk or format any individual track on both disks 1545 & 1541. An extra copy of all parameters to create a restore output disk format.
- ERROR EJECT - Quickly find & remove all bad sectors including extra & non-readable sectors. Includes a "LAST" disk from it to all. Includes data sector check & allows you to address all necessary parameters.

ONLY £9.99

DATTEL ELECTRONICS



TURBO ROM II

NEW CLONEMASTER

- The most efficient tape to tape copy method available
- Makes perfect backups of your tapes easily & efficiently!
- No user knowledge needed or an
- No more 194 tape rewinding normally done by the program is made a positive operation that cannot be done by the user because the tape is held by the device for the entire copy process.
- L.L.D. indicator shows when data is being transferred via sound sensitive tape rewinding.
- Works with almost any program including system loaders, makes a copy every second type machine. Supports tapes to 100,000 compatible data transfers.
- Makes tapes that are readable & usable. Forward to the other side of the tape.
- You can even make a backup while you are loading the program.
- This is a total hardware solution - no programs to load - just insert and start!

ONLY £9.99

TURBO ROM II IS A REPLACEMENT FOR THE ACTUAL ROMS INSIDE YOUR 64. IT PROVIDES SUPERFAST LOADING/BOUNDS.

- Loads more programs in 1/4 time normal speed
- Saves at 2-4 times normal speed or
- Improved ROM support handling 10 bits, format.
- Programs faster tape read times, etc., etc.
- Returns to normal speed in a flash at a touch.
- Easy to use with the simple
- Flash speed 1/2-loads
- No tape loss
- Fixed to maintain an existing working system (also 64's the old ROM may have to be changed)

ONLY £14.99



SMART CART

- 100% portable ROM
- Contains backup tape up to 4 pages
- Simply load the program, you require a clean disk the entire. The cartidge can then be removed just like a ROM cartidge
- Make your own cartidges including custom tapes - almost 65536 bytes. Not limited to a 16 page
- Some knowledge of tape is helpful but not essential
- 100% also open for special programming techniques

ONLY £29.99

RAM DISK

- Turn your floppy disk into a RAM disk
- 1MB of instant storage via the
- 64's RAM
- Includes commands - read, write, directory, search.
- Program data retained when computer is powered off
- Full command set - with functions.

ONLY £9.99

BLAZING PADDLES & COMPLETE LIGHTNING/GRAPHICS ILLUSTRATION PACKETS

- A fully new screen graphics package of a quality which should cost twice, more than
- Complete with a free optional lightning system for play game screens
- Screens can be saved into Amiga Display & added with Blazing Paddles
- Blazing Paddles will also work with other local device including - Apollon, Mars, Graphics Station, Terminal, etc.



EPROMMER 64™

- A top quality - easy to use screen programmer for the 64/104
- Fully uses device reference hardware package status programming/programming/loading/erasing EPROMs completely
- 800 programs 37.5K, 175K, 37.12K & 20000 chips - 10.5, 21 to 25 sides
- Fits into any port for maximum compatibility with cartidges/Eproms based on
- Full device options - all features covered like device identification.
- We believe EPROMMER is the most comprehensive, most flexible & best value for money programmer available for the 64/104
- Most computers for Eproms based cartidges/Eproms require one second operation to insert any screen based program
- Comes complete with instructions plus the cartidge holder.

ONLY £39.99 COMPLETE



BLUE CHIP DRIVE



- External power supply for real operation
- Drive runs at 57.1 double data mode (1/4 inch)
- Fully 64/128 compatible
- This drive is now probably the most compatible drive available for the Commodore - there is no other Commodore own drive
- Supports complete file set - all other to use
- At last - a top quality drive at a reasonable price

ONLY £129.99 ADD £5.00 FOR COMMON DELIVERY

ULTRA CRUNCHER™

- The ultimate cartridge based program compressor
- Compress by up to 100%
- A compressing program no one can match
- Free loading/ saving routines
- Full ROM program handling (as needed)

ONLY £12.99

see cartidges

Available as chip only for 256K Supermax Board & r

ONLY £7.99

FAST HACK'EM™

- Multi Module Disk Winder - all on the one disk
- Single Disk Winder - copy an entire disk in just 10 minutes
- Super fast File Copy - typically copies an average file in under 10 seconds

At other products to such a great "All Rounder" or this incredible low price

ONLY £9.99

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE

0782 741700
9.00am - 5.00pm
Mon-Fri

BY POST

Send cheque/PO/GBP payable to
"Datatel Electronics"

FAX

0782 744332
OR 0800-800 1000
9.00am - 5.00pm
Mon-Fri

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

PLEASED WITH YOU - These reserves price by telephone price to rise.

DATTEL ELECTRONICS

DATTEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE
GUSAN ROAD, FENTON, STONE-ON-TRENT, ENGLAND.

SALES ONLY
0782 744337

TECHNICAL ONLY
0782 744334

CONTENTS

VOLUME 5
NUMBER 9



Super Cycle



Flightmaster



3D Pool

FEATURES

- | | | | |
|---|----|--|----|
| • Ideal Home
Who did it at this famous exhibition | 14 | • Spy Hunter
Popular arcade action | 42 |
| • Oxford Basic
Expand your C64's Basic with this excellent product | 22 | • Demaris
Superb arcade action | 43 |
| • Deaddenders
No prizes for guessing what program this is a spoof of! | 32 | • 3D Pool
Computer pool as it should really look | 44 |
| • War in Middle Earth
Adventuring in Tolkien's land | 32 | • Bargain Bucket
The pick of this month's budget games | 45 |
| • Super Cycle
Motor bike racing at its best | 40 | • Disk Editing
Delve further into your disk drive | 50 |
| • The Deep
Under water arcade action | 40 | • Epson SQ-2500
We conclude our look at 24-pin printers | 72 |
| • Grand Prix Circuit
—Try your hand at Formula 1 racing | 41 | • Sketch Pad 128
Does drawing in 80 columns really work? | 74 |
| • Infiltrator
A second chance to fly with
Jumbo 'Baky' McGibben | 42 | | |

REGULARS

- | | | | |
|---|----|--|----|
| • Date Statements
More news from the world of Commodore | 6 | • Extending Basic
Add your own commands to your C64 | 66 |
| • Letters
Your chance to let us know what you think | 10 | • Tech Troubles
Your technical queries answered | 70 |
| • 1st Steps
Help for new programmers | 12 | • Routine Programming
Another handy subroutine to add to
your growing library | 76 |
| • Flow of Ideas
Write your own directory reader | 29 | • Software for Sale
If you don't want to type it — Buy it! | 78 |
| • Listings
This month's programs | 55 | | |

PROGRAMS

- | | | | |
|---|----|---|----|
| • Returner
Call up your disk menu with ease | 18 | • Help Screens
Put your C16/64 HELP key to good use | 38 |
| • Line Input
Improve on Basic's INPUT command | 25 | • ASM Assembler
A powerful assembler for C64 owners | 47 |
| • Program Analysis
Find out just what's happening inside
your C64 programs | 34 | | |

Editor: Steve Cook
Deputy Editor:
Paul Whalington
Technical Editor: Paul Dan
Group Editor:
Mark Webb
Advertisement Manager:
Paul Kinsman
Ad-Copy Control:
Kath Taylor
Artist: Alan Bartholow
Designer: Neil Swettenham
Origination: Abbey
Printer: Chase Webb

Your Commodore incorporating Year 94 is a monthly magazine appearing on the first Friday of each month. Argus Specialist Publications Limited, Editorial and Advertisement office, Four Commodore, Argus House, Broadway Way, Ipswich, Hampshire IP1 7ST. Telephone: 0447-84501. Subscription rates upon application to Your Commodore Subscriptions Department, Innes Ltd, 5 River Park Estate, Beckenham, Kent BR3 1JH, U.S.A. Subscription Agent: Wise Ltd Worldwide Publications, 404 West 20th Street, Torrance CA 90501 U.S.A.

**ARGUS
PRESS
GROUP**

The contents of this publication including all articles, designs, drawings and programs and all copyright and other intellectual property rights therein belong to Argus Specialist Publications Limited. All rights conferred by the Law of Copyright and other intellectual property rights and by virtue of conventional copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1989, Distribution MM Distribution, 6 Leighton Court Road, London SW16 2PG. Printed by Chase Webb, Plymouth. Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published for errors we cannot be held responsible for any errors that do occur.

ISSN 0269-9277

ABC
0269-9277

Free The Spirit

If there's one company above all the others that's trying to make your C128 more like an Amiga, it's Free Spirit Software. This American company seems determined to produce good, high-quality graphics packages for the C128. Their latest offering is Poster Maker 128, with which you can create posters ranging in size up to 15 pages. You need 80 col display and a C128D or C128 with the 64K video Ram upgrade. You can import Basic 8, Sketchpad 128 or Spectrum 128 files. There is also an option for reducing, thereby enabling you to perform clip art.

Touchline Free Spirit Software, PO Box 128, 38 Noble St, Astoria, PA 19010 Tel: 215-623-5686.

Spirit of Adventure

Good news for Adventure Fans. After much public pressure, Innovative Software has revisited the classic Graphic Adventure Creator. The new release will allow "Stand alone disk accessed" adventures to be produced. This will mean that you can now create adventures of incredible size and complexity, restricted only by disk size. The release date for this exciting product is April 1989. It will retail at £29.95, but existing GAC holders may upgrade for a mere £11.95.

Touchline Innovative Software Ltd, Zephyr One, Calver Park, Aldermaston, Berks, RG7 4QH. Tel: 07356 77268.

Data Statements

Chris Payne,
marketing manager
at Database, will
show the AMS range



Stop Press!

Database Software have just acquired the rights to two well established DTP packages for the C64 and C128. Both titles originate from AMS. The titles are Stop Press, and Stop Press with mouse, costing £39.99 and £69.99 respectively. Productivity software is a little thin on the ground for the 64/128, so well done Database.

Touchline Database Software, Europa House, Aldington Park, Abingdon, Oxon, OX9 4NP. Tel: 0625 870888.

Viva Geos

There is no quicker or easier way to present data than with a graph or chart, and GeoChart is a simple and inexpensive way to produce good-looking graphs and charts...etc etc, or so the blurb goes. Bentley Software have added yet another program to their ever growing array of Geos utilities. GeoChart is ideal for anyone that wants professional-looking documents and charts, but can't spare the time needed to plan and draw graphic illustrations.

Like all the Geos appli-

cautions, GeoChart is the 'What you see, is what you get' format. That is, pull down menus, windows and icons. GeoChart is for the C64 and C128 owners.

Touchline: Berkeley Software Works, 2138 Shattuck Avenue, Berkeley, California 94704. Tel: 415 644 0881.



Hubba Hubba!

Electronic Arts have enhanced that popular game from Bullfrog, Populous, by adding a new sound track from Rob Hubbard.

Rob, who left the UK in '88 to work in sunny California, has since been working on a major project. The work involves creating a code package that will allow real soundtracks to be produced. The drivers, we are told, will be usable even and over again for any product that requires a soundtrack.

Touchline: Electronic Arts, Langley Business Centre, 11/19 Station Road, Langley, Berkshire, SL1 8TN. Tel: 0514 69441.



Rob Hubbard - artist in exile

Watch Out

Microprose are about to launch an attack on your wrists with Navy Seal, a 64 game packed with action. You are a member of the Elite US Navy force, the sea-air-land commandos, who are experts in sabotage, demolition, reconnaissance and infiltration. So that you don't

miss the boat, a Navy Seal Digital Watch (Right) will be given away with every copy purchased. The release date should be in early April, at an expected cost of £12.99 ex disk.

Touchline: Microprose, 2 Market place, Farnham, Surrey, GU10 2EJ. Tel: 0483 34126.



Make a Stand

Have you noticed how computer furniture never gets a mention? Well, here goes. From MDS of Ramoth Hampstead comes the MDP5 150 printer stand. Of all the peripherals, printers seem to cause the most positional headaches, but this new stand could solve all your problems (although the £204 price tag may put you off). It is, and I quote "uniquely designed to achieve maximum efficiency, and part of a comprehensive modular desk

system which measures 150mm wide, 710mm high and 750mm deep. It is an 18 inch tall unit on the top, making the unit suitable for both top and bottom feed printers". The other good point is the electronic cable management system incorporated into this unit - no more wires and cables need get in the way.

Touchline: MDS Industries (UK) Ltd, Factory 1, Elm-rove road, Ramoth Hampstead, MF9 6QS. Tel: 0442 251905.



Change At Waterliss

Persons of Restricted Growth everywhere, here's your chance to get your own back! The latest offering from SSI, available via US Gold, allows you to assume the persona of 'Sharris' Bonaparte and change the course of history. Called *Battles of Napoleon* - a construction set, this program allows you to either fight authentic Napoleonic wars, or simply create your own. We are assured that these battles are historically recreated in every detail, with cavalry charges, bombardments, skirmishes and formations. Priced at £24.99 for the CSM, the game offers good value.

Fourfile: U.S. Gold, Units 2/3, Wyford Way, Bedford, Bedfordshire, MK7 7AE. Tel: 021 356 1388.

Special Relationship

News has reached us of a bilateral deal between Kixbits and Action Wars,

the outcome of which should prove beneficial to all you game players. The deal means that Kixbits will market AW software here in the U.K., and AW will market theirs over there. The first product off the press was *Proton*, which was released at March 16th for the Atari ST and Amiga.

The Hammy Hamsters

Fame and fortune can make people do weird and wonderful things. Howard Hughes became a recluse, a certain Mr. Klein bought a motor factory, and Mr. Wyman managed every man's dream. Now it's the turn of the Darling family. Not content with producing both possible budget software, they have recently broken out into song. To help promote their latest release, *Kick Star*, they've formed a group which, for reasons best known to themselves, is called The Hamsters. Judging by the picture, I know which Hamster I'd like as a pet!

Fourfile: Code Masters

Software Co Ltd, Lower Farm House, Stoneyslope, Southon, Warwick, CV30 0SU. Tel: 0608 894732.

Amiga Airlines

Buyers of Commodore's new 'Amiga Air Miles Pack' will be receiving free air miles along with their machines. The package offers an Amiga 500, a TV modulator, three new games (*Roger Rabbit*, *Nylnor* and *Star Ray*), computerisable point packages and 500 free air miles, all for £499.99. The offer will give Commodore customers the chance to book return flights to a number of European cities, including Paris and Amsterdam. All flight vouchers will be valid until March 1998, and can be easily redeemed via the special Commodore Air Miles Hotline, as well as local travel agents.

Fourfile: Commodore Business Machines (UK) Ltd, Commodore House, The Whiteback, Gardner Road, Macclesfield, Cheshire SK10 7LE. Tel: (0625) 73455.

Hit The Jackpot At The Commodore Show!

Commodore are giving away £200 worth of computer equipment to the Amiga user with the most innovative and unusual use for his machine. A nationwide search has been launched to discover as many weird and wonderful applications for the Amiga as possible, and winners will be judged by a panel of experts at the Commodore Computer Show, which opens at the NEC, Birmingham on June 2.

The show itself has been revamped by Database Exhibitions, and will now include seminars on making music, graphics, Commodore magazines (including, of course, PC) and game writing. 'Beat The Author' competitions are also planned, in which punters will be able to take on 'named' authors and try to beat them at their own game. But one of the big highlights of the show will undoubtedly be the innovation competition.

Commodore has already discovered a wide variety of uses for the Amiga, from bird-watching to betting on horses, and has even heard of one user who works out menus for sodas on his machine! If you think you can match these innovative applications, all you need to do is write a brief description - no more than 500 words - and send it in, along with any necessary support material on disk, to:

The Editor,
Your Commodore,
Argus House,
Brandy Way,
Hemel Hempstead HP2 7ST.

Entries should reach us before May 31st.



Mailbag

Your chance to air your views in Your Commodore

I have just received a Commodore PET 3032 Computer from a friend. With it was a 8050 Disk Drive and 4022(F) Printer. I was wondering if you or any of your readers had any information on software availability. Please send any information to me, thank you.
R.M. Crook, 28 Grange Road, Bramley Cross, Bolton, BL7 9AU

I was recently given a PET 8032, a 8050 Dual Drive and a 4022 Printer together with four good software packages complete with manuals and dangles, where necessary. I only want to use the PET for wordprocessing, and so I am sending out an appeal. Is there anyone out there that has a program for the 8032 that they don't want? If so, please contact me at my address. Thank you.

J. Simpson, 4 Gullinstones Avenue, Dunbar, East Lothian

Come on new versions, we know there are pretty serious machines, but there must be someone out there who can help!

Dear VC

Is there anything that can be done with old printer

ribbons? I feel that it's a great waste to throw them away.

Could one possibly recut/line them with ink in some way? Ink, which ink should be used, or can I send them someone to be done?
J. Osborn, Luton, Beds

Our reply

The answer, John, is yes. Printer ribbons can be recut. If you get a copy of Micro Computer Mart, you will find a firm advertising this service every week.

Dear VC

With three CDD disks, I have had some loading problems. The disk has not with yourselves, but rather with the magazines, when they stack magazines on top of each other, I have found that "Validating" the disk beforehand seems to cure this problem.

I don't know what proportion of failures are returned, but this Tip may help in some way.

J.F. Padlock Wood, Tunbridge, Kent

Our reply

Thanks for this helpful little tip. We do get some returned disks, and many of them are cured by this simple method.

Unfortunately, newspapers and bookshelves do not make how delicate disks are.

Dear VC

I have been comparing ever since the Commodore 64 was first produced. In this time I have bought various computer magazines, mostly multi-format, and I've noticed various discrepancies between reviews of the same game. For instance, *Tennis* received 50 out of 100 from one, but 97.5 from another.

The trouble with two completely different results is which one do I believe? Do I take the advice of magazine A and stay clear of it, or do I take B's advice and sell my TV so that I can buy it?

The best advice for anyone is to try and get into the reviewer's mind. Some reviewers set their hopes too high, and so if a game doesn't come up to their expectations, they give it a poor write up.

A recent review of *Thunderbolt* is a good example of this. One reviewer of this game said - "The graphics at this point, Level 1 stage 1, were very disappointing - the building tops are just stacked squares".

Personally, I think the graphics are very good. Okay, the first stage in each level is blocky, but you can't really expect hi-res graphics can you? It would be much too slow. Anyway, you've always got stages 2 and 3 in each level to make up for it.

Another annoying item is "Missing". You know the kind of thing - "The Q button doesn't work, so you have to use the F7 key". A great example of this can be found in the same review of *Thunderbolt*. The reviewer said "Unfortunately, it is hard to find any difference between cannon and air-to-

ground rockets." There are two possible reasons for this - (A) His/Her cassette disk/disk drive was faulty or (B) their copy of the game was faulty. These are the only possibilities, as my copy of the game is the opposite.

In conclusion then, what do you do if two reviews differ? The first option is to see if you can have a demonstration of the program, therefore allowing you to make up your own mind. The second option is to simply take a chance.

N. Kingsley, Chichester, West Sussex

Our reply

Thanks for such a long and critical letter. However, I think you're missing the point (as do a lot of people). Reviewers of software do not differ from any other form of reviewer, be it Films, Videos, Records, Food, Clothes Etc.

The whole point here is this - the comments passed are the comments of that particular person, NOT the consensus of a "professional critic". For example, I personally have never read *Elvis*, but for 1.3 million other users it is the best program ever.

Likewise, for me *Infocom* are the only people that produce decent adventures, but for thousands of people would not agree. Basically, one must remember that a reviewer's comments are his own personal comments. A review only really tells you what the objective is and how the game plays. There is only one person that can say whether it is good or bad, and that is you.

MIXING

Business

WITH

L, E, I, S, U, R, E,

WILL BE
A REAL

EDUCATION

Commodore computer show

Britain's brightest event for Commodore computer users is back! And there's more to see than ever before.

This show has three main themes covering some of the major uses to which Commodore machines are put. There are over 70 key companies who will be exhibiting their latest products, which means that just about everything that's new in the Commodore world will be on display!

Business

Many companies will be demonstrating their latest software and hardware specially designed to release the full business potential of Commodore computers.

As well as products for the C64 and Amiga series, you'll be able to try out applications for the pioneering Commodore PC-compatible micros.

And you'll also be able to attend seminars covering all aspects of using Commodore micros in your business.

Leisure

The C64 and Amiga computers are the most powerful 8 and 16 bit micros for producing fast-action, superb-quality games. The range of new software on show

Novotel Exhibition Complex,
HammerSmith, London W6
Friday to Sunday
June 2 to 4

Hammer Friday & Saturday: 10am-6pm Sunday

will demonstrate how these machines' power is continually being stretched, producing faster and even more addictive games with superb graphics.

If you're a keen game player, you'll find there's so much on offer at the show you're guaranteed a real treat!

Education

Commodore micros are now used as educational tools all over the country. With the development of BBC Basic on the Amiga and the advent of Desktop Video combining TV pictures with text and graphics, the range of educational applications is infinite.

At the show you'll see how the latest software

packages are making real breakthroughs in the educational sector, and be able to try them out for yourself.

Special Events

As well as special events and presentations, you'll also be able to meet some of your favourite celebrities, and maybe get a chance to talk with them about how they use micros in their work.

So for a great day out, whether you want to see what the future holds for Commodore computers, to buy the latest software or to get advice on specific applications, the Commodore show is the place to go. And if you send in the coupon today, we'll knock £1 off the price of each ticket!

For the first time we are offering a family ticket for just £11 allowing entry for two adults and two children - saving up to £7 off the usual entry price!

How To Get There

By Underground: HammerSmith (Piccadilly, Metropolitan & District)

By Bus: 260, 744, 710, 266, 30, 53, 73, 76

Car parking facilities available at the Novotel.

Advanced ticket order

Commodore
computer show

POST TO: Commodore Show Tickets,
PO Box C, Elmbridge Post,
South Mimms, EN8 9BA.

Please specify:

Adult tickets at £4 (save £1) £

Under 16s tickets at £3.50 (save 75p) £

Family ticket at £11 (save £7) £

Total £

Cheque payable to Desktop Exhibitions.

Please debit my Access/Visa card no.

NAME _____

ADDRESS _____

ENTRY DATE

SIGNATURE _____

Advertisement at show
£7 (incl. VAT)

£27.50 (incl. VAT)

Advanced ticket orders
must be received by
Wednesday, May 29

NAME _____

ADDRESS _____

ENTRY DATE

SIGNATURE _____

POSTAGE GUARANTEED BY FIRST CLASS MAIL

POSTAL ORDER NO. _____

POSTAL ORDER NO. _____

POSTAL ORDER NO. _____

POSTAL ORDER NO. _____

4583



How to make sense of Listings *By Norman Doyle*

Typing listings in can be much more of a trauma than it need be. At *Your Commodore*, we are aware that sometimes the process seems difficult or even nonsensical. Our Listings with their System Checker and special codings may ease the way, but sometimes more support is required.

Many programs are listed in several parts and each part uses the same line numbers. How is this possible and how do you set about typing them in?

First of all it should be explained that a set of these programs form a suite which operate in unison. The way to use them is to type in the first program and save it. Now the computer should be cleared by entering NEW or even by switching the computer off and on.

Proceeding to the second listing, the program is again typed in and saved. If you're starting the program on tape, you should ensure that this program is saved immediately after the first one. For disk users, it doesn't matter where it's saved as long as it's

on the same disk as the first program. After checking that all has been saved safely, the machine is again refreshed using NEW or the power switch.

Any subsequent programs should be treated in the same way as the second listing and the result will be a string of programs. One word of caution which should be heeded is that the programs should be saved under the name given in *Your Commodore* because each program occasionally uses one of these names to load in the next program in the string.

Suite Confusion

One question which we've often asked is why these multipart programs are needed. The answer is that they are a memory-saving or time-saving measure.

Many programs use redefined characters, machine code patches and sprites. The information for this has to be poked into position because they rarely occupy memory locations which Basic uses. This difficulty can be overcome in either of two ways.

The data can be poked into position each time the program is loaded and forms an integral part of the program. This has the dual disadvantage of the time the program takes to move the data in but, more importantly, reduces the amount of space for the master program itself.

Another way of achieving the same result is to use a series of set-up programs before the main program is loaded. The set-ups are loaded and run in turn and each one pokes information into memory where it cannot be touched by any subsequent loading actions. Consequently, once the program has done its job it is no longer needed and the program can even be removed by typing in NEW without raising the effect of the main program when it is eventually loaded.

One thing that you cannot do is to switch the computer off and on again! This action totally clears the memory, including the data which has been poked in.

When the first program has done its job, the next program is loaded normally and runs. Now there are two

The Case of the Electronic Home

'Holmes, I've got a letter here from a distraught magazine editor asking us to investigate the future of the electronic home, but I don't know how to go about the job.' 'Well Watson, you can always pop down to the furniture exhibition at Earl's Court and see what's new.' 'That's Ideal Holmes! (Oh God - Ed)

So it was that our two intrepid detectives went off in search of clues as to how long it would be before everything in the house could be operated at the flick of a switch.

When they arrived at the show though, they were in for something of a surprise. Whereas even a couple of years before, every machine was being heavily promoted as the all singing, all dancing model with more lights and buttons than you know what to do with, this year's displays were much more minimalist in design. It was as if it was assumed that the machine could do everything that was asked of it, and it was simply left at that.

'There are two distinct problems here' said Holmes. 'To start with, you will notice that just about every machine, from cameras to washing machines, has chips in it, which in turn gives the end user access to a plethora of functions. But although the customer may appreciate having all these extra facilities, he doesn't actually use many of them. After all, just about every single compact disc player lets you program the tracks in any order that you want. But who in their right mind is going to mess about with the order of the movements in a Mozart symphony?'

'The second problem is that nothing is compatible unless you stick to just one manufacturer. So the customer ends up with separate remote control handsets for his television, video and hi-fi. None of them will turn on his dishwasher, and he can only use them over a limited range. What is needed is some all-powerful control system.

Leaving aside his relentless search for the dreaded Moriarty, Mr Sherlock Holmes and his old friend Dr Watson visit the Ideal Homes exhibition and discover that 221B Baker Street is not all that it should be...

By Gordon Hamlett



'I notice that all the display houses have a study' commented Watson. 'The idea of people working more and more from home points, even though there is little evidence to suggest that it is actually happening. It seems that computers are still regarded as toys unless they possess those three magic letters - IBM. Moderns still have not caught on as they have done in

America. Instead, no home should be without its own personal fax machine.'

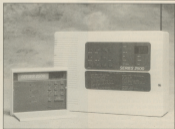
'If we are not very careful Watson, you and I could soon be out of a job. The one major area of expansion in the electronic field is in security systems. I see that Modern Alarms are offering a wireless programmable system - easy to install, no messing up the decoration, and it can be taken



with you when you move. And every type of alarm imaginable is here — magnetic contacts, heat, radio, sound, vibration, breaking glass, infra red beams and close-circuit wiring. There are even personal alarm buttons.

"How is the enterprising burglar supposed to get past that little lot, and if there are no criminals, then what are we to do? Ah, here is something that would fool even me. A machine that emits artificial barking noises. Remember that case of the dog that didn't bark in the night? This device would have totally messed up that story."

"I can find no trace of robotics here Holman. It looks as if we will have to put up with Mrs Hudson's housekeeping for at least another year. This is one field that has definitely not progressed as quickly as everybody anticipated, although I still have concerns that one of the biggest fast food chains are battling to be the first



to get their customers served by automation!

"At last, Watson, I think that we have found what we were looking for. A system that will control every domestic appliance in the house and which can be operated either from the central processing unit, from a handset or, more importantly, via a telephone link. Just think of it. Being able to phone home if we are out on a case and instruct the video to record *Criswell* and *Poker 3*!

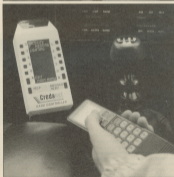
"I see that the system - *Credna* from *Creda* - works by means of a series of transmitters placed in every appliance. It then communicates with the central processing unit using existing mains-circuitry. This c.p.u. has a large display panel as well, so that you keep addresses and diary dates as well as being able to check up on share prices and train times. That would have been useful in the *Blakerville* case.

"We would be able to control the temperature in every room in the house, varying it as and when we saw fit. Lighting could be similarly controlled, ranging from individual lamps upwards. Imagine being able to turn out the kitchen light from your bed when you had forgotten to do so. All your aforementioned security devices could be checked easily, although it would lessen the impact of my detective prowess if I knew beforehand exactly who had turned up at my front door.

"Turning on the cooker, doing the washing in the middle of the night and arranging for records of my beloved *violas* music to start playing as soon as I walked through the door. And all available in the next couple of years for about a thousand pounds."

"This has got to be the way forward, eh Watson? This is where the future of the electronic house lies. It doesn't matter what make of appliances you've got - they can all be controlled from the one system. I would guess that we will be hearing of many more such control systems in the next few years, each offering more and more features and with the price coming down all the time."

"There's only one thing left for us to investigate now Watson, and that is to find out what is on the other side of that yellow door! What's so special about the yellow door *Haines*? 'It's a lesson entry my dear Watson!' (Here comes, *Mander*, no more please - a distraught Ed).



Quality. Designed from the inside out.

The Business Series



Value

24

Pin(s)



Simplex



Print



Paper Parking



Speed



Compatibility



Support

For state-of-the-art printing the multi-line LC24-80 gives you value for money as well as the top-quality of a 24-pin printer.

The LC24-80 is compatible with almost every computer's needs whatever the system. It can emulate most industry standards with ease - a standard feature that we think should always be included. For convenience over larger printers, there is even the option of a low-cost serial interface.

Printing the LC24-80 is made couldn't be simpler. At the mere touch of a button on the front control panel you can choose any one of eight resident letter quality fonts, determine your



pin(s) size or simply put the paper park into action.

For more information DMP switches to help you.

print. As with all our printers, it offers both quality and quality/price for striking leaders.

Printed 24 characters in one second and that's just the elite draft. The LC24-80 will also perform most perfectly at a rapid 51 cps producing exceptional letter quality for private correspondence and important documents. And while you print, the 18,000 buffer feeds you and your PC to get on with other things.

No other 24-pin printer in this price range can match it. Features like its generous font buffer, expandable to 38K, built-in park feature and revolutionary continuous paper parking facility are all included as standard. Another impressive feature you won't find in an optional extra is the LC24-80's remarkable range of print variations. All fonts come with the option of shadow, outline and even shadowed outline to have alternative text effects.



Behind every Star printer there's a reputation of superb reliability together with a comprehensive back-up service for peace of mind. Should you need any help at any time, you can rely on our national three-level support system which operates throughout the dealer network and is backed by the finest distributors in the country.



THE
star
COMPUTER PRINTERS

Star Micros UK Ltd.,
Crown House, 40 Colindale Avenue, London NW9 1QB. Telephone 01 848 1400.

A Division of Star Micros Co., Ltd., Japan.

Please send me all the information I'll need. Alternatively, just call Refuscon 01 540 1676.

Name _____

Company _____

Address _____

Postcode _____

Telephone _____

DMPC'S

All trade marks acknowledged.



Public sector customers may purchase at preferential terms from PWSO's. Contact 0903 1693276.

Returner

Return to the menu at any time with this ingenious program

The problem of keeping records of what's on your disk was addressed in *Commodore Gold* (See Vol 1, No 3 (Jan/Feb 88) by Menu Maker. This useful utility allowed you to load any program direct from the Menu, but to return to MENU you had to add the load statement to each program on the file, which is easy enough to do, but usually means putting check questions in and having to end the running up of the program. You could also do it by pressing Run Stop and reloading "Menu", but that's rather a waste of time.

The program Returner allows you to leave a program at any point or time, and return to the menu to select an alternative program without all the problems outlined above.

On loading and activating the 'RETURNER' program code, the first job that it does is to set up the 'RESTORE' key, and automatically load a MENU.

Obviously, for the program to work correctly, you must have a 'MENU' program on your disk. We have published quite a few such programs in the past. Alternatively, you could write one of your own. To get the best results, it is advisable to have at least two, if not more, programs on your menu.

Once you have your menu programs running, you select which program you want as normal. On running the selected program, you can press the 'RESTORE' key at any time to return to your MENU program.

If the returner won't work

There are two reasons why Returner may not work:

1. The program is necessary in using the NMI interrupt, as it is restoring the interrupt vectors to normal values. You need to allow your program (if possible) so that it's not using the NMI interrupt.

N.B. The NMI interrupt values are located at 702-793 (E118-5219).

2. The memory location where 'Returner' is located is being used by the program being run. See solution below.

Program checker

This program will provide you with alternative memory locations to hold the 'Returner' program. Load 'CHECKER'.S & run. This will give the following instructions on screen.
 File Type is N
 Then Load Menu, Load A File, & Run
 Load Menu, Load Another File, Repeat Until all Files Loaded
 Load Checker & Type Y
 Have You Wiped Before?

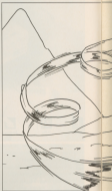
When running, 'Checker' will show a blank screen until processing is complete. This will take approximately three minutes and then show on the screen:

Ready
 Load "Menu".S

Press Return. Run the menu, then load first file, then reset computer by pressing RUN/STOP & RESTORE (or by using a Reset cartridge) Do not switch the computer off & on.

Reload the menu, run the next program, then reset as before and continue until all files have been loaded. Once all the programs on the disk have been run, reload CHECKER, run, and type in Y to the question.

Options is given to output to printer as well as screen. The result of the program is a list of free memory positions available for insertion of the program Returner. You can select the desired position to place Returner (if no space available it will tell you so, i.e. Returner cannot be accommodated on that disk.)



To change the position of returner in memory

Load "EDIT".S and run

The screen will ask for the new start address. You can enter it in Hex or Decimal numbers (the Checker program provides location values in decimal).

Once entered, press return.

The screen will then ask you to state which file you want to be automatically loaded when you press the RESTORE key. If not MENU, then also to your own requirements - press Return.

The screen will then ask what name you wish to give the Returner program. Put your own name in. The screen will then ask "Are You Sane?"

Returner



- If not, type N, if you are, type Y. N returns you to the beginning of the screen, and you will have to re-input memory location etc. When you type Y, it will save the newly-named program to disk.

Starting up your computer

When you start up your computer, you initially load and run the program you named and saved above. It will automatically install the RESTORE key function and run the program you asked to be first loaded (see above).

I recommend that it is the menu as pressing the RESTORE key will automatically return you to this program until the computer is switched off.

Note: A good place to put the 'Returner' program in memory is \$20 (\$304), as it is not normally used when using disk. It relates to the tape drive too, so you won't be able to use the tape drive.

If you have a multipart game, Returner can be used to restart the game by asking it to automatically load in the first part of the game as

part of its program (see above). The start program must, however, have a Basic line number at the beginning e.g. 10 SYS000.

Addition to Returner instructions

It is not recommended to press RESTORE during the use of the disk drive or printer, as the menu may not work properly if you do so.

If you need to press RESTORE while the printer or disk is in use and the menu doesn't work properly, then just press RESTORE again and the menu should work properly.

Technical Books

Printer Book for the 66. An original handbook which covers interfacing, setting graphics, secondary addresses, graphics dumping, keyboard settings and more. Includes software disk. 260 pages. Only £11.95

Compiler Design & Implementation 64 & 800. Learn how to design a compiler & compiler you will then learn to design a language suited to your specific problem & write a corresponding compiler. Assembly & disassembler included. Book & software disk. 890-0424. Only £14.95

Science & Engineering for the 66. An introduction to using the 66 in scientific applications. Describes variable types, computational accuracy, computers in science, linear & nonlinear regression, 128 square distribution, Fourier analysis, matrix operations. Book & disk. 240 pages. Only £14.95

Selected 128 Books

128 Quick Reference Guide. Conveniently lists all of the 128 commands, functions, etc. For applications-level usage. Includes detailed commands, program addresses, graphics commands, & input/output. Also gives a manual for all 128 users. Only £4.95

Tricks & Tricks. A collection of useful tricks and games for the C128 and then see Book & software disk. 260 pages. £12.95

Tricks & Tips. A treasure trove of programming tricks including file protection, graphics, sound, and much more. Book & software disk. 320 pages £12.95

BASE P4 Internals. A very comprehensive guide to the 128. P4s use 65 pages in the largest 128 book. £18.95 (software disk £3.95)

C128 BASIC Training Guide. A self-taught guide to programming in BASIC V1 & V2. Full examples and demonstrations. Book & software disk. 260 pages £12.95

GEOS Books

GEOS Inside & Out

The most complete guide to effectively using GEOS. GEOS Inside & Out gives the beginner a quick introduction to operating GEOS. Later chapters explain the user with GEOPaint & GEOWrite. One chapter is dedicated to practical uses - other topics included are titles & tips, creating custom windows for your programs, GEOS file format details. Plus more! A utility for converting your programs to the GEOS format. Includes optional disk with the book listings. comes with 1.4. 260 pages. Only £4.95

GEOS TRICKS & TIPS

A guide to using GEOS, use the knowledge of an experienced user. Over 60 square inches are included. From setting up documents in GEOWrite to turbo editors, printing tricks, programming your own shell messages and much more. Programs included & supported on disk are: Converter, transfer documents into GEOWrite & Forward a text editor & commands. Book & software disk. over 200 pages. £14.95

Official 128 Guide

comprehensive official programmers guide to the C128

- Inexpensive reference guide for the C128
- Covers many topics including:
 - The new BASIC V - explains new commands
 - Graphics - setting the C128 graphics power
 - Sound & Music - getting notes out of the C128
 - Machine Language - explained in detail
 - Operating System - memory management
 - Screen Editor & memory maps
 - Input/output guide - peripheral control
 - Disks - Power for non-disks
- Over 300 pages. £14.95

128 Repair Guide

Problems with your C128, are you an experienced or inexperienced user who wants to repair your computer?

- Troubleshooting guide by Howard Peers - simple fast finding on the 128
- Subsystems described in detail
- Using test equipment for fault finding
- Component descriptions & characteristics
- Tests arranged by trouble symptoms
- Perseus maintenance advice
- Easily test Perfit & repair in yourself
- Large full size book, full of diagrams. £14.95
- New Troubleshooting Guide £4. 104.95

Anatomy of the 128

Inside your C128 in great detail

- Look deep into the heart of the C128
- Programming Music & Sound
- Programming graphics modes
- Using the 80-column chip
- DR processor and local ROM
- Assembly language programming
- Full commented ROM listing
- Programming the ports
- Optional disk - contains book programs £4.95
- Over 270 pages full of information. £9.95

Anatomy of the 1571

The essential reference guide for all 6501 1571 owners

- For the Beginner & advanced user
- Fundamentals for disk drive beginners
- Assembling disk drive commands
- Creating (relative & sequential) files
- Using your 1571 under IBM SAA/SDS
- Visual access commands
- Working with Drive disk formats
- The IBM 34 system
- Reading programs in the DOS buffer
- The IBM 1571 and CRM software
- Remote disk drive functions
- Many utilities including disk & file copy
- Fully commented DOS listing
- PC/XT Display layout
- IBM & 1571 disk drive modes
- Optional disk - contains book programs £4.95
- Over 270 pages. £9.95

1541 Repair Handbook

The book 1541 users have been waiting for

- The 1541 repair & maintenance handbook
- A complete guide to setting for your Disk Drive
- Topics covered include:
 - Maintenance techniques & intervals
 - Drive motor speed adjustment
 - Testing for correct operation
 - Adjusting the read/write head
 - Instructions on digital electronics
 - Complete description of the electronics
 - Test/diagrams on most parts
 - Optional disk - contains book programs £4.95
- Large full format book with 100 pages £11.95

Hackers Book Pack 64

- Tricks & Tricks 64
- Full of quick tricks & pointers for programmers
- Program pages, sound, use (system & more)
- Control memory, string devices, bar charts
- High resolution graphics, keyboard, use port
- BASIC, extensions and machine language
- Over 200 pages packed with information
- Tricks & Tips 64
- A collection of routines & information
- Includes information covering:
 - Graphics, colour, copying, 3D lines
 - Advanced BASIC, Math, 128 & 1541
 - Interfacing & expansion options
 - Disk management & working plus much more
- Over 270 pages

• Hackers pack for including book disks £14.95

Selected 64 Books

Anatomy of the 604. Packed with information on the 64 & its peripherals. Chapters include graphics, sound, interfacing, music, BASIC, assembly language and ROM listings. 260 pages. £9.95

Anatomy of the 1541. Remove the mysteries out of your drive, learn with the set of diagrams a program how to write files, use the 1541 & ROM listing, includes disk & file control, directory reader. The protocol disk monitor, keyboard change & file transfer Complete commented ROM listing. Over 320 pages. £4.95

Meet the year 64. Make your 64 work, many practical ideas are given with the software to help you use year 64. Written for the novice, some of the projects covered include drive control, auto-loading, separate monitor & message card. Over 200 pages. £4.95

Graphics Book 64. A straight forward book that teaches you how to program, use and design graphics. Create CAD pictures, new character sets, files & multi colour pictures, sprite design & illustrate 3D graphics, animation, 2D, control and screen memory management & more. Over 200 pages. £4.95

The Official 64 Programming Guide. A complete book for the 64, published by David Books. Topics covered include sound, music, graphics, word arts, all ports & interfaces, electronic files, Commands, ASCII codes, traps & traps locations, memory maps. £4.95

Optional disk available for selected books only £4.95 per disk

Clip Art Disks

1000 Clip Art pictures

- 100 Pictures per disk, 1000 in total
- Pictures are included with every disk
- Disk 1 Variety
- Disk 2 Christmas goodies
- Disk 3 Nature life & scenes
- Disk 4 Animals
- Disk 5 American pictures
- Disk 6 More general pictures
- Disk 7 Sport
- Disk 8 Holidays
- Disk 9 People & faces
- Disk 10 More animals
- Disk 11 Vegetables & Plants, green thumb
- Disk 12 Holidays, 20th. more people
- Easily transferred to other disk drives
- Diskettes/ disks pack: disk 1 to 10 £24.95

Ideal for Label Wizard,
PrintShop, PaperClip
Publisher & Award Maker
£3.95 each

Disk Art

Ready-to-use pictures for GEM

These are not Clipart but large pictures,
ready to use

- 44-1 - Holiday, parties, animals, general
- 44-2 - Cities, life style, holidays, sea scenes
- 44-3 - Holidays, activities, 75, February, 20th
- 44-4 - Tools, garden, shapes, boats
- 44-5 - Make a face, flowers, balloons, houses
- 44-6 - Animals, stage, cars, space creatures
- 44-7 - Sports, baby stuff, animals, boats
- 44-8 - Cities, holidays, computer human
- 44-9 - US, life, sports, wilderness, real activities
- 44-10 - Churches, old & old land & houses

- Fonts - Pre-designed, ready to use fonts
- Musical - Create music sheets

- Created by professional graphic artists
- Designed for GEMPaint & Write, 64 or 64
- Work on 64 & 80 columns, mono or colour
- Disk Art disks Pack: disk 1 to 10 £24.95

High Quality picture disks
£7.49 each

64 Graphic Software

Newcomer Clip Art disks 1 to 3. High quality pictures ready to use with Renamem, 640 pictures per pack. Each Pack £4.45

Confidant Maker. Create confidants for the Award Maker and produce your Commodore picture. £14.95

Confidant Maker Library (Disk 1). Add new features & pictures to Confidant Maker. £19.95

Button & Badge Maker. A badge making factory package. More designs included £14.95

Slide-show Designer. Super! Designed to slide show creation, editor & mode show for your catalogue. Used for Action Play, Super, Power from. Only £9.95

Graphic Label Wizard

Great new graphics label printing utility for the C64

- Print labels with both text & graphics
- Easy to use menus with pop up windows
- Use PrintShop/Powermax Clipart pictures
- Off-line graphics display - see what it pictures
- Quickly change layout & print your label
- One graphics disk & lots of text per label
- Save labels to disk for later editing or posting
- Print multiple copies of your favourite label
- Print 1, 2 or 4 labels across
- 800x160 - 16 color pictures

A flexible printing tool
£19.95

Newsroom 64

A Disk Top Publisher for the 64

- Design, produce & print newsletters, posters
- Select from 640 clipart pictures
- Word processor with 4 fonts in 6 sizes
- Photo Lab, selection and edit clipart pictures
- Balloons, greetings, message features
- Copy Book, word processor with 6 fonts
- Layout Editor, quickly design your pages
- 4 disk package with an 80 page manual

A dynamic program for
journalists of all ages
£24.95

Award Maker Plus 64

It's worth mentioning, then it's worth remembering with an award they'll keep

- Create beautiful awards with your 64 & printer
- Ideal for use - four steps to create an award
- Select size from many features
- Select colour or mono-printing
- Choose a graphic from the large picture library
- Type in a message, date & add your signature
- Printouts over documents printer
- Save letters & gold awards for whole classes
- Use attractive CGA64 pictures for borders
- FREE 5000 embossed press on seals

Create a keepsake they'll be
proud to display
£24.95

Video Title Shop 64

Bring in your own Movie Director

- Use you 64 & video to film your home videos
- Create an on-line array of title-effects
- Speed feedback, down to action
- Picture only screens into captions, music a screen
- Control roll & graphics, effect & more

Polish your home movies
£19.95

Paper Clip Publisher

The most powerful Desktop Publisher for the Commodore 64

- Produce anything which uses words & pictures
- Newsletters, reports, presentations, flyers

- It's easy to use -
- What you see is what you get display
- Put down messages to get you results fast
- Icons for commonly used tools
- Built-in text editor to prepare your final text
- Built-in graphics editor to touch up artwork
- Mouse or joystick control

- It's very powerful -
- Produce multi-page documents
- Design boxes for text or graphics
- Move boxes anywhere on the page
- Rotate boxes & automatically format text
- Flow text around artwork
- Import graphics from PrintShop, Newsroom
- Use Clipart pictures
- Import text from most word processors
- Set columns, character spacing & margins
- Create custom-type machines & fill pictures

- Create Graphics modes -
- Fontwork, pencil, colour
- Zoom, pan, editor, pencil, eraser
- Graphics importer
- Textile features & fonts -
- Bold, light, superscript, left, mirrored
- Shadowed, text area, underlined, outlined
- Fonts - Helvetica, Times, Courier, Symbol

An impressive desktop
publisher for home and
professional use
£34.95

Solutions Unlimited

Photo Finish

- Near Laser Quality - available on your printer
- Optimize your titles graphics producing a technology with four times the print resolution
- Compatible with most drawing packages
- Print 50000 pictures of 160
- Photo Finish for the C64 £24.95

Icon Factory

- Convert graphics from one format to another
- Images, Fontwork, Pictures, Double
- BitMap, Pictures, Pictures etc.
- Image editor - reshape, smooth, invert, flip
- Copy and out pictures
- Convert files pictures to multi-line & text
- Icon Factory for the C64 £24.95

To Order

All products available by mail order, call now with credit card details or mail a cheque or Postal Order to the address below.

Prices include Postage & VAT.

Allow 14 days for delivery

Oxford Basic

*Despite its name,
this is a compiler with
several very interesting
extras*

By Norman Doyle

The Oxford Basic package is a suite of four programs designed to get the best from Basic and beyond. The main part of the program is a standard compiler but, before using this section, the program has to be optimized.

At the beginning of the whole process, a good set of utilities can speed things up and keep it all neat. Oxford Basic's *Yesshit* fits into the high memory slot at 48152, and offers ten new commands which are outlined elsewhere.

One glaring omission is an automatic line-numbering command, and another is a command to undo the execution of a *NEW* or to recover a program after a *reset*. It's only my opinion of course, but to me these missing commands would have enhanced an otherwise comprehensive package.

Vital Statistics

The Analyzer really gets inside a program and squares out every possible value and measurement that the programmers could think about. It takes a little time for the program to be read in, but the wait is worthwhile, because there are seven options available for analysis afterwards.

A Flow Cross-Reference revealed the following facts about the program:

```

TOTAL SOURCE LINES ----- 102,000
PROGRAM LOCATIONS ----- 102,000

LINE NO.   START   END   TYPE
-----
TOTAL UNPAID SPACE ----- 374
TOTAL PAID SPACE ----- 742
TOTAL FREE SPACE ----- 368
  
```

The first number in the GOSUB section is the line that is accessed and the line numbers that follow are the lines which call up the routine.

When a Data Cross-Reference is called, the variables are listed, followed by the line numbers where they appear:

```

DATA CROSS-REF. INDEX FOR DATA
-----
ADDRESS -----
DATA -----
LINE NO.   START   END   TYPE
-----
TOTAL UNPAID SPACE ----- 374
TOTAL PAID SPACE ----- 742
TOTAL FREE SPACE ----- 368
  
```

As a program is developed, it's easy to forget to delete lines which are no longer needed. The Deadwood Analyzer cleverly scans through the program to find lines that will never be accessed. The offending lines are then listed, and a deadwood-line listing generated if required.

Another improvement can be made through the Create Header File option, which generates a short program to initialize all of the variables used in the program. The file is then saved and can be MERGED with the main body of the program later using the *Toolkit*. Initialization may seem redundant as the *LIT* command, but it can actually improve the speed at which the program operates.

When a variable is encountered in a program, the computer checks through the list of existing variables to see if it's a new entry, or a modification of an existing one. If the label proves to be a new one, it is tagged onto the end of the current variable list. This involves a substantial degree of necessary movement and consequently wastes time.

Added to this, a variable which is commonly used may be at the end of the list, which means that the routine has to check through the full list before the current value of the desired variable is discovered. The generated header program avoids these time wastes by inserting space for the variables in descending order of frequency right at the beginning of the program.

A list of program statistics can be displayed so that memory and variable usage can be recorded. The *Concrete Stats* option also summarizes other facts such as the number of lines used, GOTOs and GOSUBs as well as the total number of variables and arrays.

Line	Original	Compressed
100	PRINT "HELLO WORLD"	PRINT "HELLO WORLD"
110	FOR I=1 TO 10	FOR I=1 TO 10
120	PRINT I	PRINT I
130	FOR J=1 TO 10	FOR J=1 TO 10
140	PRINT I*J	PRINT I*J
150	NEXT J	NEXT J
160	NEXT I	NEXT I
170	END	END

There is also a menu option for checking the disk directory, and to make sure that any files have been successfully saved.

The reports given by the state generator and the flow and data cross-reference options can be saved to disk alongside the finished program. There is no direct hard copy option, but a safety file is included which will read a stored report to the printer for future reference.

Arranged Crush

The final pre-compilation routine is the Compressor, which reduces the program to its absolute minimum size. As each line is scanned, the routine removes all REMs and spaces but the compression goes beyond this. Lines are combined in squares in as many commands onto a line as possible.

After compression, a printout of the program may contain lines which physically span three or more screen lines. This is possible because the Compressor works with raw program code so that a command such as PRINT only occupies one byte rather than five. A line could therefore contain as many as 58 PRINT commands, which would expand to 228 characters and colons or five screen lines! This would make editing (impossibly difficult, as the program must be thoroughly debugged before this operation is performed).

Although this optimises the program's memory usage, difficulties were experienced when using the compiler so I eventually decided that compression was probably better used on programs which wouldn't be improved by compilation.

Full Speed

OX-Comp is one of the best compilers that I've ever seen. It is a full-facility, four pass compiler which can cope with all of the commands in the C64's Basic library. The program also

accepts all types of variables and arrays up to two dimensions. Three-dimensional arrays can be compiled, but only after the program has been pre-processed by a special utility which is included on the disk.

A compiler converts Basic code into a special form of machine code which runs up to 40 times faster than the original program. When a Basic program runs, the operating system has to scan each line in turn, turn each command into an action, and create and modify all of the variables it encounters.

A compiler scans through the Basic program and converts all of the commands and their parameters before the program is stored on disk. The conversion also means that sufficient space can be made for all of the variables which are to be used. The finished program is therefore midway between Basic and pure code.

The initial compilation can take a long time to produce, because the full program is scanned and re-scanned four times before the final program is produced. During these passes, the compiler produces and creates temporary work files which add disk access time onto the process. A long program can take up to half an hour to prepare, but the results are worth waiting for.

Wisely, the producers insist that the master disk should be used as little as possible, and a special back-up routine is included which transfers the compiler and its associated utilities onto separate disks. Unfortunately, the back-up is not the most efficient of utilities, and each sub-program is transferred individually, regardless of whether it occupies one block or 20. The result is a constant swapping of disks, which makes the whole process very laborious and unnecessarily long.

Once the copies have been made, the compilation process is automatic, and the program to be processed is placed on the same disk as the copied compiler. This can cause problems with disk space, but this problem has been anticipated by offering suitable error message generation and options to erase programs which will no longer be required for the final compilation - even the original program can be erased once compilation has progressed beyond the first pass.

The limitations of the compiler are surprisingly few and petty. Arrays must be finitely dimensioned, which means that DIM A(2) is a legal statement but DIM A(X) would be

quoted. If an array slips through to the compiler, which lacks an explicit dimension, the program interrupts compilation to ask for a suitable array size before continuing through the rest of the program. This may limit the efficiency of memory usage, but few programs would suffer appreciably by this insistence on explicit dimensioning.

The only commands that the compiler rejects are LIST, CONT, SAVE and RUN when it has a line number parameter. LIST is excluded because there is nothing to list in a compiled program, CONT is an acceptable omission and most people could live without commands such as RUN40 when a CLR: RESTORE: GOTO40 could be substituted instead. SAVE is harder to live without but this could be remedied by including an STS in the original program to a suitable machine code patch.

One worrying problem which revealed itself under testing was that the maximum string length is 254 characters, instead of Basic's 255. This must easily be an oversight on the part of the programmers because it is not documented in the manual. Such errors are slightly disconcerting because they only show up at runtime, and I'd rather not sit with my fingers crossed while a program compiles hoping that another unforeseen error occurs.

All compiled programs are executed with the RUN/STOP key disabled. In most cases this is unnecessary but it may be necessary to allow such an interruption as a special form of REM statement has been devised to allow the key to be enabled and disabled.

Error Tracing

If errors occur when the compiled program is being tested, it can be difficult to relate the problem back to the original program. An error which stops the execution of the program will generate a message which is similar to Basic's equivalent message, but instead of a line number a memory location is displayed.

A utility on the disk can be given the value, and it will then automatically cross-reference this number in the original program to assist debugging. Of course, such an error should rarely occur because any sensible programmer would have tested and

debugged the original program before compilation, wouldn't they?

Another utility which can aid debugging was actually derived for hybrid programs which require parameters to be accessed by a machine code patch. Variables and arrays are stored differently in the compiled program so any associated code routines have to be modified. This works in favour of routines which access a current variable value, because the memory location of variable data is fixed in a compiled program. REPORT is a utility which scans the original program and prints out a table of variable locations to ease the problem of conversion.

Doty Protection

Why, in this day and age of colour photocopiers and cameras, do software houses insist on these ridiculous colour charts? I don't mind protection systems, but I object in this form because I am red/green colour blind and the companies always seem to insist on using red, green and two other

colours on these charts. Why black, white, red and yellow can't be chosen defies me.

In this case the problem is increased because each of the four programs is individually protected. In the end, I got so annoyed that I resorted to backing up the programs using a cartridge copier - so much for protection!

Despite this, the package maintains a very high standard throughout all four sections. Maybe if the space under the ROMs had been used, the toolkit could have been more comprehensive in the manual, but no computer is perfect and this compiler is comparable, though slightly inferior, to Supersoft's Basic. What this has to offer which Basic doesn't is the other three packages which make this package stand out against the opposition.

Toolkit Commands

CHANGE
DELETE
DUMP
FIND

INFO
MERGE
QUIT
RENUMBER
SIZE
TYPE

Analyzer Options

File cross-reference
Data cross-reference
Execute disk directory
Deadwood analysis
Create header file
Generate stats
Return to Basic

Compressor Functions

Removes spaces and REMS
Compress program lines

Ox-Comp Facilities

Majority of Basic 2.0 commands
Four pass operation
Error checking
Location of variables printed

AT LAST A POOLS PROGRAM THAT DELIVERS THE GOODS!!

POOLSUSTER64 IS HERE!

THE PROGRAM THAT GAVE HUNDREDS OF DIVIDENDS TO ATARI ST USERS, IS NOW AVAILABLE FOR THE COMMODORE 64. POOLSUSTER64 IS QUITE SIMPLY THE MOST ADVANCED POOLS PROGRAM AVAILABLE TODAY. LOOK AT THESE HIGH-SCORING FEATURES: 1) POOLSUSTER IS GUARANTEED THAT'S RIGHT-WERE SO CONFIDENT THAT YOU'LL WIN WITH POOLSUSTER THAT WE PROMISE TO REFUND THE PURCHASE PRICE IF YOU HAVEN'T WON SOMETHING WITHIN ONE YEAR OF THE DATE OF PURCHASE! 2) IT'S THE STATE-OF-THE-ART POOLS PROGRAM. IT USES AN ARTIFICIAL INTELLIGENCE (A.I.) SYSTEM TO FINE TUNE ITS PREDICTIONS EACH TIME YOU ENTER A SET OF SOCCER RESULTS. IT ACTUALLY LEARNS FROM THE RESULTS IT GETS WRONG. 3) IT KNOWS THE SCORE! POOLSUSTER CONTAINS A MASSIVE DATABASE OF SOCCER STATISTICS WITH DETAILS OF OVER 1000 PAST MATCHES. 4) IT'S EASY TO USE. POOLSUSTER IS FULLY JOYSTICK/MOUSE DRIVEN - THERE'S NO NEED TO USE THE KEYBOARD AT ALL. 5) IT'S VERSATILE. POOLSUSTER COMES WITH ALL THE U.K. SOCCER LEAGUES YOU'RE LIKELY TO NEED, INCLUDING GM VAUXHALL, NORTHERN PREMIER, BEAZER & HPS LOANS, AND YOU CAN ADD ANY OVERSEAS LEAGUES AS YOU WISH. 6) YOUR FORTUNE IN THE STARS! POOLSUSTER64 ALSO INCLUDES THE UNIQUE MAGIC PREDICTION PROGRAM. THIS FORECASTS LINES OF 6 POSSIBLE DRAWS ACCORDING TO THE ASTROLOGICAL POWER NUMBERS FOR YOUR NAME, DATE OF BIRTH & POOLS DATE. POOLSUSTER64 IS AVAILABLE ON 5.25" DISK OR CASSETTE.

**POOLSUSTER64 COSTS ONLY £40.1 SO HURRY
ORDER ONE TODAY & WIN THE POOLS TOMORROW!**

VERBODEN ALEN VERKOOP EN VERLEN. NIET OP TE WERKEN. COMMODORE 64, COMMODORE 128, COMMODORE 386, 486, 586, 686, 786, 886, 986, 1086, 1186, 1286, 1386, 1486, 1586, 1686, 1786, 1886, 1986, 2086, 2186, 2286, 2386, 2486, 2586, 2686, 2786, 2886, 2986, 3086, 3186, 3286, 3386, 3486, 3586, 3686, 3786, 3886, 3986, 4086, 4186, 4286, 4386, 4486, 4586, 4686, 4786, 4886, 4986, 5086, 5186, 5286, 5386, 5486, 5586, 5686, 5786, 5886, 5986, 6086, 6186, 6286, 6386, 6486, 6586, 6686, 6786, 6886, 6986, 7086, 7186, 7286, 7386, 7486, 7586, 7686, 7786, 7886, 7986, 8086, 8186, 8286, 8386, 8486, 8586, 8686, 8786, 8886, 8986, 9086, 9186, 9286, 9386, 9486, 9586, 9686, 9786, 9886, 9986, 10086, 10186, 10286, 10386, 10486, 10586, 10686, 10786, 10886, 10986, 11086, 11186, 11286, 11386, 11486, 11586, 11686, 11786, 11886, 11986, 12086, 12186, 12286, 12386, 12486, 12586, 12686, 12786, 12886, 12986, 13086, 13186, 13286, 13386, 13486, 13586, 13686, 13786, 13886, 13986, 14086, 14186, 14286, 14386, 14486, 14586, 14686, 14786, 14886, 14986, 15086, 15186, 15286, 15386, 15486, 15586, 15686, 15786, 15886, 15986, 16086, 16186, 16286, 16386, 16486, 16586, 16686, 16786, 16886, 16986, 17086, 17186, 17286, 17386, 17486, 17586, 17686, 17786, 17886, 17986, 18086, 18186, 18286, 18386, 18486, 18586, 18686, 18786, 18886, 18986, 19086, 19186, 19286, 19386, 19486, 19586, 19686, 19786, 19886, 19986, 20086, 20186, 20286, 20386, 20486, 20586, 20686, 20786, 20886, 20986, 21086, 21186, 21286, 21386, 21486, 21586, 21686, 21786, 21886, 21986, 22086, 22186, 22286, 22386, 22486, 22586, 22686, 22786, 22886, 22986, 23086, 23186, 23286, 23386, 23486, 23586, 23686, 23786, 23886, 23986, 24086, 24186, 24286, 24386, 24486, 24586, 24686, 24786, 24886, 24986, 25086, 25186, 25286, 25386, 25486, 25586, 25686, 25786, 25886, 25986, 26086, 26186, 26286, 26386, 26486, 26586, 26686, 26786, 26886, 26986, 27086, 27186, 27286, 27386, 27486, 27586, 27686, 27786, 27886, 27986, 28086, 28186, 28286, 28386, 28486, 28586, 28686, 28786, 28886, 28986, 29086, 29186, 29286, 29386, 29486, 29586, 29686, 29786, 29886, 29986, 30086, 30186, 30286, 30386, 30486, 30586, 30686, 30786, 30886, 30986, 31086, 31186, 31286, 31386, 31486, 31586, 31686, 31786, 31886, 31986, 32086, 32186, 32286, 32386, 32486, 32586, 32686, 32786, 32886, 32986, 33086, 33186, 33286, 33386, 33486, 33586, 33686, 33786, 33886, 33986, 34086, 34186, 34286, 34386, 34486, 34586, 34686, 34786, 34886, 34986, 35086, 35186, 35286, 35386, 35486, 35586, 35686, 35786, 35886, 35986, 36086, 36186, 36286, 36386, 36486, 36586, 36686, 36786, 36886, 36986, 37086, 37186, 37286, 37386, 37486, 37586, 37686, 37786, 37886, 37986, 38086, 38186, 38286, 38386, 38486, 38586, 38686, 38786, 38886, 38986, 39086, 39186, 39286, 39386, 39486, 39586, 39686, 39786, 39886, 39986, 40086, 40186, 40286, 40386, 40486, 40586, 40686, 40786, 40886, 40986, 41086, 41186, 41286, 41386, 41486, 41586, 41686, 41786, 41886, 41986, 42086, 42186, 42286, 42386, 42486, 42586, 42686, 42786, 42886, 42986, 43086, 43186, 43286, 43386, 43486, 43586, 43686, 43786, 43886, 43986, 44086, 44186, 44286, 44386, 44486, 44586, 44686, 44786, 44886, 44986, 45086, 45186, 45286, 45386, 45486, 45586, 45686, 45786, 45886, 45986, 46086, 46186, 46286, 46386, 46486, 46586, 46686, 46786, 46886, 46986, 47086, 47186, 47286, 47386, 47486, 47586, 47686, 47786, 47886, 47986, 48086, 48186, 48286, 48386, 48486, 48586, 48686, 48786, 48886, 48986, 49086, 49186, 49286, 49386, 49486, 49586, 49686, 49786, 49886, 49986, 50086, 50186, 50286, 50386, 50486, 50586, 50686, 50786, 50886, 50986, 51086, 51186, 51286, 51386, 51486, 51586, 51686, 51786, 51886, 51986, 52086, 52186, 52286, 52386, 52486, 52586, 52686, 52786, 52886, 52986, 53086, 53186, 53286, 53386, 53486, 53586, 53686, 53786, 53886, 53986, 54086, 54186, 54286, 54386, 54486, 54586, 54686, 54786, 54886, 54986, 55086, 55186, 55286, 55386, 55486, 55586, 55686, 55786, 55886, 55986, 56086, 56186, 56286, 56386, 56486, 56586, 56686, 56786, 56886, 56986, 57086, 57186, 57286, 57386, 57486, 57586, 57686, 57786, 57886, 57986, 58086, 58186, 58286, 58386, 58486, 58586, 58686, 58786, 58886, 58986, 59086, 59186, 59286, 59386, 59486, 59586, 59686, 59786, 59886, 59986, 60086, 60186, 60286, 60386, 60486, 60586, 60686, 60786, 60886, 60986, 61086, 61186, 61286, 61386, 61486, 61586, 61686, 61786, 61886, 61986, 62086, 62186, 62286, 62386, 62486, 62586, 62686, 62786, 62886, 62986, 63086, 63186, 63286, 63386, 63486, 63586, 63686, 63786, 63886, 63986, 64086, 64186, 64286, 64386, 64486, 64586, 64686, 64786, 64886, 64986, 65086, 65186, 65286, 65386, 65486, 65586, 65686, 65786, 65886, 65986, 66086, 66186, 66286, 66386, 66486, 66586, 66686, 66786, 66886, 66986, 67086, 67186, 67286, 67386, 67486, 67586, 67686, 67786, 67886, 67986, 68086, 68186, 68286, 68386, 68486, 68586, 68686, 68786, 68886, 68986, 69086, 69186, 69286, 69386, 69486, 69586, 69686, 69786, 69886, 69986, 70086, 70186, 70286, 70386, 70486, 70586, 70686, 70786, 70886, 70986, 71086, 71186, 71286, 71386, 71486, 71586, 71686, 71786, 71886, 71986, 72086, 72186, 72286, 72386, 72486, 72586, 72686, 72786, 72886, 72986, 73086, 73186, 73286, 73386, 73486, 73586, 73686, 73786, 73886, 73986, 74086, 74186, 74286, 74386, 74486, 74586, 74686, 74786, 74886, 74986, 75086, 75186, 75286, 75386, 75486, 75586, 75686, 75786, 75886, 75986, 76086, 76186, 76286, 76386, 76486, 76586, 76686, 76786, 76886, 76986, 77086, 77186, 77286, 77386, 77486, 77586, 77686, 77786, 77886, 77986, 78086, 78186, 78286, 78386, 78486, 78586, 78686, 78786, 78886, 78986, 79086, 79186, 79286, 79386, 79486, 79586, 79686, 79786, 79886, 79986, 80086, 80186, 80286, 80386, 80486, 80586, 80686, 80786, 80886, 80986, 81086, 81186, 81286, 81386, 81486, 81586, 81686, 81786, 81886, 81986, 82086, 82186, 82286, 82386, 82486, 82586, 82686, 82786, 82886, 82986, 83086, 83186, 83286, 83386, 83486, 83586, 83686, 83786, 83886, 83986, 84086, 84186, 84286, 84386, 84486, 84586, 84686, 84786, 84886, 84986, 85086, 85186, 85286, 85386, 85486, 85586, 85686, 85786, 85886, 85986, 86086, 86186, 86286, 86386, 86486, 86586, 86686, 86786, 86886, 86986, 87086, 87186, 87286, 87386, 87486, 87586, 87686, 87786, 87886, 87986, 88086, 88186, 88286, 88386, 88486, 88586, 88686, 88786, 88886, 88986, 89086, 89186, 89286, 89386, 89486, 89586, 89686, 89786, 89886, 89986, 90086, 90186, 90286, 90386, 90486, 90586, 90686, 90786, 90886, 90986, 91086, 91186, 91286, 91386, 91486, 91586, 91686, 91786, 91886, 91986, 92086, 92186, 92286, 92386, 92486, 92586, 92686, 92786, 92886, 92986, 93086, 93186, 93286, 93386, 93486, 93586, 93686, 93786, 93886, 93986, 94086, 94186, 94286, 94386, 94486, 94586, 94686, 94786, 94886, 94986, 95086, 95186, 95286, 95386, 95486, 95586, 95686, 95786, 95886, 95986, 96086, 96186, 96286, 96386, 96486, 96586, 96686, 96786, 96886, 96986, 97086, 97186, 97286, 97386, 97486, 97586, 97686, 97786, 97886, 97986, 98086, 98186, 98286, 98386, 98486, 98586, 98686, 98786, 98886, 98986, 99086, 99186, 99286, 99386, 99486, 99586, 99686, 99786, 99886, 99986, 100086.

TEL: 01/978-2280. 24HR: 01/738-8400. FAX: 01/822-1063


```

10 PRINT "PROG"
20 PRINT "RUN IT"

40 GOTO 80

50 RUN
60 LINE MISSING! HELP!!

```



Line Input

*Improve your 64's input
with this handy utility*

There are times when the standard Basic keyword "INPUT" on the C64 is a bit of a pain. In the event that a comma (",") needs to be input as part of a string "INPUT"

cannot do it, because the comma is interpreted as being a delimiter between different bits of data which are supposed to be assigned to separate variables.

If you wanted to enter such a string then it has to be enclosed by quotes so that "INPUT" will read in every character. This looks silly on the screen, and is very confusing for users

of the program who are not familiar with the technique - they can't understand why some text has to be enclosed in quotes in order to enter strings including commas, whereas other forms of input such as numeric data do not. It's even more confusing when the program later requires that the user use the comma to separate input into different fields!

The other slightly aggravating thing about "INPUT" is the fact that the "?" prompt is always printed on the screen prior to "INPUT", waiting for keyboard entry. This is sometimes a nuisance, since the message which asks for the input on the part of the user is not always of a kind where a question mark at the end is relevant - that is, it's not a question. Add to this the need to follow it with quotes - and then rub this out and type quotes again so you can use the cursor move keys to edit the line - just so a comma can be included where required in the input string, and straight away we already have two irrelevant characters on the screen before anything is actually typed in.

What's more, you always forget to add the opening quotes at the start, which itself suggests that it doesn't feel natural to have to do this. The situation is worse when inputting from an external channel - here not only the comma but also the colon and semi-colon are interpreted as carriage returns, or the end of the string, whereas in actual fact it may be nothing of the sort. Again, entering the string in quotes before it is saved to the external device prevents this, but on inputting it back the quotes will have disappeared. In either case, whether inputting from the keyboard or an external device, quotes cannot be included as part of the text. But why not, you may ask?

"GET" will input anything from the keyboard, and a string comprising all printable characters can be built up using "GET". But the Basic coding required to interactively edit the string at the same time rapidly becomes unwieldy and slow. Also, the input is not automatically echoed to the screen, so this must be done in Basic. There's no flashing cursor either.

A cursor can be forced, but it has the peculiar behaviour of leaving some characters in inverse video as soon as the cursor move keys are used. "GET" can be used to input text from an external device, but using Basic commands that retrieve one character at a time and build it into a string can be time-consuming to say the least. It's frustrating when all you want is input some text of an 'unusual' format.

If you're a machine code programmer of any skill, then this can be got around, except that you may find yourself having to re-develop the coding to suit the requirements of different programs. But if you want

to stay in the Basic environment, then while "GET" and "Input" have their place (and are very good at what they do), what is really needed is a new Basic keyword, one that will input strings of all printable characters.

Well here it is. Inspired, as it happens, by the IBM PC BASIC A's action keyword of the same name the ability is called "LINE INPUT" and is in the form of a machine code routine stored at SCACH to SCBBD high up in the free RAM area above the 64K BASIC interpreter. In actual fact, it is not a real keyword at all, but a "SYS" call which cheats by pretending to be an interpreter routine.

This is necessary because of the sort of information that the routine needs in order to carry out its task. The "USR" function only allows one numeric value to be passed on to a machine code routine, which then can only return one other value to a variable. The temporary register storage for "SYS" calls are rather long-winded to use, and in any case none of these are any use because "LINE INPUT" must return a string in a string variable.

The simple answer is for the routine, "LINE", to get these parameters direct from the current Basic line being processed using the subroutine "CHRGET" - which retrieves the next BASIC byte into the accumulator, and also incidentally is the thing that skips spaces not in quotes - in exactly the same way as all the interpreter routines get their information. This makes "LINE INPUT" very easy to program in Basic, because if you extend the presence of the word "SYS" at the front it will appear in the listing, and in operation, like a valid Basic keyword. Furthermore, if you make a numeric variable equal to the "SYS" call address and name this variable "LINE", the subroutines will be complete. The system then is as follows:

```
SYS < address > INPUT
IF < channel No.> < string variable >
```

Note the full stop between < address > and INPUT (if < channel No.> < string variable >). This is very important as it ensures that the following valid BASIC keyword, "INPUT" or "INPUT #", is crunched down to its proper BASIC token when the BASIC line which contains it is entered. Neither of these will be executed by the interpreter in the normal way. However, they exist here exclusively

for use by "LINE". In fact, the processing activity of "LINE" will make whatever of these two tokens inacceptable to the interpreter.

The keyword for "INPUT" or "INPUT #" is included because "LINE" has to write within itself to the appropriate processing for either inputting from the keyboard and screen, or from an external device. The keyword is tokenized for three reasons: 1. to save memory space, although this will be negligible; 2. because it's a professional way of doing it; 3. since "LINE" uses it to find out which form of input is required, it simplifies things greatly if only one byte can be examined instead of a string of characters. This token, and the remaining parameters, are read by "LINE" direct from the Basic text using "CHRGET".

A number of subroutines are used by "LINE" to read the parameters. They themselves get the Basic bytes using "CHRGET". One of these is a customized sub-routine called "GETVAL" located at DCB47 (\$2115 decimal), and looks like this:

```
READ EQU $ADDA
FIX EQU $B7F7
ORG $CBA7
GETVAL JSR READ
JMP FIX
```

"READ" is part of a Basic interpreter routine which picks up a numeric value from the Basic line currently being processed. The value may be written as a string of decimal numbers, or be in the form of a numerical variable. Whichever, "READ" converts it into floating point format and stores in Floating Point Accumulator #1.

"FIX" is the familiar 'fix to fix' interpreter routine. The contents of FAC #1 are converted into a 16-bit integer and stored in zero page locations \$14-\$15 hex.

Therefore "GETVAL" performs the action of reading a numeric value as a variable's contents from a Basic line and making same available in handy double-byte integer format in \$14-\$15 for a machine code to use. If "GETVAL" is handled properly, it helps in understanding how "LINE" operates, that the C64's Basic interpreter follows a special convention that every interpreter routine, including those used by "GETVAL", expects the first Basic byte that it has to deal with to already be in the CPU's accumulator. Or, to put it another



was, every Basic interpreter routine calls "CHRGET" to get the next byte of Basic text into the accumulator before it exists. Also "GETVAL" is a little bit limited, only positive numbers in the range 0-65535 are allowable otherwise an "ILLEGAL QUANTITY" error is generated. Invalidly entered numbers produce "SYNTAX ERROR".

On calling "LINE" with the "SYS" command, true to the convention, the interpreter runs "CHRGET" to load the accumulator with the next byte of Basic text. Unfortunately this doesn't arrive intact when "LINE" takes control, because the action of successive "SYS" loads the accumulator with the contents of the temporary storage for A reg. located at \$060C (260 decimal). So the accumulator has to be refreshed with a called to "CHRGET", a letter part of "CHRGET" which gets the same BASIC byte again. This byte must be a full stop (".") or EOL, if it is not a jump is made to \$A008 which prints "SYNTAX ERROR IN LINE <n>.", and terminates the Basic program. The byte following this, retrieved by "CHRGET", must either be a token for "INPUT" or "INPUT #". If it is neither of these again a jump is made to "SYNTAX ERROR".

Otherwise at this point the routine "LINE" deviates. The bulk of the routine comprises two separate processes, the one chosen depends on whether the input is from the keyboard and screen, or whether it is from an external device.

LINE INPUT from keyboard and screen -

Syntax:

SYS < address > INPUT < string variable >

Example:

```
100 LINE = $060C
110 SYS LINE INPUT AS
```

Input from keyboard and screen is carried out in three stages. Firstly the screen RAM address for the start of

the input is found by locating the cursor position with the kernel routine "PLDT", and calculating the screen address by the rather crude method of adding 48 to the screen base of 1024 "y" (row count) number of lines. "x" (column count) is added to the result. I did originally write a proper multiplication routine, but this required so much code that it wasn't really worth it. The calculated address is stored in free zero page locations as a pointer.

All you see on the screen is a flashing cursor with no "" prompt. Now a kernel routine called "CHRFN" handles all the character getting and printing with full editing support and proper flashing cursor. In use it is identical to using the Basic editor in direct mode. It has the same limitations - if quotes are keyed then cursor move keys come onto the screen as characters, but editing out quotes and re-keying stops this. "LINE" discovers the length of the line. This is done by searching backwards from the maximum string length position, the default of which is 80, looking for a non-space character. On finding a non-space character, its position ahead of the start of input becomes the new string length.

The screen line, from the start address up to the string length, is copied to a buffer. This is our old friend the cassette header buffer at \$030C. However \$030C is reserved to hold the string length count, so the buffer actually starts at \$030E. A conversion routine is used to translate screen coded characters into IBM ASCII, and the most significant bit is stripped off in case the screen characters were inverse video ("LINE INPUT" needs input a line which has just been typed, it could just as easily have been PRINTed instead and the keyboard buffer FORKed with a carriage return).

At the end of this the string is stored in the buffer and its length is \$030C. At the final stage "LINE" calls interpreter routine "LOCATE", which goes in search of the string variable through the variable storage

area - "LOCATE" gets the variable's name from the Basic line. If it does not exist then it is created. After this the string storage for the variable, if already in existence, is freed by "FREESTR" and then a fresh storage area is defined using "STRLEN" which does so according to the new string length passed to it. We now have a place in the string storage area to which the contents of the buffer can be copied.

It is impossible for characters to be put in this area directly (without being stored in a buffer first) because "LINE INPUT" does not yet know how long the string will be until this stage.

In this manner any string variable can be created or updated using "LINE INPUT" in just the same way as it can by any other INPUT function. However some care is required where the variable is one of a dimensional array. Small arrays up to a "half" line of say 10 or 11 elements would be okay, in other words:

```
110 SYS LINE INPUT T(10)
```

Where 'n' is an index from 0 to 10 in the default provision for arrays up to 11 elements, but otherwise a "working variable" such as IN1 for example would have to be used to transfer the input to the element where this is one of a large, dimensional array, e.g.:

```
110 SYS LINE INPUT IN1
120 A$(n) = IN1
```

If this precaution is not observed the process becomes too complex for the simpler than normal string building techniques used and usually crashes the 64.

Whatever the string is stored in the proper place, and the extra does not add perceptively to the processing time, even if a large string array is being filled with text.

Because "LINE" is located in RAM it can be modified. I mentioned that for input from keyboard and screen, it has a default maximum string length of 80 screen characters (two screen lines). It doesn't matter in which column across the screen the start of input occurs, the maximum remains 80 characters. You can change the maximum length by a POKE to suit your own requirements. For example, a disk file name cannot exceed 16 characters in length. If "LINE INPUT" is used to get the file name it can first be modified only to accept

up to a maximum of 16 characters. In the source listing the maximum string length is labeled "MAXLEN" and is at \$C84D (\$D045). Use the following:

```
000 LINE = 51004
```

```
0000 MAXLEN = 16001
0010 POKE MAXLEN, 16
0020 PRINT "ENTER A NAME
FOR THIS FILE"
0030 SYS LINE INPUT IN#
0040 NMB = INT(.968)
0050 OPEN 2,1, NMB
0060 etc.
```

You can type as many characters as you like for the file name, only the first 16 will be taken any notice of after you press [RETURN].

Be aware though of a couple of limitations of "LINE INPUT". One is the same as that of the normal "INPUT", namely if other characters present on the screen come within the range of the maximum string length, it will be assumed that these constitute the rest of the string, even if you don't want them. Enough blank space must exist beyond the position where input commences to prevent this.

Also, "LINE INPUT" doesn't know if the start of input occurs on the bottom line of the screen. If the screen scrolls up where text exceeds one screen line, the start address is not adjusted up with it. Also "LINE INPUT" will assume that the "rubikish" beyond the top of the screen RAM area are characters to be input! Consequently "LINE INPUT" should not be used any lower than the 24th row down.

LINE INPUT # from an external device -

Syntax:

```
SYS < address >, INPUT < chan-
nel No. >, < string variable
```

Example:

```
100 LINE = 51004
```

```
1000 OPEN 2, 1, "TEXT"
1010 SYS LINE INPUT 2,AS
1020 etc.
```

This is the only time that "GETVAL" is used, and even then only the least significant byte of the resultant integer is needed!

"GETVAL" retrieves the channel number following "INPUT # " so that

normal routines can route input from the appropriate channel. Inputted characters are copied to the buffer, a process that continues in a loop until one of three conditions are true:

1. The ST variable indicates an EOF signal from a disk drive. On detecting "EOF" inputting is terminated.
2. The character read in is found to be the termination character. The default termination character is a carriage return (13), on encountering the termination character inputting is terminated. The termination character is NOT copied to the buffer.
3. The maximum number of bytes have been copied to the buffer. The default maximum number of bytes is 128. On the 128th character being copied to the buffer, inputting is terminated.

On input being terminated, "LINE INPUT # " goes to the final stage of building the string to memory exactly as before. As far as "EOF" is concerned, it will be required that the BASIC program using "LINE INPUT # " should also monitor the "ST" variable. Re-using "LINE INPUT # " and forcing input past EOF causes a rubbishy character to be copied to the buffer, overwriting the previous string. This is not a problem if other variables have been equal to this string.

As before, because the routine exists in RAM, it can be customized to behave in a special way. "CHECKR" is a label in the source listing which represents the point where the character read is tested to see whether it's the termination character or not. It looks like:

```
CHECKR CMP # 00D
```

Where the operand is the value, in this case 13. This can be changed as in the following example:

```
CHECKR = 32065
POKE CHECKR + 1, 0
```

Now the routine will stop reading when it encounters a zero byte. Similarly the maximum number of bytes copied to the buffer can be changed:

```
CHEMAX = 52079
POKE CHEMAX + 1, < n >
```

Where < n > is any number up to a limit of 191, because this is the maximum amount of space available in the cassette buffer area.

Even better, "LINE INPUT # "

can be made to ignore any termination character. Following the "CHECKR" location is this:

ISTERM REQ PUTLEN

which is where a branch is made to the final stage upon encountering the termination character. By using

```
ISTERM = 32067
POKE ISTERM, 234 : POKE
ISTERM + 1, 234
```

This has the effect of overwriting the branch instruction with "NOP's", so the routine cannot exit if a termination character is found. This allows maximum flexibility for inputting data of a "strange" nature - "LINE INPUT # " can read in all byte values 0-255. It is possible for example to read machine code into string - supposing you wanted to do such a thing!

Because "LINE INPUT # " is completely self contained, i.e. no part of the conventional interpreter (INPUT #) routines are employed, we are allowed to do something normally quite illegal. The following is possible in direct mode:

```
LINE = 51004
```

```
READY.
OPEN 2, 1, "TEXT"
```

```
READY
SYS LINE INPUT 2,AS
```

```
READY.
1AS
```

This is incredibly useful for verifying that a Basic routine that you are trying to de-bug has saved data properly or not. It's normally impossible to get at this data in direct mode, since using "INPUT # " results in an illegal direct command error. You can get on moving the cursor up and displaying and displaying successive fields of data. This even works with numeric data saved using "PRINT # ", since those are written as decimal strings.

If you have an assembler you can enter the source listing, and if needs must the ORG statement can be changed to relocate the routine anywhere you like, but make a note of the new label locations if you then want to customize it with POKEs. It's unlikely that you will need to use "LINE INPUT # " very often, but when you do, you'll be glad it's there.

A Flow of Ideas

It's often necessary to view a directory from within a program, but from Basic this can be very difficult. There is a way, and this method can reveal more information than may at first be obvious.

Before a disk directory program can be written, it is essential to know what data the drive makes available. Using the following short program on the directory in Table 1 gives all of the information which is displayed in Table 2.

```

10 OPEN "A:1" AS #1
20 GET #1, "DIR:"; DIR$
30 PRINT DIR$, "END DIR LISTING"
40 PRINT
50 END

```

The rendered data includes the Block Allocation Map (BAM) which lies from BC000-BCXXX, as well as the directory content itself (BCXXX-BCXXX). Using GET #, each byte can be read and used to form quite a powerful source of disk information.

From this content, the aims of the program can be derived. What we will create is a flow diagram for a program which lists each directory entry, the file type, blocks used and the track and sector (the specific block location) relating to each file. As the program runs, it will keep track of the total number of programs on the disk for a general run-up screen, which is to be displayed after the individual program detail screens.

The program can also keep a tally of the total number of blocks used, and this value can then be compared with the number derived from the BAM to ensure that the disk has been validated correctly. Finally, a detailed map of the number of free sectors (blocks) on each track can be displayed using colours to differentiate between tracks which have not been used at all, and those which have lost a few sectors to file storage.

The first duty of the program is to set the screen colours and then initialise the disk under examination. This means opening command channel 15, and keeping it open for checks on disk errors throughout the program.

Next, the directory file is opened for a sequential read operation. After the drive loads the first sector into its

How easy is it to incorporate a disk directory reader in a program?
By Norman Doyle



internal buffer, variables are initialised and dimensioned ready for the main program to begin. The reading of the BAM takes a little time, so a comforting message is displayed to assure the user that this is a normal occurrence.

Exploded BAM

The layout of the BAM can be seen from Table 3. There is no single byte pair responsible for recording the number of free bytes on the disk, so this value has to be derived from the data at the beginning of the BAM. The first two bytes (the track and sector links) are not loaded when the directory is read as a sequential file, and the next two bytes are irrelevant, but some of the next group of bytes are essential to calculating the number of free memory blocks on the disk.

The information is stored in a specific map, with the bytes grouped into clusters of four. The first byte is the total number of free blocks on a particular track, and the three bytes that follow can be used to calculate which particular sectors these are. In this case, the only byte of interest is the total number of free blocks, so that is read and added to a running total while the others are discarded.

Again from keeping a record of the total number of free blocks, each

track's details must be stored in an array if the aim of producing a tracks map is to be fulfilled. There are 35 tracks on a normal Commodore disk, and it can be seen from Table 2 that there are, in fact, 35 groups of four bytes in the BAM. A loop must read each of the groups, storing the first byte in an array and adding its value to a grand total of free blocks. After this is done, the next three bytes can be ignored.

At the end of the BAM, the disk name and ID numbers can be found. This must be read and stored in a suitable form for display at the head of each screen page. First of all, the string is initialised to produce a label indicating that what follows is the disk's title and, to comply with Commodore's convention, the RVS ON character is added to display the disk header in reversed characters. A second convention is to place the disk name in quotes. This is not catered for from reading the directory file, so the opening quote is added at this point.

The disk name is allocated a space of 16 characters in the BAM sector of the directory. This means that by simply reading the 16 character group and adding it to the title string, any disk name can be catered for without any complicated checks. The resultant string is then completed with a closing quote mark.

To complete the screen page header, the next 15 bytes are added to the string. This actually reads more than enough characters to cater for the ID and the disk type descriptor (always 36 on a normally formatted disk, but is extended to cater for non-standard formatting using four character IDs and other such tricks. The difficulty here is that some of the bytes will be stored as 'null' values, which is the reason why string variables are being used in the program instead of numerical ones to avoid generating errors.

Because the bytes are stored as ASCII values, the conversion routine must look for those nulls and convert them to CHR\$(0) to avoid similar errors when the bytes are converted for use. This happens quite a lot, and a substitution is the ideal answer. Once the disk name has been included as

a string, it is printed and, because this string contains the clear screen symbol, the name acts as a title for the screen page.

Before reading in the directory entries, the column heads must be displayed. The convention that has been devised for this routine is that the program name will be followed by the file type, block count and the track and sector values for the first program block.

This information is not stored in the correct order as it must be read, stored and sorted out in a print statement. The layout of a single directory entry can be seen in Table 4. A subroutine deals with the actual reading of the individual entries, and this will be examined later.

As each directory entry is dealt with, a counter is incremented so that a check can be made for a full screen page (20 entries). When this number is reached, a 'Press Any Key' message is printed and a keypress detection loop initiated.

Directories come in all shapes and sizes, so the program cannot work on a simple loop to read in the entries. A way is needed to indicate when the directory is complete. This is done by reading the system variable, \$T. When this has a value of zero, work is still in progress, but if it has a real value one of two things has occurred.

It may just be that the directory has been read in completely, so it could be that an error has occurred. Before closing the file, the routine checks the error channel. If an error is detected, the program will halt and the message is printed; if there is no error the program continues to the next stage.

Under a screen page title of 'Vital Statistics', the details of the number of programs, and blocks used are printed, suitably labelled. Both are derived from the individual directory entries and supplied via the 'read entry' subroutine.

The free blocks' value has been calculated from the BAM record, but before this is printed we can use this value and the 'blocks used' value to check for a correctly validated disk. If the number of blocks used is subtracted from the total number of blocks free on a newly formatted disk (664), the result should equal the BAM-derived free block count.

Any inequality means that the BAM is faulty, and this is flagged by a suitable message. A word of warning - inequalities could be the result of

RLL, or VSR files being used on the disk, so ensure that this is not the case before validating, or you could lose valuable data!

After the free blocks have been displayed, a table is drawn up with the tracks listed above their unused sector values. At each stage, these values are checked against the sector capacities for each track. If none of the sectors have been used, the value is printed in brown, but any tracks which contain file data are highlighted by using light blue characters.

This program can easily be extended and improved to give a full BAM map, or reduced to produce a disk-to-screen directory printer.

The subroutines in each such directory entry is an integral to the running of the main program and, therefore, worth a closer look.

Directory Delivery

The directory read routine takes over immediately after the disk name has been read from the disk. The directory details are preceded by a series of null bytes which must be discarded until a 'real' byte value is read in. This method saves time but has one serious drawback when the first file has been deleted and not replaced.

File types are denoted by byte values as follows:

```
129 = SEQ
130 = PRG
131 = USR
132 = RLL
```

If the file is 'locked', or protected, these values are increased by 64 to give a range of 193-198. Deleting a file simply results in these values being replaced by a null byte. Using the method of discarding null bytes would mean that the first significant value would be the old track number when a deleted file was encountered.

Consequently, the program includes a check to see if the value read is in excess of 192. If this is not true then it must be a deleted file, so 'DEL' is assigned to the program type and the routine jumps to calculating the read in value as a track number. If the value is valid for a file type, a check is made to test if it is locked or not. Provision is made to denote a locked file by using a reversed 'less than' symbol, which is served as a string. If the file is unprotected, this string remains as a space.

A second string is derived from the file descriptive value. This is the file type which is calculated directly from the file type value in conjunction with a MATH statement. After the next byte has been read in, the pathway which the DEL file option took converges with the main program so that the track value can be assigned to a string before the sector value is read and similarly stored.

The disk name had 16 bytes assigned to it, and the same is true for the program name which follows the sector byte. A loop reads this in and concatenates a string. If a program name has less than 16 characters, the entry is padded out with shifted space values (160), and this is useful for formatting the screen printout later, so they are not discarded here.

After the name there are nine zero bytes which are read in and discarded, so the next significant bytes denote the number of blocks that the program occupies. This value is stored in low byte/high byte format and a suitable routine is included to reveal the true decimal value. All that remains is to print the information out in the correct screen columns. Deleted file names are shown in orange and existing files in white.

One problem with filenames is that they sometimes contain control characters such as screen clear, case switch or even colours. To avoid these messing up the beautifully formatted display, location 212 is pointed with a value to fool the computer into thinking that it is in quote mode, so that the reversed character is printed instead of the action which it represents being executed. Once the rest of the information is printed, control is then returned to the main program.

Directory Flow

The flowcharts show the logic of the program contained in the Listings pages. This is rendered as a Basic routine, but the beauty of flowcharting is that the same logic pathways can be applied to machine code or just about any other language that has been invented.

If you decide to investigate directory reading further, you'll also soon realize how useful the charts are for modifying the program. It's far better than muddling through names of listings, noting variables and then trying to make sense of it all.

THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...
NOW ACTION REPLAY Mk V PROFESSIONAL
 FOR CBM64/128 HAS ARRIVED

ONLY
£34.99
 POST FREE

A R V
 Includes an original
 manual & a 30-day
 8 Guarantee

EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

**ALL FEATURES AVAILABLE TO TAPE OR DISK
 USERS.**

ACTION REPLAY Mk V offers more advantages because it combines an 800K DISK with a 128K RAM operating system ROM. This means that ALL features are available to users of ALL FORMS.

REPROGRAMMABLE means you can install technology which exactly fits your requirements. Action Replay's mode of the user hardware gives you direct access to ROMS, RAMS, and I/O DEVICES from any other cartridge. There really is no comparison, here are just some of the features:

- ✓ **TAPE TO TAPE** Action Replay Mk V has a unique Turbo Loading - "Randomizer" & "Write Buff" - Replays data on to all drives directly & can be programmed to transfer any space & print out the contents. ✓ **TAPE TO TAPE** saves your files until called upon to disk or to the tape of a backup. ✓ **SPRITS CONTROL** full range of controls with full operations. ✓ **PROGRAMMING** operates an automatic infinite loop operation.
- ✓ **TAPE CONTACTS** allow you to program compression techniques. ✓ **TEXT EDITOR** can be used to modify the text on disk or a backup program. ✓ **REPROGRAMMABLE** (user program) allows you to modify the ROMS, RAMS, and I/O DEVICES from any other cartridge. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup.
- ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup.
- ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup. ✓ **TAPE TO TAPE** COMMANDS full range of all the power of a backup.

PERFORMANCE PROMISE

Action Replay has an unmatched range of optional features. When you buy Action Replay, if you don't find one feature to be true, then return it within 30 days for a full refund.

GRAPHICS SUPPORT UTILITIES DISK

800K DISK - View your favourite screens in a slide show type display.

800K DISK - A unique utility to allow you to make any part of all your pictures & files to up for full screen size.

SPRITS CONTROL - A complete sprits editor before you to create or edit sprits.

MESSAGE MARKER - Takes your favourite screen content with a graphics package or equipment and to Action Replay & saves it into a scrolling screen message complete with audio.

ONLY £12.99

NOTICE 1984 COPYRIGHT ACT

DATTEL ELECTRONICS has granted authorisation to reproduce the use of its products for non-commercial purposes. It is illegal to make copies of such material without the expressed consent of the copyright owner or their Successors.

TOTAL BACKUP POWER

The most powerful backup system ever devised. Outrageous backup system. Press at any point. Just press the single button to follow your program.

- ✓ TAPE TO TAPE ✓ TAPE TO DISK
- ✓ DISK TO TAPE ✓ DISK TO DISK

ONLY but a 40000 program comparison. Single push save or restore space to disk or tape. Backup software automatically & all cartridges for "program" any or more cartridge requests. THE PROCESS IS AUTOMATIC - JUST COPY THE BACKUP & SAVE.

WHAT THE REVIEWERS SAY

"The easiest, fastest and locally improved." This is many the best backup kit money ever buys. The Coverage Edge!"

Computer Week Now

UPGRADE INFORMATION

MS V2 (Enhanced) is MS V1 (Enhanced) just add 20.00 to an add some you a new MS V1 Operating System. Only for you to just plug in MS V2 (Enhanced) to MS V1 (Enhanced) and your old cartridge plus which is to use and upgrade it follow 14 items.

ALL ORDERS NORMALLY DISPATCHED WITHIN 48 HOURS

HOW TO ORDER ...

BY PHONE

☎ ☎ ☎
0292 744707
 Visa Credit
 Card Cash

BY POST

✉
 Send envelope this order payable to "Datatel Electronics"

FAAX

0292 744299

PRICES AND SPECIFICATIONS SUBJECT TO TIME OF PRINTING AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLED 0292744299 - Please receive goods by telephone order to visit.

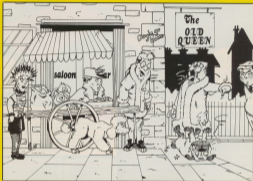


DATTEL ELECTRONICS LTD., PENTON INDUSTRIAL ESTATE
 COOKS ROAD, PENTON, BOCKING-ON-THE-HEAVES, ENGLAND.

SALES ONLY
0292 744707

TECHNICAL ONLY
0292 744299

Deadenders



War in Middle Earth

The world of Tolkien's Middle Earth once again comes to the computer screen only this time as a wargame. In it, you control the forces of the Fellowship of the Ring in a desperate fight against the evil forces of Sauron. Your objective isn't military victory, it's far more important than that, as you must clear a path through the evil forces for a hobbit to take a Ring to the cracks of Doom.

At your disposal you have the dwarves of the Iron Hills, princely elves and men of good heart as well as the Fellowship party of Frodo, Gandalf and Aragorn. Frodo is, of course, the ringbearer and can wear it at will to escape the battlefield and the area. However, it has its dangers as wearing it corrupts the soul and takes you into the realm of the Nazgûls who will attack you if they are near.

The main display shows the map of Middle Earth and a large pointer with which you can zoom in on any area and view the forces in it and issue orders for them to move into battle. Eventually, forces will clash with the undisplayed

Dandelion is an adventure based on everyone's favourite soap opera - *Neighbours*. Nah, seriously gay, it all takes place in Harbert Square, home to the Squibs, Frealigs and other unsavoury characters.

The story starts after Bill has lost his Willie. Yes, in case you haven't guessed, it's a special full of caricatures and one or two slightly naughty jokes.

You play the part of PC Donald Dance and your mission should you choose to accept it, is to find out who did the dirty on Willie. The lady tells you that poor old Willie turns up squashed, and with two holes in his neck. Leave it up to me, mate, the mission never believe this/er. It also goes on to say that you can talk to the character that you meet. "Ask Brady about Cohen" is cited as an example. I tried it, and was met with a "Yes speak, but it doesn't further your enquiries." None of the Deadenders wanted to talk to me. Was it something I said?

The jokes are obviously geared towards Easterners fans, but I can quite believe that non-followers of the series would get at least a smile from the game. Most of the humour, for me at least, came whenever I examined a character. When I examined Sissy, I was told "Purple eyes and bright green hair, she resembles a technological strand." Great stuff. All the descriptions are along these lines, and most are less than charitable.

When it comes to gameplay, I have a few reservations. The game doesn't accept some of the normal abbreviations. I had to type in "inventory" as opposed to just "I". Not all the cuts were listed, even the ones that should have been obvious, and mapping left something to be desired. When standing on the landing of the Old Queen, I was faced with the kitchen, bedroom and living room, yet no directions were given. It was a case of "some kitchen" etc, and no matter which of them I went into, the cut was always in the south. This system wasn't used consistently, though. From Harbert Square I could see the pub, the cafe and the corner shop. This time I had to input the usual

N, S, E and W, and hope I went to the building I wanted. Some things should be taken for granted in an adventure and sticking to one way of inputting directions is on this agenda.

During the course of the game, I came across two bags, one fancy and one annoying. I'd got a bag from Aster and, whenever I tried to examine it, the game crashed. However, I was allowed to "search bag" with no ill effects. Later, I discovered that I could "fixate bag" but only if I wasn't carrying it. I can just imagine how frustrating it would be for someone to find that out after going for a long time with no saved. The other bag was a silly one. I'd been searching through a dustbin, as one does, and my investigations took me to the laundrette where I was reminded how disgusting my uniform was. No problem. I took it off, washed it in a machine, got it out again, then found out I had two uniforms, one wet and one dirty.

In keeping with the spirit of the game, there's no ordinary reply when you tell the computer to get something. Instead you get a chirpy "Right, me old china." Should my copies of this game reach America, it will only reinforce their belief that we all talk like Dick Van Dyke.

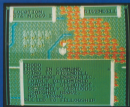
The authors have got a nice line in schoolboy humour, which I greatly appreciate. What a shame about the programming.

If you can ignore the giggles that I picked up, get down the bag an' read an' give your dach to the gear in the shop. But if you're the sort who likes to be able to use single key inputs, logical geography and no bags, knock it on the head.

One last question. Where was Kyle?

Touchline:

Title: Dandelion. Supplier: Top Ten Wms, 301, 12 Chichester Drive/Cross, Junction Road, Fife, Berks RG7 4AA. Tel: 0734 267 080. Machine: C64.



memory and the screen will snap to the battle screen showing all the combatants, who will start the fight without your help.

You can affect the battle by using a cursor to move men into the fray or you can control one character directly. This may sound like a good idea but it doesn't quite work, as to select a character you have to position the cursor over his foot and the border just takes too long. This can be annoying, particularly if the battle was only a dozen actions while the party move through the area.

Unfortunately, this spoils an otherwise interesting game which could have been so good.

Touchline:

Title: War in Middle Earth. Supplier: Melbourne House, 2-4 Grosvenor Road, Portobello Rd, London W11 3JN. Tel: 01-727 0578. Machine: C64/128 Price: £14.99 (Disk) £9.99 (Cart).

Program Analysis

*Programming can be made
simple with these three C64
performance analysers*



COMMODORE 64 PERFORMANCE ANALYSER

Basic is a programming language which makes it very easy for programmers to create complex programs with a minimum of effort. We pay a price for this programming ease, and that price is often poor performance, that is, our Basic program runs slowly. Another problem which confronts the Basic programmer is what to do when a program runs without failing, but doesn't give you the results you expect. How do you find out what your program is doing without adding PRINT statements to your program to trace execution at important locations or strategic points?

The Performance Analyser helps to overcome both problems. Not only does it trace the logic flow in a Basic program, it also determines how long each Basic line took to execute. Thus the Performance Analyser is a generalised performance analysis tool for the Commodore 64.

Performance Analyser TRACE Facility

Most commercial tracers usually amount to a window displaying five or six line numbers on the screen as your Basic program runs. The line numbers scroll in the window as each line is executed, and the window may or may not interfere with your program output. You normally cannot trace a Basic program which uses line graphics, and you certainly cannot go back and check the line number sequence previously displayed. Although you can usually close the trace display down (by the space bar for example), you have very little chance of writing down the line numbers on paper for a more detailed analysis.

The Performance Analyser overcomes all of these problems. It allows you to trace any Basic program which uses normal screen graphics, line screens, sprites or sound and does not interfere with the operation of the program. The Analyser will not slow

your program down, and allows you to give the trace display at your leisure. You may scroll backwards or forwards through the line numbers for as long as you wish.

Performance Analysis

The Analyser also provides you with a tool to determine how efficient your Basic program is. When it displays the line number it also displays the time it took to execute the line. As you scroll through the line numbers you can tell at a glance which line numbers are slowing down execution and which line numbers are executed most often. Basic programs are the same as any other program - they follow the 80/20 rule. That is, 80 per cent of the work is usually done by 20 per cent of the program. The Performance Analyser is the tool you need to tell you which 20 per cent of your program is doing 80 per cent of the work, and how long it is taking to do it. You can then concentrate on making that part of your program more efficient.

Analysing a Basic program

The Analyser is written entirely in Machine Language, and is designed to cause as little interference as possible with the traced program. The Analyser is normally loaded at 38912, and all Analyser variables and constants are contained in the 2K from 38192 to 40259. Your Basic program thus has the RAM between 2048 and 38911, and low storage locations it requires and the free RAM at 49152. Should you require the RAM at 38912, then set the top of Basic pointer (35,36) to the last RAM location available to Basic, and the Analyser will use 2K of RAM before this address. For example, if the top of the basic pointer is set to 37888, then the Analyser will load itself at 40288.

Type in the Analyser loader program and save it as ANALYSER1. Make sure you verify that what you saved is correct. To use the Analyser, simply issue a load "ANALYSER1" after setting the top of BASIC pointer if necessary. ANALYSER1 will set the required Basic pointer, POKE the Analyser Machine Language logic into the lowest RAM, relocate all required ML addresses and print messages to indicate how to start and stop the Analyser and display the trace data. The following messages are displayed on the screen by ANALYSER1 during execution:

```
LOADING THE ANALYSER AT
38912
LOAD OK
RELOCATION OK
1. START ANALYSER = SYS 38912
2. STOP ANALYSER = SYS 38913
3. DISPLAY DATA = SYS 38918
```

If the load fails, or the relocation of addresses fails, a message is issued and ANALYSER1 stops.

Obviously to start the Performance Analyser you SYS to 38912 or to the address displayed by ANALYSER1. You can do this from a program or from direct mode. The message TRACE STARTED is displayed by the Analyser, unless you start it from a program. The message is not issued then to ensure that the Analyser does not interfere with program messages or displays.

After the Analyser has been loaded, you then LOAD the Basic

program or programs you wish to analyse. The Analyser monitors execution of your program(s), and saves trace data in the trace data buffer for later display. If you only want to trace part of a Basic program, you would do the following:

```
1000 REM START THE
ANALYSER
1001 SYS 38912
1020 FORTH = ANDSTEP:1
1030 S = A*COS(TH)
1040 Y =R*COS(TH)/C
1050 NEXT
1060 REM STOP THE ANALYSER
1070 SYS 38915
1080 REM DISPLAY TRACE
DATA
1090 SYS 38918
1100 END
```

After your Basic program has finished, or you stop it executing, you can stop the Analyser if you want to. However, you don't stop it to display the trace data. You may leave it active to trace another program if you want to.

Obviously, to stop the Performance Analyser you SYS to 38915 or to the address displayed by ANALYSER1. You can do this from a program or from direct mode. The message TRACE STOPPED is displayed by the Analyser, unless you stop it from a program. Again the message is not issued to ensure that the Analyser does not interfere with program messages or displays.

Finally, you may display trace data at any time by entering SYS 38918 or SYS to the address displayed by ANALYSER1, and of course you may do this in direct mode or from a program. The message NO TRACE DATA is displayed by the Analyser if there is nothing to display. Again the message is not issued if you are under program control. This is to ensure that the Analyser does not interfere with program messages or displays.

If there is data to display the Analyser presents it in full-screen mode, that is a page or full screen data consisting of line numbers and line execution times is displayed and the Analyser ML program waits for you to press one of the function keys; F1 terminates the display, F5 scrolls back to the previous page of data and F7 scrolls forward to the next page of data.

You may scroll back and forward through the trace data for as long as

you like with function keys F5 and F7. When the end of the trace data is found, the number of lines executed and the total execution time is displayed, and the Analyser ML program waits for you to press a function key. The Analyser will only recognise F1, F5 and F7 function keys. All other keys are ignored. If you scroll forward from the end of the display, you wrap around to the start of the trace data again. You can't scroll back from the top of the trace data, you may only scroll forward.

NOTE: Trace data will be displayed automatically when the trace data buffer area is full. The trace data buffer is actually the RAM under the BASIC ROM. As much trace data as possible is stored there before the execution of the Basic program is interrupted and the trace data displayed. If you want your Basic program to continue, simply press F1 and the trace display is terminated. Your program begins execution from where it was interrupted. If you want to browse the trace data, then use F5 or F7 to scroll back and forward through the data.

How the Performance Analyser Works

The Analyser works by monitoring the execution of Basic programs via the character dispatch vector in low storage. As each program byte is interpreted, the Analyser checks to determine if the current line number (57,28) has changed from the previous byte read. When the line number changes, then the Analyser stores the line number and current time in the trace data buffer under the BASIC ROM. This is done until such time as the trace data buffer is full.

When the buffer is full, the Analyser saves the first 2K of low storage (8-2047), colour RAM and various control registers in the RAM under the KERNAL ROM. The trace data is then displayed, and when the display is stopped via function key F1, the Analyser restores the first 2K of low storage, the colour RAM and the various control registers. This allows the Basic program to restart execution from the point where it was interrupted, and the program screen is restored, as well as character colours and backgrounds.

If your Basic program uses the RAM under the Basic of KERNAL ROMS, then you cannot analyse it with this utility. Note also that if your

Basic program costs the time (TB = "000000"), then the Analyzer will not fail, but the execution times displayed will be appreciable.

COMMODORE 64 PROGRAM ANALYSIS

Commodore 64 Program Analysis (CMPANAL) is a Basic program which analyzes the contents of any Basic program and displays the information on the screen or printer. CMPANAL first displays summary information which contains the program name, the size of the program in bytes, the number of lines in the program, the total number of commands (i.e. PRINT's, GOTO's, IF's etc.) and the number of variables.

Once the summary data has been viewed, a detailed list of the commands used in the Basic program and the number of times each command is used is displayed. When you have finished viewing the command data, a detailed list of the variables and the use of each variable is displayed, and when you have finished viewing the variable data you may end the display, ask for the information to be redown or send the data to your printer.

Using CMPANAL.

CMPANAL allows you to analyze your Basic programs. It does this by running in the 4K of free RAM at 49152 to 52347, and loading the Basic program it analyzes at 2049. By not using the RAM between 2048 and 49944, CMPANAL is capable of analyzing the largest Basic program. However, with only 4K of RAM to run in, CMPANAL will not slowly analyze large Basic programs because many garbage collections will be done to assure that there is sufficient space for CMPANAL to operate correctly. Also, only 50 variables can be displayed because of space constraints.

Obviously if CMPANAL is to run in the RAM at 49152 then some changes need to be made to Basic pointers in low storage. The start of Basic and end of Basic addresses need to be changed as well as the start of variables etc. These changes are handled by the CMPANAL loader program. LOADER is the Basic loader program which automatically loads

CMPANAL. It sets the low storage pointers, and then uses the dynamic key facility to automatically load CMPANAL.

You must create and save LOADER first on tape or disk. Next type in CMPANAL and save it directly after LOADER on tape or on the same disk as CMPANAL.

Note that if you are using disk you need to change line 10 in LOADER from LOAD "CMPANAL", 1,1 to LOAD "CMPANAL", 8,1 so that CMPANAL will be loaded from disk and not tape.

Once you have saved LOADER and CMPANAL to tape or disk then simply load LOADER and RUN it. LOADER will set the various low storage pointers and then set up the screen and keyboard buffer so that when it ends, CMPANAL is automatically loaded at 49152. When CMPANAL has been loaded it begins execution automatically, clears the screen and places the first message on the screen:

LOAD FROM DISK (Y/N)?

If you want CMPANAL to load the Basic program it analyzes from disk, then reply Y. Otherwise reply N and the program will be loaded from tape. Before replying to this message, you should have the tape or disk which contains the program to be analyzed in the cassette or disk drive.

The next message to be displayed is:

PROGRAM TO BE LOADED = ?
Your answer to this message tells CMPANAL the name of the program it is to load from tape or disk to analyze.

CMPANAL then uses the KERNEL load subroutine to load the Basic program at 2049 and begins to analyze it. Since it may take some time to analyze large Basic programs, CMPANAL places the line number being analyzed in the top left-hand corner of the screen while scanning the Basic program. When analyzing is finished the summary report is displayed as follows:

—PROGRAM STATISTICS—
PROGRAM NAME = CMPANAL
PROGRAM SIZE = xxxxx
NO OF LINES = xxxxx
NO OF COMMANDS = xxxxx
NO OF VARIABLES = xxx

USE ANY KEY TO CONTINUE

You may view the summary report for as long as you wish. To move to the command report, simply use any key and the following display appears on screen:

- COMMANDS -

```
END          = 1
FOR          = 3
NEXT        = 6
DATA        = 100
INPUT       = 1
READ        = 1
GOTO        = 25
IF          = 30
GOSUB       = 17
RETURN      = 17
REM         = 8
POKE        = 5
PRINT       = 38
THEN        = 25
+           = 45
-           = 15
*           = 31
/           = 19
AND         = 1
"           = 45
MID$        = 3
```

USE ANY KEY TO CONTINUE

If all commands used in the program can be displayed on one screen, then when you press any key you will move to the VARIABLE display. If more commands are used than can be displayed on one screen, then the next screen of data will contain command data. When the list of the command data has been displayed and the USE ANY KEY message is displayed, when you press any key the list of variables appears on the screen. Note that + - * / = < and > are considered comments when used in statements such as A=+B<C/D/E OF IF X GOTO 1000.

When the comments are finished, the list of variables is displayed as shown:

- VARIABLES -

```
I           = 2
X           = 4
RRR        = 3
Z           = 1
ZZZ        = 4
```

USE ANY KEY TO CONTINUE

When the list of the variables has been listed, CMPANAL displays the following message:

R = RE-DISPLAY, X = END, P =
PRINTER

If you press the R key, then all information beginning with the summary display is redisplayed. If you press the X key then program execution is terminated and the final time message is displayed:

TIME TAKEN = XXXX.XX

This is the time in seconds it has taken CBMPANAL to analyse your program. You may then use CBMPANAL to analyse another Basic program. Press P and the information is sent to the printer.

Applying CBMPANAL

CBMPANAL has many uses. You can find the size of your Basic program, the number of variables you use and the number of lines in your program. The number of lines is important, because each line in a Basic program carries an overhead of 4 bytes (2 bytes for a link address and 2 bytes for the line number). A 500 line program uses 2,000 bytes of storage for link addresses and line numbers. If you have an excessive number of lines, you can conserve space by reducing the number of lines (also known as crunched your program). You reduce the number of lines by placing multiple commands on the same line separated by colons, removing blanks and removing REM commands.

Processing new lines also carries with it a performance penalty. The more lines in a Basic program, the longer it normally takes to run. By reducing the number of lines, you normally reduce program execution time. CBMPANAL will tell you how successful you have been at reducing the number of lines in your program. It will give the size of your program and the number of lines before crunching, and then after you have made your changes you can run it again and get the new figures.

The detailed list of commands (ie PRINTs, GOTOs, IFs etc.) can also be used to reduce program size and increase performance. For instance, if you find that you have a very large number of IF commands, then you may be able to reduce them by using the ON command. For example if you have:

```
IF CC = 1 GOTO 1000
```

```
IF CC = 2 GOTO 1100
IF CC = 3 GOTO 1200
IF CC = 4 GOTO 2000
IF CC = 5 GOTO 2100
IF CC = 6 GOTO 2200
```

then you could replace the IF commands with one:

```
ON CC GOTO 1000, 1100, 1200, 2000,
2100, 2200
```

It is also interesting to see the pattern of commands in various programs and which commands are used most frequently. In string operations the LEFT\$, RIGHT\$, MID\$ etc. will figure prominently. However, the most common commands used are the IF, GOTO, FOR and NEXT and PRINT.

The list of variables is a powerful tool to help in the construction of your Basic program. Basic maintains a list of variables, and the closer a variable is to the start of that list, the less time that is needed to find the variable when it is referenced in a statement. For example, every time IF X = 2.7 TOTO 100 is executed, the X variable must be found in Basic's list of variables to check if it is 2.7 or not. Thus the closer X is to the top of the list, the faster it is found. The order of variables makes a significant difference to the execution time of your program if you have a large number of them. CBMPANAL helps by giving you a guide as to which variables ought to be defined first so that they appear at the top of Basic's list of variables. You can ensure the order of variables by defining them in the following order:

```
X = 0.5 : Y = 0.5 : Z = 0.5 : T = 0 : 0
etc.
```

X will come first, A second, Y third in the list and so on.

If you have Basic programs where execution time is crucial (for eg. games programs) then CBMPANAL will be an important tool to help you analyse those programs and make them faster.

COMMODORE 64 SWITCH

Commodore 64 SWITCH is a short Machine Language (ML) program which resides in RAM just before the BASIC ROM. It occupies storage locations 40704 to 40926. C64 SWITCH allows you to partition your C64 into two logical machines. You switch between the two partitions or regions with a single key

stroke. With this utility, you can load two Basic programs at once and compare them or work on them. However, you cannot have both programs running simultaneously.

Using C64 SWITCH

C64 SWITCH allows you to set variable region sizes. The regions are designated zero (0) and one (1) and region 0 will extend from location 2049 to the limit you set, while region 1 extends from the end of region 0 to 4070.

To use the switching function, simply load SWITCH, which is a Basic loader program. When you run it, SWITCH will load the ML routine at 40704 and display the message:

```
ENTER REGION # ENDING
ADDR = >
```

You enter the ending address for region 0 (and then region 1 starting addr) and the final message are displayed:

```
REGIONS 0 and 1 INITIALISED
REGION ACTIVATED = 0
```

To switch between the two regions use the F1/F2 keys. F1 will activate region 0 while F2 will activate region 1. The active region is displayed in the upper right-hand corner of the screen in reverse video. To deactivate the SWITCH, simply hit RUN/STOP/RESTORE or turn the C64 off and on.

Applying SWITCH

C64 SWITCH has three main uses. You can load two Basic programs at once, and work on them or compare them. You can use region 1 as a data region which is accessed by a program in region 0 (SWITCH was originally written for this purpose). Finally, you can use SWITCH as a means of merging two programs. If you want to add code to a program in region 0 from a program in region 1, simply LIST the statements in region 1 on the screen, press F1 to activate region 0 and then move the cursor over the lines you want added and press RETURN. Each line will be entered into the program in region 0.

Help!

Enhance your help function with this handy utility

By Mark Everingham

In the old Commodore advertisement (you know, the four page epic which managed to link Charles Babbage, an elephant, and a teddy-bear called BJ to buying a Commodore computer for Christmas), special emphasis was laid on the 'HELP' function of the C16 and Plus4 computers. Commodore claimed it helps you to debug your programs, yet I have owned a Plus4 for several months now, and can honestly say I have never used the HELP function except for the novelty. The BBC Micro has a command *HELP which lists the syntax of a given command on a sideways ROM, and I decided to implement such a function on the Plus4. I decided on three features it should have -

1. It should be compatible with the Commodore C16.
2. It should not take any memory from the programmer.
3. It should not interfere with the normal HELP function.

A tall order? Well, I decided that to allow a reasonable amount of help on the C16, the program must use the Disk Drive. That way, I could put it in the cassette buffer to do not to use up any memory; and a CHERGET swidge like the DOS SUPPORT program seemed appropriate to allow for the normal HELP function. The result is a 142 Byte piece of machine code using standard PRG files on a Commodore Disk Drive such as the (DS1)

The Programs

Listing 1 is a short and sweet Screen Editor I wrote to produce HELP screens and store them on Disk.

Listing 2 is the Basic Loader program for the HELP command. It Pokes all the code into memory and redefines the function keys.

Listing 1 - The Screen Editor

When you've entered and debugged the program, save it onto a disk using `DSAVE 'HELP EDITOR'` and `RUN` the program again. You should be presented with a white screen showing the usual flashing cursor in the top-left corner. At this stage, the editor acts just as if you were editing a document - type text, in normal or reverse, graphics symbols, anything you want, even use the ESC functions to format your screen.

Pressing `RETURN` from the first level puts you into the command mode. A bar will appear at the bottom of the screen with three options. Press the relevant function key to select each one.

`LOAD` prompts for a filename and attempts to load this HELP screen from the disk.

`SAVE` prompts for a filename and saves the current screen to disk under that name.

`CONT` puts you back into level - the edit mode.

When you've created a help screen, save it onto Disk under a suitable and memorable name, and `exit` the program. The HELP command now has something to work with. If you `DIRECTORY` the disk, you will see a file filename.H If the 'H' designates a HELP file.

Listing 2 - The Basic Loader

Now that you have some data on disk (type in Listing 2, the Basic Loader, and save it on disk using `DSAVE 'HELP PROGRAM'`). When you have run it, try pressing function key 1. You should see the message `'SYS 918:REM ON'`. `SYS 913` turns the HELP command on, press function key 2 or type `SYS 990` to turn it off.

With the HELP Command on, try typing in `HELP (RETURN)`. This



simply does the normal HELP function - if your program has no error in it, nothing should happen. Now, try typing HELP 'Filename', where 'Filename' is the name you stored your Help screen under. If all is well, your screen should load and the READY prompt will appear at the bottom of the screen. If this does not happen, something is wrong - if you get an error message 'STRING TOO LONG ERROR' this means that you have tried to type in a name more than 14 characters long. If you get the error message 'DIRECT MODE ONLY ERROR' then you have tried to use the new HELP command from within a program. If neither of these, see PRINT DS3 to find the error. The Syntax, and errors returned by the HELP command are shown below.

Syntax of the HELP Command

HELP (RETURN) - Normal HELP function

HELP (Filename) - Loads help file called 'Filename' from disk
 SYS 90 - Turns HELP Command On
 SYS 950 - Turns HELP Command Off

Errors returned by the HELP Command

All DOS Errors - a fault concerning the disk
 'STRING TOO LONG' - You have typed a filename longer than 14 Characters
 'DIRECT MODE ONLY' - You have tried to use HELP in a program

Information on the HELP Program

The HELP Command resides by default in the first 142 bytes of the cassette buffer (E000-E01F). Note that when it is installed in this area of memory, the program will be erased by pressing 'RESET'. I placed it here to avoid clashes between the Plus/4

and the C16. However it may be relocated by changing the address A in line 0 of the BASIC Loader Program.

Practical use of the HELP Command

When creating HELP files, try to give them memorable and unambiguous names. It is also a good idea to make them short, though the '\$' and '?' designators may be used in filenames. For instance, if you are going to replace the manual with a HELP disk, divide the files into commands, each showing the syntax and a few examples of the command's use. E.g. Typing HELP 'CIRCLE' might bring up all the different ways of using the command 'CIRCLE'. The thing of utmost importance is to 'use your common sense!' In a system which is very powerful when used properly, but could end up not being helpful at all if help files are designed without thinking. Anyway, I hope it will be very useful to you!

GAMES UPDATE

Super Cycle



in a battle against other riders and the clock, as you have to cross the line before the seconds run out to qualify for the next course.

The early tracks are easy enough, and although the other riders can get in your way and even cause a crash, you'll easily catch up the time. Later on things get tricky, with the addition of slippery road surfaces to avoid you skidding, treacherous icy conditions, obstacles such as barricades and cones to send you crashing, and a tighter time limit where every second will count.

To add variety to the races, the scrolling landscape changes from day to night, country to city, and mountains to desert, the most impressive being the race near Cape Canaveral.

Avoids action at its best.

Twentieth:

Title: Super Cycle. *Supplier:* Atari (CIS Gold's, Data 211, Wolford Way, Hasleford, Birmingham, B6 7AS, Tel: 021-338 3388). *Price:* £2.99.

The budget bonanza continues with this classic bike racing game that topped the charts in its full-priced format. Myriad Courses wait to test man and machine

The Deep

If you tried to imagine what a game called *The Deep* would be like, you'd probably think of barrel wrecks, sharks, octopuses or perhaps submarines.

You certainly wouldn't think that you'd spend most of the game on the

sea's surface! Unfortunately, this is exactly what you do in this rather disappointing game.

It reminds me of a very old arcade game which I think was called *Submarine Attack*, in which you controlled a ship armed with depth charges that had to survive an assault of submarines that released mines and fired torpedoes at you.

In *The Deep*, your task is to destroy subs, after which small flags are occasionally released and float to the surface, staying there for three seconds. If you manage to avoid the floating mines and reach them in time, a helicopter will fly overhead and drop a package that you must catch. These packages increase your chances in the game by supplying hydrofoil speed to your ship so you can skim across the surface, smart bombs to wipe out all subs, sonar modules that can be aimed and destroy everything in their path, or submersible pods that can collect the tokens that



lie on the sea bed to clear a screen.

There are three transition games in which you must destroy a giant ship, a massive submarine and protect a fleet of escaping hostages by shooting down the missiles that are being fired at them, but then it's back to the main game again.

Twentieth:

Title: The Deep. *Supplier:* US Gold (Data 211 Hasleford Way, Wolford, Birmingham, B6 7AS, Tel: 021-338 3388). *Price:* £2.99.



GRAND PRIX CIRCUIT

Despite the split between Electronic Arts and Accolade, the string of Accolade imports continues from the EA stable. This latest has attempts to recreate the world of Formula 1 racing, and offers you the chance to drive for either the Ferrari, Williams or McLaren teams in a world championship against nine other drivers and over eight Grand Prix races.

Selecting the team you will drive for also selects the type of car you will drive – for example, the Ferrari is slower than the others, but is not as likely to spin and so is a good car to start with. When you think you can control the car, you may want to swap to a Williams or the fastest, the McLaren. A more detailed appraisal of each car is presented on-screen in a display that shows the power curves or horse-power and torque, engine size, gear box, chassis and weight, so that people who think those figures are important can use them to decide which car to use in the championship.

Perhaps of more importance is the game level you choose to play at, as this can determine whether it will be a quiet, sporting race without any mechanical problems, or a bitter fight where the hot car and driver will win. The problem with the car is keeping it going at a speed fast enough to maintain your race position, but slow enough to stop it from blowing up.

Tyres are also a major headache, particularly in the longer Grand Prix, where cornering too quickly and spinning can cause serious wear. They can be changed in the Pit, but this can cost you valuable seconds if your Pit team can't quick enough.

The first Grand Prix of the season is at Rio de Janeiro in Brazil, but before you can think about the new championship points for winning it, you have to drive round

the circuit in a qualifying lap. This not only gives you a preview of the track, but the time you take decides your position on the starting grid.

The race screen display shows your view of the track and your controls, which include a tachometer to watch for engine strain, a damage indicator that plots the condition of your car, especially braking and handling, a speedometer, rear view mirror to check that no-one's in your slipstream, and a map box that includes a line drawing of the course and a flashing dot to plot your position.

Steering can be a little tricky at first, as moving the joystick turns the wheel, and you must remember to turn it back again as it doesn't auto-rotate, unlike most driving games. However, once you've mastered the basics you have to contend with the competition, particularly at the higher game levels when any collision will put you out of the race.

The corners are the best place to overtake, where the driver with the straggled nerve will take the lead – it's all down to when you apply the brakes. If you brake after your opponent, you'll go round the bend in the lead, but if you leave it too late you'll spin off.

While *Grand Prix Circuit* is a good simulation of a formula 1 championship, an eight race season will probably prove too much for all but dedicated formula 1 fans. There is an option to race in a single Grand Prix, but there are better racing games for the casual driver. This one's for those who live on carbon monoxide car fumes.

Teacher:

Title: Grand Prix Racing. *Supplier:* Accolade (Electronic Arts) Lemply Business Centre, 12-19 Station Rd, Lemply, Berks, SL1 8TN. Tel: (0753) 49442.



Infiltrator



HAVE a second chance to fill the empty boots of the one and only Johnny "Jamba Baby" McGibbons, as he files off in his Whizzbang Enterprises Gizmo chopper to save the world at least three times. The former full-priced game has been re-released via the Klaxx label, and it's excellent value for money.

His opponent in all this is the aptly described but unnamed Mad Leader, who is threatening all sorts of

displeasing things if he isn't stopped, and Jamba Baby's the man for the job.

Each of the three missions follows a similar pattern - a three-stage game in which you sneak fly through enemy territory to find the Mad Leader's camp, then (as foot) infiltrate the heavily armed camp and search the buildings for secret plans, weapons and so on.

In fact, *Infiltrator* is three games in one, starting with a combat flight simulator in which you test bombing results, to confuse the enemy, with fighting skills to shoot them down. In the camp, you are armed with laser papers and sleeping gas to get into the camp without raising the alarm. Inside the buildings, things get tougher as you must avoid the guards and search (obviously) for every object in every room. If the guards get too noisy, showing their true papers may satisfy them, if not, use some sleeping gas and then get out before the alarm sounds and raise the alarm.

Infiltrator was a smash hit in the States, but is still underrated over here. At £2.99, it's a bargain.

Footnote:

Title: *Infiltrator*. **Supplier:** Klaxx (US Gold), Data 2/E, Welford Way, Melford, Birmingham, B6 7AE. Tel: 021 358 3384. **Price:** £2.99.



This is the re-release of the conversion of the original arcade game that began the race for car shoot-'em-ups that has resulted in a string of games including *Road Blasters* and *E.T. Streets*. It all started in 1983 with *Spy Hunter*.

Naturally, it looks a little dated with its top-down view of a scrolling road network that you must patrol, but the gameplay is just as addictive, and has certainly stood the test of time.

The action begins as the weapons van pulls up at the side of the road and the spy car rolls out into your joystick control and begins with a machine gun as its only weapon. Your mission is to stay alive as long as possible (oddly enough) and clear the road of the villains - Road Lords, Barrel Dumpers, Enforcers, Switch Blades, and Mad Bombers (helicopters).

These villains attack you with appropriate weapons, which you must answer as best you can, while staying on course on a road that turns and splits to confuse you. At some stages the road disappears altogether, but luckily your *Spy car* then becomes a *Spy boat* and takes the battle on to the waters.

Destroying enemies not only ticks up the points, but also earns you tendersons with the weapons van and an upgrade for your weapons system, which includes rockets to take out bombing helicopters, and oil slicks and smoke screens to deter enemies that get too close.

Even after six years, *Spy Hunter* is still a great shoot-'em-up.

Footnote:

Supplier: Klaxx (US Gold), Data 2/E, Welford Way, Birmingham, B6 7AE. Tel: 021 358 3384. **Price:** £2.99.



Denaris is a planet with a problem. For years its scientists have been developing super-advanced machines, and so it was almost inevitable that one day the machines would get so advanced that they wouldn't need the men any more.

By the time the Denarians realised what had happened, it was too late to launch a direct assault. They tried anyway, but it just made the machines more powerful. Their only chance to break the tyranny and escape from their underground prison is you, living a small fighter.

The fighter is, of course, highly manoeuvrable, and can be impressed by collecting items and debris from enemy ships. Before you start thinking that you've heard it all before, and that this is just another *Neoswiss* clone, I'm delighted to tell you that it isn't. For one thing, two of you can play together.

The fighter is a DS-173 Eagle Fighter that begins the game unprepared for the onslaught ahead. You'll face a barrage of meteorites, swarms of mechanised flies and have to brave the firepower of land-based gun emplacements. The other player controls a satellite that's activated by player one collecting a crystal early in the game. From then on, it can either move independently or dock with the fighter (which is what happens in the one player game).

As with *Neoswiss*, you can pick up objects to improve your ship, but these are harder to come by. They have specific effects on your ship, and come in the shape of crystals, extra weapons and bombs. Crystals are split into two groups, with the Derga crystal activating and improving

the firepower of the satellite, and the Zerkla stars increasing and decreasing the speed of the alien ships.

There are five types of balls that add a cumulative effect on your ship - they include a red ball to increase shooting power, green to add to the number of missiles, blue to activate a temporary shield, grey to add 1000 bonus points and yellow, which acts as a smart bomb destroying everything currently on the screen.

Extra weapon symbols are less common, and include systems known as the Scatter Shot, Lightning Bite Shot and Power Shot. You can also collect protection satellites that fly above and below your fighter and deliver incoming missiles. You will need all this firepower and protection if you are to survive the level, as well as the inevitable monster machine that guards the way to the next screen.

Perhaps the most important thing to learn is control over the fire button, as this is not a game where an auto-firing joystick is the way to get a high score. Instead you must choose the right amount to wish the fire button for a short burst of energy, or press and hold the button to build up the power shot that can take out a medium-sized asteroid or dent a super machine.

Obviously, *Denaris* is inspired by games like *Neoswiss*, but includes enough subtle differences to make it worth a second look.

Troubleshooter:

Title: *Denaris* **Supplier:** Rainbow (no US Gold), Unit 213 Wolford Way, Maccfield, Broomsgrove, MK 7 4JE. Tel: 027 319 2392. Price: £4.95.

3D Pool



Pool and snooker games are very popular, but they've always had the problem of being played mainly from a top-down perspective, making it difficult to shoot accurately. Now, thanks to Firebird and the skills of pool champ "Malrose" Joe Barstara, you can get into the action in one of the first true 3D pool games.

It's quite a remarkable piece of programming — you can walk around the table, pull out for an overhead view and then get down close to aim your shot. There are no aiming lines or cursors, it's up to you to get down close and judge the angles.

You'll need to practise a little on the controls before you can expect to clear the table in a single break. Left and right controls move you around the table, while up and down zoom in and out. Pressing the fire button up and down increases and decreases the power of the shot, moving it left and right controls the horizontal position where the cue will strike the cue ball. This decides the "side" as the ball, and whether it will curve left or right after it's hit another ball or a cushion.

Top or spin is added by hitting the cue ball above or below its centre, and this, curiously enough, is controlled by the same controls that zoom you in and out. Therefore, to view the ball back, you have to look down on the table from overhead.

Eventually, you'll sort that one out and be ready to take part in the tournament against seven other players for the right to play Malrose Joe himself. When you realise that Joe is pool's equivalent of Steve Davis, you'll appreciate why you have to qualify first.

This is also the first C64 pool or snooker game which includes an actual tournament where each round is played over more than one frame. For example, you begin your challenge in the quarter final in a best-of-three match, then it's best-of-five in the semi, and seven in the final.

The computer opponents provide a reasonable game, based on the fact that they know what they're doing, whereas you're learning. As you get better, you'll get closer to the challenge match with Malrose Joe. If that's too much for you, or you find it too easy, then you can opt to play against another human, or try one of the 13 trick shots with which you can amuse your friends.

The game plays the popular pub version of pool, in which you must pot all seven of your coloured balls before sinking the black to win the match. However, any foul shots (and the computer opponents occasionally play what I would describe as deliberate fouls) are punished by awarding the other player an extra shot and a free ball.

This game is definitely a cut above the usual 2D style of pool game, but it does have some rough edges. Although the balls have shadows and reflections, they slide rather than roll across the table, but that's probably more to do with the technology of the hardware than the programming. The odder thing is the lack of cushions. Visually, they're there, but the balls seem to drift through them and hang off the edge of the table. However, I doubt there'll be too many complaints about these minor graphical points.

As a first 3D pool game it is exceptional and far from the notorious seven cushion show that plagues many 2D versions.

Firebird:

Table 3D Pool, Supplior, Firebird, 64-76 Ave Oxford St, London, WC1A 1PS. Tel: 01-267 7547. Price: £9.99 (inc.) (£7.99 cash).





Bargain Bucket!

Six new budget games of varying quality are given the once over by Gordon Hamlett

Four titles this month, offering six different games, so everybody should be able to find something to suit both pocket and taste.

Missing Omega from Big Byte sees you trying to explore and deactivate an alien object - Omega. The problem is, you only have one hour of real time to accomplish this. There are four different reactors that need to be shut down in this period in order to ensure the safety of life, the universe and everything.

You start off by designing your own robots, each made up of four different components - base, weapon, sensor and pointer. You must then explore Omega, controlling your robot either manually, on automatic or by programming it.

This is an intriguing game, but it's let down somewhat by an inadequate set of instructions so that even after prolonged playing, I still had little idea of what exactly I was trying to accomplish.



Three games for less. Here's a pound pack in the heart of ZZ from Silverbird. Pick of these titles is an undoubtedly *Rock 'n' Wrestle*, your chance to try out such moves as flying body press and airplane spin on the likes of Vicious Vitoan and Gorgeous Greg. Bouncing off the ropes to build up speed, you should attempt to soften your opponent up a bit before attempting to pin him to the canvas.



The other two games are somewhat less noteworthy. *BMX Rally* has you trying to qualify for the next race by finishing in the top three in your current one. It is not sufficient just to complete the course, you have to perform stunts and tricks as well. Collect cans of pop to boost your energy and wheels to repair the damage caused by collisions with the other riders.

The final game is *Ninja Warner*, which is a truly dread ful martial arts game, but one so bad that it's almost worth having for that fact alone. Dodge arrows, karate chop legs, fend off death stars with your sword and finally slant down savanties with your blow-pipe, if you can master enough katas.

P.R. Squared, also from Big Byte, is a strange game. You must chase round eggs trying to pick up the different elements of assorted mathematical formulas. Pursuing you round the perimeter of the eggs are the chasers. Contact with these decreases your IQ, although you can restore this by collecting books. Other objects may be beneficial or hazardous to your journey. I didn't particularly enjoy this game first time round, and I'm afraid that time has not mollored my opinion.

The final game this month, and by far the best of the bunch, is *Star Gate II* on Mastertronic's Riscodot label. Perhaps the most notable aspect of this sequel is that it is actually different from the original - something most unusual in this industry. The evil Melkon has developed a race of Superbeans, and it's up to Dan to penetrate the four levels of the Melkon's ship and sabotage the control beam.

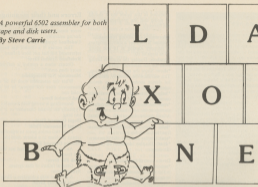
You can also choose to play the Melkon, attempting to mislead the Superbeans into outer space. In either case, you only have a limited amount of time to accomplish your task before moving onto the next level. This is a good-looking game that plays exceptionally well, and if you don't already have it in your collection, I suggest that you go out and pick up a copy straight away.



Bargain Bucket!

A powerful 6502 assembler for both tape and disk users.

By Steve Currie



The ASM Assembler

The ASM assembler is a dual mode system whereby 6502 assembly language programs may be compiled from either tape or from disk files and/or memory. It provides a set of commands to control its various functions, and also allows output to a Commodore printer.

The system has two modes of operation: disk mode and memory mode. In disk mode, the source code file is read from the source disk drive, and its compiled output sent to another file on the destination disk drive. In memory mode, the source code is read from memory (where it is edited in either mode) and output to memory. Code relocation facilities exist to allow a program to be assembled to run at one address but placed at another.

The ASM system has two error modes, fatal and nonfatal. In fatal mode, an error will cause the assembler to halt, whereas in nonfatal mode, the assembler will process the whole file, listing errors as it goes. Some facilities also exist for control of disk drives and errors reporting from those devices. A printer re-direction facility allows all output which is normally sent to the screen to be sent to the printer at device number 4 or 5. Source and destination drive numbers may be set before assembly commences. Whilst using the editor, any Basic direct mode command may be issued.

Whilst ASM was designed primarily for use with disk drives, the fact that it supports memory assembly allows tape users to make use of it. The memory mode was originally

designed to allow short routines to be tested without having to resort to disk usage. Even in memory mode, you may still make use of files from disk as the file-disk/file-include facility still works.

Getting it all in

Listed here as a Basic loader program, ASM represents a considerable typing task. You may type in the program directly as it stands, but don't run it until you have read the next bit! Before running, (assuming you have saved a copy to tape/disk first) execute these direct BASIC commands first. These will set the memory configuration correctly.

POKE 43, 0: POKE 44, 15: POKE 8960, NEW

Now reload the Basic loader and run it. ASM will be POKE'd into memory at the correct address and saved to whichever device you set the device no. to (1 or 1/2/10) (I use the latter). Reset the machine and load and run ASM. You should get a sign-on message and a flashing cursor.

When ASM is loaded and run, it installs a small module into the BASIC system. This has two important effects:

- (1) Edited program lines are no longer tokenized by Basic. This means that you cannot edit a Basic program. This is similar to the EDIT program in my "Constructing a Compiler" series in a previous issue of *Four Commodore*.
- (2) A set of additional commands are introduced via the special character *. These commands allow you to easily access the facilities provided.

The additional commands

The extra commands are as follows:
Assembly:

***asm**: Start the assembler in the mode set by the mode commands ***mem** and ***disk**. The operation is as follows. In disk mode, you are asked for a filename whose default extension is .asm. The output file will have extension .asm. In memory mode, the source code is expanded from memory and output is to memory. During assembly, output of messages, listings, etc to the list device will follow the mode set by commands ***print** and ***printoff** while the treatment of errors will be defined by ***fatal** and ***nonfatal**. Assembly may be halted at any time by pressing the RUN/STOP key.

Assembler control commands

***disk**: Sets assembler disk mode.

Source code is expanded from disk and output is to a disk file. (see ***asm**).

***mem**: Sets assembler memory mode.

Source code is expanded in memory and output is to memory.

***fatal**: If an error is encountered during assembly, the assembler will stop.

***nonfatal**: If an error is encountered during assembly, it is displayed but assembly continues with the next line of source code. Note that output is still

produced in this mode but it should be noted that the program may not run correctly.

Information and editing commands

***reset**: Resets a source program in memory starting at 10 and going up in steps of 10.

***symbolic**: Display the symbols being from the last assembly operation.

***show**: Show the mode of operation. This will show the assembly, error and list device modes as well as device selections.

***help**: Lists all the available commands.

***reset**: Set the ASM system to default startup mode. All values are set to default status: i.e. printer off, fatal error mode, memory assembly.

***info**: Display assembly information from last assembly operation.

***stat**: Display source device directory.

***dest**: Display destination device directory.

Device control commands

***source**: Set source device. Argument is the device number which must be in range 8 to 11.

***dest**: Set destination device. Argument is the device number which must be in range 8 to 11.

***send**: Send a command to source device. The command must be in quotes, e.g. ***send** "ofdisk,d" will format a disk.

***show**: Same as ***send** but for the destination device.

***err**: Display device error for source device.

***err**: Display device error for destination device.

***printer**: Enable printer output. Argument is the select code for the list device (4 or 5).

***printoff**: Disable printer output.

Note that certain functions such as listings are controlled from within a source program, e.g. sym, list (see directives).

Available Operands

The following are valid in an expression:

- B Hex value, e.g. aa&C000
- % Ascii value, must be two e.g. lda a 'a'
- < Low byte, e.g. lda a < symbol
- > High byte e.g. lda a > symbol
- + Addition e.g. symbol eqs number+2
- Subtraction e.g. symbol eqs number-6

Errors and their meanings

The following is a list of the various error messages which may be printed during assembly:

Undefined Symbol Error:

This occurs if a symbol has been referenced but has not been defined.

Redefined Symbol Error:

Occurs when a symbol is defined more than once.

Mnemonic Not Recognized:

What the assembler took to be a mnemonic does not appear to be a valid one.

Bad Symbol Error:

Something is wrong with a symbol. Typically an invalid character or it is too long.

Illegal Operand Field:

Illegal Mnemonic Field:

These two errors point to a general syntax problem in a source code line.

Missing Operand Error:

An operand was expected but was not found.

Disk File Error:

General failure of disk system.

Syntax Error:

A problem with a directive is likely.

Illegal Quantity Error:

Some overrange condition has occurred, typically a 16-bit value in a byte mode instruction.

Illegal Addressing Mode:

An instruction was used in an incorrect way.

Not X or Y Index:

Only X and Y index registers are valid.

Symbol Table Full:

Fretty fatal one this. It indicates that the space set aside for symbols has been exceeded.

Branch Range Error:

Branch instructions are relative and may only operate within a certain range.

Linkfile name length error:

The argument to an link directive is too long.

Linkfile name missing:

The argument isn't there at all!

Bad directive in memory mode:

You have used some directive not valid in memory mode.

Bad directive in disk mode:

You have used some directive not valid in disk mode.

Cannot open another linkfile:

Trapping another message when you try to link another file whilst already linked.

No such select code for this device:

You have tried to assign a device


```

10 : Example A
20 : Memory Mode
30 :
40 and #0000
50 lsr
60 :
70 .index rts #0
80 .char rts index+2
90 .screen rts #0400
100 :
110 lda #0screen
120 ldx #0screen
130 stx index
140 rts index+2
150 ldx #0
160 ldy #0
170 lda .char
180 .fill rts (index)*2
190 clr
200 brw fill
210 ldx index+1
220 dex
230 brw fill
240 rts
250 :

```

ready.

```

10 : Example B
20 : Uses lsr fill
"VAR_ASM"
30 : use in memory or
disk mode
40 :
50 and #0000
60 lsr
70 :
80 lsr "var_asm"
90 :
100 lda #15
110 ldx #0
120 stx vic+021
130 stx vic+020
140 rts
150 :

```

ready.

Type This In And Save to disk

```

10 : Module "VAR_ASM"
20 : Substitute for Example B
30 :
40 .vic rts #0000

```

ready.

code other than 4 or 5 for a printer (*prints) or a code rather than 8, 9, 10 or 11 for a disk drive (*source, *hex). Occurs in editor mode only.

Device Communication Failure:

Communication to a disk drive failed. May indicate wrong device number. Equivalent to Basic's "Device Not Present" error.

Things to look out for...

When ASM is installed and running, the following information is relevant. The Basic cheat code is inserted to a new routine within the ASM code to allow the inclusion of the new commands. The program loads like a Basic program into memory starting at address \$0801. When it has been run, the start of Basic is shifted up to about \$2500. You may still type any Basic direct command such as LOAD, SAVE, POKE, etc but caution is advised using POKE on addresses between \$0801 and \$2500.

In both memory and disk modes, code is edited above about \$2500. In memory mode, the symbol table begins in memory after the source program. This also applies in disk mode hence any program in memory will be preserved. This means that you should type "new" before commencing a disk mode assembly to maximize symbol space. During assembly, the BASIC ROM is overlaid out. The space from \$C000 to \$CFFF is left free. Symbols that occupy the space from the end of any program in memory up to \$EFFF.

ASM should operate peacefully with the Basic interpreter. The "reset" command may be used to resolve certain situations where the system is not operating correctly. However it has a limited effect, and it may become necessary to powerdown should the system still operate incorrectly.

ASM is source code compatible with my earlier PCL system assembler published in a previous *Your Commodore* and also my PL154 assembler. ASM's facilities are in effect, a superset of the PCL assembler's facilities and ASM could therefore replace the PCL assembler if desired.

To help you become familiar with the system, I have included some example source files listing which may be assembled using ASM. The comment fields at the beginning indicate which mode they should be run in. Good luck!

DMA Assembler Directives

BYT Byte value directive. Single values or strings in single quotes values.
e.g. byt \$1, \$4, \$01, 'abcd', 0

WORD

Word directive
e.g. word \$0000, \$001A, \$0000

EQZ

Zero page equate. Used to assign a zero-page value to a symbol.
e.g. pointer equ \$10

EQA

Absolute equate. Used to assign an absolute value to a symbol.
e.g. var equ \$0800

ORG

Set code origin. In disk mode - also sets code load address.
e.g. org \$C000

RES

Reserve memory.
e.g. res 60 (reserves 60 bytes)

LST

Causes assembler to list during pass 2.

SYM

Causes assembler to display symbols upon completion of the assembly.

LNK

Chain to another file. When the file has been assembled, the current file resumes assembling.
e.g. ldk "symbols.asm"

REL

Relocation offset. The code origin is set by the org directive. This directive allows you to assemble code to run on one address (org) while being used to another memory area (rel).
e.g. org \$0000
rel \$C000

Disk Edit

Delve further into your disks with the help of this article.

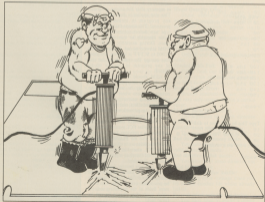
By Fergal Moore

Disk editing is what separates a casual disk user from a professional. Once you can edit disks, a whole world of seemingly impossible tasks becomes possible. Files can be locked, unscratched, cloned, relocated, and renamed when you have the commands and the know-how.

Firstly, a word of warning: don't edit a disk with important programs on it, unless you know what you're doing. Use unwanted disks for practice, and take backups of valuable disks. A good Disk Editor will make things a lot easier: you have no need for complex commands. There is a good example in the December 1987 edition of *Your Commodore*. This is not essential though - you can make do with the commands and DISPLAY TAB on the demo disk you get with your drive.

Commands

The commands regarding direct disk access are called the Block commands (a Block is another name for a Sector).



Your disk drive manual contains more detailed explanations, but a summary follows.

To use these commands, you'll need to have two files open — one for commands and the other to a buffer for data. The command channel you will probably be familiar with:

```
OPEN15,0,0
```

The data channel can be any other number, but 2 or 3 are usually used:

```
OPEN5,2,0 *
```

After these open commands, PRINT 15 will send commands, and PRINT 5 will send data to the channel.

Note that when 'drive' is mentioned, this means 0 for a single drive. The device is usually 0, but can be changed. See the examples on the disk for more information.

Block-Read

```
SYNTAX: PRINT #15, "B-R:";
channel;
drive;
track;
sector;
```

This command transfers the required sector into the data channel (in our case 5). Then use the GET 5 command to read the information into a variable.

It's important to note that Block-Read will only read up as far as the Block-Pointer, which is usually 0. The USER1 command is usually used, as this sets the pointer to 255 automatically, allowing the sector to be read in one operation.

USER1

```
SYNTAX: PRINT #15, "U1:";
channel;
drive;
track;
sector;
```

Block-Write

```
SYNTAX: PRINT #15, "B-W:";
channel;
```

```
drive;
```

```
track;
```

```
sector;
```

To use this command, fill up the channel with information to write, using PRINT 5, then use the command to write to the required sector. This is the exact opposite of Block-Read, so again USER1 is usually used.

USER2

```
SYNTAX: PRINT #15, "U2:";
channel;
drive;
track;
sector;
```

Block-Pointer

```
SYNTAX: PRINT #15, "B-P:";
channel;
location;
```

By using this command, you can specify where exactly in the sector you want the next read or write to begin. This allows you to read or alter individual bytes in a sector, starting at 'location'. See the Disk Name program for a demo.

Block-Allocate

```
SYNTAX: PRINT #15, "B-A:"; drive;
track;
```

```
sector;
```

This allocates a bit in the Block Availability Map to show a sector is in use. It is used in connection with random access databases.

Block-Free

```
SYNTAX: PRINT #15, "B-F:"; drive;
track;
```

```
sector;
```

This is the opposite of Block-Allocate, and frees up sectors for use without destroying the actual data on them. If a scan is made, the data will probably be overwritten, as the RAM has marked the sector as empty.

Disk Maps

Before you can use these commands, you will need some information on disk structure. The maps will provide this information, and information on file structure.

Editing

There are a number of files provided here for demonstration purposes. The best way to learn is to study these programs with the maps close at hand. They are heavily REBbed, but here are some notes explaining what's going on. Even if you don't learn anything, they are useful utilities to have.

Protect File

This program 'locks' (i.e. prevents it from being scratched) the first file on a disk by setting bit 6 of the file type to 1, effectively OR'ing it with 80. This prevents accidental erasure, and has a < beside its name in the directory. By adding 12 to the last pointer number, and changing the sector number, any program in the directory may be protected.

Disk Name

This allows you to change the name of the disk without erasing the contents. It makes use of the fact that the disk name is stored at byte 104, track 11, sector 0.

Load Address

This changes the load address of any program to a given address. It searches for the first sector of storage, and bytes 2-3 contain the load address. It is most useful with spinite data.

Unscratch

On scratching a file, the filetype in the directory is usually marked as being deleted. This program searches the disk for a scratched program and restores the filetype, unscratching the file. You are advised to save the unscratched program to another disk in case of another accident. Note that this will probably not work if something has been saved to the disk since the SCRATCH, as it may have been saved over the old program. Enjoy your disk editing!

BLOCK DISTRIBUTION BY TRACK

Track number	Block range	Total
1 to 17	0 to 30	21
18 to 24	0 to 18	19
25 to 30	0 to 17	18
31 to 32	0 to 16	17

1540/1541 BAM FORMAT

Track 18, Sector 0.		
BYTE	CONTENTS	DEFINITION
0,1	38,01	Track and block of first directory block.
2	45	ASCII character A indicating 4040 format.
3	0	Null flag for future DOS use.
4-143		Bit map of available blocks for tracks 1-32.
		*1 = available block 0 = block not available (each bit represents one block)

1540/1541 DIRECTORY HEADER

Track 18, Sector 0.		
BYTE	CONTENTS	DEFINITION
144-161		Disk name with shifted spaces.
162-163	160	Disk ID.
164		Shifted space
165-166	50,65	ASCII representation for 2A, which is DOS version and format type.
166-167	160	Shifted spaces.
177-255	0	Nulls, not used.
Note: ASCII characters may appear in locations 180 thru 191 on some diskettes.		

SEQUENTIAL FORMAT

BYTE	DEFINITION
0-1	Track and block of next sequential data block.
2-256	256 bytes of data with carriage return as record terminators.

PROGRAM FILE FORMAT

BYTE	DEFINITION
0-1	Track and block of next block in program file.
2-256	256 bytes of program info stored in CBM memory format (with key words colonised). End of file is marked by three zero bytes.

RELATIVE FILE FORMAT

DATA BLOCK	
BYTE	DEFINITION
0-1	Track and block of next data block.
2-256	254 bytes of data. Empty records contain FF (all binary ones) in the first byte followed by 00 (binary all zeros) to the end of the record. Partially filled records are padded with nulls (00).
SIDE SECTOR BLOCK	
BYTE	DEFINITION
0-1	Track and block of next side sector block.
2	Side sector number (0-5).
3	Record length.
4-5	Track and block of first side sector (number 0).
6-7	Track and block of second side sector (number 1).
8-9	Track and block of third side sector (number 2).
10-11	Track and block of fourth side sector (number 3).
12-13	Track and block of fifth side sector (number 4).
14-15	Track and block of sixth side sector (number 5).
16-256	Track and block pointers to 120 data blocks.



WIN WITH COURSEMASTER

NEW

THE COMPUTER HORSE-RACING PROGRAMME

- **WATER-LOVE RACE IN SECONDS** - BUY DAILY NEWSPAPER IS ALL YOU NEED
- **WAGER** out of date - Both H, Hand and Flat - Fast data entry
- **AMAZING ACCURACY** - How you CAN BEAT THE ADVISORY
- **Works on the simple principle that FASTER HORSES BEAT SLOW ONES**
- **Clearly identifies race selection in every race plus those AMAZING Features:**
 - First, Second and Third place shown for forecasts and Throats etc. (Forecasting more accurate than 100 ft bet)
 - **Actually works out your WINNINGS** of 1000 popular bets including SINGLES and DOUBLES, win and each way, PATENTS, TRAPSETS, CANADIAN, HEINZ etc. Good BACKWASH and LONG ODDS bet (clearly shown)
 - **NEWBORN** - but what BETTING SLIP for you
 - **Maximum 4 BANK ACCOUNTS** - BET Like PROFESSIONALS & Get Records of your bets in any number of accounts. Keep a complete record of your betting or compare YOURSELF/US/US against your favourite jockey.
 - **PLUS!** - The **AMAZING COURSEMASTER SYSTEM** - This superb betting system is included in the programme & system which regularly produces huge wins from small stakes. Try it out - you'll find it really makes you win - on the top 20!
 - **Supplied with 35 pages BETTING GUIDE** and **MARSHAL FREE COPY TIP OF THE MONTH SENT TO EVERY PURCHASER** (Supplied on DISK at £14.95 for:-
 - All Spectrums, Commodore 64/128, Amstrad CPC's, BBC B
 - IBM, Amstrad CPC's, BBC B

Specialist on DISK at £14.95 for:-
 All Spectrums, Commodore 64/128, Amstrad CPC's, BBC B
 IBM, Amstrad CPC's, BBC B
 Special Also available for Spectral DL on Microdrive at £18.95.
 Please state machine and disc size

INTRASET LTD. (DEPT KC), FREEPOST (No Stamp needed) 8 Glendale Close, Gosce Green, Blithfield, Warrington, Cheshire, WA3 2BN. Or large SAE for further details.

PRINT ENHANCER Plus/4 disk £19.99

Print desktop publishing Programmability enhanced. High quality printers, from B&W, P&I and graphics. Supplied with 16 fonts and font-design program. MFPS-801/800, VIC 1000.

RS232 INTERFACE 64, 128 or Plus/4" ... £24.99

Measuring only 14x16x10mm, the smallest and nearest unit available. Connects to Commodore 1214, color-print standard. Connections with B&W, Easypage, Superprint, Superline, Slim Office, Super Plus etc. Connect to PDP20 printers, modems and other devices. Supplied with 1 metre of cable (500 The extra metre) terminated with a male (printer) 25-way D-subconnector, or a custom cable made to your requirements. Our after sales service for unusual printers guaranteed 10 day into use and running. Supplied with a terminal emulator program, file transfer and other utilities on tape/disk.

IBM PC: File Transfer Utility £9.99

Captures text or binary files via the COM or RS232 port. Supplied on a 5.25" disk format 1-1/2" disk. Simple instructions provided by on-line help.

VIEWDATA TERMINAL 64 or Plus/4" £14.99

Access: Prodat, Microdat, Cityline etc. Works with any user port RS232 interface and 1000/10 modem.

CENTRONICS CABLE 64, 128 £18.99

Use with Easypaper, Superprint etc. Drive for B&W or dot or laser - 6002 ASSEMBLER 64 or Plus/4" tape/disk £12.99

A sophisticated two-pass typographic assembler and text editor which supports tags and gas tags. The assembler, editor, source and object code may all be resident in memory simultaneously. Outstanding speed and interactive code development. Very fast edit/assemble/translate cycles.

286 EMULATOR/ASSEMBLER 64 disk £12.99

A unique integrated DOS development package. The emulator simulates 286 source code into optimized 8088 which runs on the 64 at about one sixth the speed of a 286/4.286. The cross-assembler generates hex or binary 286 object files. The disk contains a powerful editor and example programs.

Please specify options, send cheque/P.O. or order by credit. Prices include P&I and P&P. Overseas orders add 25.00. Allow up to 1 week for delivery.

YORK ELECTRONIC RESEARCH
 The Fitzhugh Centre, Dept Y10, 4 Fitzhugh's, York YO1 4AB
 Tel: (0904) 510120



A SUBSCRIPTION IS ONLY A CALL AWAY...



What could be a better way of keeping up to date with the latest news and developments in the world of the Commodore range of computers, than by ordering a direct subscription to 'YOUR COMMODORE', delivered direct to your door each month.

We've now made ordering a subscription easier than ever before by calling our CREDIT CARD HOTLINE. Simply give us a call, quoting your credit card details and delivery address and we'll do the rest! Remember, a subscription delivered to any address in the UK is POST FREE, all overseas subscriptions include postage.

Subscription Rates: UK £15.00; Europe £23.35; Middle East £28.50; Far East £26.42; Rest of the World £24.00 or USA \$42.00. (Airmail Rates on Request).



Telephone 0442 876661/4
 Between usual office hours.



LISTINGS

63	3680	6874	778,9,41,2,1,74,7,2,4	1,59,5,70,5,3,28,5,7,5,4,10,2,1,1	74	2420	6208	1,99,1,77,21,8,2,210,2	1,64,1,188,70,1,22,48,123,210,1	80	3220	5874	28,12,82,20,149,32,9
64	3680	6874	3,30,8,1,78,10,2,2,30,7,1,12,8,2,3,10,8,4,1,78,10,2,3	2,79,2,1,2,2,2,2,2,2,2,2,2,2,2	75	2520	6374	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	81	3220	5874	32,12,82,20,149,32,9
65	3680	6874	3,30,8,1,78,10,2,2,30,7,1,12,8,2,3,10,8,4,1,78,10,2,3	2,79,2,1,2,2,2,2,2,2,2,2,2,2,2	76	2520	6374	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	82	3220	5874	32,12,82,20,149,32,9
66	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	77	2520	6374	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	83	3220	5874	32,12,82,20,149,32,9
67	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	78	2520	6374	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	84	3220	5874	32,12,82,20,149,32,9
68	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	79	2520	6374	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	85	3220	5874	32,12,82,20,149,32,9
69	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	80	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	86	3220	5874	32,12,82,20,149,32,9
70	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	81	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	87	3220	5874	32,12,82,20,149,32,9
71	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	82	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	88	3220	5874	32,12,82,20,149,32,9
72	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	83	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	89	3220	5874	32,12,82,20,149,32,9
73	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	84	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	90	3220	5874	32,12,82,20,149,32,9
74	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	85	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	91	3220	5874	32,12,82,20,149,32,9
75	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	86	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	92	3220	5874	32,12,82,20,149,32,9
76	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	87	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	93	3220	5874	32,12,82,20,149,32,9
77	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	88	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	94	3220	5874	32,12,82,20,149,32,9
78	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	89	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	95	3220	5874	32,12,82,20,149,32,9
79	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	90	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	96	3220	5874	32,12,82,20,149,32,9
80	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	91	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	97	3220	5874	32,12,82,20,149,32,9
81	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	92	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	98	3220	5874	32,12,82,20,149,32,9
82	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	93	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	99	3220	5874	32,12,82,20,149,32,9
83	3680	6874	7,2,3,1,20,1,7,2,4,2,2,1,1,1	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	94	3220	5874	2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	1,19,1,121,2,2,2,2,1,2,2,2,2,2,2	100	3220	5874	32,12,82,20,149,32,9

60	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	67	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	74	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
61	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	68	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	75	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
62	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	69	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	76	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
63	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	70	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	77	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
64	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	71	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	78	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
65	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	72	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	79	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100
66	8880 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	73	8910 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100	80	8970 2474 5,200, 100, 0, 100, 5 888, 200, 0, 470, 0, 80, 50, 100, 100 100, 100

Binders

Organise and protect your disk with Commodore Disk User disk binders and data disks.

Why not keep your Commodore Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder comes complete with 10-disk sleeves to organise and protect your program disks. Why not buy a disk binder to house all of your data disks? We can even supply Commodore Disk User data disks. The Commodore Disk User tags immediately identify your disks and there's room to file them and document the disks details. Send for your disks and binders now!

Prices are as follows:

Commodore Disk User Binder (4.95, including 10 sleeves. Order code **BDYU**)

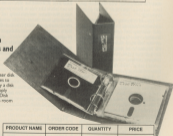
Commodore Disk User Binder with 10 sleeves and 10 disks, (9.95) Order code **BDYU2**

10 sleeves for insertion in binder, (1.50). Order code **BDSS**

20 sleeves for inclusion in binder, (2.75). Order code **BDSS2**

10 Commodore Disk User data disks, (3.95). Order code **BDSS3**

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, 9 SMALL ROAD, HEMEL HEMPSTEAD, HERTS HP2 7BH. Please allow 28 days for delivery.



PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Cheapest postage add £1.00			
TOTAL			

Extending Basic

Declaring and using labels in Basic is not as difficult as it may seem

By Burghard-Heury Lehmann

One advantage of writing machine code programs with an assembler is that you can declare labels for jumps and branches. An example of this is "Speedy Assembler", written by yours truly, and still available from Readers' Services for everybody who wants to get into serious machine code programming!

Instead of having to give jumps and subroutine calls an absolute address, and having to calculate relative branches, you simply declare a label on the left-hand side at the start of the routine to be jumped or branched to, and the assembler does the rest. It stores the label in what is called "the symbol table", with the address of the location at that point next to it.

Later on, when it finds that label next to a jump or branch instruction, it fetches the address from the symbol table and, in case of a jump or subroutine call, assembles it as the location to be jumped to or to be called on, in the case of a branch instruction, calculates the length of the branch. The programmer doesn't have to trouble himself with any of this. Nor does he have to go to the trouble of calculating branches. Instead he just attaches a name to the routine or subroutine in question.

And what's more, labels like this add tremendously to the readability of the program, because you can give each routine and subroutine a name that suits you. This helps quite a bit when a bug has developed in the

program (and doesn't it always!), and the programmer has to spend ages finding it!

Labels in Basic

Because of all this, I felt for a long time that it would be nice to be able to use labels like this in Basic programs too. No more memorizing of line numbers, and, most of all, no more remembering of GOTO's and GOSUB's whenever one changes the program.

It is actually surprisingly easy to introduce such a facility to the rather poverty-stricken Commodore 64 Basic. In the last article of this series we've developed a routine which allows us to give GOTO and GOSUB with variable names. So we've already got the basic facility to labelize jumps and subroutine calls, but we still have to declare the value of the label at the beginning of the program with a line

like "Subroutine = 1000". To do this job for us, the computer has to build a symbol table of sorts, and this has to be done before the program is actually run. This is because, during execution, when a GOTO or GOSUB is encountered, the computer has to know where to jump to.

You may know that most assemblers are called "Two Pass Assemblers". This is because in order to deal with jumps and branches, the assembler has to do its job in two goes. First it goes through the whole of the textfile and builds the symbol table, and then it has a second go in which it is able to assemble the source in earnest.

To use labels to the full in Basic, we have to do a similar thing. Before the program is run in earnest, the computer has to sift through the whole of the textfile and collect all the line numbers to which it has to GOTO and GOSUB later on. This means that it

Figure 1

A\$71	BasicRun	PHP	Save status register
A\$72		LDA #	This is direct run mode
A\$74		JSR \$FF90	Set kernel flags to direct run mode
A\$71		PLP	Retrieve status register
A\$76		BNE LineNo	If Z=0 then RUN plus line number
A\$7A		JMP	Do CLR and start program
A\$7D	LineNo	JSR \$A660	Do CLR
A\$80		JMP \$A697	Jump to GOTO and start program

takes a bit of time until running proper starts. But I don't think that this is a major handicap, and it certainly is worth it!

Running a Basic Program

So we have to intercept the RUN command. To do this, let's look first of all a bit closer at what it does under normal circumstances.

After you've typed in your Basic program and then given the RUN command to execute your program, the Basic interpreter jumps to the routine at BAS71. Figure 1 gives you a disassembly of that routine.

First the flag register is saved on the stack. Then a call to a Kernel routine is made. This routine puts the computer into the direct run mode by loading the system variable \$RD with zero and setting ST, which is the status variable.

Then the flag register is pulled from the stack again. If the zero flag equals one, that is, if the last value in the accumulator has been zero, then there were no parameters with the RUN command. In this case the routine continues, otherwise it branches forward, because a line number has been given with RUN.

If no line number is given with RUN the routine jumps to the CLR routine and doesn't return. If a line number has been given with RUN, the routine calls on the CLR routine and then jumps to the GOTO routine, because in the real, RUN 100 is similar to GOTO 100. The only difference is that RUN clears all the Basic variables, while GOTO leaves them untouched.

The CLR routine clears all the Basic variables and gives Basic a fresh start, so that variables and arrays can be built up anew. This is mainly done by setting the string storage pointer and the array storage pointer to the end of the Basic textfile, because, as you might know, Basic stores all the variables and arrays declared in a program directly after the end of the textfile.

Modified RUN

Our modified RUN routine starts at line 1520 (Listing 1). First we deal with the RUN command more or less in the same way as the ROM routine. That is, we set the Kernel flag to the direct run mode (line 1520-1530), and then we clear the Basic variables (line 1540).

I haven't bothered about RUN line numbers, so this won't work with the routine as it is. If you give a line number with RUN, it will just be ignored. But nevertheless, if you do want this little-used facility, it shouldn't be impossible for you to add it with the help of the explanation of the normal RUN routine, which I've given above.

Next, we set zero page 294/295 to the start of the Basic textfile (line 1580-1610). Zero page locations 291 to 293 are never used by the Commodore operating system, so these locations are absolutely safe to be used for your own purposes. There are many other zero page locations which are used by the Kernel or the Basic interpreter, but which are usually quite safe to use. For example, if you don't do any floating point arithmetic, you may use locations 504 to 50F without any trouble. But the point is always to think before you use a zero page variable, otherwise the system might do very funny things indeed!

After that we go into the main loop (SEARCHLP), which looks at each Basic line to see if it contains a label (line 1630-1610). To understand SEARCHLP, here is a short explanation of how a Basic line is stored in memory:

First of all the line number is given in the usual low byte/high byte fashion. Next there are two bytes which contain the so-called link pointer. Each link pointer points to the beginning of the next line. This makes it very easy to search through a Basic textfile, because you just have to jump from one link pointer to the other, each time looking at the line number preceding it, and in next to no time you've found the line you're looking for.

Each Basic line is finished with a zero, which is the standard terminator used by Commodore (strings too are always terminated with zero). At the end of the textfile there will be two zeros in the locations where otherwise the next line number would be. So this is how the computer will know when it has reached the end of the textfile.

If you look at SEARCHLP in our program, you'll find the routine testing for those two zeros right at the beginning of the loop (line 1630-1660).

Declaring a Label

Naturally we have to tell the computer when it has found a label. For that

we have to make a label stand out in some way.

To do this I have chosen the following way of declaring a label: a label has to be at the beginning of a line, after the line number, and it has to be preceded by a full stop. Of course, you are free to experiment with methods which might suit you better, because this is the whole purpose of this series of articles, to enable you to develop extended Basic routines which suit your particular needs!

Anyhow, in the routine given, the computer looks for a full stop and that tells him that it has found a label (line 1690-1720). So it jumps to the routine which I called LABELFOUND. First of all the current location in the textfile is saved in STA/TH, so that afterwards the computer can continue searching the textfile for more labels (line 1990-1990).

Then it goes forward five bytes to point at the label itself (line 2000-2000). Remember, all we are doing here is a simple LFT operation, like "LET LABEL = 1000" (or "LABEL = 1000" if you omit the LET). So the next ROM routine (SHOWR) we call validates our label, that is, finds out if it is a permitted variable name (as you know, a valid Basic variable name has to start with a letter).

If the name is valid, the first two characters of it are stored in the variable area, which starts immediately after the end of the Basic textfile. On return from SHOWR, the low byte of the variable location is in the accumulator, and the high byte is in the Y-Registers. We store this in zero page 240/244, which is the system variable pointer (line 2110-2120).

Next we get the line number and store it in zero page 262/63, which is the floating point accumulator # 1 (line 2160-2160). The next routine (SRV99) which we call converts the line number into a proper floating point number (line 2250). We have to do it in this rather roundabout way because variables of this type have to be stored as floating point numbers in order to be recognized later on!

Finally we call \$BBD0 (line 2260), which stores the value contained in register # 1 in the variable area itself. Now our label has been stored like any other Basic variable in the variable store, including the number of the line on which it appeared. We can return to our main search loop and look for the next label.

The fact that each of the labels we

declare is stored like any other Basic variable means, of course, that only the first two characters of the label will be recognized. The rest of the label is ignored, which limits the use of labels

rather, because it doesn't give much scope for declaring meaningful names which are recognized.

Therefore, in the next article I'd like to develop a routine which builds

and recognizes a symbol table away from the Basic variable area. This will enable us to declare labels with six or more characters which are fully recognized.

Listing 1

```

10          GOSUB 40000
20          END
30          :
40          CHARGET EQU $007F
50          EXECUT EQU $0000
60          :
70          :
80          :
90          :
100         :
110         :
120         :
130         :
140         :
150         :
160         :
170         :
180         :
190         :
200         :
210         :
220         :
230         :
240         :
250         :
260         :
270         :
280         :
290         :
300         :
310         :
320         :
330         :
340         :
350         :
360         :
370         :
380         :
390         :
400         :
410         :
420         :
430         :
440         :
450         :
460         :
470         :
480         :
490         :
500         :
510         :
520         :
530         :
540         :
550         :
560         :
570         :
580         :
590         :
600         :
610         :
620         :
630         :
640         :
650         :
660         :
670         :
680         :
690         :
700         :
710         :
720         :
730         :
740         :
750         :
760         :
770         :
780         :
790         :
800         :
810         :
820         :
830         :
840         :
850         :
860         :
870         :
880         :
890         :
900         :
910         :
920         :
930         :
940         :
950         :
960         :
970         :
980         :
990         :
1000        :
1010        :
1020        :
1030        :
1040        :
1050        :
1060        :
1070        :
1080        :
1090        :
1100        :
1110        :
1120        :
1130        :
1140        :
1150        :
1160        :
1170        :
1180        :
1190        :
1200        :
1210        :
1220        :
1230        :
1240        :
1250        :
1260        :
1270        :
1280        :
1290        :
1300        :
1310        :
1320        :
1330        :
1340        :
1350        :
1360        :
1370        :
1380        :
1390        :
1400        :
1410        :
1420        :
1430        :
1440        :
1450        :
1460        :
1470        :
1480        :
1490        :
1500        :
1510        :
1520        :
1530        :
1540        :
1550        :
1560        :
1570        :
1580        :
1590        :
1600        :
1610        :
1620        :
1630        :
1640        :
1650        :
1660        :
1670        :
1680        :
1690        :
1700        :
1710        :
1720        :
1730        :
1740        :
1750        :
1760        :
1770        :
1780        :
1790        :
1800        :
1810        :
1820        :
1830        :
1840        :
1850        :
1860        :
1870        :
1880        :
1890        :
1900        :
1910        :
1920        :
1930        :
1940        :
1950        :
1960        :
1970        :
1980        :
1990        :
2000        :
2010        :
2020        :
2030        :
2040        :
2050        :
2060        :
2070        :
2080        :
2090        :
2100        :
2110        :
2120        :
2130        :
2140        :
2150        :
2160        :
2170        :
2180        :
2190        :
2200        :
2210        :
2220        :
2230        :
2240        :
2250        :
2260        :
2270        :
2280        :
2290        :
2300        :
2310        :
2320        :
2330        :
2340        :
2350        :
2360        :
2370        :
2380        :
2390        :
2400        :
2410        :
2420        :
2430        :
2440        :
2450        :
2460        :
2470        :
2480        :
2490        :
2500        :
2510        :
2520        :
2530        :
2540        :
2550        :
2560        :
2570        :
2580        :
2590        :
2600        :
2610        :
2620        :
2630        :
2640        :
2650        :
2660        :
2670        :
2680        :
2690        :
2700        :
2710        :
2720        :
2730        :
2740        :
2750        :
2760        :
2770        :
2780        :
2790        :
2800        :
2810        :
2820        :
2830        :
2840        :
2850        :
2860        :
2870        :
2880        :
2890        :
2900        :
2910        :
2920        :
2930        :
2940        :
2950        :
2960        :
2970        :
2980        :
2990        :
3000        :
3010        :
3020        :
3030        :
3040        :
3050        :
3060        :
3070        :
3080        :
3090        :
3100        :
3110        :
3120        :
3130        :
3140        :
3150        :
3160        :
3170        :
3180        :
3190        :
3200        :
3210        :
3220        :
3230        :
3240        :
3250        :
3260        :
3270        :
3280        :
3290        :
3300        :
3310        :
3320        :
3330        :
3340        :
3350        :
3360        :
3370        :
3380        :
3390        :
3400        :
3410        :
3420        :
3430        :
3440        :
3450        :
3460        :
3470        :
3480        :
3490        :
3500        :
3510        :
3520        :
3530        :
3540        :
3550        :
3560        :
3570        :
3580        :
3590        :
3600        :
3610        :
3620        :
3630        :
3640        :
3650        :
3660        :
3670        :
3680        :
3690        :
3700        :
3710        :
3720        :
3730        :
3740        :
3750        :
3760        :
3770        :
3780        :
3790        :
3800        :
3810        :
3820        :
3830        :
3840        :
3850        :
3860        :
3870        :
3880        :
3890        :
3900        :
3910        :
3920        :
3930        :
3940        :
3950        :
3960        :
3970        :
3980        :
3990        :
4000        :
4010        :
4020        :
4030        :
4040        :
4050        :
4060        :
4070        :
4080        :
4090        :
4100        :
4110        :
4120        :
4130        :
4140        :
4150        :
4160        :
4170        :
4180        :
4190        :
4200        :
4210        :
4220        :
4230        :
4240        :
4250        :
4260        :
4270        :
4280        :
4290        :
4300        :
4310        :
4320        :
4330        :
4340        :
4350        :
4360        :
4370        :
4380        :
4390        :
4400        :
4410        :
4420        :
4430        :
4440        :
4450        :
4460        :
4470        :
4480        :
4490        :
4500        :
4510        :
4520        :
4530        :
4540        :
4550        :
4560        :
4570        :
4580        :
4590        :
4600        :
4610        :
4620        :
4630        :
4640        :
4650        :
4660        :
4670        :
4680        :
4690        :
4700        :
4710        :
4720        :
4730        :
4740        :
4750        :
4760        :
4770        :
4780        :
4790        :
4800        :
4810        :
4820        :
4830        :
4840        :
4850        :
4860        :
4870        :
4880        :
4890        :
4900        :
4910        :
4920        :
4930        :
4940        :
4950        :
4960        :
4970        :
4980        :
4990        :
5000        :
5010        :
5020        :
5030        :
5040        :
5050        :
5060        :
5070        :
5080        :
5090        :
5100        :
5110        :
5120        :
5130        :
5140        :
5150        :
5160        :
5170        :
5180        :
5190        :
5200        :
5210        :
5220        :
5230        :
5240        :
5250        :
5260        :
5270        :
5280        :
5290        :
5300        :
5310        :
5320        :
5330        :
5340        :
5350        :
5360        :
5370        :
5380        :
5390        :
5400        :
5410        :
5420        :
5430        :
5440        :
5450        :
5460        :
5470        :
5480        :
5490        :
5500        :
5510        :
5520        :
5530        :
5540        :
5550        :
5560        :
5570        :
5580        :
5590        :
5600        :
5610        :
5620        :
5630        :
5640        :
5650        :
5660        :
5670        :
5680        :
5690        :
5700        :
5710        :
5720        :
5730        :
5740        :
5750        :
5760        :
5770        :
5780        :
5790        :
5800        :
5810        :
5820        :
5830        :
5840        :
5850        :
5860        :
5870        :
5880        :
5890        :
5900        :
5910        :
5920        :
5930        :
5940        :
5950        :
5960        :
5970        :
5980        :
5990        :
6000        :
6010        :
6020        :
6030        :
6040        :
6050        :
6060        :
6070        :
6080        :
6090        :
6100        :
6110        :
6120        :
6130        :
6140        :
6150        :
6160        :
6170        :
6180        :
6190        :
6200        :
6210        :
6220        :
6230        :
6240        :
6250        :
6260        :
6270        :
6280        :
6290        :
6300        :
6310        :
6320        :
6330        :
6340        :
6350        :
6360        :
6370        :
6380        :
6390        :
6400        :
6410        :
6420        :
6430        :
6440        :
6450        :
6460        :
6470        :
6480        :
6490        :
6500        :
6510        :
6520        :
6530        :
6540        :
6550        :
6560        :
6570        :
6580        :
6590        :
6600        :
6610        :
6620        :
6630        :
6640        :
6650        :
6660        :
6670        :
6680        :
6690        :
6700        :
6710        :
6720        :
6730        :
6740        :
6750        :
6760        :
6770        :
6780        :
6790        :
6800        :
6810        :
6820        :
6830        :
6840        :
6850        :
6860        :
6870        :
6880        :
6890        :
6900        :
6910        :
6920        :
6930        :
6940        :
6950        :
6960        :
6970        :
6980        :
6990        :
7000        :
7010        :
7020        :
7030        :
7040        :
7050        :
7060        :
7070        :
7080        :
7090        :
7100        :
7110        :
7120        :
7130        :
7140        :
7150        :
7160        :
7170        :
7180        :
7190        :
7200        :
7210        :
7220        :
7230        :
7240        :
7250        :
7260        :
7270        :
7280        :
7290        :
7300        :
7310        :
7320        :
7330        :
7340        :
7350        :
7360        :
7370        :
7380        :
7390        :
7400        :
7410        :
7420        :
7430        :
7440        :
7450        :
7460        :
7470        :
7480        :
7490        :
7500        :
7510        :
7520        :
7530        :
7540        :
7550        :
7560        :
7570        :
7580        :
7590        :
7600        :
7610        :
7620        :
7630        :
7640        :
7650        :
7660        :
7670        :
7680        :
7690        :
7700        :
7710        :
7720        :
7730        :
7740        :
7750        :
7760        :
7770        :
7780        :
7790        :
7800        :
7810        :
7820        :
7830        :
7840        :
7850        :
7860        :
7870        :
7880        :
7890        :
7900        :
7910        :
7920        :
7930        :
7940        :
7950        :
7960        :
7970        :
7980        :
7990        :
8000        :
8010        :
8020        :
8030        :
8040        :
8050        :
8060        :
8070        :
8080        :
8090        :
8100        :
8110        :
8120        :
8130        :
8140        :
8150        :
8160        :
8170        :
8180        :
8190        :
8200        :
8210        :
8220        :
8230        :
8240        :
8250        :
8260        :
8270        :
8280        :
8290        :
8300        :
8310        :
8320        :
8330        :
8340        :
8350        :
8360        :
8370        :
8380        :
8390        :
8400        :
8410        :
8420        :
8430        :
8440        :
8450        :
8460        :
8470        :
8480        :
8490        :
8500        :
8510        :
8520        :
8530        :
8540        :
8550        :
8560        :
8570        :
8580        :
8590        :
8600        :
8610        :
8620        :
8630        :
8640        :
8650        :
8660        :
8670        :
8680        :
8690        :
8700        :
8710        :
8720        :
8730        :
8740        :
8750        :
8760        :
8770        :
8780        :
8790        :
8800        :
8810        :
8820        :
8830        :
8840        :
8850        :
8860        :
8870        :
8880        :
8890        :
8900        :
8910        :
8920        :
8930        :
8940        :
8950        :
8960        :
8970        :
8980        :
8990        :
9000        :
9010        :
9020        :
9030        :
9040        :
9050        :
9060        :
9070        :
9080        :
9090        :
9100        :
9110        :
9120        :
9130        :
9140        :
9150        :
9160        :
9170        :
9180        :
9190        :
9200        :
9210        :
9220        :
9230        :
9240        :
9250        :
9260        :
9270        :
9280        :
9290        :
9300        :
9310        :
9320        :
9330        :
9340        :
9350        :
9360        :
9370        :
9380        :
9390        :
9400        :
9410        :
9420        :
9430        :
9440        :
9450        :
9460        :
9470        :
9480        :
9490        :
9500        :
9510        :
9520        :
9530        :
9540        :
9550        :
9560        :
9570        :
9580        :
9590        :
9600        :
9610        :
9620        :
9630        :
9640        :
9650        :
9660        :
9670        :
9680        :
9690        :
9700        :
9710        :
9720        :
9730        :
9740        :
9750        :
9760        :
9770        :
9780        :
9790        :
9800        :
9810        :
9820        :
9830        :
9840        :
9850        :
9860        :
9870        :
9880        :
9890        :
9900        :
9910        :
9920        :
9930        :
9940        :
9950        :
9960        :
9970        :
9980        :
9990        :

```

PROGRAMMING

1390	LEA #3A	2030	STA #7A
1395	FBA	2040	LEA #7B
1399	LEA #3B	2050	ASC #0
1399	FBA	2060	STA #7B
1399	LEA #400	2070	
1399	FBA	2080	
1399	JEE #07D: #71	2090	STORE LABEL
1399	JMP #ATVA	2100	JEE #0000
1399		2110	STA #80
1399		2120	STY #AA
1399		2130	
1400	STORE LABEL	2140	GET LINE NUMBER
1410		2150	
1420	SPFC #T JEE #0A00T	2160	LDY #C
1430	SEE SPFC #T	2170	LEA (254).Y
1440		2180	STA #B7
1450		2190	INT
1460	STO	2200	LEA (254).Y
1470		2210	STA #C2
1480	MODIFIED 'SUB'-ROUTINE	2220	
1490		2230	AND STORE IN RLP ACCU
1500	DO 'SUB'	2240	
1510		2250	JEE #0000
1520	END #T LEA #C	2260	
1530	JEE #FF90	2270	STORE VALUE IN VARIABLE STORE
1540	JEE #A050	2280	
1550		2290	JEE #0000
1560	DO THROUGH FIG & ASSEMBLE LABELS	2300	
1570		2310	FINISHED
1580		2320	
1590	LEA #00001	2330	
1600	STA #254	2340	JMP #STARTLFE
1610	LEA #00001	2350	
1610	STA #254	2360	
1620		2370	
1630	SEARCHLFE LDY #C .END OF FIG	2380	TEST FOR REST OF 'OFF'
1640	LEA (254).Y	2390	
1650	SEE SEARCHLFE:	2400	OFF #T JEE #0A00T
1660	INT	2400	OFF 'F
1670	LEA (254).Y	2410	SEE NORMAL1
1680	SEE REST	2420	JEE #0A00T
1690	SEARCHLFE LDY #A	2430	OFF 'F
1700	LEA (254).Y	2440	SEE NORMAL1
1710	OFF 'F	2450	
1720	SEE LABELFOUND	2460	EXECUTE 'OFF' COMMAND
1730	SEARCHLFE LDY #C .NEXT LINE	2470	
1740	LEA (254).Y	2480	JEE #EXTRASOFF
1750	FBA	2490	
1760	INT	2500	GET NEXT CHARACTER AND
1770	LEA (254).Y	2510	JUMP TO REST OF SUB-ROUTINE
1780	STA #254	2520	
1790	PLA	2530	JEE #0A00T
1800	STA #254	2540	STO
1810	JMP SEARCHLFE	2550	
1820		2560	
1830		2570	
1840	ALL DONE - NEXT CHARACTER	2580	NORMAL1 JMP #ATE
1850	(PRINTED AND START PROGRAM		
1860			
1870	EXIT JEE #A00E		
1880	JMP #ATEA		
1890			
1900			
1910			
1920	STORE LABEL IN VARIABLE AREA:		
1930			
1940	POINT TO BEGINNING OF LABEL		
1950			
1960	LABELFOUND LEA #254		
1970	STA #7A		
1980	LEA #254		
1990	STA #7B		
2000	CLC		
2010	LEA #7A		
2020	ASC #0		

Listing 2

```

110 BEGINNING
120 DRG=(INT:RND11)*100
130 PRINT " DRG="; DRG; " "
140 PAPER=INT:RND(11)*100
150 PRINT TAB(15); "PAPER="; PAPER; " "
160 DRD=(INT:RND(11)*10)
170 PRINT TAB(10); "ROUNDER="; DRD
180 COLOR INT:PAPER/DRD
190 COLOR DELAY
200 GOTO UNDERDRD
210 DELAY
220 FOR #=1 TO 100
230 NEXT
240 RETURN

```

Tech Troubles

*A selection of the problems solved and readers' hints
from this month's mailbag*

By Andy Andros

Why do computers use integers as well as several decimal values, and where do floating point numbers come into it?

David Kennedy, Eastgripold

Dear David,
Another term for an integer is a 'whole number' or, as you point out, a number without any decimal point. Other numbers are referred to as 'floating point' numbers because of the way in which the mathematics is performed in the computer's memory. Floating point calculations are difficult to explain in the space allowed here but, luckily, the fact that they do work is all that you really need to know for Basic.

The reason that computers use integers is because it helps to speed-up arithmetic, and is more accurate than floating points when a certain amount of rounding up and down of values occurs. The speed benefit is gained because there is no need to calculate the position of the decimal place, or to check if the value should be expressed in exponential form, when an extremely large or small number is expressed by a value, and the power to which it has to be raised to reveal the actual number.

A popular myth is that integers use less memory than floating point values, but this is not entirely true. Integers use four bytes - two for the variable name, and two for the actual number. Floating point numbers similarly use two bytes for the variable name, but five more bytes are needed for the value. So, on the face of it,



integers do need less space, but the integer value is followed by three unused bytes, which means that both types of variable physically occupy seven bytes of memory.

I thought the *Your Commodore Serious Users Guide 1988*, and found the 64 tips for the 64 quite informative. One of the tips is how to simulate a PRINT AF command without using rows of cursor down and left. I thought that you may be interested in a technique that I've been using for the past few years:

```
10 PRINT "A", "B", "PRINT HELLO"
20 PRINT "C", "D"
30 PRINT "HELLO"
```

Execution 20 is the vertical position of the cursor on the screen (value 1 in 24). The PRINT command in line 10 is used to update this new row in the Commodore's memory, and a cursor up to replace the cursor onto the required line. Next the cursor is placed horizontally by pointing a value in execution 20 and the message can then be printed.

Two final words on this machine relate to printing on the top and bottom lines. For using the top line, pointing a cursor home with a semicolon will set the row position and then the column value can be pointed into 20. If the last line is used the screen will scroll up unless you follow the message printing command with a semicolon.

I hope this is of use to someone, somewhere.
Carol Seidman, Wigan

Dear Carol,

Thanks very much for that tip. I found it very informative, and if anyone else has a technical tip I'd be very pleased to hear about it.

I m a listing I came across the following syntax which I don't understand

```
IF C=RESTORE THEN GOTO 100 ELSE 200
```

Could you explain what it achieves?
Clive Fowler, Midhurst

Dear Clive,

This is an unusual but useful application of a comparison command. It is the sort of routine which would convert a decimal value to its hexadecimal equivalent.

To understand how it works, I'll have to explain the principle behind the greater than and less than commands. When a comparison is made, the result is a zero if the condition is false and a minus one if the condition is true.

In the example that you've supplied, the value given by the comparison would be zero if X had a value between zero and nine but minus one if the value exceeds nine. What the equation is doing is taking the value in A and adding 48 to give a value whose CHR\$ equivalent is the number character which corresponds to X. This works well up to a value of nine, but values 10 to 15 would produce punctuation marks and symbols because there is a gap of seven characters between the CHR\$ value for nine and the CHR\$ value for the letter A. The program has to be able to add seven to values over 10, but not to values below that.

The solution is to compare the value with nine, and then multiply this by seven. This would mean that the low values would subtract two from the CHR\$ value but high values would subtract $(-1) \times 7$ multiplied by 7. Subtracting a negative value follows the mathematical law that states that two minuses gives a plus, so the program actually adds seven to the CHR\$ value.

To test this try setting up a loop for A having values of zero to 15. Place the equation inside the loop and a statement to print each HEX\$ value, and the result will be a list of hex equivalents from 0 to F.

I have written a program which uses several groups of DATA statements but the only problem with it is that RESTORE can't be set to point to a particular line. In there any way around this, because at the moment I have to RESTORE and then read

in DATA until the correct point is reached?
Harry Ivy, Hampton

Dear Harry,

There are two ways to tackle your problem. The first is to read the data into several arrays and manipulate these instead of using data READs.

The second method is to create a special form of the RESTORE command. This is done by loading in the data until the line where the program is to be restored to. The actual memory location can be saved by pointing locations 63 and 64 and storing this in a table in a safe part of memory.

Each suitable point is similarly stored until the table is complete. When the data reading has to be restarted to any of these points, they can be pointed from their storage point and pointed back to 63 and 64.

Can you tell me where I can find a good book containing a breakdown of the C128 Basic ROMs? I have a hunch that such a book may not exist, because I have a book which gives a breakdown of the Kernal, but which states that the Basic ROMs themselves would fill a very large volume.
Richard Terry, York

Dear Richard,

You'll be pleased to hear that such a book does actually exist, though you are right in assuming that it is a weighty tome. C128 Basic 7.0 Breakdown, published by Abrams, runs to over 600 pages. Of this, 450 pages contain an annotated ROM disassembly, and the rest is packed with useful information about the workings of the system, alongside some useful programming hints and routines. The price is £18.95, and it is available from Precision Software, 6 Park Terrace, Worcester Park, Surrey KT4 1JZ, but please include £1.49 for post and packing.

If you have a problem let us know and Andy will try to help. Write to: Tech Trouble, Your Commodore, Argus House, Boundary Way, Hemel Hempstead, Herts HP1 1ST.

THE EPSON SQ-2500

The first major task is getting the SQ-2500 out of the box - ideally, it's a job for two people. The SQ-2500 might be expensive, but it has a wide carriage and you certainly get a lot of printer for your money. Styling is recognizably Epson, but the greatest impression from the first is that this printer wastes quality. Construction is superb. I didn't actually try it, but I'm sure I could jump on the SQ-2500 without damaging it (and I weigh over 16 stones).

Setting it up

This is an ink-jet printer so there's no ribbon, just an ink cartridge, paper feed knobs, the mains lead and paper guide. Open a cover on the right side towards the rear of the printer, push in the ink cartridge, close the cover and that's done. On the left side is a similar cover, but unless you want to use either of the possible two extra font cartridges, you needn't open it.

The interface and power lead connect at the rear, as usual for Epson, and the power switch is on the right side. The paper guide easily slots into position, and when an new ink cartridge is fitted, the SQ needs priming (this is the long cleaning cycle mentioned later, but it just means switch on and press a button). Allow about half a minute if's ready to go.

Both parallel and a serial interface are standard, with provision for a third one, and any of the extra Epson interfaces (listed in the LQ-580 report) can be used too. The standard buffer is 8K. The manual is 178 pages, in contrast to the LQ's 258, but contains all the necessary ingredients, including both types of command summary and a quick reference card.

If you can afford an SQ, buy one! If you do, don't waste time looking for dip-switches - there aren't any! Four buttons (and one of those is an off line) control EVERYTHING! This explains the short manual. There's very little on trouble-shooting.

To conclude last month's feature on the world of 24 pin dot matrix printers,

Robin Burton admires the highly impressive Epson SQ-2500

The only way I can envisage anyone having trouble with the SQ is if they can't read, in which case the manual wouldn't help much.

I'm not being flippant! It's all set-up, and manual selection is by question and answer, but with a difference. There's a liquid crystal display at the front of the printer next to the selection buttons, and the SQ asks the questions. You answer by simply pressing a button, and the settings are stored. This is a dream machine.

Superb though this is, there's more! The SQ also has four macros. Each of these is a complete, permanently stored definition containing everything you could want the printer to know about a job. Any one of the four can be loaded automatically at power-up; you choose which one you want in the configuration defaults, which remain permanently set unless you alter them.

You can manually load any macro by a couple of button presses, and can also amend them manually, by software or both, either temporarily, or permanently. Simply re-save a macro at any time (by pressing a button) if you want the changes to be permanent. If not, they're forgotten, either when you switch off or when you load a different macro. You might also want to print out the settings, in which case - press a button.

This is all so comprehensive yet simple that describing it is difficult. I've therefore included a print of my configuration settings separately, so you can see for yourself. Everything is stored - left and right margins, font, size, pitch, page size, etc. etc.

As standard, the SQ-2500 is a cut sheet machine, but an optional tractor unit can be added for continuous paper. You may have gathered by now that the SQ-2500 is intended for high quality, high speed, high volume output. The fact that the tractor is top-mounted and therefore without paper parking is, in context, irrelevant.

Don't misunderstand - swapping between continuous and sheet-feed is easy enough, but if you need to do it very often you don't need an SQ-2500. It wasn't built for the average home user's mixed low-volume needs, and used that way would be like doing the grocery shopping in a Formula One car, theoretically possible but...

If cut sheet is the major use a double-bin automatic sheet feeder can also be attached.

Specification

The SQ-2500 is of course above all an Epson, so the compatibility comments for the LQ-580 all apply to the SQ, including the 13 national character sets.

Six fonts are standard for the SQ-2500, such as usual with Epson. Additional effects are limited to double width, double height or both. All fonts are available in the usual 10, 12 and 15 pts, and all can be condensed except 15 pt.

There's also provision for two font modules, but so far as I could see all the Epson LQ fonts are standard in the SQ except OCR-B. Minimum vertical spacing is 180ths of an inch, and horizontal is 360ths.

Using the Epson

When the SQ powers-up it automatically goes through a self-cleaning cycle. The printer informs you (via the LCD) when the ink cartridge is getting low, and when it's exhausted. According to the manual, a cartridge lasts for 3,000,000 characters in LQ, and 6,000,000 in draft. This seems fair enough - I've gone through well over 1,200 sheets, largely in LQ, (roughly 6,000,000 characters) and not a word so far.

Using semi-automatic sheet feed is simple and quick, and it has been absolutely reliable with all weights of paper. Just drop a sheet into the guide, press 'form feed' and the sheet is lined up perfectly (I even tried it with a 9 x 4 inch envelope - notoriously difficult to keep straight - with no trouble at all).

If the tractor is fitted, it simply attaches on top of the SQ - there are no covers to change. Usefully, along with the tractor, Epson provide a matching base on which the printer sits. Continuous paper is kept inside this, out of the way. A paper rest is also included to keep the paper clear of the leads.

Manual line feed, form feed, self-test and hex, dump are controlled by the buttons, and the self test includes the current configuration and macro settings too. As mentioned before, just the configuration and the four macros can also be listed on demand (or alternatively checked directly via the LCD). Without altering any of the current settings you also can switch between draft or LQ at any time by pressing either the line-feed or the form-feed buttons.

Print quality is frankly so superior to that of an impact dot-matrix printer that a comparison is pointless. Because there are no wires passing through a fabric ribbon, the individual dots are much smaller and more precise. I've

even asked when I bought a laser-printer by people who didn't know I had the SQ. It might not be quite that good, but it's obviously near enough, and you'd have to put the two side by side to notice a difference. Also unlike any impact printer, the quality never varies. There's no deterioration or fading of print quality, because there's no ribbon to wear out. Characters are always perfect, and uniformly sooty black.

The SQ is the fastest matrix printer I've ever seen. The figures don't adequately tell the story - you have to see it to appreciate just how quick it is. In fact its speed was at one point an inconvenience. With such dense character images, double strike is almost unnecessary. Of course I tried it, but output was quick enough to allow the paper to re-fold in the collection basket before the ink had fully dried. I have now lowered the basket.

The final difference about the SQ is noise, specifically the absence of it - only ink touches the paper. You don't realize just how much you are normally conditioned to brace yourself for the onslaught of noise from an impact printer, at least not until you start printing and it doesn't happen. It's rather uncanny at first, and after years of impact printers, it took me a week to get used to this. Needless to say the SQ easily passes the telephone test, in fact if you were surrounded by them it would be no problem.

Quite aside from its obvious desirability, this aspect of the SQ's performance (along with the others) offers very serious benefits for offices where several printers are at work. I would think the amount of noise from the SQ (just the carriage moving back and forth) would be acceptable even in places like libraries and hospitals too.

The SQ-2500 has operated perfectly throughout.

Conclusion

It hardly needs saying that the SQ-2500 is Epson's top-of-the-range conventional dot matrix printer, and it has been for a few years now. It is built like the Ford Bronco, and has by far the easiest manual control of all operational features of any printer I know. Macros are excellent, and allow an automated switch between four completely different setup definitions in literally a couple of seconds.

Of course the recent arrival of low-cost laser printers must have eroded SQ sales, but there are still plenty of jobs that laser printers don't do very well or at all, (like printing A3 pages sideways), and their running costs are also higher than the SQ's. One laser printer limitation however is shared by the SQ-2500. Because it's not an impact device, multi-part sets are impossible, though with its speed you'd probably just print extra copies as needed.

In its target market, I doubt that the SQ-2500 has any competition. Quality is beyond question in construction, operation and output. It must be the latest, quickest way of getting high quality print onto paper, with all the flexibility of a conventional printer and a virtual absence of operator skill or training.

The recommended retail price is £1,349. Options include a tractor unit at £90 and a double-bin automatic sheet feeder at £170. (£1,519, £5, £100.50 and £425.50 including VAT.) Ink cartridges are about £24, and have a shelf life of two years.

Checking current advertisements I found that the SQ-2500 can be purchased for around £975 plus VAT, with proportional reductions in the optional fittings.

TABULATION FOR YOUR COMMODORE/MATX PRINTERS:-

	Dimensions WxDxH	Time/Speed to print 5000 chars	Weight
Star LC14-88	17" x 19" x 4.5"	56 sec/38 cps Draft	14.2 Lbs
Epson EQ-900	18.5" x 17" x 8"	52 sec/34 cps Letter	15.5 Lbs
Citrus HQP-48	24" x 17" x 4.75"	49 sec/111 cps	16.5 Lbs
Epson SQ-2500	23.5 x 15 x 8	32 sec/156 cps	25 Lbs

Sketchpad 128



Gordon Davis examines a rare specimen – an 80-column graphics package for C128 owners

Skechpad 128 is a rare animal. Almost all Graphics packages for the C1280 operate in 80-column high-resolution mode, but this one is different – it offers a full graphics capability in 80-column mode – a resolution of 640 pixels horizontally.

The package has been made possible by the release of Walnut's Basic 8, a language package which we reviewed last year. Basic 8 is designed as a language which offers a graphics-oriented environment to the systems designer. It's no surprise, therefore, that the salient features of Sketchpad are exactly those which Basic 8 supports most readily.

A Commodore 128i mouse is a necessity for Sketchpad. This, I think,

is unfortunate, since it isn't the sort of item that Commodore owners are likely to just have hanging around. Although, as the authors point out, a mouse is by far the best device for doing graphics, it would have been nice to allow a joystick to be interfaced with the program, just to get people started.

Little messing

Fortunately, we managed to find the office's 125i mouse, so that play could be done. There's very little messing around following loading. Sketchpad moves straight into the drawing screen following the obligatory Basic 8 loading screen.

Free Spirit, which, as you might guess, is a California software house, has obviously done its Apple Macintosh homework. The main screen looks very Mac-like. All mouse selections are made from the drawing screen by clicking the left mouse button. The right button is reasonably consistent in returning the user to the main menu.

The main elements on the main menu are exactly those which have been made familiar by that animal program, MacPaint. Naturally, you can draw firsthand on the screen with a variety of different pens. The indispensable airbrush is also there. The annoying thing about this is that the airbrush area cannot be sized, which is something that you can see even with C64 packages.

Sketchpad supports a suitably fast fill. This is a little worrying in the way it operates, since it seems to leave many lines unfilled along the way. But never fear, the filling algorithm gets back and fills them all back in too.

A multi-font option for writing is also available. Different fonts can be loaded from disk, and can be typed anywhere on the screen in one of five different sizes. A useful touch is that Return will align the text cursor exactly one line below the text you've already put in.

This is not a multi-colour graphics program, quite rightly, because unless you are designing loading screens or demos, anything other than two-colour screens is functionally useless

for all owners of black-and-white printers. Even so, you can store the foreground/background to give you more screen real-estate.

Although clearing the screen is fairly dandy on Sketchpad - there is no check on this, you can lock your program into memory, which will put it into a spare area of Ram for safety. This is also useful if you've ascertain about the changes you're making to a picture.

The graphics shapes options are the ones that show the most difference over most graphics packages. All the traditional favourites are there: lines, boxes, circles, and so on. Preserving these is more or less as straightforward as the commands on any other graphics package. A unique extra, however, is the addition of the so-called Cylinder shapes.

Solid graphics

Cylinder shapes are views of shaded solids of three types: cylinder, sphere and torus. They are named after the programmer who designed a set of

algorithms for representing them easily.

Basic 8 later incorporated these calculations, and here they are at last in a graphics package. The shading on the Cylinder shapes can be varied between pixel dots or smooth shading, and whatever you use them for, they are an interesting addition.

Besides these shapes, another useful feature is the inclusion of arcs. You can provide arcs of any pitch and angle. Rays can also be provided, subtended at any angle from a central point. Both of these make the production of piecharts an easy matter.

One vital thing I expect ages looking for was some form of Zoom option. It's very unusual using a graphics package, and some means of adjusting things on a pixel-by-pixel basis can be useful. This is, annoyingly, not available from the main screen, but from a sub-menu called the Gadgets menu.

Besides a pixel menu, Gadgets also supports a Cut/Paste option which allows you to move chunks of the picture around, or simply remove them. You can also scroll your picture - an interesting feature which few

packages support. This menu also supports a number of patterns which can be used to fill areas. Unfortunately, you have to draw the clip patterns in yourself - there is no automatic fill for these.

Five Spirit deserves complimenting for the number of features which have been included in Sketchpad - I haven't covered everything here. The main flaw in the program is that it is a trifle slow, due to being written entirely in Basic 8.

I'm also not sure about some of the design priorities - some of the essential are very hard to get to. The program documentation is brief and unhelpful. These days, this is something I expect software houses to get right. On the other hand, C128 fill-column drawing packages are fairly hard to come by, so it seems unfair to criticise too much.

Footnote:

File Sketchpad 128, Supplier: PESS/ Five Spirit, 28 Wight St., Redditch, Worcestershire, W10 3JG. Tel: (0582) 531132. Price: £24.95.

DON'T GET IN A TANGLE...

take out a subscription

Make sure of your regular copy of *Micro Music*, with all the news, views and reviews from the computer music scene, delivered to your door at no extra cost in the UK. Just fill in the form, enclose your remittance and you're in touch.

4 Issues	UK	£10.50
4 Issues	Europe	£14.00
4 Issues	Mainland East	£14.15
4 Issues	Far East	£18.00
4 Issues	Rest of the World	£18.00
4 Issues	USA	\$25.00

Air Mail rates given on request. Overseas subscriptions include postage.

MICRO
Music

Please complete and return this form with your remittance to:
NAME (Mr/Ms/Ms/Ms)
ADDRESS

Signature
Send your remittance to:
P.O. BOX 1, 5 River Park Lane,
Barnet, Herts. EN4 3JL.
MMP/2



Routine Programming

*A bubbly routine
to sort lists into
orderly sequences*

By Eric Doyle

Programs often include lists of numbers or strings which have to be displayed in an orderly way. This could be alphabetically or numerically. This bubble sort subroutine can be used, suitably modified, for either purpose.

Bubble sorts work by comparing neighboring list items and swapping them over if one exceeds the other. Take this list as an example:



The first number is compared to the rest of the numbers in the list one at a time. If the number under comparison is smaller than the first number, a straight swap takes place. In this way the smallest value rises to the top.



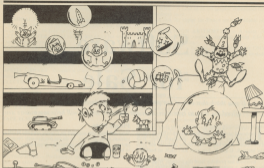
After this process is completed, the second number is compared to all of the following figures in a similar way.



This is repeated, gradually passing down the columns until the last two characters are compared and swapped if necessary, and the sort is then complete.



The two elements of the sort are a loop which selects the character to be tested, and a loop within this, a nested loop, which selects the remaining characters in the list for comparison.



```
60200 FORN1=1000-1
60210 Z=Z+1
60220 FORN2=Z*1000
```

The value Z is the number of items in the list. Line 60200 forms the main loop for the character to be tested. It only runs from the first to the master predicate character because there would be no point in comparing the final character with itself.

Line 60210 calculates the first character to be tested by the nested loop which is initiated in line 60230. Line 60220 tests the two values, and if they are already in the correct order no action is taken and the exchange routine is bypassed.

```
60230 IF(Z<N1+1)GOTO(7)60200
```

If an exchange is necessary, this is done by using a temporary store for one of the values, so that values can be swapped with ease.

```
60240 Z(Z)=Z(N1)
60250 Z(N1)=Z(Z)
60260 Z(Z)=Z(Z)
```

Z(N) takes the value held in Z(Z) and Z(Z) can then be given the value held in Z(N). The stored value can then be moved from Z(N) to Z(Z), and Z(N) may then be discarded.

Now that everything has been done, the loops can be closed and

eventually control is handed back to the program with a RETURN command.

```
60270 NEXTZ,Z1
60280 RETURN
```

The listings shown contain an example of numerical sorting using this sub-routine and there is also an alphabetical sort routine which shows how this routine has to be modified for this use. Really all that has happened is that the Z(N) values are changed to Z(N).

Parameters for Main Program

Setting:

- Z: Number of list items
- Z(N): List item values

Returns:

- Z(N): The sorted list

Other Variables Used

- Z1: Main loop variable
- Z2: Remainder of list items
- Z3: Nested loop variable

PROGRAM: NUMBER SORT	
01 10 0-100-50 0-1000000.000000	01
02 20 0-1000	02
03 30 1000-1000-1000-1	03
04 40 1000-1000-1000-1	04
05 50 1000-1000-1000-1	05
06 60 1000-1000-1000-1	06
07 70 1000-1000-1000-1	07
08 80 1000-1000-1000-1	08
09 90 1000-1000-1000-1	09
10 100 1000-1000-1000-1	10
11 110 1000-1000-1000-1	11
12 120 1000-1000-1000-1	12
13 130 1000-1000-1000-1	13
14 140 1000-1000-1000-1	14
15 150 1000-1000-1000-1	15
16 160 1000-1000-1000-1	16
17 170 1000-1000-1000-1	17
18 180 1000-1000-1000-1	18
19 190 1000-1000-1000-1	19
20 200 1000-1000-1000-1	20
21 210 1000-1000-1000-1	21
22 220 1000-1000-1000-1	22
23 230 1000-1000-1000-1	23
24 240 1000-1000-1000-1	24
25 250 1000-1000-1000-1	25
26 260 1000-1000-1000-1	26
27 270 1000-1000-1000-1	27
28 280 1000-1000-1000-1	28
29 290 1000-1000-1000-1	29
30 300 1000-1000-1000-1	30

31 310 FORN1=Z*1000	31
32 320 Z=Z+1	32
33 330 FORN2=Z*1000-1	33
34 340 Z=Z+1	34
35 350 FORN3=Z*1000	35
36 360 FORN4=Z*1000-1	36
37 370 FORN5=Z*1000	37
38 380 FORN6=Z*1000-1	38
39 390 FORN7=Z*1000	39
40 400 FORN8=Z*1000-1	40
41 410 FORN9=Z*1000	41
42 420 FORN10=Z*1000-1	42
43 430 FORN11=Z*1000	43
44 440 FORN12=Z*1000-1	44
45 450 FORN13=Z*1000	45
46 460 FORN14=Z*1000-1	46
47 470 FORN15=Z*1000	47
48 480 FORN16=Z*1000-1	48
49 490 FORN17=Z*1000	49
50 500 FORN18=Z*1000-1	50
51 510 FORN19=Z*1000	51
52 520 FORN20=Z*1000-1	52
53 530 FORN21=Z*1000	53
54 540 FORN22=Z*1000-1	54
55 550 FORN23=Z*1000	55
56 560 FORN24=Z*1000-1	56
57 570 FORN25=Z*1000	57
58 580 FORN26=Z*1000-1	58
59 590 FORN27=Z*1000	59
60 600 FORN28=Z*1000-1	60
61 610 FORN29=Z*1000	61
62 620 FORN30=Z*1000-1	62
63 630 FORN31=Z*1000	63
64 640 FORN32=Z*1000-1	64
65 650 FORN33=Z*1000	65
66 660 FORN34=Z*1000-1	66
67 670 FORN35=Z*1000	67
68 680 FORN36=Z*1000-1	68
69 690 FORN37=Z*1000	69
70 700 FORN38=Z*1000-1	70
71 710 FORN39=Z*1000	71
72 720 FORN40=Z*1000-1	72
73 730 FORN41=Z*1000	73
74 740 FORN42=Z*1000-1	74
75 750 FORN43=Z*1000	75
76 760 FORN44=Z*1000-1	76
77 770 FORN45=Z*1000	77
78 780 FORN46=Z*1000-1	78
79 790 FORN47=Z*1000	79
80 800 FORN48=Z*1000-1	80
81 810 FORN49=Z*1000	81
82 820 FORN50=Z*1000-1	82
83 830 FORN51=Z*1000	83
84 840 FORN52=Z*1000-1	84
85 850 FORN53=Z*1000	85
86 860 FORN54=Z*1000-1	86
87 870 FORN55=Z*1000	87
88 880 FORN56=Z*1000-1	88
89 890 FORN57=Z*1000	89
90 900 FORN58=Z*1000-1	90
91 910 FORN59=Z*1000	91
92 920 FORN60=Z*1000-1	92
93 930 FORN61=Z*1000	93
94 940 FORN62=Z*1000-1	94
95 950 FORN63=Z*1000	95
96 960 FORN64=Z*1000-1	96
97 970 FORN65=Z*1000	97
98 980 FORN66=Z*1000-1	98
99 990 FORN67=Z*1000	99
100 1000 FORN68=Z*1000-1	100

PROGRAM: NAME SORT	
01 10 0-100-50 0-1000000.000000	01
02 20 0-1000	02
03 30 1000-1000-1000-1	03
04 40 1000-1000-1000-1	04
05 50 1000-1000-1000-1	05
06 60 1000-1000-1000-1	06
07 70 1000-1000-1000-1	07
08 80 1000-1000-1000-1	08
09 90 1000-1000-1000-1	09
10 100 1000-1000-1000-1	10
11 110 1000-1000-1000-1	11
12 120 1000-1000-1000-1	12
13 130 1000-1000-1000-1	13
14 140 1000-1000-1000-1	14
15 150 1000-1000-1000-1	15
16 160 1000-1000-1000-1	16
17 170 1000-1000-1000-1	17
18 180 1000-1000-1000-1	18
19 190 1000-1000-1000-1	19
20 200 1000-1000-1000-1	20
21 210 1000-1000-1000-1	21
22 220 1000-1000-1000-1	22
23 230 1000-1000-1000-1	23
24 240 1000-1000-1000-1	24
25 250 1000-1000-1000-1	25
26 260 1000-1000-1000-1	26
27 270 1000-1000-1000-1	27
28 280 1000-1000-1000-1	28
29 290 1000-1000-1000-1	29
30 300 1000-1000-1000-1	30

31 310 FORN1=Z*1000	31
32 320 Z=Z+1	32
33 330 FORN2=Z*1000-1	33
34 340 Z=Z+1	34
35 350 FORN3=Z*1000	35
36 360 FORN4=Z*1000-1	36
37 370 FORN5=Z*1000	37
38 380 FORN6=Z*1000-1	38
39 390 FORN7=Z*1000	39
40 400 FORN8=Z*1000-1	40
41 410 FORN9=Z*1000	41
42 420 FORN10=Z*1000-1	42
43 430 FORN11=Z*1000	43
44 440 FORN12=Z*1000-1	44
45 450 FORN13=Z*1000	45
46 460 FORN14=Z*1000-1	46
47 470 FORN15=Z*1000	47
48 480 FORN16=Z*1000-1	48
49 490 FORN17=Z*1000	49
50 500 FORN18=Z*1000-1	50
51 510 FORN19=Z*1000	51
52 520 FORN20=Z*1000-1	52
53 530 FORN21=Z*1000	53
54 540 FORN22=Z*1000-1	54
55 550 FORN23=Z*1000	55
56 560 FORN24=Z*1000-1	56
57 570 FORN25=Z*1000	57
58 580 FORN26=Z*1000-1	58
59 590 FORN27=Z*1000	59
60 600 FORN28=Z*1000-1	60
61 610 FORN29=Z*1000	61
62 620 FORN30=Z*1000-1	62
63 630 FORN31=Z*1000	63
64 640 FORN32=Z*1000-1	64
65 650 FORN33=Z*1000	65
66 660 FORN34=Z*1000-1	66
67 670 FORN35=Z*1000	67
68 680 FORN36=Z*1000-1	68
69 690 FORN37=Z*1000	69
70 700 FORN38=Z*1000-1	70
71 710 FORN39=Z*1000	71
72 720 FORN40=Z*1000-1	72
73 730 FORN41=Z*1000	73
74 740 FORN42=Z*1000-1	74
75 750 FORN43=Z*1000	75
76 760 FORN44=Z*1000-1	76
77 770 FORN45=Z*1000	77
78 780 FORN46=Z*1000-1	78
79 790 FORN47=Z*1000	79
80 800 FORN48=Z*1000-1	80
81 810 FORN49=Z*1000	81
82 820 FORN50=Z*1000-1	82
83 830 FORN51=Z*1000	83
84 840 FORN52=Z*1000-1	84
85 850 FORN53=Z*1000	85
86 860 FORN54=Z*1000-1	86
87 870 FORN55=Z*1000	87
88 880 FORN56=Z*1000-1	88
89 890 FORN57=Z*1000	89
90 900 FORN58=Z*1000-1	90
91 910 FORN59=Z*1000	91
92 920 FORN60=Z*1000-1	92
93 930 FORN61=Z*1000	93
94 940 FORN62=Z*1000-1	94
95 950 FORN63=Z*1000	95
96 960 FORN64=Z*1000-1	96
97 970 FORN65=Z*1000	97
98 980 FORN66=Z*1000-1	98
99 990 FORN67=Z*1000	99
100 1000 FORN68=Z*1000-1	100

Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from *Four Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little bug slips through unnoticed.

The *Four Commodore Software Service* makes available all of the programs from each issue on both cassette and disk at a price of \$6.00 for disk and \$4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFUNET LTD, 3 River Park Estate, Betchingham, Herts,
HP4 3HL
Tel: (04427) 7660

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied in totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strap containing the article type, C64 Program etc. So that you can see which programs are available in which format, you will also find a couple of symbols after this strap. The symbols have the following meaning:



This symbol means that the program is available on cassette.



These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the programs, we do not just disk-only programs on tape. There is no sense in placing a program that expects to be loading from disk on to tape.

JANUARY 1989

PREPARE SPRITES - A powerful sprite editor for the C64. **HAZPRO** - A simple but helpful text processor for the C64. Available on disk and cassette but will only save files on tape.

UDC COMPRESSOR - Save on memory when using UDC's in your programs. For C64 only.

WILLIAM TELL - Our popular arcade game for the C64. **4-1 AUTOBURN** - Improve tape loading on your Plus/4 cassette. Only available on disk.

MINIBASE - A database for C128 owners.

ORDER CODE
DISK YJAN89 06.00
TAPE YJAN89 04.00

FEBRUARY 1989

TAPE MENU - Add a menu system to your program cassette (C64).

SONIC EFFECTS - A superb sound editor for the C64.

F DUMP - Dump your C64 text sources to printer with ease.

DATA LOADER - A simple way to enter those teams of C64 DATA files.

SPRITE LIBRARY - A collection of birds to your growing library (C64).

PLAY THE GAME - A superb fruit machine program for the Plus/4. (Available on disk only).

ORDER CODE
DISK YFEB89 06.00
TAPE YFEB89 04.00

MARCH 1989

PERSONAL FILE - A cross between a wordprocessor and a database that allows you to set up "cards" that can be quickly altered (C64 Disk only).

LETTER WRITER - An 80 column text editor for writing those personal letters (C64).

BASIC WORKSHOP - A single key entry system, just like a Spry! (C64).

HEAD FOR HOME - Our version of a popular board game for C16 and Plus/4 owners - available on disk only.

SPRITE LIBRARY - Geometric shapes from this month's instalment (C64).

ELECTRONIC NOTEBOOK - A personal diary on disk (C64 disk only).

WILLIAM TELL - Our very popular arcade adventure for the C64.

ORDER CODE
DISK YDMAR89 £6.00
TAPE YCMAR89 £4.00

APRIL 1989

BASEX - Give your C64 new sound, graphics and control commands as well as a machine code assembler with this Basic extension.

AUTOSCROLL - Professional text screen scrolling with this C64 utility.

BALANCE SHEET - Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).

HDN 64 - Add icon control to your own programs with this C64 utility.

88 COLUMNS - A suite of 88-col graphic commands for C128 80 column users.

RELS - A disk file copier for Plus/4 users. Available on disk only.

MULTI-PRECISION ARITHMETIC - A suite of powerful mathematical routines expanding your C64's number crunching abilities.

STORAGE SPACE - Use the RAM behind the C64's Kernel and Basic ROMs for data storage (available on disk only).

ORDER CODE
DISK YDAPR89 £6.00
TAPE YCAPR89 £4.00

MAY 1989

ANTI-FREEZE - Protect your C64 programs from cartridge-based thieves.

MULTI-COLOURED LIST - Brighten up your C64 Basic listings by adding on-screen colour.

PRINTX - A suite of powerful screen printing commands to add to your Plus/4's Basic (disk only).

BANKER 128 - Money Management for C128 disk users.

TURBO TOUCH - A superb typing trainer for C64 disk users.

ORDER CODE
DISK YDMAY89 £6.00
TAPE YCMAY89 £4.00

Cassettes or disks are available from March 1986. Please ring the editorial office (01-477 0626) for details of these.

ORDER FORM - PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
JUNE 89		TAPE (£4.00)	YCJUNE 89	
JUNE 89		DISK (£6.00)	YDJUNE89	
OVERSEAS POST E				
				TOTAL

NAME.....
 ADDRESS.....
 POSTCODE.....

I enclose a cheque/postal order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.
 All orders should be sent to: PCW COMMODORE READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP1 7ST
 Please allow 28 days for delivery.

Repairs Guide

0442
86850

CROYDON COMPUTER CENTRE

111 Bognor Road, Bognor Regis, Sussex, BN8 7JA Tel: 01 303 3348

COMPUTER SERVICING

1974 price 1982
We repair — all IBM computers — quick turnaround
• Commodore
• BBC & BBCw (Approved Service Centre)
• Amstrad & Tronix (Approved Service Centre)
• Disc Drives, Printers, Monitors
Mail Orders by phone. Access 8 Year warranty

COMMODORE REPAIRS

(1980-1982) estimated guarantees

COMMODORE 40 £26.75
COMMODORE 128 £26.75
COMMODORE 640 £26.75
C16-4/810 30 £26.75

Price is all inclusive 100% we guarantee the whole computer for 12 months

IBM SERVICE

24, Pelican Common, Springfield, Chislehurst, Essex CM1 2EP
Tel: 0245 - 888342

COMMODORE SPARES & REPAIRS

REPAIRS OFFERS: C16 to C128 44
1041 £26.00 1042 to 1047 44
These include warranty, value guaranteed

MODEL	PRICE	MODEL	PRICE
1041	£26.00	1042	£26.00
1043	£26.00	1044	£26.00
1045	£26.00	1046	£26.00
1047	£26.00	1048	£26.00

SALES COMPUTER SERVICES
Dept 22, 18 Kingsway Road, Watlington, Wiltshire, SN4 6SL, Tel: 0300, 648817 ext 104

"COMPUTER KAPUT"

WITH OUR
Specialists
THE EXPERTS IN REPAIRING
IBM, TRONIX, COMMODORE
AND ALL OTHER
COMPUTERS

PER YICOR
PLUS-CR
CR-CR
SERVO...
FAST
EFFICIENT
SERVICE

FOR REPAIRS CALL
01-801 2850
WILMINGTON NUMBER

VTS COMPUTER REPAIRS

01 - 407 8820

Simply send your unit to the address below securely packed with fault description and

COMMODORE 40 £26.00 (C16, C128, C40, C64, C80)
12000 £26.00

VTS COMPUTER REPAIRS

12 Lily House, 61 Swaney Street, London, SE7 4JG
FAST RELIABLE TURNAROUND

PROBLEMS WITH YOUR COMMODORE ?
FOR FAST, RELIABLE AND PROFESSIONAL
REPAIRS AT COMPETITIVE PRICES

P M ENGINEERING



UNIT 280, MILTON SCIENCE
PARK, CAMBRIDGE CB4 4WE

ST. IVES (0223) 420007



We can also supply Hardware, Software, Disk Drives,
Accessories and Spare parts.
C/M Repairs £25.00 inc. VAT (only PMEs)

REPAIRS

AMIGA ATARI etc.
MONITORS & ALL ADD ON'S
AMIGA UPGRADES AND MOD'S

All estimates FREE

Prices from only £19.50

Call or post FAST TURNAROUND

PVS ELECTRONIC COMPONENTS

244 Deansgate Court,

Deansgate, Manchester M3 4NQ

Tel: 061 831 7056 Telex: 868898

Fax: 061 832 6374



Computer Repairs

Fixed Super Low Prices!

Inhouse of parts, labour and VAT



FAST AND
FOR
TECHNOLOGY

1 week turnaround! Call for delivery available for local area

* SUPER OFFERS *

SPECTRUM	£74 inc. - Free Glass	ELECTRON	£20
104	£22 inc. - Free Glass	COMMODORE 154	£20
128	£26 inc.	1084 DRIVE	£28 inc.
VICAR, C-4	£26 inc.	1084/1085	£28 inc.
880C	£26 inc.	C&R ALL TOP GALS	£28 inc.

Please contact supplier with unit - 1 week turnaround on repair. Electronics subject to stock.

W.T.S. ELECTRONICS

7, C. Shute Market House, Chesham Lane, Luton, LU1 1EC. Tel: 0582 491840

HINDLEY ELECTRONICS

The Computer repair specialists

BASED IN
SURREY
(KINGSLAND)



100
TURNING
(POST 11)

EXPERT COMPUTER REPAIRS

COMMODORE — SPECTRUM — AMSTRAD — ACORN — BBC
STANDARD CHARGES (INC VAT) AND FULLY INSURED RETURN POSTAGE

Commodore 40	£24.00	Commodore 128	£24.00	Commodore 64	£24.00
Commodore 128	£24.00	Commodore 64	£24.00	1084/1085	£24.00
1084/1085	£24.00	1084/1085	£24.00	1084/1085	£24.00
1084/1085	£24.00	1084/1085	£24.00	1084/1085	£24.00

1 MONTH "NO QUibble" WARRANTY ON ALL REPAIRS

100% SATISFACTION GUARANTEE OR FULLY INSURED RETURN POSTAGE

100% SATISFACTION GUARANTEE OR FULLY INSURED RETURN POSTAGE

HINDLEY ELECTRONICS

DEPT A 97 MARKET STREET, HINDLEY, WILKINS RD, LANC. Tel: 0547 82240

Computer repairs include supply of the correct spare parts (where applicable) and 100% satisfaction guarantee. Standard charges for return postage included. All prices include VAT. All prices include postage. All postage charges for return postage included. All prices include VAT. All prices include postage. All postage charges for return postage included.

B A E A H

Readers Problems

Though the *Commodore* is one of the world's most popular microcomputers, it can be very difficult to find specific information about your particular machine.

At the *Four Commodore* office we receive literally hundreds of letters from you, our readers, on a wide range of subjects ranging from the simple "Can you give me the telephone number for...," to the more complex "I'm trying to write a program that uses a split screen. How do I do it?"

Unfortunately, the volume of mail received has become so great that it is impossible to answer every letter and still manage to publish a magazine each month.

For this reason we have felt it necessary to produce a number of guidelines for getting information from us:

- 1) We cannot guarantee to answer every letter sent to the magazine. Should it become apparent that a number of readers are suffering from the same problem, then we will reply to the letter via the Letters page.
- 2) A new helpline has been set up. This will be open for your queries on

Tuesday and Thursday afternoons between 2.00pm and 4.00pm. We will not be able to deal with our telephone queries at any other time. If our technical adviser is not available when you ring, then a message will be taken.

3) If you are having problems with one of our listings, can you please let us know in writing. This will enable us to see if a number of people are having the same problem. When a common problem becomes apparent with a program, then a correction sheet will be issued. Enclose a self-addressed, stamped envelope and we will send you a copy of the correction sheet as soon as it is available.

We are sorry that it has become necessary to mitigate these rules. However, we are sure that you will agree with us that the more time that we can spend making *Four Commodore* the most informative magazine around, the better.

For program queries write to:
Program Corrections
Four Commodore
Angus House,
Boundary Way,
Hemel Hempstead
HP1 1ST
Tel: 0442-66551

CORRECTIONS FOR MARCH '89

READ FOR HOME

Once again those intractable grannies have caused havoc.

The program as it stands will not run correctly. This is due to the fact that quite a bit of code is missing from the loader program. An amendment is on its way. The update will be published as soon as it reaches our office. We apologise for any inconvenience to our Plus 4 readers.

CORRECTIONS FOR APRIL '89

BALANCE SHEET

Unfortunately there appeared a couple of errors in the Balance Sheet program from the April Edition.

- 1) Page 101, right hand column, paragraph 4 should read Load "LIST4" and not Load "LIST1".
- 2) The second line of Listing 1 should be POKET76,131:POKE771,164
- 3) All the REM statements should be taken out of Listing 1.
- 4) Again on page 11, right hand column, paragraph 1. This should read:
% 87557812 @ SHEET".LJ

Commodore Where Are You?

At the *Four Commodore* office we are repeatedly asked for the address and telephone number of Commodore

U.K. Many people, after referring to their computer manuals, believe them to be based in Carby.

The Commodore plant at Carby was closed down some time ago. Reproduced here you will find the correct

address for Commodore U.K.
Commodore Business Machines, (UK),
Commodore House, The Switchback,
Bucks SLS 7TA,
Tel: (0625) 770888
Gardner Road, Maidenhead.

The Nibbles *By Alan Batchelor*



AMIGA

ATARI
ST

IBM PC

APPLE
MAC

MUSIC X-The Review

MICRO MUSIC

Find out about the magazine
the computer music world has been
waiting for

THE SPECIALIST
MUSIC MAGAZINE
FOR COMPUTER USERS

KCS-Just what the
Doctor ordered!!

ACORN

ORIGAMI
64

AMSTRAD
CPC

ZX
SPECTRUM

THE TIME HAS COME . . .

Britain's First Music Magazine for the Computer User!
Featuring Reviews on Hardware & Software across all formats - Micro Music is
the magazine the Computer Musician has been waiting for.
Available 10th March from all good Newsagents

Emlyn Hughes



INTERNATIONAL



**GAME
OF THE YEAR!**

**WINNER OF CCI OSKARS FOR
BEST SPORTS SIMULATION
AND BEST GAME OF THE YEAR!**

*Quite simply the best football
simulation available for the 64.
Nothing short of superb.*

ZZAP 64

**AVAILABLE NOW FOR
COMMODORE 64
£9.95 tape £12.95 disk**
Coming soon for Spectrum & Amstrad

Audiogenic

Audiogenic Software Limited, Winchester House,
Canning Road, Harrow HA3 7SL, England

Order by phone on 01 861 1166

