

YOUR

COMMODORE

AUGUST 1988

£1.20

GEOS — *Being used at last!*

Wordprocessors
Which one's for you?

Expert V3.2
As fast as a parallel
DOS

**Writing
Structured
Programs**

**UNBEATABLE
PROGRAMS:**

Plus/4 SIMPLE
C64/C128 Terminal
Tape Organiser
Educating SID
A Short Interlude



Games Reviewed: Karnov ▲ Patton vs Rommel
Wizard Warz ▲ Infiltrator II ▲ Jinxs

TOTAL BACKUP POWER TAPE Now ACTION REPLAY MEMORY

NOW EVEN MORE POWERFUL, MORE FRIENDLY AND WILL BACKUP

ONLY
£29.99
POST FREE

Action Replay works by taking a 'SNAPSHOT' of the program in memory so it doesn't matter how the program was loaded - from tape or disk - at normal or turbo speed.

• **WARP 25. THE WORLD'S FASTEST SERIAL DISK TURBO - NOW EVEN FASTER! LOADS 200 BLOCIS IN 6 SECONDS! 240 BLOCIS IN 7 SECONDS!** - that's even faster than some parallel systems. Built into the cartridge - no extra hardware or software required. Includes expert test, diagnosis, search, storage recovery. Incorporated with normal disk turbo for supercompatibility - can do normal or turbo format or no special load commands are required. Turbo and WARP 25 speed are entirely independent of the cartridge with WPC8000T.

• **SIMPLE TO USE.** Just press the button and make a complete backup - tape to disk, tape to tape, disk to tape, disk to disk, tape to disk or vice versa... just press the BACKUP & ERASE.

• **TURBO RELEASE.** All backups will release at turbo speed, COMPLETELY INDEPENDENTLY OF THE CARTRIDGE.

• **SPORTS KILLER.** Make yourself invulnerable. Disable sports collisions - make win every program.

• **PRINTER DUMP.** Freeze any game and print out the screen. Eg. leading players, high scores screen etc. Works with most printers. WPS 100, 101, 102, 103, 104, 105 etc. Double size, 10 shades, reverse print option. Very versatile - no user knowledge required.

• **PICTURE SAVE.** Save any three multi-line screens to disk or to the peak of a screen. Compatible with Missing Pictures, Goals, Art or 4x, Image Freeze etc.

• **SPORTS MONITOR.** Freeze multi-screen allows you to freeze the screen and store all the statistics, record the results collection, save or delete any sports. Load options from one game into another to make extended programs.

• **FOURSCREEN MODE.** Press the button and enter three games for extra time etc., then transfer the program or make a backup. Ideal for custom games.

• **MOVIEWAY TRANSFER.** Easy transfers multi-stage programs from tape to disk, the extra game feature - a unique feature. Enhancement disk available for any standard multi-line games (see below).

• **EXTRA COMPACT.** More efficient program compression techniques. Each program saved as a single file. 2 programs per disk side - 4 programs per disk, if you use both sides.

• **TEXT MODIFY.** Change title screens, high score screens etc. Put your own name into a game that needs it or make a backup to tape or disk. Very simple to use.

• **MONITOR.** Full feature 'Flamingo' type MACHINIC CODE MONITOR, all standard features plus many more - assembly, disassembly, line dump, compare, variable compare, 64 bit, number comparison, built-in printing, indirect, indirect etc. Time as memory full printed output.

• **DISK MONITOR.** A special monitor for use on the R&M inside your disk drive. All the usual commands - a useful backup tool.

• **WHOLE DISK COPY.** Copy a full unformatted disk in under five minutes with only one drive.

• **FAST FILE COPY.** Works with standard and WARP 25 files of up to 140 blocks. Converts format to and from WARP 25.

• **FAST FORMAT.** Under 20 seconds.

• **TODAY'S COMMANDS.** A whole range of useful new commands including: SETP-LINE (HYPERLINK), DELETE, MERGE, APPEND, COPY, LINK/UNLINK, etc. FRAMESLASH - list any program (including drivers) already on disk to prevent or screen without corrupting memory.

• **REDEFINED FUNCTION KEYS.** Single-byte commands for operation of many common commands including: LOAD, SAVE, WAP, Load from directory - no need to type in filename.

• **TAPE TESTS.** Designed to make turbo backups for most new programs. No screen blanking during loading.

WHAT THE REVIEWERS SAID
"The reviewer summed up the product as follows: 'This is not really a backup, but it really is the best value for money I've ever seen.' The reviewer gave the Commodore Disk Drive a 5 star rating."

ACTION REPLAY ENHANCEMENT DISK

The biggest and best collection of special parameters and the COPY programs for transferring any standard multi-line tapes to disk - just as the LAST MAN, CALIFORNIA, GAMES, LEADERSBOARD, DRAGON'S LAIR - 800000 titles in all. Almost all major titles covered. Latest edition includes: GUNRAY WINGS, PLAYBOY, FREEDOM, CARTRIDGE, THUNDER, SHAY, ON THE APOLLO, THE TRUMAN and many more. Titles for which time lives etc. The GRAPHIC EDITOR - latest edition displays multi-line programs on loading screens saved by Action Replay in 100 lines. Art Package - Missing Pictures, Goals, Advanced Art Studio, Art or 4x etc. Lots of fun. Only £7.99. Register - send £1.00 plus old disk.

REMEMBER all features are built in and available at the touch of a key. All features work with both TAPE and DISK. (Except multipart transfer & disk file utility).

TAKES A QUANTUM LEAP MK IV HAS ARRIVED!

FOR
CRM64/128

BACKUP MORE PROGRAMS THAN ANY RIVAL UTILITY.

BUT THAT'S NOT ALL ... NOW AVAILABLE FOR THE SERIES PROGRAM HACKER ACTION REPLAY IV 'PROFESSIONAL'

● All the features of the normal Action Replay IV but with an amazing on board LSI LOGIC PROCESSING CHIP. Plus 32K operating system ROM and 8K RAM CHIP. The first RAM-ROM based cartridge of its type!

ALL THE MK IV FEATURES PLUS ...

● FULLY INTEGRATED OPERATION

The MK IV 'Professional' has all the features of the MK IV plus an enhanced version MS LOGIC PROCESSING CHIP that integrates the whole range of utilities and makes them available at the press of a button at any time.

● EXTENDED MONITOR

The 'Professional' has an on-board powerful machine code monitor. Because it has both ROM and RAM at its disposal the Professional can freeze any program and then examine the WHOLE OF COMPUTER MEMORY in the frozen state including system RAM, EPROM (PAGE) and STACK.

Full feature disassembly, compare, fill, transfer, jump, relocate, jump etc. etc. to reveal all the features of the new fully frozen monitor available. Return to the frozen program at the press of a key at the point you left it. An absolute must for the program hacker - or even the programmer who needs to debug his program.

● IMPROVED HARDWARE

The Professional hardware is unmatched anywhere in the world today. Its special logic processing chip runs even with premium methods as they appear for existing to its environment.

● BUILT LOADER

In addition to Wzap III, the MK IV Professional now has BARGLOADERS. Making use of microboard III from the Professional MK4, also had commercial disks already easy to use is recommended. Remember the features in addition to AR4's unique Wzap III feature that enables all hardware to 35 times speed.

WARP 25

Replaces an average 1000's to 10's seconds!

MK IV PROFESSIONAL
ONLY £34.99
POST FREE

UPGRADE INFORMATION

MK IV PRO-MK III: Improved MK III and MK III will work with the new MK IV. MK IV will work with your existing MK III's as well.

MK IV MK IV PRO: "POWER KARAOKE" based on your old hardware plus a lot of new bits will work with a new Professional MK III.

MK II Action Replay cannot be used as part hardware system with the MK IV or Professional. Send old hardware for return.

PERFORMANCE PROMISE

Action Replay will backup any program that you allow, including any loading and saving to diskette or unformatted output without hardware. Before you buy check our performance claims and what else there are and how many of the Action Replay bits of hardware are useful and check to have a free copy of our Engineering Guide etc. When you buy Action Replay check our 100% money back guarantee to be sure that we're serious. 14 days for a full refund.

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS
HOW TO ORDER ...

BY PHONE	BY POST	OR ORDERING POST FREE
  		
0202 744777 or 0202 744777 ext.	Postage & P&I extra International Order (overseas)	TERMS: OVERSEAS ADD 10.00 50% DS
		CALL 0202 744777

DATTEL ELECTRONICS

DATTEL ELECTRONICS LTD, FELTON INDUSTRIAL ESTATE,
CROYAN ROAD, FELTON, ENGLAND ON TRENT, ENGLAND

SALES ONLY
0202 744777

TECHNICAL ONLY
0202 744324



Editor: Stuart Cooke
 Deputy Editor:
 Sam Joyce
 Technical Editor:
 Fred Reid
 Advertisement
 Manager:
 Paul Kinnangh
 Advertisement Copy
 Control: Andrew
 Selwood
 Originations: Ebony
 Typesetting:
 Artist: Alan Batchelor
 Designer: Neil
 Sweetman

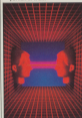
Your Commodore
 incorporating Your 64 in a
 monthly magazine appearing
 on the first Friday of each
 month. Your Advars in
 published every second month
 within the pages of Your
 Commodore. Angus Specialist
 Publications Limited Editorial
 & Advertisement Office, Your
 Commodore, No 1 Golden
 Square, London W1R 3AA.
 Telephone: 01-407 0800 Telex:
 903 0900.

Subscription rates upon
 application to Your
 Commodore Subscriptions
 Department, Editorial Ltd, 1
 Bruce Park Square,
 Barkinghurst, Herts, HP14
 0HL, U.K.B. Subscription
 Agent, New Con Worldwide
 Publications, 3214 West 126th
 Street, Torrance, CA 90505,
 U.S.A.



FEATURES

- **GEOS Goes To Work** 10
 Catch up on this mouse orientated environment
- **A Short Interlude** 24
 Mix and match your interrupts
- **Communications - The Way Forward** 26
 Clive Grace runs up a phone bill in the pursuit of truth



Communications



Geos goes to work

- **The Moving Cursor Writes** 36
 A wordprocessor roundup
- **Structured Program Design** 42
 Improve the standard of your programming
- **Trilogic's Rocket Attack** 60
 With Trilogic's new Rocket system, who needs a parallel DOS?

UTILITIES

- **Keep It Simple** 14
 Turn your Plus/4 into a WIMP!
- **Sprite Library** 17
 Take it with this month's installment
- **Educating Sidney** 46
 Teach your SID chip to be more sociable
- **Tape Organiser** 50
 Better communication between your cassette deck and your computer
- **Split Band Rate Terminal** 58
 Not one but two communication programs

	Y64	Y16	Y128	Y7
GEOS Goes To Work	•	•		
A Short Interlude	•	•		
Communications - The Way Forward	•	•	•	•
The Moving Cursor Writes	•	•	•	•
Structured Program Design	•	•		
Trilogic's Rocket Attack	•	•		
Keep It Simple				•
Sprite Library				
Educating Sidney	•	•		
Tape Organiser				
Split Band Rate Terminal	•	•		

REGULARS

		2	3	5	7
• Data Statements	6	•	•	•	•
Our regular look at the Commodore world					
• Competition	8	•	•	•	•
Win a goods bag from Telegic					
• Games Update	18	•	•	•	•
What's going on in the world of games					
• Making Music	22	•		•	
This month Pete Gerrard takes a look at interrupts and background music					
• Karnov	25	•		•	
A Russian fire breathing circus strongman takes the lead role in this game					
• Byting into the 6510	31	•		•	
ROM routines explained					
• Polton versus Roumel	40	•		•	
Put your wits against the world's greatest military leaders					
• Wizard Wars	52	•	•		
An opportunity to brush up on your spell casting					
• First Steps	54	•	•	•	•
Computers are fallible creatures, learn to sort out what the error messages really mean					
• Infiltrator II	57	•		•	
Captain Johnny 'Jumbo-Jumbo' McGilberts rides again					
• Software for Sale	62	•	•	•	•
• Jinx	65	•		•	
Where Breakout meets Pinball!					
• Relative File Programming	66	•	•	•	•
The fourth gripping installment in our regular programming series					
• Listings	71	•	•	•	•
• Back Page	90	•	•	•	•

**VOLUME 4
NUMBER 11**

**ARGUS
PRESS
GROUP**

**SEPT ISSUE
AVAILABLE
5th AUG 1988**

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 international copyright conventions are specifically reserved to Argus Specialist Publications Limited and any reproduction requires the prior written consent of the Company. © 1988. Distribution: 5M Distribution, 61 Leighton Court Road, London E15 2PJ. Printed by Clunas Web, 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 review we cannot be held responsible for any errors that do occur.

**ISSN
0269-8277**

D A T A S T A T E M E N T S

Commodore Beats
Amstrad!

COMMODORE HAVE announced a dramatic price cut on their PC1 personal computer, from £499.99 to £349.99, including mouse monitor, Able software and VAT. This puts the Commodore PC1 some £30 more attractive than Amstrad's comparable machine, and makes it the cheapest PC

"deal" on the market in this country! The PC1 uses an 8888 processor running at 4.77MHz, and comes with the usual 512K RAM, expandable to 640K. Ready to run business software, MS-DOS 3.2 and a built in 7 $\frac{1}{2}$ " 800K floppy complete the picture.

The PC1 is probably one of the

smallest desktop machines around, measuring just 12 inches square, it should prove popular with those short of desk space, the Able software includes wordprocessing, spreadsheet and database management software.

The ball's in your court, Amstrad...

Busby gets Busy on Amiga

TELECOM SOFT have been busy this month with five major game releases for the Amiga from Rainbow Warrior's jewel in the crown is undoubtedly Starfighter 2, the sequel to Jet Set's blockbuster, Starfighter. Your mission (should you choose to accept it) is to annihilate the Elytra army overseas for all, and destroy the beam projector on their home planet, Novema. The odds are stacked against you, you can only strain Elytra systems unarmored. On top of that, your Hallacraon projector is on the blink and it's generating some pretty hairy images of hideous monsters and mutated creatures.

Next up, Carrier Command, a strategic shoot 'em up on the high seas. Take the helm of a futuristic aircraft carrier complete with fighter planes and amphibious assault craft. Whether you attack the enemy installations with planes, tanks or both, you'll have to run the gamut of surface to surface and surface to air missiles, lasers and enemy tanks and planes.

Legend of the Sword, a mega adventure game, tells of the chronicles of Anas, a mystical sword and a shield whose magical aura protected the inhabitants of Anas for many centuries. But now Anas has been plunged into a state of turmoil and fear by an invading force of telephonic engineers - oops, mutated humanoids under the evil wizard Saxon. Only with the combined power of the sword and shield is it possible to defeat the dark forces.

Where did Napoleon keep his armies? Up his sleeve! This could be your chance to find out for yourself! The Universal Military Simulator is an entirely new concept in war games. Re-enact some of the world's greatest



Legend of the Sword



Virus



Starfighter 2



Universal Military Simulator

military conflicts including the battle of Hastings, Marston Moor, Waterloo and Gettysburg against the computer or with a friend. The graphics are rendered to be superlative in 3-D, view from any angle, or zoom in for a spot-check on your troops. You're not stuck with real battles, you can create your own situations, maps and armies, even match heroes from different times and/or troops. You'd like to pit Monty's Desert Rats against Alexander the Great?

Firebird's first contribution this month is a superior 3-D shoot 'em up by David Braben, co-author of Elite. Virus is actually a conversion of Zark, currently knocking 'em dead on the Archimedes. Invading aliens are attacking your planet, polluting the surface with a deadly virus. Your job is to destroy the alien craft and thus prevent the spread of the virus. Having seen the Archimedes version, I can't wait to see it on the Amiga.

Also from Firebird is Whirligig, a space battle extravaganza. Whirligig tells the story of ships with brains, kept glowing and mechanically healthy by slaves - small humanoid bio-things, skurrying around the ships to tend to their every need. You are determined to steal the to be born (?) and boldly seek out brave new worlds. The whirligig is actually an interplanetary shortcut to other worlds and times, but you'll need to blast your way through fleets of alien spacecraft, picking up new weapons as you go....

Available: Starfighter 2, Carrier Command, Legend of the Sword and Universal Military Simulator, all at £24.95 each, from Rainbow. Virus and Whirligig cost £19.95 each, from Firebird.

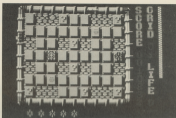
D A T A S T A T E M E N T S

Commodore v Chelsea

THOUGH COMMODORE are to be commended for their sponsorship of Chelsea FC, recent events must pose a dilemma for both company and club. The appalling behaviour of Chelsea supporters at the recent play-offs against Middlesbrough not only reflects badly on both of the football clubs but also on the companies who support them.

The television coverage showed the full horror of the incident as Chelsea supporters surged across the pitch towards the Middlesbrough supporters. Property, punches and abuse were thrown with equal abandon as police hurried to control the melee. The greater tragedy is that this incident not only discredits football in this country but will also feature strongly as the European debate on the necessity of English clubs into full international competition rages on. Having this debate video-recorded of the incident will no doubt be shown and re-shown with Commodore's name featuring strongly and receiving some of the torches. Subliminal though this inference may be, it will exist. Can Commodore afford to be connected with a team who is supported by a strong hoodlum element?

Chelsea and other League clubs maintain that they are doing



Oop! News from The Big Apple
"Oop!", a thoroughly addictive arcade style game, the first offering from London's newest software house is about to hit the streets. The Big Apple label plan to release several more titles in the near future, watch out for

Delphin and Newton on all formats in the near future. Oop! is available mid-June on the C64 (£9.95) and Amiga (£19.95).

Touchline:

The Big Apple Entertainment Co. Tel: 87-568 5141.

everything humanly possible to eradicate the violent element from the sport but this is blatantly not working.

Whether Commodore can afford to maintain their support in their decision, Chelsea cannot be held totally responsible for their fans at all times but the sponsors must ensure that there is clear both for the reputation of English football and the business of the company name.

Programmer's Olympics

THE FIRST COMPUTER Olympiad is all set for August, at London's prestigious Park Lane Hotel. For the first time, you will be able to see machines pitted against machine, program against program and programmer against the bar! This unique event will feature tournaments for chess, bridge, backgammon, draughts, poker, and many other 'thinking' games, the human operators doing no more than telling their own computers what moves have been made by their opponents.

The first London Conference on Computer Games will take place as part of the Olympiad chaired by Professor Tony Marsland from the computing science department of the University of Alberta, Canada, papers will be invited on all aspects of programming computers to play 'thinking' games.

The Computer Olympiad is organised by International Chess Master David Levy, who is president of the International Computer Chess Association. Anyone wanting further information on the event should send a large stamped addressed envelope to Computer Olympiad, 11 Loudoun Road, London NW9 8LP.

Electronic Arts in \$7.3 Million Lawsuit

BETHESDA SOFTWARE, developer of Gridiron (an American football simulator) is accusing California based software publisher and distributor Electronic Arts of forcing their game off the market. Bethesda claim that Electronic Arts offered them firm development and distribution contracts in order to promote their own American football game, 'John Madden Football'.

Under the 1987 contract, Electronic Arts gained exclusive marketing rights to the Amiga and Atari ST versions of Gridiron! and prohibited Bethesda from developing it for any other computer, according

to documents filed in the California federal court.

As a condition of the marketing agreement, Electronic Arts also called for major parts of Gridiron! to be incorporated into a new game featuring former Oakland Raiders coach and current CBS commentator, John Madden, according to Bethesda's suit.

In a match, Bethesda Software are accusing Electronic Arts of naming them into designing Gridiron, and then using the game as a basis for their own 'John Madden Football'. Would anyone from Electronic Arts care to reply?

Win Expert Prizes from Trilogic

Spot the differences and you could be the proud owner of the Expert cartridge, the Voice Digitiser and the Data Sector Doctor

We've teamed up with Trilogic for this month's competition. The first entry picked out of the hat will win the Expert cartridge, the Voice Digitiser and the Data Sector Doctor. The nine runners up will each receive a copy of either the Voice Digitiser or the Data Sector Doctor. (Please state preference on entry coupon.)



Trilogic Entry Coupon

Name

Address

..... Postcode

Number of differences found

Voice Digitiser

Data Sector Doctor

(please tick)

Closing date: 1st August 1988

Post to: Trilogic Competition
 Your Commodore
 1 Golden Square
 London W1R 3AD



The Graphic Environment Operating System, that was once set to be bundled with the Commodore 64C has at last come of age

GEOS

In its original format GEOS consisted of a single disk that contained the Amiga style desktop as well as GeoWrite and GeoPaint applications programs. However, there were incredibly basic programs and were little more than demonstrations of what could be done with a GEOS environment.

In December 1986, Berkeley Software, the author of GEOS, released GEOS 1.2 with updated versions of GeoWrite and GeoPaint and also the Writer's Workshop (turned GeoWrite into a full wordprocessor), Postpack 1 (28 new fonts), GeoDEX (read index system), Desktop (calendar, graphics grabber and line editor) applications which were followed later by a full database program, GeoFile and GeoCalc the spreadsheet.

Armed with all these a dedicated user could turn GEOS into a workable system, however you were soon confronted by its limitations.

Now, four major factors have come together which will spread the use and variety of GEOS. Firstly, Berkeley Software has made further improvements to existing packages and has added more to the range including GeoPublish, GeoSpell and GeoProgrammer. Secondly, this expanding GEOS range is now being distributed in greater quantities through a joint venture with Micropro UK which has meant dramatic cuts in costs. Thirdly, GEOS is being bundled with new disk drives such as those supplied by Eventum Micros and finally in this article we review the first third party books and software.

What is GEOS?

GEOS is a C64 and now C128 disk operating system that attempts to mimic the icons, pull down menus and workbench system pioneered on machines such as the Apple Macintosh, Atari ST and Amiga. It's a "You get what you see" or "if you

want it, click it" system where a click of a mouse or joystick button can lead to files, select options or swap disks.

Each package or program that you add to the system must be installed or keyed into your own copy of GEOS. This not only prevents piracy but is the beginning of a main file copy system in which you create work disks containing all the programs, files and type fonts you will need.

This is important as GEOS is a fully integrated system in which you could create graphics in GeoPaint or grab them from Print Shop, Print Master or Newsroom disks and mix them with text in GeoWrite documents which can then be used for a standard letter sent out by GeoMerge to some or all of the names and addresses stored in a GeoFile database or GeoDev card file.

Upgrades

1988 sees a more streamlined GEOS system with many of the old upgrades now included in the new standard packs. GEOS itself is now supplied on two disks so you now have an automatic backup copy (once it's been installed) and a built-in disk copy utility to ease the creation of

work disks. This is all backed up by an improved and enlarged manual.

The new GEOS system and all subsequent application programs including the ones mentioned below include the 1.3 version of desktop as standard with its faster disk access and keyboard shortcuts. For frequently used commands such as opening or closing disks, pressing two keys is a lot faster than selecting options from pull down menus. In new GEOS style the keyboard shortcuts aren't buried away in a manual but are alongside the appropriate function or command in the pull down menus.

Desktop 1.3 also includes more printer drivers so that more printers and interfaces can now be used to print GEOS text and graphics. A full table and guide to help you install the right printer driver are included in the manual.

Desktop Plus is the updated version of Desktop and includes the original combination of Calendar, Icon Editor, Blank Jack and Graphics Grabber but now also includes the GeoDev and GeoMerge programs that were sold separately as GeoDev.

Similarly, the old Writer's Workshop has been updated to form the GeoWrite Workshop which is now available in both C64 and C128

GEOWRITE WORKSHOP

FOR THE COMMODORE 64 AND 128. THE ONLY WORD PROCESSOR THAT LETS YOU SEE WHAT YOU GET. WITH A FULL RANGE OF FONTS, A PULL-DOWN MENU, AND A WINDOW MANAGER, GEOWRITE WORKSHOP IS THE MOST POWERFUL AND EASY TO USE WORD PROCESSOR AVAILABLE FOR THE COMMODORE 64 AND 128.

Safford

GEO PUBLISH

FOR THE COMMODORE 64 AND 128. THE ONLY PUBLISHER THAT LETS YOU SEE WHAT YOU GET. WITH A FULL RANGE OF FONTS, A PULL-DOWN MENU, AND A WINDOW MANAGER, GEO PUBLISH IS THE MOST POWERFUL AND EASY TO USE PUBLISHER AVAILABLE FOR THE COMMODORE 64 AND 128.

Safford

GEOS Goes to Work

versions and has the additional ability to include customized headers and footers and three types of printers in almost any shape or size.

Expanding the Range

Berkeley Softworks has released four new packages to extend the GEOS range - GeoPublish, GeoSpell, Fontpack Plus and GeoProgrammer.

GeoPublish adds a full desktop publisher to the GEOS range and can be used to create newsletters and magazines.

Creating a page or a longer magazine couldn't be easier as each step is described in detail in the manual and demonstrated through worked examples.

Before the task can begin you must design a master page format by deciding the height of the headline, what will appear at the foot of each page and the number and size of the columns of text. This can range from a single column spread across the page at two, three or four times columns for a more professional look.

Once you have created master page formats, or selected one of the sample formats included on the GeoPublish disk you're ready to make up the pages.

Switching to Page Layouts mode allows you to box off parts of the page and assign them to either text or graphics. You don't actually have to type in the text, you simply assign a file either created by GeoWrite or converted by the text grabber. GeoPublish will then print in the text in the space you've allocated with any remaining words carried onto the next page.

Similarly, you can assign a space for graphics and then fill it with anything stored in a GEOS picture album. These can range from GeoPaint masterpieces to clip art "grabbed" from Newsroom, Print Master or Print Shop.

That's only half the story as each box can be moved to anywhere on the screen, altered in size and shape and filled with a picture that can be cut or cropped to fit, reduced to fit or stretched to fill the allocated space.

Whenever you change a page layout, GeoPublish automatically re-adjusts the text through the page leaving you to concentrate on the design and layout.

GeoPublish introduces a new command tool to the existing pull down menus and dialog boxes which usually pose you no questions but can be used to select brush widths and fill

patterns. This is a toolbox which contains up to 32 icons representing additional commands that are specific to the screen mode you are in and are activated by a simple click of a joystick or mouse button. They are at first confusing and will have you diving for the manual, as you create your first masterpiece but they soon become second nature and you wonder how you ever managed without them.

New with the pages filled with all the text running neatly around the appropriate graphics you can add a headline (up to 192 point 2½ inches high) or zoom in to edit part of a page.

In edit mode you can add and delete words and even change the font and point size to add cross heads and captions to your text and graphics. You can also use a mini graphics utility to add the final touches such as ruled lines, circles or boxes that can be drawn in a variety of thicknesses and filled in a selection of patterns.

The final proof should be a masterpiece, if not you can add or edit it or start again and then print it out on a dot matrix printer or through LaserPublish and a laser printer which will greatly improve its appearance. This isn't as drastic as it sounds as the words and pictures will still be secure in disk files as GeoPublish only decides the order and positions that they are used in.

Fontpack Plus can improve the style and variety of any GEOS document as it includes 45 new fonts and ten of the best from Fontpack I.

These new fonts include some that consist of symbols instead of letters and numbers so you can easily add scientific and electronic notations as well as trees, musical notes and house contents. In other words everything including the kitchen sink!

If a total of 55 fonts isn't enough, and it should be for most Geowrite documents and to appear up more GeoPublish pages then you can create your own with GeoFont which is also included on the Fontpack Plus disk.

This works the same as any character editor and can either be used

GEOSPELL™
 THE SPELL CHECKER FOR GEOS

FOR THE GEOS SYSTEMS AND COMPATIBLE HARDWARE

Berkeley Softworks

Fontpack Plus™

FOR THE GEOS SYSTEMS AND COMPATIBLE HARDWARE

Berkeley Softworks

to alter existing fonts or to create your own including pictures which can add character graphics to GEOS.

GeoSpell also includes GeoFont as well as a 28,800 word, upgradable spell checker which can be used to correct even the most spilling misspellers in any GeoWrite file including any grabbed by the text grabber and converted to GeoWrite format.

Twenty-eight thousand may sound a little low for a dictionary but you can freely add words to user dictionaries and swap between them at will.

GeoSpell makes up for this apparent lack of words through ease of use as whenever it finds a word it doesn't know you have the choice to accept it as it is, use the alphabet keys to search for any word that might be stored in any of your dictionaries or use the FIND key to produce a scrolling list of possible options that can then be used to replace the word as well as every other line that word appears.

The transition from the user friendly end user application GeoSpell to the assembly language development package GeoProgrammer is about the biggest jump you can make.



GeoProgrammer is a scaled down version of Berkeley Software's own UNIX development system and can be used to create GEOS application programs. The program is supported by a massive 400 page manual that describes in great detail the three parts of the system, GeoAssembler, GeoLinker and GeoDebugger.

GeoAssembler takes 650 assembly instructions and creates linkable object files which are then linked by GeoLinker to form either

B·e·c·k·e·r B·A·S·I·C

For programming applications under GEOS™



Abacus 
A Data Becker Product

stand alone CBI programs or GEOS applications that can be tested by GeoDebugger which allows you to instantly toggle between the GEOS bit-screen and the text mode debugging screen.

Each section boasts an impressive range of features such as the support for overlay modules in GeoLinker and GeoDebugger's 88 commands that allow you to assemble, disassemble, single step, display variables and set breakpoints.

GeoAssembler code is written into a normal GeoWrite file so you can add different fonts to make your code easier to understand without affecting the code. For example, you could highlight labels in italics or in dots if you really want them to stand out! You can even use the DeskPark Icon Editor to create icons and GeoPaint for graphics which are automatically converted into binary data.

If you don't think you're ready for GeoProgrammer then BeckeR Basic would be more your style.

This Abacus extended basic breaks

new ground as one of the first pieces of third party software and brings the ability to create GEOS programs complete with pull down menus, icons and dialog boxes to the reach of the Basic programmer.



Becker Basic consists of 175 new commands that are used in the input and testing systems to create a Becker Basic program that can then be run as a GEOS application by double-clicking its desktop icon or run on its own through the third part of the program.

Each of these commands can be returned to their actions can be made clearer. For example, PRINT could become OUTPUT or even PAPER. This may not seem important but a remaining session could save you a lot of time delving in the manual for the right command.

These commands include programming tools such as TRACE, and RENUMBER, structured programming controls IF/THEN/ELSE, REPEAT, WHILE and LOOP, X commands to create and move sprites and IS to create the modulations, alter the filters, change the waveforms and set the envelopes to make beautiful music.

The menus and dialogic boxes that will give your programs that GEOS touch are built and added to your programs through the pull-downs menus and dialogic box construction sets.

With GEOS and Becker Basic in memory it is quite remarkable that there is 16K remaining for your program. However, if you add pull down menus and dialogic boxes to your code you'll be left with about 1K as they require a second screen bitmap. Conversely, if you avoid hires graphics altogether you would gain an extra 1K but the result wouldn't be a GEOS program.

Creating a pull down menu or dialogic box couldn't be simpler as the construction set generates the code you need and saves it on disk in response to simple questions such as the number of menu items or dialogic box options and the text they should contain.

You can even have sub-menus for your pull down menus that can run either horizontally or vertically down or across the screen.

Dialogic boxes can include up to six options ranging from yes or no to which way to go at a junction in an adventure. A simple branch command will then direct the program to the right section of code.

Here at last are two different ways to create GEOS programs, GeoProgrammer for the assembler and Becker Basic for the others. Both offer ease of use and the chance to

use GEOS's menus and boxes in their programs. These not only provide two good ways of programming on the C64 but also opens the door for more GEOS applications.

GEOS Books

The Official GEOS Programmer Reference Guide, published by Rastan Computer Books, is the official source of facts and addresses for programmers who wish to delve into GEOS without the aid of GeoProgrammer.

Armed with this manual the assembler programmer can unlock the GEOS menu routines to create icons, menus, dialogic boxes, fonts as well as new printer interfaces, graphics libraries and multi-tasking applications.

Four hundred and fifty pages that can provide the GEOS equivalent to the C64 Programmer's reference guide. Having said that, it shouldn't scare off first time assembler programmers as the book describes these routines through labels that are indexed at the end of the book, and therefore can be substituted for the hex equivalent when you start coding.

GEOS Tricks and Tips, from Abacus the company behind Becker Basic, covers the whole spectrum of GEOS from hints and tips for GeoWrite and GeoPaint users, a guide for programmers and listings for three GEOS programs.

The hints and tips cover common sense and shortcuts which make using GEOS programs a lot easier, such as writing your own error messages and creating GeoWrite form letters.

If you use one of the 50 hints and tips crammed into the book you could type in either a consumer or lost editor which do the same job as GeoFast and the Desktop's text grabber or tackle Edmon, a machine code monitor to delve into programming and converting GEOS.

This new burst of GEOS activity which has included the development of a GEOS desktop publisher and spellchecker, the release of programming tools backed up with information aimed at every level of user will attract more and more interest in this valuable but underrated operating system.

GEOS has now evolved from a gimmick to a mini 16 bit machine to a system with its own wordprocessor complete with spellchecker, spreadsheet, graphics package, database and DTP package and also has the ability to incorporate others programs, files and pictures into the system.

To paraphrase someone who was actually describing the Amiga (but its equally appropriate here), GEOS was first a child full of promise but couldn't stand on its own, then it was a teenager as it became more organised and useful but still capable of throwing the occasional tantrum, now it has come of age. ☐

Supplier: Micropress UK, 1, Market Place, Tisbury,
Glouc
Tel: 0666 54126

Becker Basic (with software)	
GEOS Tricks and Tips	£11.95
The Official GEOS Programmer Reference Guide	£17.99
GEOS 1.3	£29.95
GeoCalc	£29.95
GeoPublish	£29.95
GeoWrite Workshop	£24.95
FontPack 1	£19.95
FontPack Plus	£24.95
GeoProgrammer	£39.95
GeoFile	£29.95
GeoSpell	£19.95
DeskPack Plus	£24.95
GEOS 1.2	£39.95
GeoWrite Workshop 1.2	£39.95
GeoCalc 1.2	£29.95
GeoFile 1.2	£39.95

40 gos to,x,y
50 goto 20

WINDOW,x,y,w,h requires four parameters and will set up a window on the screen, draw a border, and clear the window. 'x' and 'y' are the coordinates of the top-left corner of the window, 'w' is the width of the window, and 'h' is the height of the window. The ranges for these are as follows:

- 1 < x < 38 - Range of X-Coordinate.
- 1 < y < 23 - Range of Y-Coordinate.
- 0 < w < 38x - Range of Width.
- 0 < h < 24-y - Range of Height.

For example:

```
10 window=5416:desk=5204
20 gos desk
30 gos window,3,5,18,18: print
  "Window #1"
40 gos window,10,10,10,10: print
  "Window #2"
```

POSITION,x,y simply sets the coordinates of the pointer to the inserted values. It is used in conjunction with the LOOP command to set the start position of the movement. 'x' and 'y' are the coordinates of the pointer and are in the range below:

- 1 < x < 40 - Range of X-Coordinate.
- 1 < y < 24 - Range of Y-Coordinate.

For example:

```
10 goto 5991
20 input "Pointer Coordinates": x,y
30 gos pos,x,y
40 "The Pointer is at"x,y
```

SHOW & **HIDE** respectively print and erase the pointer. When SHOW is executed, the two characters under the pointer are saved before it is printed. HIDE then retrieves the characters and puts them back on the screen. For example:

```
10 desk=5204:ptr=5991:show=5618
  hide=5680
20 gos desk:gos ptr,20,12
30 clear 1,1,1, "A Flashing pointer!"
40 gos show:gos hide
50 gos hide:gos hide:40:goto 40
60 for i=1 to 200:next i:return
```

SPEED,s - as all the routines are written in machine code, if there were not a delay loop in the routines to move the pointer etc., it would zip around the screen and be totally uncontrollable. Also, people have different reaction times, and may prefer to have the pointer moving at a slower speed than others. This command will determine how slow or fast the the pointer moves, 'y' which is the speed, must be in the range below:

- 1 < s < 126 - Higher the value, lower the speed.

For example:

```
10 desk=5204:ptr=5991: speed=5712
  move=5729
20 sound:print "Pointer Speed:599 to
  end?"
30 if s=999 then stop
40 gos speed,x,y:desk:gos ptr,20,12
50 gos move:goto 20
```

MOVE is the most important command in SIMPLE's vocabulary. It draws the pointer, and allows it to be moved around the screen and the select button is pressed without erasing everything it moves over. For an example of it in use, see SPEEDS.

MAKE "Optimised/Optimised/Optimised...", sets up a pull-down menu. Pull-down menus have a fixed size of 8 by 9 characters with up to eight options, and the last option is always 'Exit'. Each line of option text must

be separated by a ')' character, 'm' is the menu number and 'y' is the x-coordinate at which you wish the menu to be pulled down.

- "01/02/03..." 72 Characters - Option text length.
- 1 < m < 4 - Menu number.
- 1 < i < 31 - Tab position.

For example:

```
10 make=5837
20 gos make "These are the options/
  for/menu/number/length" ,1
30 print>Type in 1837"
40 monitor
```

PULL,m pulls-down menu number 'm' and allows you to select an option by moving the reverse-video bar up and down, then pressing the select button. The parameter range is shown below:

- 1 < m < 4 - Range of Menu number.

For example:

```
10 desk=5200: menu=5204:
  make=5837: move=5729: pull=5937:
  hide=5680: s=5281: ftab=5314
20 gos desk:gos menu "9 1 2 3"
30 gos make "Menu 0/Op0/Op0/
  Op4/Op2/Op6/Op2/Op8"0,1
40 gos make "Menu 1/Op0/Op0/
  Op4/Op2/Op6/Op2/Op8"1,3
50 gos make "Menu 2/Op0/Op0/
  Op4/Op2/Op6/Op2/Op8"2,5
60 gos make "Menu 3/Op0/Op0/
  Op0/Op2/Op6/Op2/Op8"3,7
70 gos move:gos desk(216):y:gos desk(217)
80 gos hide:if 0 then 70
90 gos:gos (572+1)*40-483: m 1
  then 70
100 gos:gos:gos pull,move:hide:goto 70
```

In Use

ENABLE / **DISABLE** respectively enable, and disable the inverse-video which keeps the character-set pointers



Screen-Code	Function
64	Desk Pattern
91-92	Pointer on a plain background
93-94	Pointer on a desk background
95	The I-line (Text pointer)
96-97	Border characters for menu heading
98-105	Surround characters for windows

Figure 1

pointing to the SIMPLE font. This is so that when you make a mistake in a program, the screen doesn't clear to a interference-type mess!

For example:

```
10 SYS 648:END
20 SYS 648:END
```

Type RUN 10 then type some rubbish. The screen will blink, and your error will be displayed.

Type RUN 20 then type some rubbish. The screen will go crazy! Type RUN 30 to re-enable the interrupt.

Co-ordinate Sensing - after a MOVE or POSITION command, locations SD8 and SD9 hold the current coordinates of the pointer. Thus, they can also be changed by FORKING directly.

Option Sensing - after a FULL command, the Option that was selected is stored in SD9 - the Y-Coordinates of the pointer. This is so that the pointer remains in a logical place after pulling a menu down. Before moving the pointer again, you need either assign the option number to a variable or store it elsewhere in RAM.

The SIMPLE Font - to improve the Reverse Video, and create a space for icons, a new font is defined by SIMPLE starting at the address \$2000. This font can be redefined in the usual way by using a Character Designer.

The Icons - to square all fifteen icons into a character set, the characters which make up the icons are not stored in a strict sequential order - they skip around alphabets etc... For a list of screen codes which make up each icon, see the reference table (Figure 1). Again, these can be redefined using a Character Designer.

Special Characters - there are some character codes which are neither icons or normal alphanumeric characters. These are shown below:

Re-defining the Keys - the listing presented here is configured for a joystick in port 1, using the fire-button as a select button. If you do not have a joystick, or would prefer to use the keyboard, you can change the keys quite easily. The third section of the reference table shows the addresses you need to change the values in, and the relevant values for a joystick in port 1, and the cursor keys + RETURN. However, you can use any keys by pointing their ASCII codes into the locations shown.

Starting the System - to take up as little memory as possible, SIMPLE is installed at the bottom of memory. It operates only in text mode, and you must not attempt to enter a Hires mode as this will corrupt the program. As it is at the bottom of RAM, the start of program pointers must be set before entering the SIMPLE Loader program, or before writing your own SIMPLE program. This is done as follows and is very important:

```
POKE 10240,0:POKE 44,40:NEW
```

This sets the pointers. Now, you can begin to type in the loader or begin to write your own programs.

When you have typed in and debugged the loader, RUN the program and follow the instructions to save the code.

MONITOR

```
5 "simple",8,1000,2000 - Save To Disk.
```

```
5 "simple",1,1000,2001 - Save to Tape.
```

X - Exit Monitor.

Then to reload the code, enter the above codes and type the following:

```
LOAD "simple",8,1 - Load from Disk.
LOAD "simple",1,1 - Load from Tape.
```

Type SYS 648 (ENABLE) to set up the interrupt and now program to your heart's content. It's SIMPLE when you know how!

Using the Demonstration Program

Before typing the demonstration program in, you must set the bottom of memory pointers as explained previously. The program starts by loading the code file "simple" which should already have been saved as explained before. Tape users must change line 10 to:

```
10 IF C10 THEN C=LOAD "SIMPLE",1,1
```

Once the program has been run, you can move the pointer anywhere on the screen. Experiment with pointing the select button. You will find that everything on screen will give a reaction of some kind!

Clicking on one of the icons at the base of the screen will result in a small window to indicate what each icon represents. Clicking on the desk-top, title-window, or menu-header will have the same result.

The four pull-down menus are Screen, Pointer, Data, and Cookies. The first three consist of lists of the SIMPLE routines. Selecting any of these will show a large window telling you the syntax, etc of the routine. These windows are closed by clicking on the reverse-video button saying "OK".

The fourth pull-down menu has four options Input, Speed, Info, and Quit. By selecting Input, you can choose whether to use Joystick or Keyboard. This is done by clicking on the relevant icon.

When you select Speed, a window will appear in the centre of the screen with a bar showing the present speed. You can change this setting by pointing at the position on the bar corresponding to the required speed.

Selecting Info just shows a window of general information about the program, which is removed by clicking on OK in the usual way.

Finally, selecting Quit will exit the program and reset the computer, but be warned the next performed is a cold one, and the program cannot be recovered.

Please note that I would be interested to see any programs you write using SIMPLE. You can send these to me at the address below, on tape or disk, enclosing an SAE. Send to: Mark Dunningham, 17 Collingwood Road, Redfern, Bristol, BS6 6PD. 70

See listings on page 73

Sprite Library

This month our Sprite Library series takes to the sky with jets, helicopters and hang-gliders

By Mike Benn



This month the Sprite Library will take to the air in three different forms of transport. Starting with a small jet followed by a helicopter flight and finally a white truck to ride under a hang-glider. Both the plane and the helicopter approach the viewer from the distance and turn to the right. The hang-glider is not animate as space did not permit. However, there should be enough key frames for you to fill in the missing animation.

All the sprite definitions require two sprites.

Getting it all in

Type in the basic loader as published and SAVE IT-DON'T RUN IT or it will self destruct. Before running the loader program you will need to reset the computer and type the following: POKÉ43,0; POKÉ44,64; POKÉ43M4,0; NEW and press return.

This will trick the computer into believing that the basic now starts at 54000 instead of 50000. Load in the basic loader and run it, if error free, the program will automatically save itself as a block of data. If you reload that data in the future remember to add a 4 after the device number. The data is saved in the following location: 52000-53777.

The sprites run from 160 to 225 in a compromise to avoid the area 52000 traditionally set aside for re-defined character graphics and to avoid the need of typing in line after line of data.

If only one or two sprites are required then use the following formula: (Sprite Block No.-160) *40 + 190 = the data line number at which that sprite block's data starts. Remember to type in the following three lines of data and alter the variable RL to the number of data lines you have in your finished program, less 1.

The small basic program AIRCRAFT DISPLAY will variably animate the sprites in both non-expanded forms on the screen simultaneously. To hold a sprite enter the same number for Start and End.

Any Sprite Editor program will enable you to change and adapt the individual sprites to your own requirements.

See Listings on page 72

HEX.	DECIMAL	DESCRIPTION
AB-07	160-183	Jet plane approaching and turning to the right
BB-CA	184-207	Helicopter approaching and turning to the right
CB-CC	192-195	Helicopter blades turning while helicopter hovers facing the viewer
CA-CE	200-203	Helicopter blades turning while helicopter hovers facing right
CE-D6	208-213	Hang-glider turning to the right
D6-D9	214-215	Hang-glider flying upwards to the right
D6-D9	216-217	Hang-glider getting ready to land or has just taken off
DA-DB	218-219	Hang-glider on the ground just about to take off or has just landed
DC-DD	220-221	Hang-glider on the ground, pilot under canopy
DE-DF	222-223	Hang-glider on the ground

Games Update

Well, as to be expected all is quiet on the games front this month. Few releases are around and we all eagerly await the build up to the Autumn releases



Quester II

A quick visit to my local W.H. Smiths reveals that their software shelves have large numbers of compilations and sequels among their top titles, with very few new releases. Hopefully, Autumn will bring a host of exciting products in what has, so far, been a disappointing year as far as games are concerned.

One new feature this month is the introduction of a roundup of Amiga games. The sixteen bit invasion marches ever on.

Commodore 64

Strategy is very much the flavour of the month with one fantasy role playing game, two wargames and one classic board game comprising the main offerings.

Quester II (US Gold \$55) sees you trying to destroy the Book of Evil. The only way to achieve this is to travel back in time and prevent it from ever being created in the first place! This involves you seeking out six mad sorcerers and your journey will take you over two continents, through dungeons, catacombs and castles before you achieve your goal.

This game is entirely menu controlled via either joystick or keyboard although I found the keyboard much easier. Estimated playing time is thirty to sixty hours. This is one of the simpler fantasy role playing games currently available and

would probably appeal more to the novice or younger player of this type of game.

Two wargames have also been released from the US Gold 1000 stable. *Panzer Strike* cannot claim to be short of ambition, covering as it does, the entire Eastern Front campaign, the Western Front in 1946 and the North African campaign. Should that be insufficient for your needs, you can always design your own battles and campaigns. Despite the size, symbols still represent individual tanks and to add extra realism, armor ratings on these vehicles have been squeezed into the front and side of the hull and turret and top. Every conceivable ground weapon has been included in this simulation from artillery to tanks, mortars to tanks. Not surprisingly, this game is recommended for advanced players only!

Somewhat easier is *Sons of Liberty*, which contains introductory, intermediate and advanced scenarios. Set in the War of Independence or, as the Tanks like to call it, the Revolutionary War, you get the chance to re-enact the three major battles - Bunker Hill, Saratoga and Monmouth.

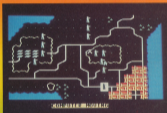
Bunker Hill was a relatively small conflict. Only a few thousand men fighting over a small area. Not only did the Brits come second at Saratoga, but the major debate caused the French (boo, hiss) to decide that this might be the perfect opportunity for them to also declare war upon us so that we now had to fight on two fronts. The battle of Monmouth was the largest of the three battles and the time when George Washington really came into his own as a leader of men. So here is your opportunity to rewrite the history books and ensure that the good old US of A really should be coloured red on all the maps!

From Lenora Dennis comes that old favourite, *Monopoly*. This game has probably caused more inter-family rows than any other in history. Monopoly players seem to come in two fairs. Those that play to win at any cost and those who don't. The two factions are totally incompatible with neither side being able to see the point of view of the other.

The computer version allows you to play against either human or computer opponents or any mix of the two. You can opt for a long or short game and games can be saved midway through. Control has been made as



Panzer Strike



Sons of Liberty



Monopoly



Seven Cities of Gold



Pink Panther



The Hobbit



Bruce Lee

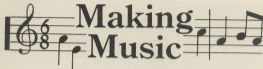
user friendly as possible and all the features of the original game - mangaping, answering and going to jail have been included. My only disappointment is that with the computer looking after the money, it is no longer possible to cheat! (No prizes for guessing to which faction you belong - Ed.)

Few budget games are noteworthy this month. On Microtron's Ricochet label comes *The Hobbit* a 3-D maze game featuring some excellent graphics for the occasion. You need to find and collect jewels and fireballs before being allowed past the dragon and onto the next level.

On the Americana label comes *Bruce Lee* a platform game with martial art elements. Bruce has to collect all the letters in an area before a door opens allowing access to the next level. At the same time, he must avoid the ministrations of ninjas and the Green Yarns. The game looks very dated but is still good fun to play.

Four games for under a pound each in the form of Microscan's third compilation. The games are *Splosion*, an arcade adventure, *Dashball*, a shoot 'em up, *The Big KO*, a boxing simulation and a platform game, *Mouse Trap*.

Finally, with timing that is about as good as one of Vinny Jones' tackles comes *F4 Cup Football* also on the Ricochet label which arrived in the office two days after Wimbledon had faked their win over Liverpool. As a game, it is about as interesting as Don Howe's tactics which, for those of you who care little for football is not very much.



Making Music

Continuing our music series, this month we look at the use of interrupts when playing background music.

By Peter Gerrard

There are many ways of playing background music using interrupts, and in the next few pages we'll be looking at just two of them. Both use the same kind of technique, so we'll see how that works before getting down to the serious business.

When playing a tune in Basic, the information is read as a series of data statements and POKE'd into the appropriate registers. In machine code we will have to have the music stored as a collection of numbers in memory, and use strange incantations and terms like "load the accumulator" and "store the accumulator" in order to make the appropriate notes. To those of you who are frightened by machine code programs, "load the accumulator" and "store the accumulator" can be thought of, at least as far as this program is concerned, as being analogous to something like A-PERK(register) and POKER(register). A, respectively. In other words we're going to be looking at a collection of memory locations, getting information from them, and then storing that information in the correct registers to make a sound.

As in the Basic example we will need a dummy set of data, or dummy information in memory, to tell the program that it has reached the end of the current data for the particular tune that we're set it playing, and since storing a -1 in a memory location is rather difficult (you try POKE \$32, -1 and see what happens!) we'll be using the value 255, since a high frequency value of 255 makes little sense to the SID chip and simply produces an extremely high pitched whine. If you really want to make a high pitched whine, then use value 254 instead and have the program to finish of its own accord.

Using Voices

In the program that follows we will be using voices two and three to play the background, interrupt driven music. Voice one will be left free for other devices and, as you might readily appreciate, voices two and three can also be used for miscellaneous noises and sound effects, since an interrupt drives tune, once started, cannot be diverted unless we tell it to be diverted. That is, voice three (in our example anyway) is being used to play a sequence of notes, and it will play those notes over and over again regardless of anything else we might tell it to do.

The waveform is stored in memory by the interrupt routine, the values for the high and low value frequencies are read in by the interrupt routine, and as anything else that we might do outside the routine is readily overridden by the routine itself. We might decide to make a noise using voice three set to the white noise waveform, but our machine code program never returns from this and carries on without skipping a note. So, we can play our two voice tune, have one voice left over to do whatever we want, and still have voices two and three to use if we really want to.

A second important fact to remember about the interrupt routine is that it can be played at a varying rate. I once heard someone else's music program, and was quite impressed by the way in which one could speed up, or slow down, the rate at which a background tune was being played. It was only in devising this program that I realized how trivial an exercise that was!

Consider the following - the interrupts on the Commodore 64 are serviced every fifth of a second or

so, and if we're playing an interrupt driven tune, this means that fifty times a second the routine is going to be activated and a note (or two, since we're using two voices) will be played. This is obviously much too fast, since few of us are capable of registering that many notes per second. We're not all fans of The Ramones you know!

So, we insert a delay loop so that a note is played only every (say) tenth interrupt. This not only slows the rate of note playing down to an acceptable level, but it also allows other things to be serviced by interrupts as well. For example if we wanted to we could have another interrupt routine in memory that was moving sprites around, or rotating user-defined graphics a la Tony Crowther!

Such a routine might look like this:

```

C000 LDX $CFFF
C001 INX
C002 STX $CFFF
C003 CPX # $0A
C004 BNE $C012
C005 LDX # $08
C006 STX $CFFF
C007 JMP $range1
C008 JMP $range1

```

Assume that we have diverted the Hardware Interrupt Vector so that it jumps off to \$C000 instead of \$EAC0 as usual. Then, everytime it gets there we load the X register with the content of memory location \$CFFF (X=POKE(\$304)) if you like). Then we increment the X register (X=X+1) and store the new value at \$CFFF again (POKE \$304,X). We then compare this new value with \$0A (Does X equal 10?), and if it is not equal then we jump to location \$C012 if X < 10 then go

A Short Interlude

*Overcome the problems of using more than one
interrupt routine with this handy utility*

By Michael Tinker

The 6510 processor used in the Commodore 64, like other processors is able to handle interrupts. The Commodore 64 makes the use of interrupts particularly easy by placing the ROM interrupt handling routine vector in RAM. This is located at \$0314/\$0315 (lo/hi format).

The interrupts take place 60 times per second and are a powerful programming tool. There are many short programs which use interrupts, available in printed form, from books and magazines.

Using Several Interrupts

The aim of this short machine code program is to make the use of several interrupt routines a lot easier. (I developed the program initially for a game I was writing which required several interrupt routines to operate at once.) The routine is however very useful for utility routines. First let us examine what the problems are in using more than one interrupt routine.

Each program is usually published as an individual, stand alone program which will usually have the following format:

```

SEI          ;set interrupt disable
LDA #0      ;
STA $0314   ;place address of new
LDA #AC0    ;location into interrupt
STA $0315   ;vector
CLI          ;clear interrupt disable
RTS         ;

*+C000      ;start of routine
...
...         ;main interrupt routine
...
JMP $EA31   ;link to ROM interrupt
            ;routine
  
```



This format means that two interrupt routines cannot be used together unless they are installed by changing the JMP \$EA31 at the end of the first routine to point to the second. The start routine to change the interrupt vector at the beginning of the second routine then becomes superfluous.

The short length of most interrupt routines will allow more than one to be in operation without any noticeable reduction in the computer speed, and before clashing with the next interrupt. Sixty interrupts per second, assuming an average of three clock cycles per instruction, allows about 3,000 instructions to be carried out before the non-interrupt program gets to a halt.

The solution I have come up with is to keep a table of the interrupt routine start address in the form of a small array similar to the BASIC DIM A(3). The control program is called on each interrupt, and in turn will call each of the routines which have their address in the table.

To use this method, only small adjustments to the interrupt routines are required. First of all remove the usual start routine which usually changes the interrupt vector, as this

task is carried out by the control program. Secondly, change the JMP \$EA31 at the end of the routine to RTS. This is because the routine is called by the controller with a JSR instruction.

The routines can be added into the table in any order; they are each called in turn starting with routine 0 at the start of the table.

(N.B. ensure that the interrupt vector is not pointing to the controller when you change the table, as it will attempt to call a routine which is not present, if the address is only partly changed when there is an interrupt.)

The controller is particularly easy to use because there is no need to predetermine how many routines you wish to use; entries in the table of \$0000 will not be called, therefore if you wish you can even stop all the routine calls. This is particularly useful when starting to use the computer when you may not know how many routines will be used.

Getting it all in

If you enter the Mikro Assembler listing first save a copy in case there has been a catastrophic error and then assemble a SYS call to 49152 will start the program. The basic listing is easier to use as there is a checksum error check as part of the listing. Also when the program is ran, the cursor will disappear over the command SYS49152 after a short delay, so all that is required is a further press of the return key and it is away!

Next month we will look at a program of more general use which will have commands to add and remove routines from the table and also list the contents, showing the addresses which are being called. Until then, happy interludes!
See listings on page 73



Step out on the Russian steps with a fire-breathing circus strongman

Have you ever been killed by a skeleton riding an ostrich? It's pretty safe to bet the answer will be not! In fact there's only a minimal risk of this owing to the idea of you and me at the time of writing. Karnov however, an Russian-in-circus strongman, appears to have an affinity for this and other bizarre disasters, guaranteed to occur as frequently as wet afternoons in August. The game needs to remind me of wet afternoons too, what levels of the most pathetic graphics you can imagine - more later!

The game opens as Karnov is ripped into existence by a bolt of lightning. Your task is then to battle your

way past marauding creatures of every description, and a few that defy description, collecting useful objects along the way. At the end of each level, you'll need to load the next part from tape - an annoying little nuisance. The rest of the plot is as follows: the Russian village of Chernova (Karnov's reluctant home) has a secret - it's the hiding place for one of the world's most awesome treasures (not Paul Daniels' magic pocket), stolen by Rya, an evil wizard, who left behind a few monsters to punish the villagers for hiding the treasure in the first place! Karnov naturally decides he's the man for the job (naturally after the insurance company), and sets off to find the pieces of the map that will lead him to Rya, the treasure and the game's end.

New to the graphics. The background features are solid but colourful while the moving objects (usually two-dimensional) are well detailed. Where the whole thing falls to pieces is the animation. All the sprites are animated using just two positions, giving them a very jerky disposition, but worse of all is the big black areas behind them! OK, if this was a PlayStation or Spectrum game, this is to be expected, but there's no excuse for sloppy graphics on the C64!

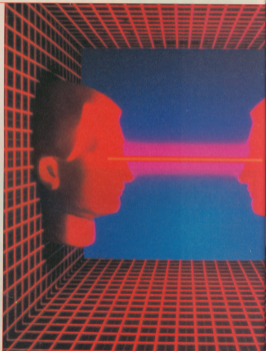
As for the sound, equally pathetic - not even worthy of a VIC 20. Associated clicks and squeaks with a brief smattering of monophonic blasts make this you die. Having said all that, I did find the game challenging and totally addictive, although not addictive enough to persuade me to buy it! P.H.

Touchline:

Title: Karnov. Supplier: Circuit Breaker Software, Grosvenor House, Grosvenor Terrace, Southampton, Hampshire SO9 1PE. Tel: 0703 226094. Machine: C64. Price: 29.99 (C64), 19.99 (Disk).



Communications



s the way forward

We look at the history of computer communications, where it came from, where it is now, and where it is going

By Mary Branscombe

The essential technique of sending data down a telephone line has a history going back 160 years, and some of the common jargon terms that have evolved alongside the more traditional computer buzz words can be traced right back to the days of the Telegraphic transmission systems in the Victorian era!

The earliest form of telegraphy - being the first form of totally electrical and "wireless" communications systems, and probably the earliest form of electrical message sending, was the remote actuation of electric relays to punch holes in a piece of paper. The letters of the alphabet were defined by a stream of these dots each representing a 0 or a 1 and were thus defined between a mark and a space signal.

This goes back as far as 1844, and the first reliable machines to do this were traditionally kept for military purposes, although the government found that the new devices were very useful for transmitting information from one part of the country to another.

The mark and space have been replaced by holes punched into large paper tape, larger holes for mark, and smaller ones for space - this code is called Baudot, and is in common use today in some of the more backward corners of this planet. After this

development, longer distance communications was made possible by synchronising the sending and receiving stations by beginning each letter with a start bit and concluding it with a stop bit, so over the years the standard was to send seven bits of data; the start bit, the stop bit, the code for the letter and, of course the mark.

When after 100 years or so - technology was increasing dramatically - there was a need to access computers remotely. Telegraphy was the obvious way of doing this as the technology could easily be interfaced with the new computer systems and the information was readily translatable. But back in the 1950s computers were not the machines we know today, processing was never expected to come up with instant results; jobs were assembled in batches (hence the term batch processing), fed into the computer by paper tape and then run. Instant collection and processing of data was then considered quite amazing!

So the first major application which computer communications had was to ensure that computers were fed with up to date data, the paper tape was then quite a fast way of inputting large amounts of data and indeed this saved a lot of time with computer-entry staff, who were employed exclusively

to enter data as quickly and as accurately as possible.

Barbaric wasn't it?

Typical communications speeds were at 30 or 75 bits per second and this is NOT the baud rate of a communications system. Baud rate and bits per second are more or less identical at lower communications speeds, but at higher rates bits are transmitted in different ways other than changing the state of a signal from high to low (and vice versa); additional information is transmitted by detecting the phase state of a signal, so 1200 bits per second full duplex is actually achieved by a 600 baud signal using four phase angles (two for each end of the communications stream).

In days gone by there were no friendly on-line systems, no real remote access, terminals were still yet to have screens (let alone faster storage devices) and there were literally thousands of protocols, variants and alternative communications systems cropping up all over the shop.

The Telephone Network

By this time, all sorts of changes were taking place. The Telex and Telegraphy network, originally set up to handle data, were taken over by voice grade circuits (the earliest was Bell's telephone design from 1836). And likewise, communications systems were being adapted to make the most of the telephone system by using audio tones to represent the state of high and low in a pair of twisted wires. Initially this was merely one directional transmission and receiving (half duplex as the terminology had it), but by cleverly using two different signals, full duplex (or two way communication) was possible.

Improved digitizing and sampling systems after the war (primarily) developed for the encryption systems set up (in England) and America to communicate via the war led to faster baud rates - although 96/75 bits/s were possible. The next step was to move to 110bits/s, this soon moved up to 300 bits/s and only recently has the limit been reached for acoustic data transmission along a voice grade telephone line - 2400 bits/s.

Over the years little has actually happened to the way the data is transmitted along a line; the start and stop bits of the baudlet code have been

retained, but instead of a simple 3 bit signal used to transmit the letters of the alphabet, a new system has been employed whereby a 7 bit extended code can be used to transmit the letters of the alphabet. This in fact was as a direct need to transmit data other than letters - I2B moves in fact.

To reduce the number of errors in transmission due to line noise (remember we are still talking about voice grade telephone lines - and think how noisy they can be!), a further bit was added to the stream of data - the parity bit, which adds up all of the bits in the main characters and then, depending on whether the result is odd or even, generates a 0 or a 1.

This system gave birth to the world's most adulated to standard - ASCII, the common name for the alphabet - naturally IBM came along with their own variant, but as the PCs use ASCII instead of their own EBCDIC system (based on mainframes), I think you will agree that IBM are flicking their wounds and trying to keep quiet about this one!

Higher Speeds

The fastest speed possible on an ordinary voice grade telephone line is usually regarded as 1200 bits/s. Beyond this, noise on the line, due to "noise" exchanges and poor cabling has made anything faster very difficult. 2400 bits/s is becoming more common, and indeed the standard transmission speed of TELETEXT services, the high speed version of telex.

Services

Communications these days is easy for the microcomputer user - most computers have a common facility built into the hardware, be it by using the standard RS232C interfaces, or by some such similar serial communications.

Software too abounds for many computers. This is a far cry from the first days of computer communications, when the required "software" for communicating with another computer involved making direct links to a processor's data bus, and then feeding the computer into thinking it was reading from a very fast keyboard!

These days communications

packages are very sophisticated and can handle many different baud rates. They can operate in either full or half duplex modes; they can effectively emulate many different types of terminal, from DEC's VAX and VT100 and the Hamline to the more modern graphics terminals and professional workstations that support graphics.

Software is fast becoming more sophisticated than the hardware that we need to communicate with the telephone line, and the faster they are, the more sophisticated the software needs to be. For example, the new generation of 2400 bits/s modems can now support special hardware encryption devices that ensure your data is unreadable as it goes down the line. The only way you can to retrieve this data is to use a similar modem with the same decryption device at the other end.

Communicating with a micro has become more popular than with, say, a data terminal. The reason is that a computer can be programmed in software which would cost many hundreds of pounds to achieve in hardware! Many of the initial sales of Amstrad PCs was not in the business sector, but to those requiring cheap and reliable terminals - a VT100 terminal can cost anything up to \$1000 - but with a piece of I2S software running on a 6400 machine, you have a terminal and a computer with on-line storage, plenty of buffer memory with all the extras attached.

Communications has really come home for the micro.

Bulletin Boards

For the micro user, a service has sprung up since the mid-1970s called the Bulletin Board, the ideal place equivalent of the bulletin boards you may have at work or in a social club.

Hackers are remarkably solitary creatures, only talking with their own type - where possible, and in this end, the bulletin board is the ideal place to exchange ideas and programming hints and tips, so it is small wonder why these services haven't taken off and spread all over the world.

On a Bulletin Board, you are likely to find a great many messages - have a browse through a few and then you will see that a great many of them are requesting specialised information about the inner workings of a computer of a complex. Occasionally

you will find a message saying something to the effect that they will be starting a BBS (Bulletin Board Service) of their own, so why not ring them up on their number with their modem set to such and such and have a chat!

I am an owner of a small number of wall sized computers - all of which are hooked up to modems at some time or another. I ring up BBS's daily, and this is why I run up 1050 telephone bills a quarter! But the returns are much more valuable than £150. I have downloaded priceless public domain software instructions on how to build my own Winchester drive for my computer using surplus stocks (this one saved me £400!) and the interchange of information is friendly, informal, intelligent and varied.

Every now and then we receive calls from "hackers" asking for a few numbers to try out, and I often oblige with a few of the best known file services. These people often log on for a few minutes - have a look round and then disappear forever - this isn't hacking - and I would contest that breaking into government computer establishments is hacking - that is sheer vandalism (sorry, but my favourite BBS system was brought to its knees by a bunch of nerds). Real hacking is carrying on the information transfer and communication among a close knit group of computer buffs. You don't have to be clever, but if you know something and you have something to say or offer, then welcome to the club!

It is true that BBS's have special user groups, or access to certain parts of the BBS's hard disk that are unobtainable to the usual people, and entry to these is honorary and entirely at the discretion of the Sysop (file manager of the BBS). I am a member of one and I wear it like a medal because it shows to each and every other member of the SIG (Special User Group) that we all have something special to offer each other, be it technical help, knowledge of a language, access to certain information regarding the BBS software (a great deal of which is maintained by ourselves and friends) or simply because you can offer some very useful information about improving services.

In short, accessing a BBS is like starting a group of friends that communicate all over the world - in much the same way that a pen pal does with paper, pen and ink!

Professional Services

For the businessman there are a great many services on offer. Financial data services are available, but they have to compete with the excellent services offered by Firstel and CEEPAK although specialist services offer less general information. Scientific databases offering access to files and programs all around the world are hooked up (in the UK at least) by JANET - the Joint Academic Network, and access to American computer systems at Berkeley, MIT and others are possible by linking up JANET through PSS, British Telecom's Packet Switching System and the international version called IPSS.

University computers are mainly about powerful processing, and it is often a good thing to try and gain access to these systems if you have particularly large applications to run on these machines. Or if you need to access specific languages or information services that support the sciences and engineering facilities around the world.

Polytechnic computers are probably the least interesting computer systems available to the traveller. They offer simple applications, but often offer a back door entry into university computers as they often have a PAD - very useful way of accessing other computers from a remote computer!

PADs are also very good at hiding your tracks if you are a hacker interested in prying into the inner workings of a computer system. This isn't merely done by logging on and typing in a few commands, but by reading up on the operating system manuals, getting information out of a library, and asking your friends on a BBS whether they have had any experiences of such and such.

The spirit of hacking is learning... so go on and learn something!

The Future

Not being particularly good at interpreting the future, I can only state what may happen and not what is going to happen. Computers are going to have to get faster and faster. It is true that even micros that move away from IBM's self-imposed code, are using more and more powerful processors and have more and more memory that needs to be filled up more

and more quickly. After all, if you have ever heard the old adage, "The program expands to fill the available space", then you will no doubt see the futility of accessing a database with a modem running at 300 baud! How are you going to upload IBM's program file using a slow speed?

The way around transmitting large amounts of data down the line is to go digital. By using a special digital data line, you can theoretically go as fast as 300,000 bits/s (that's a far cry from the 56 bits/s 100 years ago!) and even after speeds are possibly by splitting the telephone line so that the Rx line (the receiving line) is on one number, while the Tx line (transmitting) is on the other line!

Large maintenance establishments are using parallel communications over eight telephone lines nowadays, and with the data traffic being measured in gigabytes, the future of parallel communications has to come down to more level in the next few years.

Digital traffic requires special leased lines at the moment, but with the inclusion of X25 (which is just about working - snigger) this leads the way to parallel transmission down varying frequencies, so you could transmit a signal running on a carrier of 600 Hz and 1200 Hz simultaneously, reusing the Rx and the Tx signal at the same time down the same cable with no loss of data.

Light transfer is another option. Using fibre optical cable, modems may soon be just transmitting bursts of light down a cable straight into the telephone network. With this sort of technology, only the speed of the hardware is the limiting factor, so transmitting at 300,000 baud may just be a slow option - certainly we are aiming for data transfer rates that are faster than disk drives by the year 1990.

Of course this will all happen overnight! Data cables using fibre optics are already in wide use worldwide, but to be used over any great distance requires a revolution in manufacturing the optical cable, which is very expensive at the moment.

Thankfully the cable manufacturing companies are developing fibre optics so there will be competing interests artificially keeping the price of this new technology high.

Faster computers, faster communications lines and faster modems, when will it all end? Never, I hope!

In my opinion, the best way to learn machine code programming is to jump straight in the deep end, and start designing the routines yourself!

However, it would be silly not to take advantage of the library routines already resident in your C64, namely the Kernel and Basic interpreter routines. The saving in time and memory should be obvious. After all, why re-invent the wheel?

However, on the other side of the coin, the ROM routines in your C64 have been written specifically to operate the computer without it constantly crashing. Therefore a great number of safety checks have been built into these ROM routines. This has the drawback of making the routines slower and less efficient than those you design yourself.

However, if you ask yourself the question, "Is it really vital for the routine I need to execute that it be second quickest?", then you can make a prudent decision, i.e. whether to write a routine yourself or use its slower counterpart already in ROM.

As a whole, ROM routines, when used correctly, execute quicker than Basic. And this is an advantage you should not forget!

The other disadvantage in using ROM routines is if you want to make your programs compatible with other computers. For example the BBC computer uses the 6502 processor which is completely compatible with 6501 (in fact, the 6501 is the parent of the family of microprocessors of which the 6501 is part of). So at least in theory, any routine written for the 6501 can be executed on any BBC or other compatible machine.

But obviously, even if you don't use any of the Kernel ROM routines, there are still other difficulties to be considered. For example, the screen locations of the BBC computers are at a different place in memory than those of the C64.

Nevertheless, before using ROM routines you should ask yourself, if the program needs to be portable.

Using ROM Routines

My advice is initially to try yourself out with a decent disassembly of the Commodore 64 Kernel and the Basic ROM.

The one which I use like a Bible is called *What's Really Inside The*

Byting into the 6510

ROM routines can save you a lot of work and hassle.

But first you'll have to learn how to use them - read on

By Burghard-Henry Lehmann

Commodore 64 and written by Milan Baturus. It has the advantage of having lots of remarks (some disassemblies I've seen haven't got any remarks!) and it's published by DataCap, 15 Trinius, B-4545 Proser, Belgium. I bought my copy from Buntz.)

Studying such a disassembly thoroughly is a very good way of learning how a professional machine code program is written.

Secondly, it tells you all about the routines in your Commodore ROM. This allows you to use ROM routines not only from the starting points which are listed in many books and magazines (and also in this article), but you can also use ROM routines in your own way, maybe like nobody else has used them before.

Calling a ROM routine

Most ROM routines are called with a JSR instruction, because the

majority of them are subroutines and end with an RTS instruction. This returns the program flow back to your own routine.

Before calling a ROM routine you have to know which registers will be used by it. Then, if you need any of the values later on, you have to make provisions to save them.

As we know there are two ways of using a variable. Transferring it into the accumulator and then pushing the contents of the accumulator onto the machine stack or saving the variable in an address (if possible, a zero page address).

I prefer saving in memory, because it prevents the initial bugs associated with the instructions of the machine stack and I know at all times where everything is. Furthermore, I can recover things from that variable as often as I like without having to worry about pulling pointers.

It is important to be aware, which memory locations are used by the ROM routine you want to call. This

is especially important when using a zero page location. You then have to know the exact calling address of the ROM routine you want to use.

Although this may seem obvious, some ROM routine listing calls give the address of the vector and not the proper start of the routine.

For example, one of the most frequently used ROM routines is called **CHROUT**, which sends an ASCII character contained in the accumulator to the current device (mostly the screen). This can be called from four different points: **\$FDD2**, **\$F1CA**, **\$A847** and **\$E716**.

The start of the routine proper is **\$E716** and is the best one to use if you don't want to waste any time. **\$A847** does some error checks before jumping to the routine proper. **\$F1CA** is where the routine starts when it is called from the vector at **\$0326**.

Finally, **\$FDD2** is the vector on top of the computer which in turn sends the vector at **\$0326**, which again starts the routine proper at **\$F1CA**.

ROM Typewriter

To give you a practical demonstration of how ROM routines can be used I have redesigned our little nonprocessor program to work entirely with ROM routines. This gives you a demonstration of some of the most often used ROM routines of the C64.

Also I have added an extra facility which gives a printout of the current line number and column number at the top of the screen. This makes the program more like a wordprocessor.

Figure 1 shows each ROM routine I have used, in detail and what you have to do to properly call them.

Figure 1 - ROM Routines

Print (\$E716)

Prints any ASCII character onto the screen. Also executes non-printable characters such as delete, cursor movements, cursor home etc. All registers are saved at the start of the routine and recover again at the end, including the content of the accumulator. So you don't have to bother about saving anything.

Pointer (\$A847)

This routine prints a whole string of characters (maximum = 256 characters), including non-printable characters at the current print position. Before entering the low byte of the start of the string is memory into the accumulator and the high byte into the X-register. Zero is used as the end-marker of the string. Don't forget! All registers are corrupted by the routine and have, if necessary, to be saved beforehand.

Prints (\$D8C0)

Prints a 16-bit NUMBER at the current print position whose low byte is contained in the accumulator, while the high byte should be in the X-register. This is used by basic to print line numbers on the screen. All registers are corrupted by this routine.

Plot (\$FDD2)

Plots the current plot position. The current print position is contained in the system variables **\$D9** (column) and **\$D8** (line). If the carry flag is set, transferred from these variables into the X- and the Y-registers. If the carry flag is clear, the values in the X-register (column) and the Y-register (line) are plotted onto the screen, that is, made into the current print position. All registers are corrupted by this routine.

GetIn (\$F1CA)

Gets the value of the last key out of the keyboard buffer and loads it into the accumulator. If no key has been pressed, 0 is loaded into the accumulator. All registers are corrupted by this routine.

Clr (\$E544)

Clears the whole Commodore screen and places the current print position to the top of the screen. All registers are corrupted by this routine.

Let's now look at the most important points of the program which you'll find listed as always at the back of the magazine.

In lines 340-350 I turn the system

cursor on by loading the system variable **\$CC** with 8. If it should contain a number larger than 0, the cursor would be turned off.

The main loop of the program consists of testing the keyboard, exiting from the routine if F1 has been pressed, printing the ASCII character on the screen or executing a non-printable character, such as delete, updating the line number if a new line has been started and, finally, updating the column number.

Since, as far as the operating system is concerned, each line consists of 80 characters, even though the screen can only portray 40 character lines, we have to make an adjustment in the next 40 column line before character 39 has been printed. This is done in lines 370-450.

The ASCII character contained in the accumulator is saved on the machine stack. Then the current row as contained in system variable **\$D6** is incremented and the beginning of the line plotted back to column zero. Finally, the ASCII character is recovered into the accumulator. Then the ASCII character is printed.

After this the current line number, contained in system variable **\$D5**, is saved in **\$31** and the current column number, contained in system variable **\$D3**, is saved in **\$32**. This is because the print positions have to be repositioned in order to print the line number, and column number at the top of the screen.

Next the current line number is printed at the top of the screen. Since the header takes up five lines, 5 is subtracted from the current row as contained in **\$D6** (lines 880-930). Next the current column number is printed at the top of the screen.

If a new line has been started, the former column number is blanked out with two spaces, otherwise one would get columns 11, instead of column 1 (lines 940-990).

Finally, the current print position is recovered from **\$31** and **\$32** and reported. Then the routine loops back to get the next keypress.

The program as it stands has several drawbacks, such as the deletion routine not working properly from one line to the former. But I'd hope all you people out there will get busy mending these insufficiencies and make a fully working program out of it.

See listings on page 23

DATEL ELECTRONICS



3 SLOT MOTHERBOARD

Now you can test on your Expansion Port!

- ▶ 3 expansion slots
- ▶ 16MB memory bus
- ▶ 2MB cache memory
- ▶ 16MB DRAM
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$16.99**

DATA RECORDER

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$24.99**



SMART CART

Now you can have an IBM or PC cartridge that you program like IBM then use like IBM!

- ▶ IBM or PC compatible
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**

- ▶ IBM or PC compatible
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$29.99**

- ▶ IBM or PC compatible
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**

ONLY **\$29.99**

SMARTCART UTILITIES

SmartCart Utilities are the only utilities that can be used on both IBM and PC compatible systems. They are the only utilities that can be used on both IBM and PC compatible systems.

ONLY **\$14.99**

- ▶ IBM or PC compatible
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**

- ▶ IBM or PC compatible
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**



PRINTER IV

Now you can test your PRINTER IV on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$19.99**

ONLY **\$19.99**

ONLY **\$19.99**



BLAZING PADDLES

A complete light pen/graphics accelerator package.

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$24.99**

ONLY **\$12.99**



TOOLKIT IV

The ultimate disk tool for the IBM/PC

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$9.99**

RAM DISK

Now you can test your RAM DISK on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$9.99**

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$9.99**



TURBO ROM II

Now you can test your TURBO ROM II on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**

ONLY **\$14.99**

ONLY **\$14.99**



DIGITAL SOUND SAMPLER

Now you can test your DIGITAL SOUND SAMPLER on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$24.99**

ONLY **\$24.99**

ONLY **\$24.99**

ONLY **\$24.99**



COM-DRUM

Now you can test your COM-DRUM on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ONLY **\$29.99**

ROBOTEK 64

Now you can test your ROBOTEK 64 on your Expansion Port!

- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$14.99**

Model 8 Robot Control made easy

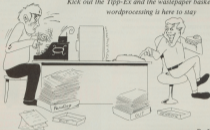
- ▶ 16MB memory
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache
- ▶ 16MB cache

ONLY **\$39.99**

ONLY **\$39.99**

The Moving Cursor Writes

*Kick out the Tipp-Ex and the wastepaper basket,
wordprocessing is here to stay*



By Eric Doyle

In the Tenth Century, wordprocessing was a thing of fables. In a review for the *Kahayan*, Umar Khayyam wrote, "The moving finger writes; and, having writ, moves on; nor all thy piety nor wilt shall here it back to cancel half a line, nor all thy tears wash out a word of it." Not very user friendly. When Edward Fitzgerald translated the *Kahayan* into florid verse, the age of the typewriter was dawning and the writing was on the wall for the moving finger system.

Today the wordprocessor has supplanted the typewriter and few authors pained platters to beat their weary prose into a semblance of literary worth. The wordpro revolution means that corrections, updates and re-arrangements can be made with ease and there are few people who would not benefit from its use.

Even for a humble job application, the power of the wordpro can be harnessed to produce flawless text and the days are numbered for liquid paper correction fluid. Even though the use

of the wordpro is so well accepted, there is still the need to show what is essential, what is desirable and which features are rarely needed.

What is a Wordpro?

A wordprocessor is basically like a typewriter. It allows text to be typed in at the keyboard of a computer to give a hardcopy on paper. Where a wordprocessor differs lies in the fact that it acts as an interface between the keyboard and paper, expanding the possibilities far beyond a typewriter's wildest dreams.

The essential features of a wordprocessor can be broken down into three categories. There is the actual entry of text, the ability to specify layout, and the facility to access a printer and storage device. In addition to these essentials, the use of a spelling checker can make life easier and, for a very few people, integral modern software can read the completed text to any printer in the world.

Editing Facilities

The basic requirements of text editing is to be able to type in characters at a comfortable speed, see the text displayed on the screen, and to correct errors before the text is printed. All of this can be achieved with many of the microchip typewriters currently available but a wordpro can do more.

Nobody's perfect so the wordpro should allow the user to scan through the text making deletions and insertions. Major rewrites and the addition of large blocks of text ought to be possible.

Sometimes this modification process means that the whole structure of the document needs to be changed. In the bad old days this would mean that yet another ball of paper would whiz towards the wastepaper bin. With a wordprocessor it is usually possible to mark a block of text and move it from one place to another or to repeat it if it is a clause that is used several times. This saves a lot of ripping and consequently saves time and tribulations.

excited rather than being displayed, so the raw text is revealed:

This is an example of an embedded command.

The theory only proves true to a limited degree because WYSIWYG is not available. WYSIWYG simply means What You See Is What You Get, and refers to a screen display which looks exactly the same as the printed-out page. For many reasons relating to efficient use of memory and character definition, it is not possible to utilize WYSIWYG displays on the Commodore. Instead a program may highlight the characters which will receive special annotation by reversing out, the display or by some other device.

Stylish Characters

Modern printers allow various styles of characters to be used within a single document. Enlarged characters, italics, N.L.O. different typefaces should all be readily accessible to the user.

The overall appearance of the document can be affected by the effect known as justification. This means that characters can be aligned to the left margins leaving the right edge looking ragged as each line has a different number of characters. Right justification can also be employed which means that extra spaces are added to successive lines to make all the lines the same width so that a uniform block of text appears. For Commodore's pages right and left justified. Glance through a few paragraphs of this text and you can see the effect that this has on individual lines.

If the line width is short, the justification process can result in two words being separated on one line by several spaces if the third word is too long to fit beside them. The way that the third word is pushed onto the next line is known as wrap-around and avoids words being split in peculiar ways. Let's see what happens when I give the typewriter a headache by using a very long word.

A long word like antidiagonalization causes problems on a line which has a maximum character count of about 80 characters per line. There are two ways in which this can be handled by the typewriter. First, after the four short

words a new line is started for the long word. This means that sixteen characters are ranged across a line which normally contains twice as many characters. The result is a gappy line. The second method of treating this situation is to break the longer word but there is a convention to be observed. Words should only be broken down into syllables. The computer program cannot tell where syllables occur so some programs have a device known as a 'soft hyphen'.

Soft hyphens are characters which may or may not be displayed according to a single rule. Take the word 'somewhere' as an example.

If a soft hyphen is used the word is written as some-where. When the word some-where appears at the end of a line it splits into its hyphenated form. If there's enough room, somewhere isn't split and the soft hyphen is ignored by the program.

The choice of whether the soft hyphen facility is useful or not depends on the user. To ensure every word in its hyphenated form would be a hind but to be able to use it when previewing the page before printing can avoid problems occurring if changes are made elsewhere in a paragraph which pull the word back into the middle of the line. A hard hyphen would remain but the soft one disappears.

When writing a letter, my text looks just as it does in this article; but when submitting this article my original text was double spaced. This means that instead of having a single carriage return at the end of the line, two returns were sent leaving a blank line between each printed line. This is done so that there's plenty of space for marking up special features such as italics or for correcting grammar and misspelt words.

Any wordprocessor for my use must have the facility to double space lines and the one I use can actually triple space as well.

On the monitor each character is given equal spacing. Some characters like the letter 'm' fill the space comfortably but letters like 'Y' and 'T' leave large gaps between themselves and neighbouring characters. Examine the text you're reading and you should observe that the spaces between letters are approximately the same on any given line. This means that characters next to the letter 'Y' have been automatically moved closer together. This is known as proportional spacing.

The better printers can

proportionally space lines so the best wordprocessors are given commands which allow access to this facility.

Patterns on a Page

The main features necessary to set up a page for print are the margins. These not only appear at the sides but also at the top and bottom. To add to this the page may not be A4 in height. So there are five parameters which must be expressed: page length, left and right margins, and the bottom margin.

Left and right margins determine the number of characters per line. The normal value is 80 but there may be occasions when a wide platen printer is used and 132 characters or more can be fitted onto a line. On other occasions A5 paper may be required so a much smaller page width might be required.

The ability to specify long and short line widths is not the only feature required because the preview option needs to be able to range across this number of characters. If it only has a maximum of 80 characters, the preview screen is useless for wider documents.

Once a line width is set, it is far more interesting if the document has indented sections to highlight particular features. Some wordprocs allow temporary margins to be set. This could be done by simply inserting spaces at the beginning of each line but these spaces are saved when the program is stored and means less space for the document.

There are also occasions when each paragraph may have a heading, correctly called a sub-heading. One way of doing this is to indent the following text to highlight each sub-head. To insert spaces on already every line would be boring especially when the power of the computer can be called upon to do this dull task for you.

Longer documents need page numbers and for some users each page must have an identifying piece of text across the top or bottom of the page. It's a bind if you have to add this text each time a new page is started or finished. Even if the user accepts this limitation, the real problem becomes clear when a block of text is added to the text later which completely covers up the start and end of the pages following. Each page would

have to be laboriously altered. For this reason a header and footer facility should be available.

The header and footer is a command which allows a string of characters to be designated at the beginning of the text. This string will then be repeated at the top or bottom of each page and the facility normally allows a number character to be entered anywhere along the line. As each new page begins the page number is automatically incremented and inserted.

The final essential is a centering facility. To find the correct position for the start of a centered heading means counting the number of characters in the heading, dividing it by two and then subtracting this value from the maximum number of characters per line. A centering facility does this automatically but takes care when using expanded characters or proportional spacing because I haven't found a program which copes adequately with these situations.

Hard Links

Some documents are longer than the available memory allows. To create longer texts a linker can be useful. This permits a line to be added to a file which automatically searches for, loads and prints the next document in the chain.

When saving a document, there is a need for a save and replace function. The great disadvantage of a wordpro is that it stores the text as volatile electrical impulses which can be wiped out easily if the power supply is interrupted. Regular saving of the text is recommended if disaster is to be avoided. If the text is saved under a new name each time, the disk soon fills up with redundant files.

The alternative is to use the save and replace function to store an unmodified document under the same name as before. The disadvantage with this is that the program often uses the built-in save and replace function of the disk drive. This is a bugged feature and can result in the loss of a file or the corruption of another file.

It's difficult to know if a wordpro can save and replace or if it screens the file before doing a normal save. It is better to use two filenames and address each an alternative save.

The computer has a limited memory which is further reduced by

the inclusion of the program itself. This means that computer storage space is at a premium and often a document will overflow the allotted memory space. Longer documents have to be saved over several files and a linker facility to chain them together makes life a bit easier. A special code inserted at the end of each document file combines the files into a long chain so that when each one finishes printing, the next part loads and prints automatically.

Special facilities to call up the directory and to send disk commands are useful especially when formatting a new disk for file storage. Equally, the ability to send special codes to the printer lengthens the life of a wordpro because printer technology is advancing rapidly with new commands and features being added with each new machine.

Printers fall into various categories. Commodore has its own codes, Epson have a different set, compatibles may vary from the Epson standard in small but important ways, and there are still many manufacturers doing their own thing where codes are concerned.

There are three ways of combating the diversity needed. The onwards way out is to support one machine type and list names of the others basic with the problem by sending out character-string commands themselves or relying on an external interface to cope with the problem. The workman system is to incorporate an interpreter which asks for the information for creating special effects. The information is then stored as a block in memory which asks for the information for creating special effects. This information is then stored as a block in memory which can be saved and loaded when required. The heroic, idiot-proof system is my favourite where a format is added to the program asking what type of printer is being used. According to the response, a ready-made file is loaded from the master disk and the system is up and running from day one. This system also incorporates the work-man-like facility for save, create species of printers.

The more expensive wordprocs usually allow user defined codes to be used. Special keys are reserved and these can be pressed either to send extra codes to printers or to issue strings of wordprocessor commands. In all cases, colours of screens, header and characters can be customized and printers should be easily altered for

those times when some error becomes obvious at a late stage.

Extra Powers

Many wordprocs are now accompanied by a spelling checker which will sift through the text and query any word which it doesn't recognize. The best checker that I've used is the one accompanying Legatum's Writer 1285. It is both fast and has a good initial vocabulary which can be edited and expanded. Some spelling checkers are laboriously slow and inflexible incorporating words which, though difficult to spell, are hardly ever used in common parlance.

A spell checker is a utility and not a magic wand. For example, I seem to have developed a nasty habit of writing 'their' when I mean 'there'. I know the difference but just can't seem to break the habit. No spelling checker will correct grammatical or syntactical errors so a command of the English language is still a requirement of the user.

For journalists like myself, a word or character count is essential. Many of the programs go beyond this and give paragraph and sentence counts.

One feature which can be useful if your work requires the manipulation of figures is a built-in calculator. Normally these are limited in accuracy to two decimal places for financial calculations.

The final facility is one which is purely a business or club feature. When a circular is to be sent to some set of mail slots, it can be extremely time consuming changing addresses and names to personalize these communications. A mail merge function allows keywords from a specially constructed file to be substituted where indicated in the text allowing a high degree of customizing to take place.

Most people will at some time be subject to the cunningly phrased Fowler's Digest special offer using copy phrases like 'the Doyle family has been specially selected', or 'imagine the expression on the faces of your friends, Eric'. Look carefully at the construction of the message and you'll soon realize that you're a victim of a mail slot. How many children receive special offers for themselves and their wives or husbands! How many 18-year olds have received the chance to win a car of their own. Computer intelligence is limited!

The mail merge is almost a standard in small but important ways, but some also link through to databases which saves the unattractive prospect of creating two-inch files. The C128 version of SuperScript can reside in memory alongside SuperBase. For mail merge processing this has a distinct advantage.

What to Look For

When buying a wordpro the first consideration is to sit down and work out which facilities you need, anticipate those which may be of use in the future and then start ploughing through the range to find a system that suits your needs and your pocket.

Just because the program has everything that you need doesn't mean that it's definitely the one for you. If possible ask for a demonstration because although most wordpros offer similar basic features the way that the end effect is achieved may not be appealing.

Check that the program supports the printer that you intend to use. A bad selection here would be an expensive error.

Some wordpros allow files to be saved as ASCII files. The advantage of these programs is that commonly used files can be transferred to a new wordpro if a change in system is considered and files can be transmitted to any computer through a modem.

If modem linking is an essential feature, PaperPit 128 has built-in terminal software. A comprehensive program such as this means that program hopping between wordpro and terminal can be avoided.

Lateness should be avoided. It's easy to say that buying the wordpro that was everything, will cover all future developments but the manual will also be complex. This makes it difficult to sift out the essential features that you require.

For most wordpros the manual is an essential companion. Eventually the commonly used codes will become familiar but the lesser used commands will have to be looked up. A manual which has a printed guide to commands can be a bonus. SuperScript has a very useful menu display which can be called up to readily access all of the commonly used features. A program with a Help screen can also be a good buy.

SuperScript

Supplier: Precision Software, 6 Park Terrace, Worcester Park, Surrey KT4 1JZ.

Price: C64 disk £24.95
C128 disk £28.95

Comments: This is my favourite because it's the most user-friendly system. It has a wide range of printer files, a reasonable spelling checker and a calculator. The readily accessible command menu is brilliant. C128 version has 40/80 columns and can reside in memory with SuperBase.

Easy Script

Supplier: Commodore Business Machines, Commodore House, The Switchback, Gardner Road, Maidenhead, Berkshire SL6 7XA
Price: C64 disk £30.50 (was a freebie with the 1341)

Comments: This is really a less user friendly SuperScript without the menu or the spelling checker. The price makes SuperScript the better buy.

Writer 128

Supplier: Logotron, s/o Vantage Services, 13 Denington Road, Wellingborough, Northants NN8 3LR
Price: C64 disk £12.85

Comments: Don't be fooled by the price, this is an excellent new wordpro with the best spelling checker that I've seen. The features are worthy of a package costing twice as much.

PaperPit

Supplier: Was Aristonsoft, can still be found in shops.

Price: C64 and C128 disk £44.95

Comments: Undoubtedly an excellent package but inhibited by price. The C128 version has 80 column screen capability and modem terminal. Excellent 15,000 word Spell Checker is not available on all versions so buy with care.

Tasword

Supplier: Tasman
Price: C64 disk £14.95

Comments: The only true 80 column C64 wordpro but otherwise nothing startling. A good work horse.

Word Perfect

Supplier: Softsoft) ASI, Software, Winchester House, Canning Road, Walsleywood, Middlesbrough TS1

Price: C64 Disk £19.95, cassette £12.95
C18 disk £14.95, cassette £12.95

Comments: Word Perfect is a good basic wordpro and has the distinction of being the only one available for the C16. The printer interface is set up for Commodore only but special commands can be sent to other types of printer but no save facility is available so automatically saved those commands when the wordpro is used again.

Viva Words/ Viva Classic

Supplier: Calco Software, Lubricant House, Kingston Hill, Surrey KT2 5QT

Price: Write C64 disk £10.95 cartridge £48.95

Classic: C128 disk £50.95

Comments: The editor's favourite. Phenomenally expensive yuppie C128 package but has a full range of facilities. The Classic is only available for the 80-column mode. Help screens may be loaded from disk rather than referring to the manual.

Mini Office II

Supplier: Database, Europa House, 68 Chester Road, Hazel Grove, Stockport, SK7 5NY

Price: £16.95 (C64) £18.95 (Disk)

Comments: This is a fully integrated system incorporating wordpro, database, spreadsheet, Business graphics, forms pack and label printer. Its sales have made it a blockbuster. The wordpro is surprisingly good for such a low cost system. If a full pack is what you need, you could buy worse at three times the price.

Car's Paste

Supplier: Was Aristonsoft, can still be found in shops.

Price: C64 disk £21.95

Comments: Extremely user friendly but basic package. Main feature is the text transfer referred to in its title.

STRUCTURED PROGRAM *Design*

Although the advantages of structured programming are often mentioned, there seems little information which explains how to achieve it! So for those of you who would like to improve the standard of your programs - read on!

By Derek Barrett

Traditionally, and I am the first to admit to having done this, many computer programmers write their programs by sitting at the keyboard and just typing away. If a new idea for a routine comes into their head they slip in a GOTO and add the new piece at the end. This leads to 'Spaghetti' programs where GOTOs are sprinkled randomly throughout the code. Anyone who tries to read such a program often ends up in a mess trying to unscramble it.

This is not to say that the orderly use of GOTOs is wrong, sometimes the only way round a problem is to use them. It is only when they are used indiscriminately that they lead to confusion.

If you're used to designing your programs by flowcharts, you'll know that even in the early stages, you must

think in terms of the smallest detail of code that will be needed in the final program. Constant redefinition and refinement is needed to the design before it can be run.

The technique described in this article allows you to start from simple 'root' ideas and refine each step as you progress through the design phase. The final design will look rather like an inverted tree, with the branches forming the paths of the design. That is why the technique is called *top down design*.

Each branch can be designed independently and in any order. Often the easiest solution to a problem is to start by thinking of how to produce the output that will be required and then work backwards through the steps that are needed to produce that output.

Language Independent Code

An advantage of this method is that the designs you produce are completely portable and can be coded into any language on any computer. So whether you program in Basic, Pascal, Fortran or even assembler, the same rules apply. You can also use the techniques for everyday decisions, totally unrelated to computer programming.

Think about this - you have just produced a brilliant program. A friend sees it and would like to run it on his machine. However his computer is not compatible, with a different dialect of Basic, and anyway he prefers to program in Pascal as he uses that at work. With top down programming there is no problem. Just give him a copy of the design and he can code it himself. This is much better than trying to redesign the program from your scrambled Basic.

The beauty of top down design is that no complicated new terminology or diagrams are used. If you can draw a simple rectangle you can start to design well-structured programs.

Right, let's get on with it. There are only three elements to structured design:

- Sequences
- Selection
- Repetition

Sequences

A sequence is the basic building block of top down design; each box indicates an action that will be carried out. The top box indicates the program title and is sub-divided into lower boxes which indicate the processes needed for the program to perform its function.

If you look at Figure 1 you'll see A is a sequence of B to F which are the main stages needed in a program to compare the latest reading of a car mileage with a previous reading and printing the difference.

At this stage of the design, the program has been broken down into manageable chunks with no attempt to define the detail that will be needed to code the program. Imagine you are writing a book. Box A is the book title and B to F are the chapter headings. The contents of these chapters will be written later. Similarly boxes B to F will be broken down as the design

continues until the final solution is reached. Each box contains one action only and the temptation to group ideas into one box must be avoided as it will lead to confusion.

Although Figure 1 is shown as being self-contained it could easily be part of a much larger program, for perhaps, a vehicle fleet servicing system. Very often a large program is best tackled by splitting it into sections that can be considered separately.

For those of you more familiar to reading flowcharts the position of the linking lines from A to B, A to C etc. may seem strange, but you will see the reason for this when we expand the ideas further.

Selection

This is the second of the structures to be learnt and has the same function as IF...THEN in Basic. The only addition to our simple box is the addition of an 'n' in the upper right of the box.

Figure 2 shows a simple example of IF...THEN. You'll see that the actions are a sub-division of box A, which serves as a 'Dummy' box (i.e., it should be used as a comment line in the final code to aid legibility).

Box B is the IF part of the statement and Box C contains a line to show that nothing is to be done if the conditions of the IF statement is not met.

Box D forms the THEN part of the statement. You will notice that the instructions are written in pseudo code to make it easier to use the features of the language you will eventually code in.

Figure 3 expands this example to illustrate the IF...THEN...ELSE construction found in most languages. In this case there is an alternate action to be taken, with the ELSE part of the statement in box E. With this type of construction, flow is only allowed to go from A to B or A to C.

In other words, if the condition of B is TRUE, program flow will continue at D (C and E will not be carried out). Conversely if B is FALSE C and E will be carried out. You can see from this that C in Figure 3 really means ELSE DO NOTHING. A is a selection of EITHER B or C but not both. Figure 4 extends this to form the CASE statement. In this case flow will go to one only of B,C,D or E.

Multiple conditions can also be

shown in the design (Figure 5) but you must be careful as it is easy to design a condition that can never be met, in which case the program could 'hang up' at that point. Box C can be coded using the ELSE IF construction if your language permits it. It is probably better, at least until you are confident in using the technique, to use the nested condition statements (Figure 6.)

Boxes F, J and L must be included to allow flow to continue if the conditions are not met. Incidentally I have used letters adjacent to the boxes purely for ease when explaining the principle. Some software designers use nested numbers instead and I tend not to bother at all. It is entirely up to you.

Walking Through The Design

Before we go any further it is essential that you check the design thoroughly before you start to code it. The usual method is to 'walk' through the design, using dummy data, to ensure that it has the desired result and that no hidden traps have been put in.

The rule for reading a top down design is to start at the top and work down each branch, working from the left to the right.

Draw up a table of dummy data that will allow all the conditions of the stage to be tested. Now 'walk' through the design using this data to see where you end up.

We have entered the design at A, which is a sequence of B,C and D. Our tests are only concerned with the detail for C at this stage.

```
Int test, Tank no.=3, Batch Total=200,
Total Quantity < MAX, (MAX is
defined elsewhere in the program.)
```

```
If is TRUE to go on to G
D is also TRUE so carry out Action
I at J.
```

We have now reached the end of a 'branch' so we must go back to E. As this is an IF...THEN condition we cannot proceed to H or I as we end back to C which is also an IF...THEN so we go on to D. This is vital to the concept of Top Down and before we it is easier to do than describe.

2nd walk through

```
Tank no.=3, Batch Total=150, Total
```

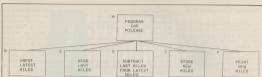


Figure 1 - example of sequence construction



Figure 2 - selection with single choice



Figure 3 - selection with alternative action

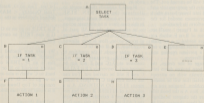


Figure 4 - structure of CASE statement

Quantity < Max.

E is TRUE so go on to G
 G is FALSE so go to H
 H is TRUE so go to K

K is TRUE so do Action J at M then
 Action J at N. I have added a sequence
 to show the possible lines of design. J, M
 and N would be sub-divided in a real
 program.

3rd walk

Task no.=3, Batch Total=266, Total
 Quantity < Max.

E is TRUE so go to G
 G is FALSE so go to H
 H is FALSE so go to I
 I has no action so carry on at D.

4th walk

Task no.=4, Batch Total=390, Total
 Quantity > Max.

E is FALSE so go to F
 No action at F so carry on at D.

Get the idea now? But was it the result
 you expected? If it was you have
 succeeded in your design. If it wasn't
 then go back and re-design the bits
 that didn't work. A lot of refining is
 often needed at this stage, but the more
 time spent getting the design right leads
 to less frustration when you run your
 masterpiece and find that it doesn't
 work as you expected, or, even worse,
 doesn't work at all.

Repetition

On to the last of the three elements.
 That of repetition, also known as
 iteration. You've all used the familiar
 FOR...NEXT construction of Basic,
 and possibly the allied
 REPEAT...UNTIL or WHILE...DO
 of more advanced languages. These are
 all forms of repetition. Repetition is
 merely the forming of a loop to do
 the same action many times over.

Once again we will use the familiar
 rectangle but this time add an asterisk
 (*) in the upper right of the box that
 is used as the control element.

I'm afraid space doesn't allow us
 to run the whole article, and so look
 out for the second installment in the
 near future.

See listings on page 59



Figure 1 - selection with multiple conditions

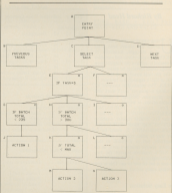


Figure 2 - alternate structure for Figure 1

Educating Sidney



CAT
DOG

*SID has never been the friendliest of characters,
teaching him to be a little more sociable can't hurt.....*

By Richard Head

Using sound in your Basic programs has always been a bit of a struggle, hundreds of POKES, data statements and variables are needed to achieve the simplest beep - things could certainly be easier. Due to the physical structure of the chip, most of SID's registers are immune to PEERS. This means that accepted bit switching techniques won't work and your programs become a mess of variable arrays, or worse!

For example, POKES4285, (PEEK(5400)AND2-5) will set bit 5 of the VIC control register, enabling VIC's bit map mode. POKES4285, (PEEK(5400)AND2-5) would be an easy way to select voice 2's sawtooth waveform, but as a PEEK to that location always returns a zero, so do all SID's registers (except the two paddle registers at 5407 and 5408), it's not a lot of use to us. This is one problem easily solved....

An Easy Solution

Enter SIDREAD. When initialised, this short machine code program

Table a - Individual Voice Controls

Register No.	v1	v2	v3	Description
0	7	14		Frequency-low byte
1	8	15		Frequency-high byte
2	8	16		Pulse Width-low byte
3	10	17		Pulse Width-high byte(s 0-15)
4	11	18		Control Register
5	12	19		Attack / Decay control
6	13	20		Sustain / Release control

Table b - Overall Controls

Register No.	Description
21	Filter cutoff/low nybble (0-7)
22	Filter cutoff/high byte
23	Filter input byte
24	Filter mask / volume byte

Table c - Other SID facilities

Register No.	Description
25	Paddle input 1
26	Paddle input 2
27	Random number generator
28	Voice 3 envelope output

wedges itself into the 6th interrupt system. All it does is set up a dummy SID in RAM, copying its contents into the real SID every 60th of a second. Since the program is installed, any value poked into the RAM table is automatically copied into its corresponding SID register. Thus you can PEEK the RAM table, perform your math on the value, and POKE it back again. The earlier example will now work, since the address has been changed to point to the RAM table.

The code was originally written to sit at \$C000-\$D152, and load in direct from disk, but to make things easier still, I've produced a Basic generator program that will install a working version of the code anywhere in RAM, and save a copy to disk. SIDREAD is the program generator, and before you run it, make sure you set the base address in line 60 (it defaults to 49152 (\$C000)). Beware though, if you try to put the code under the ROM, or any reserved RAM area (zero page, screen RAM etc), it won't work.

When you Run the generator, the start and table address are displayed and it's probably worthwhile writing them down. Disk users get the option here of saving a working copy of the code to disk, just follow the screen prompts. To load the code back again, use LOAD"filenames".8.

With the code in place, SYS(base address) will clear SID and initialize the table. Any poke to the table will automatically be copied to the corresponding SID register. Should you happen to hit RUNSTOP/RESTORE, you will need to re-initialize the table before using it again, use either SYS(base address) or SYS(base address + 10). The latter will re-initialize without clearing SID.

Getting it all in

SIDTEST is a demo program, load and run it, and nothing happens! This is because the program relies on PEEKING SID. Next, load and run the generator program, and re-load SIDTEST. Now change line 10 to read SID=(base+48) and RUN (if SIDREAD was Run unaltered, (base+49) should equal 49201). If all has gone well, you'll know about it!

Once you've created a working copy of the code at a suitable address, you're not restricted to using it in Basic programs; make use of it in your machine code programs as well! It



Figure 1 - SIDREAD cycle

your machine program also uses interrupts - no problem. SIDREAD automatically uses the contents of \$B000 and \$0015 for its jump back to the normal interrupt system, so all you need to do is initialize any other interrupt routines before initializing SIDREAD.

Using the Tables

The tables are arranged in three groups. Tables (a), (b), and (c) map

out the registers in their various groups (individual voice channels, control controls and output registers), while tables (d) to (h) show what each section does within certain registers. Finally, table (i) shows the ranges that SID's functions can be set to.

To set any single bit in a control register to a '1', use this simple formula (in conjunction with the SIDREAD program):

POKE BASE + REG, PEEK (BASE+REG) AND VALUE

Table d - Control byte: Registers 4, 11, 18

Bit Value	Bit no	Function
128	7	Random noise
64	6	Pulse waveform
32	5	Sawtooth waveform
16	4	Triangle waveform
8	3	Test bit
4	2	Ring modulation
2	1	Synchronous
1	0	Gate

: 1 = ON
: 0 = OFF
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 1 = ON
: 0 = start Release

Table e - Attack/Decay: Registers 5, 12, 19

Value	Bit nos	Function	Range
240	7-4	Attack Time	0-15
15	3-0	Decay Time	0-15

Table f - Sustain/Release: Registers 6, 13, 20

Value	Bit nos	Function	Range
240	7-4	Sustain Level	0-15
15	3-0	Release Time	0-15

and to turn a bit off, use

POKE BASE + REG,PEEK (BASE + REG) OR 255-VALUE

Where BASE is the start of the RAM table, REG is the register number and VALUE is the binary 'weight' of the bit in that register. (The figure in the 'value' column of the table.)

For example: to select voice 7s sinusoidal waveform, use:

POKE BASE+11,PEEK(BASE+11)AND 32

Or to set voice 1's output to bypass the filter, use:

POKE BASE+23, PEEK (BASE+23) OR 255-1

Figure 1 is a graph of volume against time, showing how SID's envelope generator works. The volume or 'amplitude' envelope of a sound is one of the primary factors deciding what the sound will 'sound' like. Try experimenting with different values and note the differences.

See *Settings* on page 73

Table g - Filter input byte: Register 23

Value	Bit no.	Function
240	7-4	Filter resonance : 0-15
8	3	Filter control : 1 = off
4	2	Filter voice 3 : 1 = on
2	1	Filter voice 2 : 1 = on
1	0	Filter voice 1 : 1 = on

Table h - Filter mode/volume byte: Register 24

Value	Bit no.	Function
128	7	Voice 3 output : 1 = off
64	6	High pass mode : 1 = on
32	5	Band pass mode : 1 = on
16	4	Low pass mode : 1 = on
15	3-0	Master volume : 0-15

Table i - SID Function Ranges

Function	Range	Value
Oscillator frequency	0-6K Hz	0-65535
Attack time	2ms-6s	0-15
Decay time	6ms-24s	0-15
Sustain level	0-peak volume	0-15
Release time	6ms-24s	0-15
Pulse width	0-100%	0-65535
Filter range	30Hz-12KHz	0-2547

Mail Order
For all your
software needs

Cottage Software

All prices
include VAT
and delivery

OUR AIM IS TO SUPPLY YOU, THE CUSTOMER, WITH THE BEST VALUE FOR CREDIT

GAMES STRATEGY

	Disc	Cass		Disc	Cass		Disc	Cass
Airborne Ranger	14.99	7.99	Storm Ranger	9.99	5.99	Yankee	14.99	7.99
Apple II	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Armadillo (Macintosh)	9.99	5.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Armadillo (PC)	9.99	5.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain II	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain III	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain IV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain V	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain VI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain VII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain VIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain IX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain X	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XIV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XVI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XVII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XVIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XIX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXIV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXVI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXVII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXVIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXIX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXIV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXVI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXVII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXVIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XXXIX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XL	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLIV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLV	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLVI	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLVII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLVIII	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain XLIX	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99
Battle of Britain L	14.99	7.99	Walt for Red October	14.99	7.99	Yankee	14.99	7.99

Just because you don't see it — doesn't mean we haven't got it — For full details, send a stamped, addressed envelope

Please make cheques or postal orders payable to:

Cottage Software

P.O. Box No. 8, Shaw Old Road Telephone 0766 44336

64/128

64/128



PATTON VS ROMMEL

The Normandy landings produced many heroic anecdotes, however it didn't provide a direct confrontation between Patton (who arrived one week after the landings) and Rommel (who was injured one week before). Electronic Arts, not letting facts get in the way of a good simulation, has produced a game that matches these two in a head to head fight.

The game was written by Chris Crawford, who wrote the all time classic *Eastern Front*, and provides the same simple but realistic game mechanics and a tough computer opponent.

Unlike other wargames where the object is to defeat the enemy forces, *Patton vs Rommel*, is a battle for territory in the shape of towns. The allied forces have just a few days to break through the German lines and capture as many towns as possible by moving through them. The German player must halt their progress and maintain their thin line of defence for as long as possible.

The state of the game, which is played in turns, is represented as a victory point total which will begin and stay negative until the allies take towns, when it will gradually increase until a positive score announces an allied victory. A German player must keep the score negative to win the game either by halting the allied advance or by retaking captured towns.

The screen display is entirely black and white, which is never explained except it may be to mirror the *Machine*. Though why ignoring the CMC's excellent graphic capabilities is beyond me.

Most of the screen is filled by a fraction of the battle map and can be scrolled across it by moving a window over the map icon. The dark areas represent high ground and slow down progress that enters it. Conversely, the white roads connecting the towns form the quickest way to travel.

There are just two types of units in the game infantry

and tanks although these symbols actually represent more accurate forces, such as supporting artillery which are displayed whenever a unit is selected. By clicking other icons the units can be displayed instead as arrows showing which way they are facing, circles representing the total and actual strengths and by dots or widening cracks showing the damage they have sustained. A badly cracked unit could mean it is met a determined and even weaker opponent.

To issue an order to a unit is a simple case of selecting (clicking) it and then pointing to where you want it to go. In the basic game it will try and move there as quickly as it can but will stop and fight any enemy units it gets near. In the intermediate level you must give more specific orders by clicking icons that command a unit to turn clockwise or anti-clockwise and move forward in one of seven modes which range from the rapid movement, but open to attack road to the artillery assault of static attack mode.

Whatever your orders the computer will then carry them out before either Patton or Rommel, and where appropriate will comment on your performance and may even pass on the odd tip.

Once you have broken through or held the lines of defence several times you may be looking for the new challenge that's ready and waiting, and allows you to rebirth the game to produce a much tougher Expert level game.

Patton vs Rommel has a few quirks such as the menu display, but it will thought out and will challenge even experienced computer gamers.

T.H.

Touchline:

Patton vs Rommel, Supplier: Electronic Arts, Langley Business Centre, 11-49 Station Road, Langley, St. Slough, Berks SL2 7PN. Tel: 0753 49047.

Machine: C4X. Price: £24.95 alk only.

TAPE ORGANISER

The Commodore tape system is renowned for its reliability but unfortunately it is equally well known for its lack of speed. Here's your chance to load programs from tape almost as quickly and as reliably as when using a disk drive

By D.J. Cook



The program uses a directory program which is saved as the first program on a tape and stores all the names of all the programs on the tape presenting them as a menu. Once a program is chosen the tape is wound on at fast speed to the start of that program, as this wind-on is controlled entirely by the directory program there is no need to sit watching the tape counter.

The program can then be loaded using the fast loader which is automatically installed by the directory.

Creating The Program

First of all type in program 1. This program contains the machine code fast load routines. When the program is correct SAVE a copy for later use and then run the program.

Having run program 1 you should delete the program using NEW and then type in program 2. Do not run the program at this stage or you may cause the computer to crash with the STS command. SAVE a copy of the program.

When you are satisfied that the program is correct you should put a blank tape into the recorder with the junction of the leader tape and the magnetic recording tape positioned on the left pad of the cassette (accurate positioning of the tape each time the directory is altered and saved means the directory never overwrites other programs on the tape).

With the tape in position type in SYS40007. This calls a machine code routine to transfer the fast loader to the end of the BASIC program and then saves both the BASIC and machine code as a single composite program. The composite program can be loaded and saved in exactly the same way as a normal BASIC program but since the machine code is added the BASIC program cannot be edited. If there are any errors in the program they must be corrected before calling SYS40007.

Saving a Program Using the Directory

To save a program from an ordinary tape onto a directory controlled tape you should first load the directory from the beginning of the cassette and overwrite the appropriate DATA

statement with the name of your program. Take care not to alter the length of the DATA statement and check that PEEK(19070) returns the number 100. Once the DATA statement is correctly altered you should then rewire the altered directory in the same position on the tape as the original (by positioning the junction between the leader tape and the magnetic tape on the left pad of the cassette). Do not press stop, or rewind the tape. Run the program and follow the screen prompts. The tape will wind on automatically to the start of the sector in which the program is going to be saved. Remove the tape without rewinding and keep to one side. Now load the program you want to save and when correctly loaded replace the directory tape in the recorder and fast save the program using IS "name of program". The program is now saved on the appropriate sector and can be directly accessed by the directory.

Loading Programs Using the Directory

This is the easy part. Load and run the directory. Follow all the screen prompts (especially the one that says stop the tape). Use CL to load your program and press the space bar or Commodore key once the POUND ——— appears and that's all there is to it. This program is very efficient at finding and loading programs and is the most invaluable program I have ever used.

I never use cassettes larger than C20 so I find longer tapes take more time to wind on to programs and they stretch and break more rarely. One side of a C20 can store up to ten 20K programs which I find is enough on one tape though the directory does have sufficient DATA statements for up to 20 programs if you want to use longer tapes.

The sectors defined by this program are large enough to store programs up to 20K, if a longer program is saved it will occupy more than one sector so be careful when using long programs not to overwrite any programs already stored on the tape. If most of your programs are longer than 20K then the sector size can be increased by increasing the 250 in line 130 to suit your own requirements, or if your programs are much shorter you can save time and

tape by decreasing the 250 appropriately.

How the Directory Works

The tape motor can be controlled by FORKING registers 1 and 192 with the appropriate numbers (POKE 1,38 sweeps the tape and POKE 1,7 starts it again) as in lines 110 to 140. Unfortunately the operating system constantly checks to see if a key is depressed on the cassette and resets both registers appropriately. The way I have got round this is to have a right loop constantly FORKING the registers until the space bar is pressed. This allows the cassette to be changed from PLAY to FASTFORWARD etc without the tape motor starting up. When the space bar is pressed the tape motor is started (by POKE 1,7) and the tape is wound on at fast forward speed to the beginning of the chosen sector.

The timing of the wind on uses TL the internal clock of the 64 which is incremented every 1/60th of a second. How long to wind on, and therefore how far along the tape the sector is, is determined by the number of the chosen program and a constant (250 in this case). This divides the tape into sectors of approximately 128cm length which is about 50 secs of recording time at 4.25cps at fast forward speed. Each sector can hold about 20K of program using the fast load routines or slightly less than 2K using a normal save.

The machine code loaded is transferred into memory from 49152 to 50076. This area is not normally used by BASIC and should not cause problems with BASIC programs but the fastload will be incompatible with any machine code routines which use this area of memory. The program alters the character dispatch vector (5308 & 309) to allow the CL, IS and EV commands to be implemented. If you want to disable the fast load routines then SYS50041 will restore the normal vectors while SYS44215 will reinsert the fast load.

The directory program can be used without the fast load using the normal save and load routines but the 250 in line 130 needs to be greatly increased to accommodate the greater tape length needed by the slow save. If the machine code is omitted then the STS command must also be omitted or the computer will crash.

See *Diagrams* on page 72



Have your chance to become a junior wizard, capable of wielding limited powers who sets out to become the chief wizard by beating the best seven magicians in the land.

Armed with only five spells in your spellbook you stride out into the wilderness which scrolls inside an unusual circular window which is flanked on either side by the spellbook, image and physical, spiritual and mental levels of you and your opponent.

The task ahead is split into three levels that must be completed in order. First of all you must explore seven towns that are spread across the land and do battle with six monsters. Defeating these monsters will grant you access to the treasure they guard which can be traded with the relevant towns for food to top up your three energy bars. Once you have killed all six monsters and have delivered the treasure to all six towns the seventh leads to level 2.

Level 2 provides a test to see whether a wizard is strong enough to challenge the seven master magicians. The contest comprises 30 minutes each with unique strengths and abilities. Three of these possess magical artefacts that must be collected if the player is to gain access to level 3 and a dual to the death with the seven wizards.

To complete the game the player must fight and defeat each of the master magicians, beginning with the Wolf Lord

and Bear Lord and ending with a final battle with the Dragon Lord.

The key to success in your quest lies in the spells that you wield from the four that you can select at the beginning of the game, to the others that you earn by defeating the monsters in level 2. Any spells that you gain must be stored on an empty page in your spell book, as there are times in the game where you will have to face the difficult decision as to which spell to raise from your book.

The spells are split into three groups and represent the magic drawn from physical, spiritual and mental power. Physical spells include the more usual fireballs, walls of ice, magic missiles and rock showers associated with fantasy games. However in Wizard Wars you can wield the Fear, Evil eye, the Vision and protection from evil spiritual spells and the mental trickery of Forget, Invisible and Mind lock.

The result is a mixture of arcade action and roleplaying that creates one of the first true arcade adventures. **T.H.**

Fourfiles:

Title: Wizard Wars. **Supplier:** GSI (US Gold) Dept 2/ J Walford Way, Moford, Birmingham B6 7AN. Tel: 021-276 2338. **Machine:** C64/128 **Price:** 69.99 (C64)/89.99 (128).

**BACK UP!
TRANSFER!
CONVERT!
ANY PROGRAM**

**TAPE TO DISK
DISK TO DISK
DISK TO TAPE
TAPE TO TAPE**

FOR ONLY
£29.99
P.C.C. MAIL
& POSTAGE



**THE AMAZING EXPERT CARTRIDGE
WITH NEW EXPRESSE SOFTWARE V 3.2**
EXPRESSE - 5.25" 4 COMPRESSION - FASTER, SMALLER, EASY COPY -
RECORDS - REMOVES UNWANTED FILES ON DISK OR TAPE - ONLY 1 PLAN

ALSO INCLUDES ALL FEATURES BELOW & MORE

- | | |
|-------------------------------|----------------------------|
| • PROTECT FILE TO FILE SYSTEM | • SMART COLLATION VALUE |
| • SMART EXTRACTION & SAVES | • SMART EDITOR |
| • INTERNAL & EXTERNAL COMMAND | • JOYSTICK POINT OPERATOR |
| • AUTOMATIC ALTERNATE MODE | • FULL 64K RESOLVE MONITOR |

- | | |
|--|-------------------------------|
| • Disk Copy Utility | • Smart Directory & File List |
| • Disk Compression | • Expert Command Language |
| • File Compression & File Recovery Options | • Disk Copy Utility |
| • Compression/Decompression Utility | • File Copy Utility |
| • Compression/Decompression Utility | • File Copy Utility |

PLEASE CHECK DISK OR TAPE SOFTWARE IDENTIFICATION

TRILOGIC Trilogic Dept. VC, Unit 1, Fax: 0274 600002
253 Newn Works Road
Bradford, BD12 0QP, Tel: 0274 600115

NO OTHER DISKETTES, TAPES OR OTHER MEDIA ARE INCLUDED WITH THIS SOFTWARE PACKAGE. PLEASE CHECK THAT THE FILES ON ANY DISKETTES/TAPES ARE CORRECT.
© 1988 TRILOGIC. ALL RIGHTS RESERVED. TRILOGIC, TRILOGIC, TRILOGIC AND THE TRILOGIC LOGO ARE TRADEMARKS OF TRILOGIC.

**BUY TEN
TOP QUALITY
BRANDED FLOPPIES
AND GET TWO FREE!**

An unbeatable offer from Computec for
5.25" floppies, the most reliable and
dependable diskettes on the market.
Look at the prices below, let alone the
freebies. They even include VRT and
same day despatch.



5.25" PRICES
FOR 10

+2 Free Disks

- | | |
|--------------------------------------|---------------|
| Single sided,
double density 400k | £7.47 |
| Double sided,
double density 400k | £8.05 |
| Double sided,
double density 800k | £10.25 |
| Double sided,
high density 900k | £17.25 |

3.5" PRICES
FOR 10

+1 Free Disk

- | | |
|---------------------------------------|---------------|
| Single sided,
double density 1200k | £14.95 |
| Double sided,
double density 1200k | £18.40 |

Every disk is thoroughly inspected and carries a
'no quibble' lifetime guarantee.

**DEALERS WANTED
EDUCATIONAL ENQUIRIES
WELCOMED**

To order just phone us on
0258 59330 9-5 hours
Fax: 0258 59008

SKC Authorised
Distributor

Write to the address below
Or make an appointment and
drop in on us. We accept
ACCESS and VISA cards.
Cheques or postal orders. And
don't forget - all prices include
VRT and same day despatch.

COMPUTEC



Computec, Dept. C1,
P-I-R-E-P-O-S-T
Pimphese, Blandford
Dorset DT11 7BR

DISCOUNT SOFTWARE

- | | |
|------------------------------|---------------|
| Fontmaster | £34.95 |
| (This is the latest version) | |
| Toyshop | £39.95 |
| Printshop | £29.95 |
| Certificate Maker | £19.95 |
| Create A Calendar | £19.95 |
| Hack Pack 128 | £12.95 |
| Petspeed | £12.95 |
| Helper 128 | £14.95 |
| Viza Star 128 | £74.95 |
| Viza Rite Classic | £54.95 |

ADAMSOFT

18 Norwich Avenue, Rochdale,
Lancs. OL11 5JZ

Access card orders accepted - Tel. 0706 - 524364

First Steps



Computers can be useful but mysterious messages await the unwary user

By Norman Doyle

Cast your mind back to the early days of computing. A computer utopia was dawning, the ZX 81 had mutated into the Spectrum and the VIC-20 was being superseded by the Commodore 64. Magazines were full of articles relating to the revolutionary benefits of the new technology. Home accounts would be held on computers, databases would hold all the names, addresses and birthdates of friends and relatives, and the computer would even control the improved household gadgets which would result. To brutally paraphrase Plato, the reality has proved to be less than the dream.

Small businesses could certainly benefit through the use of the C128 and many would even find the C64 and disk drive beneficial but the problem here is one of credibility. Amstrad's Alan Sugar observed recently that business people prefer to use equipment which is not identified with the domestic market. Eighty

such as this probably explains why the Commodore PET was so successful with large and small businesses but, in the shadow of the PC, the C128 has not prospered.

Application Error

Computopia was not achieved for one main reason - in most cases the tasks are not complicated enough to merit the time and trouble of loading and using the software. Until hard disks become cheaper or an even better instant retrieval system is devised, home accounts will be worked out on card/index packets, hi-tech databases will still be called *Filedos*, and home gadgets will remain an unattainable ideal of their own. The only area where a computer has proved beneficial is in the realm of wordprocessing and leisure pursuits.

One of the most rewarding pastimes in programming - it stretches the mind and usually rewards the

programmer with something that can be used over and over again. One of the less pleasurable aspects is debugging. Everyone would like to create something that works first time but that isn't always possible.

Whatever the make of the computer, one thing they all have in common is the fearsome error message generator. The Commodore 64 has 53 of these messages and the C128 has 44, as shown in the panel. Over the next few issues *First Steps* will look at each message in turn and consider the reasons why these occur and how to find, correct and avoid them.

Too Many Files

Overview: rare

When this error is reported the program has attempted to open more than two files. The simple solution is to check that all the files are necessary and erase out those which are not needed or redundant.

If such an inordinate range of files is necessary, it is unlikely that they will be receiving information simultaneously. In multiple file programs, it is always good practice to open a file, read or write the information necessary and then close it again immediately.

File Open

Occurrence: program error

When opening a file there are three values following the OPEN command. The first is the file number which can, with few exceptions, be any number between 1 and 255. The second value is the device number and the third is the secondary address which determines the kind of operation that is to take place.

The file number is the one used when communicating with a device using file handling commands such as PRINT#, INPUT# and GET#. This error means that an open file is already using the value. If you like, the file number is a connection from the computer to one peripheral device. Although several of those lines can be attached to the same device, it is not possible to connect more than one peripheral on a single line.

The solution to this problem is to allocate a different number to the file. With 255 numbers to choose from this shouldn't be difficult because only ten files can be open at any one time (see too many files).

File Not Open

Occurrence: program error

This also relates to the file number and occurs when a file handling command uses a value which has not yet been allocated to any device. The answer is this is simple, either open a new file or find a file which has already been opened and allocate the value to the handling command.

File Not Found

Occurrence: user error

This is principally a disk error and simply means that a file for loading or saving has not been found on the disk currently in the drive. This can result when the wrong disk is in the

drive, if the directory has been corrupted in some way, or through typing errors.

The only way around the corrupted disk error is to shrug your shoulders and promise to be more careful in future. Format a new disk and start again. Otherwise the answer is to check the directory and then insert the correct disk or correct the spelling of the file name.

Be careful and this error should never occur.

Device Not Present

Occurrence: user error

This is displayed as a file command issued to a peripheral which is not connected or is turned off. Care should be taken when this occurs. If the device is connected it can usually be rechecked on with no problem. If it has not been connected the only safe solution is to switch the computer off and start again after connecting the device.

If the device is the cassette recorder or disk drive and a program is in memory, which will be lost if the computer is switched off, drastic measures must be taken. Ensure that the device is switched off and that any other peripherals connected through it are also off. Next check that the plug is correctly positioned for insertion and then confidently push the device connector into the relevant computer port. DON'T wiggle the plug about. Finally switch the device on.

This procedure is contrary to any safety regulations and should only be used as an emergency measure. Lack of care will result, at best, in the computer doing a warm start or reset - at worst possible damage the input/output chips inside the computer, peripheral or both!

Cartridge and user port connections should ONLY be done with the computer powered off. It is very easy to misalign these connections and short across terminals which should never ever meet.

Table 1 : Commodore Error Messages

Peripheral Errors	Operational Errors
TOO MANY FILES	NEXT WITHOUT FOR
FILE OPEN	RETURN WITHOUT GOSUB
FILE NOT FOUND	OUT OF DATA
DEVICE NOT PRESENT	OUT OF MEMORY
NOT INPUT FILE	UNDEF'D STATEMENT
NOT OUTPUT FILE	REDIM'D ARRAY
BAD DATA	ILLEGAL DIRECT
MISSING FILE NAME	BREAK
ILLEGAL DEVICE NUMBER	CAN'T CONTINUE
MISSING FILE NAME	SYNTAX
LOAD	
VERIFY	
FILE DATA	
	Additional C128/Plus Errors
	CAN'T RESUME
	LOOP NOT FOUND
	LOOP WITHOUT DO
	DIRECT MODE ONLY
	NO GRAPHICS AREA
	BAD DISK
	Extra C128 Errors
	SEND NOT FOUND
	LINE # TOO LARGE
	UNRESOLVED REFERENCE
	UNIMPLEMENTED COMMAND
	FILE READ
Mathematical Errors	
ILLEGAL QUANTITY	
DIVISION BY ZERO	
FORMULA TOO COMPLEX	
UNDEF'D FUNCTION	
OVERFLOW	
Input Errors	
STRING TOO LONG	
TYPE MISMATCH	
EXTRA IGNORED	
REDO FROM START	
BAD SUBSCRIPT	

THE MPS803 PRINTKIT IV



AT LAST!

Top quality performance from your MPS803 Printer. Simply fit "PrintKit IV" directly to your MPS803 printer and you will get:

- TRUE DESCENDERS • BOLD OPTION
- BOLD LINES • 72x110

And up to 50% speed increase!

All facilities are fully CBM compatible.
"PrintKit IV" is a very impressive upgrade kit. CBM New '87. Send just £10.00 to join the hundreds of satisfied customers.

WE'VE DONE THE IMPOSSIBLE!

AMIGA OWNERS. Connect your MPS803 DIRECTLY to your AMIGA. All the facilities of PrintKit IV on your AMIGA. Don't buy an expensive new printer just send for the AMIGA version of PrintKit IV at just £39.95 plus £14.95 for the lead (you'll need a new lead anyway). £115 discount for PrintKit IV owners.

We also stock top quality tractor feeds for the MPS803 with acoustic hoods, available now £33.95.

SOUND SAMPLER

A NEW BREAKTHROUGH!

New breakthrough brings to you an Amiga sound Sampler at an affordable price. Directly digitizes sound signals, allowing them to be stored and modified inside the computer. Can digitize sounds of up to 100,000 samples per second, 3.5mm socket for audio input at line levels. No external power supply necessary, draws power directly from the computer. "My small unit does not take up any desk space. Plugs directly into Parallel Printer port. Please state when ordering whether you require the AM1000 or the A500/A2000 version.

Public Domain software provided free of charge, also compatible with most commercial software.

Price at **ONLY £29.95.**

Ribbons for the MPS803 £3.99.

We will design the printer driver to couple any printer to your Amiga. Call us for details.

Telephone: 01274 250000

AMIGA PRINTER TECHNOLOGY

SHINDORF HOUSE, 4 HOWARD ROAD, SOUTHVILLE,
BRISTOL, BS4 1QH. Telephone BRISTOL 02720 3541115
Telegrams SHINDORF BRISTOL
7 days delivery. Full guarantee.

I.C.P.U.G

the Independent
Commodore Products Users Group
is the largest and most friendly
computer club in the country

Back issues of '88' available to non-members at well as members, at £1.50 each, postage paid.

- Many local groups with regular meetings
- News magazines included in membership - 100 plus pages of reviews, news and information every two months.
- We support all Commodore Machines old and new: PET, WCRS, 64,16, +4, PC, 128 and all AMIGAs.
- Free Software Library of public domain programs for all the above machines available to members on supply of blank disk or tape and payment of p.p. Here members, do not send for this until membership number is received.
- Help and Advice
- Discount scheme
- Subscription only £70 per year (UK) plus £1 joining fee

If you are seriously interested in joining or programming any Commodore computer, then joining I.C.P.U.G. is a must! For full details, send a stamped, addressed envelope to:



ICPUG Membership Secretary, Jack C. Cohen,
30, Brinsford Road, Newbury Park,
Epsom, Essex, SS9 7EP

OFFICIAL COMMODORE/AMIGA DEALER

COMPUTERS

AMIGA BUSINESS SYSTEM 4860 - 10M - MPS803 -

Hardware + Peripherals (no software) £299.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Amiga 500 + 10M Floppy + 5.25" Disk £129.00

Delta Pi Software Ltd

10 Parkway Lane, Southfield, Leeds, LS2 1NG
Tel: (043) 268383 (Mon-Fri)

Infiltrator II



The over-the-top hero Captain Johnny "Jumbo-Baby" McGilbert is back for another series of adventures in Infiltrator II.

The world class printer, TV and film mega star, consultant to every first world leader and all round nice guy has been called back into action to once again thwart the Mad Leader's evil schemes. The call came just in time as you were sliding from the front page 25. Not exactly the ideal situation for a hero who craves the limelight. Now the Mad Leader has reorganised his forces and you're back in business in another line of impossible missions.

Each mission consists of three parts. First you must pilot your Whizzbang Enterprises Glomo DORN-Attack helicopter through enemy lines, infiltrate an enemy base to complete a ground mission and then fly back to headquarters and capture an airplane. Fly three such missions and you're out again the hero you think you are.

In the first mission you must discover the Mad Leader's chemical labs and neutralise a vast of deadly nerve gas, then return in part two to discover and deactivate his missiles and cause a meltdown in the leader's nuclear reactor before completing the game by tracking down the mad man himself and by installing a brain plant which turns one a valuable member of society!

Flying the Glomo to the target requires a combination of skill to keep in the air and to make sure you arrive at base with enough fuel to get home, strategy to plan and discover your route and skill deciding your reactions to other aircraft that guard the skies.

Through a series of passwords that aircraft exchange, it is possible to decide who is friend and foe. However,

you are the Infiltrator which happens to be your password for friendly forces, but you can lead the enemy with the password ID Overload. If you give the wrong response then you'll need all your wits as well as your Glomo's guns, rockets, flares and chaff launchers to survive the battle.

At the base, the screen display and game changes from a flight simulator that incorporates disorientation to a tense ground mission. Whichever mission you are on, you will be up against overwhealing odds and so you will have to use cunning and smarts to get away rather than attempt a do-or-die assault. Armed with joystick and selected fabric papers you should be able to get past most guards, however our hero has a sleeping gas canister up his sleeve in case the papers don't impress and a stock of grenades if things get really tough.

Inside the buildings the display changes once again as McGilbert searches through the rooms for enemy weapons, a change of uniforms, security passes and the key to his mission either the set of blueprints, missiles or the Mad Leader himself.

The game is both challenging and enjoyable as you relish the chance to be a hero that's completely over the top, and to add to the value the original Infiltrator is included in the game pack. Two games for the price of one can't be had!

T.H.

Together:

Titles: Infiltrator II, Supplines: US Gold, Units 2/3, Redwood Way, Redwood, Bovingdon, Herts. Tel: 021-354 3186. Machines: C64/128. Price: £9.99 (C64), £14.99 (Disk).

SPLIT RATE

Save on phone bills with this intelligent
modem software

With the prices of hardware constantly falling, bulletin boards are using intelligent modem equipment that accept a variety of incoming baud rates. The caller wants to use as high speed as possible to minimize both his waiting time and telephone bill. Cheap RS232 type modems can often cope with both 300/300 and 1200/75 baud and unless you are planning to upgrade software, 1200/75 is by far the best to use - after all, most of your time is spent reading rather than writing.

A scrolling terminal emulator is very easy to program if you restrict yourself to 300/300 since the Commodore RS232 implementation can cope with this directly. There is even a perfectly usable BASIC version in the Programmer Reference Guide. However, to allow the computer to send at a different rate to the one at which it is receiving requires a little more work. 1200 baud is also too fast for BASIC and machine code needs to be used to make sure that the receiver buffer does not overflow.

The real benefit of the latter reception is when it is used to download software from a bulletin board. However, there is always the problem of telephone line noise, and so, some form of error checking is needed. This program uses the very common modem protocol.

How it Works

The program consists of a BASIC section that does all the setting up required. There is no rush for this. Terminal emulation and the receiver part are in the machine code section.

Terminal emulation consists of sending the character codes of the keys pressed to the RS232 output and printing the received character codes onto the screen. Bulletin boards work in proper ASCII which has the upper and lower case letters reversed, compared with Commodore. This conversion needs to be performed for both transmitting and receiving. Any control codes may be sent to the RS232 output, but they are filtered out of the input to prevent misinterpreted characters affecting the display. Standard KERNAL routines have been rewritten to allow an independent baud rate.

The modem protocol consists of sending or receiving 132 byte packets. Each packet consists of SOH (ASCII 1) followed by the packet number and the 2% complement of the packet number. Then, there are 128 bytes of data and finally a single byte checksum calculated by taking the least significant byte from the sum of the data bytes. Transfer is initiated by the receiving computer sending NAK (ASCII 21).

The sending computer then sends

out the first 132 byte packet. If the packet number and the checksum are correct, the receiving computer will then send ACK (ASCII 6) and the sender will send the next packet, but if either of these values are wrong, then NAK is sent to tell the sender to repeat that packet. The final packet has its data padded out to 128 bytes by control-Z (ASCII 26) characters, and when the sending computer is asked for another packet, it sends EOF (ASCII 4) instead of SOH and waits for a reply of ACK before stopping.

This may all seem a little long-winded, but it does mean that any transmission error should be detected, and the whole file should be transferred correctly. A mistake of a single byte can mean that a program will not work.

Getting it all in

There are two versions of the program, one for the C64 and one for the C128. These are both BASIC programs which poke in the machine code program each time the program is run. The machine code section has built in error checking and the BASIC section can be modified as required. Simply type it in. The C128 version works in both 40 and 80 column modes and automatically switches to full mode where possible.

BAUD TERMINAL

By William Sellers

In Use

When the program runs without any errors, you will be confronted by a menu. The baud rate and protocol options are self-evident. If neither is used, then the RS232 port is set to 300/300 baud, 8 data bits, 1 stop bit and no parity. This can be changed if required.

Online opens the RS232 channel and enters the terminal mode. This will switch most modems online, but you may have to do this manually. Normally, you need to dial the required telephone number, wait until you hear the computer answer tones and then switch the modem to online. You may also need to hit return a few

times to get the other computer to acknowledge you. To get back to the main menu, press Commodore-Q.

Xmodem send and receive are (surprisingly) for sending and receiving files with the xmodem protocol. To use these, access the bulletin board with the online option and activate file transfer at that end. Then press Commodore-Q to return to the menu and select the send or receive option as required. You will then be prompted for a filename and the display will show the status of the file transfer. When this has finished, you will be returned to the main menu. To abort at any stage, press Commodore-Q. Xmodem on the host is often cancelled with Control-X. Note: xmodem will

only work with 8 data bit protocols.

ASCII/Pet conversion is for converting between the two character coding systems. Conversion from ASCII to Pet codes also removes any line feeds from the file since these are usually unwanted. Other control codes are left unaltered.

Other Uses

Xmodem is often used for file transfer along RS232 lines between computers so this program is quite handy for getting your Commodore to talk to any other computer you may own - assuming you have suitable software for them.

See *Setups* on page 71





*Can the new Expert cartridge
disk system broadside
WARP 25?*

By Eric Doyle

TRILOGIC'S ROCKET ATTACK

When TriLogic launched its Expert cartridge the boast was that a RAM cartridge can do anything that a ROM could do. TriLogic proved true to its word with the release of Danc's Action Replay Professional IV cartridge with WARP 25 fast disk loader. The Expert just couldn't keep pace.

Now the RAM cartridge is heading up the cart wars with a new operating system, Expert V3.2R, which includes a new program called Rocket TurboLoad which intends to take on WARP at its own high speed game.

The principal at stake is that The Expert is based on a RAM chip which can be programmed from disk or tape

but Danc's ARP IV is ROM based. This means that an update for The Expert costs the price of a new disk (\$3.99) but an update for ARP IV means the price of a new cartridge (currently \$24.95).

Although The Expert offers many more utilities, the ARP IV cartridge has an extremely fast disk loader which gives it a slight edge for some users.

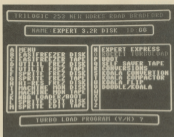
Rocket TurboLoad attempts to redress the balance in its favor but how do the two systems compare?

Round One

A program was chosen which filled up 202 blocks of computer memory

(\$990-5C1FF). The saving disks were freely formatted so that the search time for both system's resellers would be the same. To use WARP all that needs to be done is to plug in the cartridge, load the program for saving, switch the cartridge in and then save. After this a loader program is required which loads as a no block boot.

The Expert is more fiddly to use. First the operating system has to be loaded into the cartridge, then the routine follows a similar path as the WARP system but its final step has to be added. A fresh disk has to be formatted, a second program (Rocket TurboLoad) has to be loaded, and the program for saving is transferred from



the normal Expert disk to the newly formatted disk through Rocket TurboLoad.

On balance the first round goes to ARP IV for convenience but Expert seems slightly better because fewer program conditions trip out the system. ARP IV efficiency against Expert reliability.

Round Two

Compression of the program rendered a reduction from 262 to 172 blocks on ARP IV. Expert initially reduced the program to 128 blocks but after processing through the Rocket converter this increased to 187 blocks. A slight points advantage to Trilogic.

Reloading the program through the ARP IV WARP system always means loading the loader program first. This displays a menu and the cursor has to be moved onto the program required. The RETURN key then initiates the load. The load took about five seconds.

Rocket is easier to use as long as you know the program's name. Using the normal LOAD command followed by "R" program 3.1 automatically boots the program. This load took about seven seconds.

Although it is obvious that WARP will have a slight edge on speed, the need to select through the loader affects the overall speed depending on how quickly the user loads and runs the menu, selects the desired program and presses RETURN. On balance the second round goes to The Expert for convenience of use.

Round Three

The final round is a measure of the stamina of both the contestants. Which can survive the longest?

Both cartridges offer special 'shaping' utilities through which pipes and infinite lines can be added, and both have respectable memory monitors. Once again The Expert has the edge with joystick port swapping so that all games will use the port of your choice, a Sprite Extractor program to allow the viewing and saving a much improved sprite editor, and an infinite life creator which doesn't work on all programs but will work on many.

The knockout punch is named home because The Expert is reprogrammable and there is still room for future additions to the sprites at low cost. ARP IV additions would have to be added onto a new chip and sold at ten times the cost of an update disk.

Naughty but Nice?

The controversy surrounding cartridges wends with ever increasing intensity. Are these cheating keys for use by thieves to unlock protected programs or are they useful tracking utilities which can lay bare the bones of programming structures? The fact that the cartridges can be used to copy most software programs does open up the piracy field but, as Alan Sugar proved recently with his twin cassette decks, it is not illegal to advertise or

sell equipment which has a legitimate use.

Let's face it, many programmers and programming bosses make use of The Expert and its kin for developmental and exploratory purposes. Can anyone justify condoning private use while condemning a product in public? Cartridges MUST never be used for piracy but how can anyone learn progressive programming techniques without a cartridge?

The Expert's Sprite Extractor is a good example of the positive use of cartridges. Using this utility, sprites can be lifted from commercial software for examination at animation routines. The animation characters can then be saved to tape or disk. Using the new Sprite Editor program these characters can be reloaded and altered to create new characters.

The sprite program gives full facilities to view up to eight animation sprites but allows the creation of over 348 in all. The sprites can be animated to check for glitches, viewed in multicolour or hi-res, enlarged, flipped, flopped and generally manipulated into the required shapes. This is a boon to programmers who want to learn animation, few books or magazines can teach the principles in a more graphic form.

Cartridges may be potentially harmful but what is the alternative for serious programmers?

Round Up

For sheer adaptability and flexibility, The Expert is the best possible buy. For convenience it doesn't seem so well, the program is stored in a volatile RAM chip and disappears if power is disrupted for too long. The RAM is dynamic and will hold the program if the computer is switched off and on quickly, however.

Every time the cartridge is used the operation system has to be booted up and this is the price that must be paid for cheap upgradability (well, this and £3.50 for the new disk.)

The addition of the Rocket system has improved The Expert to a point where it can be considered to have no equal. Some cartridges may have the edge here and there, but who really cares if one system loads a more second or two faster than another? What is important is the overall benefits offered and in this department The Expert cannot be beaten. ☐

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 *Your 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 *Your Commodore* Software Service makes available all of the programs from each issue on both cassette and disk at a price of \$5.00 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:

INFONET LTD, 5 River Park Estate, Northamstead, Herts HP4 3HL.
Tel: (04427) 76651

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working systems, 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 stamp containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this stamp. 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 program, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

MARCH 1988

LABEL LINKER - Create a library of C128 programs (C128 disk only).

MULTI DUMP - Epsilon screen dump for C64 owners.

WEEK 11 - Continuation of program from February 1988. Complete program on this disk or tape.

MUSIC LOAD - Play music while your programs load (C64).

PLUS/4 ASSEMBLER - A machine code assembler for Plus/4 users (available on disk only).

ORDER CODE

DISK 3DMARKS 05.00

TAPE 3DMARKS 04.00

APRIL 1988

AUTO START MAKER - Give your disk programs that professional look by making them auto-start (C64 Disk Only).

CLEAR WITH BASIC - A series of short Basic routines illustrating various ways of clearing your C64.

TABULATE - Format your numerical printouts with ease (C64).

Commodore Modem (Revealed) - A small basic program showing how you can write your own programs for your Commodore modem.

MAKING GEOS BRITISH - Have you ever wished that GEOS had a £ sign or that GEOSes had British addresses - modify your GEOS programs to do just that (C64 disk only).

EXTENDED BACKGROUNDS - Alter the background colour for every screen line - a full display of 256 characters (C64).

4TOSPEED - Transfer your single part programs from C64 tape or disk to cartridge.

ORDER CODE

DISK 4TOSPEED 05.00

TAPE 4TOSPEED 04.00

MAY 1988

DESIGNER — A powerful drawing package for C64 owners.

80 COLUMN VIDEO — A simple 80 column character designer for the C128 plus disk.

CHARACTER GRABBER — Grab your favourite C64 character sets and use them in your own programs.

ADVENTURE KIT V — The fifth part of our continuing library of C64 adventure writing routines.

C128 KEYS SWAPPER — Store four sets of function key definitions with this handy utility.

SCREENDRAW 64 — A powerful C64 character and background editor for producing your own scrolling backdrops (disk only).

ORDER CODE
DISK YDMAY88 06.00

TAPE YCMAY88 04.00

JUNE 88

DIRECTORY EDITOR — A superb utility that allows you to alter your disk directories as well as add comments (C64 disk only).

90K RAM DISK — Turn unused memory on your C64 into a parallel disk.

BASIC LISTER — List a Basic program stored on disk without having to load it into memory (disk only).

EASY BASIC TOOLKIT — Give your C64 over 36 new bases — including the ability to store libraries of subroutines.

DISK SECTION EDITOR — Examine the contents of your Commodore 1341 disks (disk only).

ORDER CODE

DISK YDJUN88 06.00

TAPE YCJUN88 04.00

JULY 88

EASYPRIENT — Now C16 and Plus/4 owners can access their printer functions with a set of 80 easy to use Basic commands (available on disk only).

HANDY INPUT ROUTINES — Make your Basic programs idiot-proof and more attractive with these two Basic input routines (C64).

MAKING MUSIC — A simple C64 synthesiser to accompany our popular series.

DH PARALLEL INTERFACE — Use this software and a lead of your own manufacture to drive a Centronics printer (C64).

CP/M CENTRONICS INTERFACE — Use your 128 in CP/M mode access to a Centronics printer.

CASSETTE INLAY PRINTER — Use your printer to produce neat inlays for your cassette collection (C64).

DISK ASSEMBLER — Disassemble your programs direct from disk with this program for the C64 (Disk Only).

ORDER CODE
DISK YDJUL88 06.00

TAPE YCJUL88 04.00

Cassettes or disks are available from March 1988. Please ring the editorial office (01-437 0626) for details of these.

ORDER FORM — PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
AUGUST 1988		TAPE (£4.00)	YCAUG88	
AUGUST 1988		DISK (£6.00)	YDAUG88	
OVERSEAS POST £1				
				TOTAL

NAME.....

ADDRESS.....

POSTCODE.....

I enclose a cheque/postal order for £..... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, 9 HALL ROAD, HENEL HEMPSTEAD, HERTS HP3 7BH.

Please allow 28 days for delivery.

Pagebuilder

A full desktop publisher for the 128 in 80 columns, this unique package allows you to produce newspapers, forms, letter heads, labels, etc. It supports the 1280 RAM Reader and 1281 displays. Features include: electronic fill patterns, line styles, border styles, 12 different fonts, import text and graphics from page illustrate, fill or write for more details. Only \$25.95

Page Illustrator

A graphics drawing package. Offer using cursor, mouse, keyboard or joystick. Features include: zoom, mirror, fit, border, import text from any word processor, built-in label maker, import graphics from illustration, Fontmaker and PageMaker systems. Many more features. Fill or write for details. Only \$25.95

Disk Drive Problems?

15711241 Drive Alignment package by Free Spin Software repairs the alignment condition of the disk drive system performance adjustments. On screen help is available while the program is running. Includes features for speed and head alignment. Allows you to test each full and half track as you align. The Drive Alignment manual, user manual and alignment tool 1571 and 1547 drives. Error includes instructions on how to load the alignment program when nothing else will load. Works on the C24, 2524 and 128 in 64 or 128 mode. Price only \$8.95

SF73 0201 0000/0005/0000 versions available - Ring

New COMMODORE Version 5 ROM.

Are you experiencing any of the following problems with your 1571 computer:

- Errors when using Superbase?
- Device not shared/lost?
- Major problems when 128 or more open files?
- Fails to recover to recognize a floppy disk?

If you have one or more of these problems then you need 1571F10 ROMs, developed by Commodore Int. USA, do not settle for cheap alternatives, just plug in, saves over 12 main programs. only \$4.95

The Next Generation Basic - Basic 8.

We are proud to introduce Basic 8 with Basic 8 Print, Basic 8 Calc and Basic 8 Sorts. Basic 8 is the first 128 software package specifically designed to unleash the hidden graphics power of your 128, using a special edge technique. Basic 8 achieves performance rivaling that of 16 bit microprocessors your 128 in 80 columns producing a resolution of 640 x 200 in mono and 640 x 192 in 16 colours without any additional hardware!

Basic 8 adds over 65 Hi-Res graphics commands to Basic 7 and is completely compatible with its advanced monographic commands. You work in true 3-Dimensional environment, controlling such parameters as perspective viewing angles and the origin & depth of the view, and many many more features. only \$4.95

Basic Print, Calc and Write was all written under the new Basic 8 environment. These provide working examples of the power of this unique package. Basic 8 is available on a chip and a disk. More disk upgrade is wanted.

READ AND WRITE IBM FILES ON YOUR 128 OR 64.

The Big Blue Reader is ideal for those who use IBM PC compatible computers at work and have a 128 or 64 at home. The Big Blue Reader is not an emulation, but rather it is a unique and easy to use file saving program designed to transfer word processing, text and dBASE III files between your locally different file formats. The Commodore 128 and 64 and 1200GB disk formats. The Big Blue Reader requires a 1671 on a 64 or 128 and cannot use a 1641. The Big Blue Reader only \$9.95

Timeworks Software.

The popular selection of software for the 128 is now available at the low retail price of \$6.95 each. WordPerfect 128, a full featured 48 or 80 column word processor. SoftCalc 128, spreadsheet with Solworks printing. DatabaseManager a complete database.

Financial Systems Software Ltd

2nd Floor, Ambrian House, St Mary's Street,
Worcester, WR1 1HL.

Telephone (0206) 750217

All prices include VAT and P&P

Foreign Orders add £2.50 per package.

commodore

AMIGA

- | | |
|---|---|
| ■ Amiga 2000 with 2MB disk
and 207 KB. Retail value £399 | ■ Amiga 2000 with 1MB disk
and 207 KB. Retail value £349 |
| ■ Amiga 2000 with 2MB disk
and 192 KB. Retail value £399 | ■ Amiga 2000 with 2MB disk
and 192 KB. Retail value £349 |
| ■ Amiga 2000 with 2MB disk
and 192 KB. Retail value £399 | ■ Amiga 2000 with 2MB disk
and 192 KB. Retail value £349 |

PERIPHERALS
AMIGA

- | | |
|--|--|
| ■ Amiga 2000 mouse £12 | ■ Amiga 2000 mouse (UK)
with Amiga 502 connector £12 |
| ■ Amiga 2000 keyboard £12 | ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 |
| ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 | ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 |
| ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 | ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 |
| ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 | ■ Amiga 2000 keyboard (UK)
with Amiga 502 connector £12 |

CREATIVITY
AMIGA

- | | |
|--------------------------|--------------------------|
| ■ Creative Graphics | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |
| ■ Creative Graphics (UK) | ■ Creative Graphics (UK) |

PROFESSIONAL
AMIGA

- | | |
|------------------------------|------------------------------|
| ■ Professional Graphics | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |
| ■ Professional Graphics (UK) | ■ Professional Graphics (UK) |

128

- | | |
|-----------------|------|
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |
| ■ Commodore 128 | £129 |

EVERY YOU WANT IT CHEAPER... CALL US TODAY! 01 462 1616 999

Calco Software

BRISTOL HOUSE, KINGSTON HILL, BANBURG, N7 5ST, TEL. 01 462 1616

JINKXS

Imagine a cross between *Breakout* and *Pinball*, set over a horizontal scrolling playing area and you will have some idea of what *Jinks* is all about. Needless to say though, the software industry prefers dressing matters up as fancy and there is the usual space scenario.

You have to guide your rocket craft across the surface of the planet *Mari* in what is described as a unique exploratory research mission. The planet is beautiful yet remains an enigma. I'm not surprised seeing that it will resemble a cross between a pinball machine and a breakout game.

Jinks has been released simultaneously on both the C64 and the Amiga and so both formats can usefully be compared.

The object is to guide your probe around the planet using your spaceglider. Translated, this means hit the ball with your bat. Points are awarded for hitting bricks, sorry objects that happen to be lying around. As well, there are special objects such as mini worlds that when hit, can help your cause by adding extra lives, restoring your bat size, etc.

The bat is straight on one side and angled on the other. Flipping between these two options allows you to control the path of the ball. In practice, the ball tends to disappear off screen and you spend large chunks of the game chasing after it. This is by far the worst implemented part of the game. One of the hazards that needs to be avoided is letting your ball fall through a hole in the ground. Frequently, I found that I was getting "game over" messages with the ball nowhere in sight because I couldn't move the bat quickly enough to catch up. This is especially true on the C64 version which is rendered largely unplayable.

As you travel along the playing area, you have to avoid touching any of the moving objects. The first touch reduces your bat to half its usual size. The second touch kills you. Bat size can be restored by guiding the ball onto the appropriate object.

There are four different screens to try your luck on. Guiding your ball onto a rotating ball, special objects, take you into a 'bonus screen' although no bonus is given to your score. From here, you can select any of the other screens to visit by guiding your ball into the appropriate slot. Returns made to previously encountered screens results in more moving objects to avoid and many holes to fall through. These are effectively nine different skill levels as you can alter the speed of the ball and the effects of gravity upon it.

Graphics look quite slick on the Amiga although each screen takes an age to load in. Conversely, on the C64, all the screens are in memory at the same time but they look fairly rough.

Despite what I have said, the idea behind *Jinks* is quite a good one and gives more thought in the gameplay, it could have been a good game. As it stands, the Amiga version is average while the C64 version is quite dreadful. **G.R.H.**

Touchline:

Title: *Jinks* **Supplier:** GOF Dist 213 Welford Way, Melbourn, Huntingdonshire MK45 **Price:** C64 £4 - £8.99 (Cay £11.99) (Amiga) £14.99

	C64	Amiga
Originality:	7/10	5/10
Graphics:	5/10	7/10
Playability:	5/10	5/10
Value:	5/10	6/10



SCORE: 230



SCORE: 1000

JINKXS

Relative File Programming

Sort out your troublesome relatives

By Eric Ramsey

Following the first three articles in this series we are now at the stage in which a relative file has been set up and records entered into it. These can be displayed in a browse mode, amended and even printed out. At the moment, however, they are sorted in any order, appearing as they've been written into the file.

The first record you entered will be record number 1, the last entered record is the last to appear on the screen. Obviously, a file such as this has only limited use, so we must decide on a keyfield routine, and then a sort which will show the records in order according to the keyfield.

The first routine we need is one which will declare the keyfield for the file. You have the option of making this permanent, but I have always programmed so that the keyfield may be changed so that the records can be sorted in any field. Parts of all the user has to be able to change the keyfield, and to do that it would be a good idea if we displayed the field details for him to look at while he decides. Declare keyfield routine:

```
400 print@screen": rem **** clear
the screen
410 print@FIELD": tab(7): "Name":
tab(7): "Length"
420 goto:for a=0 to c
430 print a+1,tab(7):FIELD$ (a),
tab(7): FIELD(a)
440 next a print:print"Which FIELD
to key?"
```

```
450 go:to:5450: rem *** input number
Routine
450 key=n-1:
470:go:to:4370:rem *** branch to sort
Routine
480:go:to:0 [ Whenever you have the file
routine]
```

So far we have the number of the desired field to be keyed in the variable KP. One advantage of doing it this way is that just after a file has been set up and before this routine has been run for the first time, KP will be 0, which happens to be the number of the first field.

Among all the other statements at the beginning of the program, you might like to create a graphic image of the KEY, with two left-graphic R and a 0, like this:

```
1000:(cR)R-R
```

I chose the 0 for the graphic instead of the right-graphic on the W key because this graphic symbol does not work in lower case; you get a W instead of the round symbol. You might like to show the graphic symbol in the request for the keyfield number:

```
440 next a print:print "Which FIELD
to key?"
```

Now we have set up the keyfield variable; it is used inside the field array of the current file, then by stating the keyfield string from each record into

an array to be sorted. We will call the sorting string array ESKER(). Since this will be an array, it will have to be DIMensioned in the beginning of the program much like all the others we have already dealt with.

Dynamic Arrays

A word of warning - you cannot DIMension this or the other sort array which we will come to in a moment, using a real number. If we did so, you would only have to write a new record over that number and you would get a syntax error. For this reason, you must use a dynamic array because you cannot tell for each program file how many records you might be going to use, hence the variable number for the DIM. Which variable you use in the dynamic array depends on you.

If your program to halt the run at the 49th new record created, remove the `hoverslepping`, `CLR` and then `re-run` the program; you would be able to use the RN variable inside the DIM:

```
dim run$(rn*50)
```

Or, simpler but setting aside much unused variable space, is to use the very maximum number of records that can be written, the `AVAIL` variable:

```
dim run$(avail)
```

after which every possible record you might enter has already been

DIMENSIONED for. You would never get the dreaded "Bad Subscript Error" message with that, but you might have 100 records in the file and have the arrays DIMENSIONED for 4000!

The method I use is to POKE all the variables to a free page of RAM, starting at DEC-0854. Everything, field names, lengths, RN and all the other variables not yet mentioned are POKE'd there. When RN has increased by 48 lines when the program was last RUN, the program executes a CLR and then pokes all the locations in memory to get the variables back. The arrays are then re-DIMENSIONED, the housekeeping is then saved as a REAVE to disk for the next time the file is accessed in a program RUN.

With this method it is not possible to exceed the DIMENSIONED numbers in the arrays, but neither is it necessary to set aside huge amounts of variable memory when it is not required. However, I digress slightly.

You must have DIM'D the array RSRTE and RSKTE, either with %VAL or RN*50 or 100. Now we will write the first of the sort routine. We have to read the keyfields of all the records currently in the file, (and deleted ones) into the string array RSRTE, then sort them into alphanumeric order, ending up with the actual record numbers stored in the array RSKTE.

Why the %? Well, in case you have not used integer arrays before, they use much space on the disk when they are stored. An ordinary number is stored in several decimal places, which is wasteful of space, but an integer is all we need for this application (since there is never a need to read record number 4087!).

Let us begin the routine. You must have opened the file. The screen is cleared, and the message placed to inform the user what is happening. I have also programmed a display of the record number which is being processed so that the user has something to watch. Time seems to pass more quickly while you watch a number change!

```
4160 rem ***** Sort RECORDs
routine *****
4170 print(CHR$(2000)); print
"RECORD Processing Now"
```

The loop, / to the current record number used plus, most important, the variable storing the number of records currently deleted is added to it, DREC

and then the cursor is HOME'd (without clear) to show the record being processed, which of course, since PG is the loop counter, is PG.

```
4180 loop=1 to rndrec:print (home)
:print
```

Then the record to be read variable is declared as PG: BASIC 2, you have your own, don't forget:

```
4200 rem/pointer=0: gosub 4180:rem
*****dec is delete re
4260 (BASIC 2 user) open: gosub
4114: gosub 4180
```

Although we haven't seen this yet, the following line will be explained in greater detail later. Basically, the loop will detect records deleted, and will find the set character, CHR\$(255). Since this will mean nothing to the sort routine, we declare the sort string for this record as the lowest possible value, a string of "Z"...

```
4400 if disk% = chr$(255) then
drec=drec+1: read% (drec)open:
read%open "ZZZZZZZZZZZZ": goto
4420
```

ROUT%(i) is the array which stores the deleted record numbers for future use. They are stored at the time of deletion, but if this is a new sort, the sorted record numbers change, and you would end up overwriting good records if this array was not updated! Now we can look inside the DREC. Notice there is no gosub to the field string routine. What for? That would merely slow things down. Instead, MID\$(and the field pointers array with the length of the field areas, FIELD% is used to pick out the field to sort. KF. Then, because this field might be 80 characters long, LEFT\$(is used to take the left 12 characters to sort. The loop is then ended. You may use more characters if you wish by increasing this number inside the LEFT\$(.

```
4410 read$(p)=mid$(mid$(field$(, i)
(, k),
FIELD$(KF),12)
4420 next p
```

At the end of this loop, the entire field has been scanned, the relevant field has been stored in the array RSRTE and we are ready to sort. FIELD% SEPARATED SCAN FOR SORT - ALL BASIC.

Yes, don't panic. You should have realized that this routine would not work for you. You require a separate read routine. You will remember that your read record routine reads the record inside a loop, and ends with the record neatly inside the array DSRP%. This routine could be used, except that it is a waste of time reading the entire record when you only need to look at the single field. So if you have used the field separated method, you may take advantage of the fact that you can only read one field at a time. For this application this method has an advantage. Make these changes to the above:

```
4380 drec=4:rem ***** DO NOT use
the Read Routine. 4180
4390 (BASIC 2 user) open: p=p: gosub
4114
4392 RECORD =3, (p), (p)sk(k)
: gosub 3810
4394 RECORD =3, (p), (p)sk(k)
: gosub 3810
```

BASIC user:

```
4390 print #3, "p"chr$( 100)+chr
$(n)-chr$(k)
+chr$(p) (k)go sub 3810
4394 print #3, "p"chr$( 100)+chr
$(n)-chr$(k) +chr$( (p)sk(k)):
: gosub 3810
```

Then read the keyfield into the sort string array RSRTE%, no need to use DSRP%.

```
4396 open #3: read$(p)=go sub 3810
```

And you have achieved the same result: the keyfield of every record in the file is stored in the string array RSRTE. The RSKTE for records which have been deleted are stored as a string of 2 characters, making sure that they will be right at the end of the sorted array.

You may like at this stage to store the keyfield string array into another sequential file, to which new records as they are entered into the file may be added. This saves scanning the file every time the file is sorted, but you might not wish to do this. Close the relative file, and use this routine to store the sorted string array:

```
4430 a$(k)=mid$( name$(, k)data=...,
10: dopen #3, "w" + (a$(, k), #3, a$,
: gosub 3810
4430 for p=1 to r-drec: print #3,
read$(p): next p
```



4450 Delete

Now, because this routine will be used for sorting records without scanning the file, the next line skips the INPUT of the sort string array. I shall come to that later.

4460 goto@31 from *** skip read file

Now we go into the sort routine. This is the only routine which is not my own. It is based on a routine called 'Pur5ort' which is public domain and written by Nick Maropoulos. In fact the original needed some working on, since it suggested that the record number to the sort string and then stripping off the record numbers after the sort was finished, then reading the sorted array into another array for the sort result.

This was all quite cumbersome and actually unnecessary, since the effect of the sort is to shift the pointers in the array (RSRT%) and does not move the actual data contained in the strings at all.

I removed the need of any second array, which when you might have 4000 records is a terrible waste of valuable space, and the routine works very well for this application. Double check that you have typed it in correctly or the routine will simply not work at all.

```
4510 print@screen) "Sorting now...
Please Wait."
4520 array=dim array *** DBEC will
be explained later
4530 for i=1 to numof%()=1:next
4540 w=1/2.55:z=1:u=0 then return
4550 if w >= z then i=1:if w > 17 then z=
15: if u >= 39 then k=199: if w > 199
then k=573
4560 for j=i to i+1:300: form=0:ok=1
4570 for n=1 to a step i:form%()=
for j=i+1 to step-k
4580 if out%()=j() >= out%() then
n=j%()=n+1()()=n+1
4590 out%()=j()+w:next n:next i:if i > 1
then 4580
4600 return
```

Now the records are sorted, but the effect will not be apparent until you have changed the lines which request the records read routine. At present they are simply reading the record number of the file, not the sorted record number. In case you find this difficult to grasp, let me explain this further. Your records are presently on the disk, filed in the order in which

you typed them in. They look like this:

RECORD

```
1) ) 2) ) 3) ) 4) )
5) ) and so on. But the data they
contain might be:
```

```
1) 5) 2) 3) 1) 4) 3) 5) 4) and
so on.
```

At present when you request record number three in the file, you always see the actual record three. Now we have sorted them we have the sorted order in an array: the array for the above RECORDs would be:

```
RSRT% (RECORD REQUESTED=ACTUAL RECORD
RSRT% (1)=3
RSRT% (2)=2
RSRT% (3)=4
RSRT% (4)=1
RSRT% (5)=1
```

You must remember that any array is really a label: when you request a number inside the brackets of an array you are asking for the label of a particular package, which is something completely different. A browser of the file shown above, using the sorted record array RSRT%() would look at record number three first, then record two next, then record four, and so on.

You will be amazed of this because you will see record number 1 on the screen while the file record, that actually read, will be three. If you want to see the file record number you are actually accessing, you should add this to the screen display routine:

```
4300 print "Sorted RECORD No:
"loop" of".
:;:"Virtual", n%()=j%()
```

In the new part of the line, we have used the sorted array RSRT% in reverse to get the virtual, the file, record number.

Now the effect on the file has been to arrange the records for display in alphanumeric order. But why did we want the deleted records to show a "XXXXXXXXXXXX" as the RSRT%? Simple. The loop included not only current record numbers, but also deleted records. We shall go into more detail in the next article but if you have declared a loop, FOR PO=1 TO 3+1, then PO Loops 7 times. But if you

read back a string created inside that loop using just 5, FOR PO=1 TO 3 then the loop will only read the first 5. Now the string containing "XXXXXXXXXX" will have been sorted to the end of the array and will not be displayed because they are outside the RN range.

Now you must go through your routines carefully, and change every request for a File Read or Write.

WRIC=RN or WRIC=DISP

```
to
WRIC=RSRT% (RN) or
WRIC=RSRT% (DISP)
and of course, for the BASIC 2 users,
to
RP=RSRT%(RN) or
RP=RSRT%(DISP)
```

before going to the record pointers routine. To help you, here is a list of the relevant line numbers. Basic 2 users, you declare RP as the relevant RSRT%() before you branch to the record pointers routine.

In the other records routine, the write record reads the record number as RN, which is correct. But because this record is a new one the sorted array, RSRT%() does not exist for this record. Since the other records routine adds new records to the end of the file until the next sort, the new record sorted number will be the end record number, which here is RN. Thus we add:

RSRT%(RN)=RN

to the Line 2580

```
2580 w=cc%()=n%()=rn
```

Now we add the RSRT% to the browse record routine so that the record accessed is the sorted record.

```
1740 r=cc%()=j%()=rn *** go
RECORD number
```

We amend the edit record routine, otherwise we will be amending the wrong record!

```
3000 w=cc%()=j%()=rn: go@=4870
```

Then the lines in the copy routine:

```
3140 r=cc%()=w:cc=cc: r%()=m%()
3170 d=pr%()=r%()=j%()=rn: for i=1
```

And if you wish, you may show the virtual record number in the browser routine from the browse menu:

```
1210 print #4, "RECORD Number
"diag;" of "rec"
Virtual: ?; set%diag)
```

Finally, now that you have a keyfield you might like to display on the screen and on the hardcopy of the file which field is the key. The following lines and changes will do that for you.

```
1230 if%di=0 then print # 4, k%F;
else print # 4, ;
4300 print-print for a=0 to c: if a%k then
print #4, k%F; else print # 4, ;
```

Now we have to amend the house-keeping file. We need to add a short routine to store and read back the sorted record numbers, as well as the new program control variables. There are more of these to come yet. Here are the complete routines. BASIC 1, I am sure it is not necessary for me to write them for you as well. Just add the new line numbers, the open statements are the same.

```
200 n%L=LEFT$( "HSE"
~NAMES";", 16)
210 open #3, " " & n%L, w%asub
360
220 print #3, r%k; goto 3610
230 for a=0 to c
240 print # 3, FIELD%a; goto 3610
250 print # 3, FIELD%a; goto 3610
260 next a
270 for p=1 to m
280 print # 3, r%r% (p)
290 next p
300 close: goto 3610
340 return
And the Read Housekeeping.
```

```
150 n%L=LEFT$( "HSE"~NAMES";"
-----", 16)
360 open # 3, (n%L), r%asub 3610
370 input # 3, r%r%k; goto 3810
380 for a=0 to c
390 input # 3, FIELD%a; goto 3610
400 input # 3, FIELD%a; goto 3610
410 next a
420 for p=1 to m
430 input # 3, r%r% (p)
440 next p
450 close
460 return
```

Searching on the Keyfield

Now you have gone to all this trouble to sort the records, you may take advantage of it. After the sort, the records appear on the screen, in alphanumeric order as you browse, one after

another through the file. This is all very well, but if you have 4000 records, it would take a lot of browsing to find the particular entry you need.

Of course, you could use the goto option on the browser menu, but locating the individual record would still take some time. What we need is a search on the keyfield. This is the search routine I promised in previous article, which will find any entry in the keyfield in 9 or less reads from 1000 records!

How does it work? It is actually quite simple. Imagine you had a card file of names in alphabetical order. There is no marking in the card file showing where each letter begins as read; how would you look for one name in that file?

You would look at a card about the middle of the card file, and compare the name there with the name you are looking for. If the name is "less than" or a lower alphabetical name, you would look in the previous half of the file for another sample. If the card you selected was higher, you would look in the next half of the file. When you get close to it, you would look at individual cards until you found the name you were looking for or you knew it was not present.

Keyfield Search

Do not forget, you must have opened the relative file for accessing. BASIC 2 users, you must have opened the Error Channel 15.

First, the once extra read counter, the N%ldge counter is set to 0. This will be explained in a moment. The request string is input into n%.

```
3850 n%="0-yy-0"
3860 print "Enter Search Data:"
3870 input n%
```

Now we come to the processing. BN is the beginning number which is the number of records divided by 2, in other words, the routine find the middle of the file. The B%N variable is the number which will be processed from now on.

```
3900 b%=(n%+1)/2; bin=b%
```

The record number is assigned into R%REC. Note the R\$FIELD%() array find the sorted record number. This is important.

```
3970 rec=r%r% (j) in (bin)
```

BASIC 2:

```
3970 r%r%=(j) in (bin); goto 4110
```

Then the record is read. Field separated method users would probably be as well to copy the single field read method I used in the third article for the sorted keyfield string R\$RTS: or to write their own subroutine:

```
3975 RECORD # 5, (rec), (f%a%k%);
goto 3910
3977 RECORD # 5, (rec), (f%a%k%);
goto 3910
3980 input # 5, (di%k%); goto 3910
```

Using the in% function the keyfield string is searched for the request string. If the match is found then the RREC is declared as the number for reading and the display number is quoted for the display record routine. (Otherwise there will be blanks on the RECORD screen.) The character to show a positive search result, "Y", is set to 1, and then the routine branches back to the display part of the browser routine so that the searched record is displayed.

```
3990 goto 4120; goto 4080; if not
(di%k% in (k%L))
-> then rec=r%r% (j) in (bin); (di%a% in
(di%a%)) goto 3750
```

But the second read might not provide a match. In that case the routine continues into 3990, where the BN variable is again halved, having the effect of quartering the file. A quick check is made in case the BN variable has been halved to less than 1. If it has then the mdge variable is set to one for one last try. This is because some files will have many records with very similar keyfields, taking a disproportionate number of entries in a particular part of the file. The mdge allows one more read.

```
3990 if=(bn/2) in (bn) < 1 then
mdge=1: bn = 1
```

If the mdge variable has already been set to one, then the routine ends, displays the file not found message, and returns to the browser with the record number changed to the nearest record found to the match request. This is useful to move the file display to a particular part of the file even when you know the requested string will not be found.

```
3100 if n>1 then print "File Not
Found." : sleep 2
: rrr=rrr% (int(hn): disp(int) (int):
: goto 2150
```

Now we do the comparisons: if the requested search string, NS is "greater than" i.e. a higher alphanumeric value, then the keyfield string just read, then the halved processing number BN is added to the BIN number to give the next record number to inspect, and the routine branches back to the next record read.

Likewise, if the value of the string read is "less than" the string requested then the halved number BN is deducted from the BIN number. In this way the file is repeatedly halved until the record is found or the flagged variable is greater than 1, showing that the record is not present in the file.

```
3110 if n > disp(hk) then
bin=bin+bin:goto 3150
3120 if n < disp(hk) then bin=bin-
bin: goto 3050
```

You remember that a display counter Y% was set to 1. This is incremented to

display the keyfield of the searched record in reverse, for underline, Commodore 128 88 column screen. If you wish, you make like to change the lines in the display records routine to effect this.

```
4000 key= (and:rkf)binprn: "in here
please reverse or underline character":
4000 meta: y%binprn
```

Basic 2 Keyfield Search

Without the very fast and useful INSTR routine, BASIC 2 users will have to use their own small comparison routine at the end of the search. The lines are as follows:

```
3002 goto 3122: if n=1 then
rrr=rrr% (int(hn):
disp(int(hn): y%1) goto 2150
```

The remainder is the same except that you need this small subroutine to compare the strings, and to return the result SR.

The search result is first set to 0. If the length of the search string is longer than the length of the field string to be searched it would be a waste of time to continue, so if this is the case

then the routine returns

```
3122 sr=0: if n > l then (disp (hk):
) then return
```

Then the search loop is started to the length of the field string minus the length of the search string:

```
3124 lrr=l-lln: (disp(hk):) lrr n%>1
```

Using MID\$, the field string is scanned from left to right until there is one character less than the length of NS. If a match is found then the loop is closed and the SR result is set to 1:

```
3136 (mid$( (disp(hk):), n, lrr) =
) then sr=1:
(disp(hk):) lrr-1: n%1) return
```

After the loop has finished, DISP\$ has been searched without result, so the routine RETURNS with SR as 0:

```
3138 sr=0: return
```

There is the keyfield search. In the next article I will provide the routines for deleting records, and a sequential search method which will find any data anywhere within a file. ☺

TRYING TO USE YOUR COMPUTER!...

YOUR

COMMODORE

CAN HELP.

16 Lines, 1 kb, \$13.95
24 Lines, 2 kb, \$19.95
32 Lines, 4 kb, \$29.95
48 Lines, 8 kb, \$49.95
64 Lines, 16 kb, \$79.95

Send me the following Commodore software:

16 Lines, 1 kb, \$13.95

24 Lines, 2 kb, \$19.95

32 Lines, 4 kb, \$29.95

48 Lines, 8 kb, \$49.95

64 Lines, 16 kb, \$79.95

NAME (please print)

ADDRESS

CITY

STATE

ZIP

PHONE

DATE

MAIL TO: COMMODORE

INTERNATIONAL

16 Lines, 1 kb, \$13.95

24 Lines, 2 kb, \$19.95

32 Lines, 4 kb, \$29.95

48 Lines, 8 kb, \$49.95

64 Lines, 16 kb, \$79.95



Send this form with your remittance to:
COMMODORE, INC., 5300 West Park Drive,
Ann Arbor, MI 48106

Listings

Get it right first time with our deluxe program system
for the C64.

You may have noticed that our listings are free of those horrible little black blinks which send you searching around the keyboard for a suitable graphic symbol. You may also have noticed the funny numbers by the side of each line of the listing. For no reason, it's all part of our easy entry aid.

Instead of those nasty graphics and rows of countless spaces in PRINT statements and strings we use a special coding system. The code, or mnemonic, is always contained in square brackets and you'll soon learn to decipher their meanings.

For example, [BA] would mean type in a Shifted A, or an ace of spades in layman's terms, and [BAH] would mean a row of ten of those symbols.

[S+2] means hold down the shift key and press the plus key twice. It doesn't take a great leap of logic to realize that [C+2] means exactly the same thing except that the Commodore key (bottom left of the keyboard) is held down instead of the shift key.

If more than two spaces appear in a statement then this will be printed as [SPC4] or, exceptionally, [SSPC4]. Translated into English this means press the spacebar four times or in the latter case hold the shift key down while you do it.

A string of special characters could appear as:
[CTRL N, DOWNLEFTS, BLUE, F.C.C.]

This would be achieved by holding

down the CTRL key as you press N, press the cursor key down tenx, the cursor left key five times, press the key marked BLUE while holding down the CTRL key, press the F3 key and, finally hold the Commodore key down while pressing the number two key [C2] would of course make the computer print in brown).

Always remember that you should only have a row of graphic characters on your screen with no square brackets and no commas, unless something like this appears:
[SS][C*]

In this case the two characters should have a comma between them.

On rare occasions [REV T] will appear in a listing. This is a delete symbol and is created by entering the line up to this moment. Then type a closing quotation mark (SHIFT + ") and delete it. This gets the computer out of quote mode. Hold down CTRL and press the number nine key (BVSIGN), type the relevant number of reversed Ts and then hold down CTRL and press zero (EVSOPF).

Next type another quotation mark and delete it again. Now finish the line and press RETURN.

A list of these special cases is given in the table but remember that only one of these mnemonics will appear outside of a PRINT string; the symbol for pi. This may appear when its value is needed in a calculation so this may look something like:

CTC=[PI]*R;

Ignore the square brackets and just type in a shifted upward pointing arrow (in the pi symbol).

PROGRAM SYSTEM CHECKER

1 RUN SYSTEM CHECKER - GATE CODE

20 BL=0 (14074) 40000000

30 FOR I=0 TO 10:BL=0:FOR J=0 TO 10

40 BL=0:IF 10000000000000000000

50 IF 10000000000000000000000000

60 CO=0:FOR I=0 TO 10:FOR J=0 TO 10

70 IF 10000000000000000000000000

80 FOR I=0 TO 10:FOR J=0 TO 10:FOR K=0 TO 10

90 FOR L=0 TO 10:FOR M=0 TO 10:FOR N=0 TO 10

100 FOR O=0 TO 10:FOR P=0 TO 10:FOR Q=0 TO 10

110 FOR R=0 TO 10:FOR S=0 TO 10:FOR T=0 TO 10

120 FOR U=0 TO 10:FOR V=0 TO 10:FOR W=0 TO 10

130 FOR X=0 TO 10:FOR Y=0 TO 10:FOR Z=0 TO 10

140 FOR AA=0 TO 10:FOR AB=0 TO 10:FOR AC=0 TO 10

150 FOR AD=0 TO 10:FOR AE=0 TO 10:FOR AF=0 TO 10

160 FOR AG=0 TO 10:FOR AH=0 TO 10:FOR AI=0 TO 10

170 FOR AJ=0 TO 10:FOR AK=0 TO 10:FOR AL=0 TO 10

180 FOR AM=0 TO 10:FOR AN=0 TO 10:FOR AO=0 TO 10

190 FOR AP=0 TO 10:FOR AQ=0 TO 10:FOR AR=0 TO 10

200 FOR AS=0 TO 10:FOR AT=0 TO 10:FOR AU=0 TO 10

210 FOR AV=0 TO 10:FOR AW=0 TO 10:FOR AX=0 TO 10

220 FOR AY=0 TO 10:FOR AZ=0 TO 10:FOR BA=0 TO 10

230 FOR BB=0 TO 10:FOR BC=0 TO 10:FOR BD=0 TO 10

240 FOR BE=0 TO 10:FOR BF=0 TO 10:FOR BG=0 TO 10

by Eric Dayle

Checksum Program

The hexadecimal numbers appearing in a column to the left of the listing should not be typed in with the program. These are merely checksum values and are there to help you get each line right. Don't worry if you don't understand the hexadecimal system, as long as you can compare two characters on the screen with the magazine you can use our line checking program.

Type in the Checksum Program, make sure that you've not made any mistakes and save it to tape or disk

immediately because it will be used with most of the present and future listings appearing in Your Commodore.

















At the start of each programming session, load Checksum and run it. The screen will now beaver with yellow characters and each time you type in a line and press the RETURN key a number will appear on the screen in white. This should be the same as the corresponding value in the magazine.

















If the two values don't relate to one another, you have not copied the line exactly as printed so go back and check each character carefully. When you find the error simply correct it and

press RETURN again.

If you want to turn off the checker simply type SYS49152 and the screen will return to the familiar blue colours. You can then do whatever it was you wanted to do and if this doesn't use the area where Checksum lies you can go back to it with the same SYS command.

No system is foolproof but the chances of two errors cancelling one out are very small. Many of the listings are presented in lower case made press the Commodore key and the SHIFT key at the same time. 75

Mnemonic	Symbol	Keypress
[RIGHT]		CRSR left/right
[LEFT]		SHIFT & CRSR left/right
[DOWN]		CRSR up/down
[UP]		SHIFT & CRSR up/down
[F1]		F1 key
[F2]		SHIFT & F1 key
[F3]		F3 key
[F4]		SHIFT & F3 key
[F5]		F5 key
[F6]		SHIFT & F5 key
[F7]		F7 key
[F8]		SHIFT & F7 key
[HOME]		CLR/HOME
[CLR]		SHIFT & CLR/HOME
[RVSON]		CTRL & 9
[RVSOFF]		CTRL & 0

Mnemonic	Symbol	Keypress
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[PURPLE]		CTRL & 5
[GREEN]		CTRL & 6
[BLUE]		CTRL & 7
[YELLOW]		CTRL & 8
[POUND]		£
[LBARROW]		←
[RPARROW]		→
[PI]		SHIFT & ↑
[INST]		SHIFT & INST/DEL
[REV T]		rev tco
[Clear]		CBM + letter
[Shift]		SHIFT + letter

DON'T MISS OUT

Fill in your name and address and give this form to your newspaper.

Please order me a copy of **YOUR COMMODORE** and receive/deliver me a copy every month.

NAME

ADDRESS

.....

.....

Newspaper: This magazine is made available to your wholesaler through S.M. Distribution Ltd
6 Leigham Court Road
Surreyham
LONDON
SW16 2PG

Tel. 01-477 8111



Listings

MAKING MUSIC



PROGRAM: MOODIE

```
00 1. REST MOODIE
01 0
02 3 PRINT (MOODIE);
03 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
04 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
05 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
06 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
07 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
08 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
09 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
10 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
11 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
12 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
13 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
14 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
15 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
16 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
17 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
18 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
19 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
20 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
21 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
22 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
23 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
24 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
25 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
26 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
27 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
28 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
29 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
30 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
31 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
32 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
33 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
34 0.000;MOODIE;MOODIE;MOODIE;MOODIE;MOODIE;
```

LISTINGS

```

17 000 000000,000,100,000,100,
000,000,000,000,000
50 000 000000,000,000,000,000,
000,000,000,000,000
58 000 000000,000,000,000,000,000,
000,000,000,000,000,000
70 000 000000,000,000,000,000,000,
000,000,000,000,000,000
88 000 000000,000,000,000,000,000,
000,000,000,000,000,000
96 000 000000,000,000,000,000,000,
000,000,000,000,000,000
80 000 000000,000,000,000,000,000,
000,000,000,000,000,000
92 000 000000,000,000,000,000,000,
000,000,000,000,000,000
94 000 000000,000,000,000,000,000,
000,000,000,000,000,000
98 000 000000,000,000,000,000,000,
000,000,000,000,000,000

```

PROGRAM INTERRUPTS

```

94 3 0000 0000
96 3 0000 0000
98 3 0000 0000
100 3 0000 0000
102 3 0000 0000
104 3 0000 0000
106 3 0000 0000
108 3 0000 0000
110 3 0000 0000
112 3 0000 0000
114 3 0000 0000
116 3 0000 0000
118 3 0000 0000
120 3 0000 0000
122 3 0000 0000
124 3 0000 0000
126 3 0000 0000
128 3 0000 0000
130 3 0000 0000
132 3 0000 0000
134 3 0000 0000
136 3 0000 0000
138 3 0000 0000
140 3 0000 0000
142 3 0000 0000
144 3 0000 0000
146 3 0000 0000
148 3 0000 0000
150 3 0000 0000
152 3 0000 0000
154 3 0000 0000
156 3 0000 0000
158 3 0000 0000
160 3 0000 0000
162 3 0000 0000
164 3 0000 0000
166 3 0000 0000
168 3 0000 0000
170 3 0000 0000

```

```

32 000 000000,000,000,000,000,000,
000,000,000,000,000,000
34 000 000000,000,000,000,000,000,
000,000,000,000,000,000
36 000 000000,000,000,000,000,000,
000,000,000,000,000,000
38 000 000000,000,000,000,000,000,
000,000,000,000,000,000
40 000 000000,000,000,000,000,000,
000,000,000,000,000,000
42 000 000000,000,000,000,000,000,
000,000,000,000,000,000
44 000 000000,000,000,000,000,000,
000,000,000,000,000,000
46 000 000000,000,000,000,000,000,
000,000,000,000,000,000
48 000 000000,000,000,000,000,000,
000,000,000,000,000,000
50 000 000000,000,000,000,000,000,
000,000,000,000,000,000
52 000 000000,000,000,000,000,000,
000,000,000,000,000,000
54 000 000000,000,000,000,000,000,
000,000,000,000,000,000
56 000 000000,000,000,000,000,000,
000,000,000,000,000,000
58 000 000000,000,000,000,000,000,
000,000,000,000,000,000
60 000 000000,000,000,000,000,000,
000,000,000,000,000,000
62 000 000000,000,000,000,000,000,
000,000,000,000,000,000
64 000 000000,000,000,000,000,000,
000,000,000,000,000,000
66 000 000000,000,000,000,000,000,
000,000,000,000,000,000
68 000 000000,000,000,000,000,000,
000,000,000,000,000,000
70 000 000000,000,000,000,000,000,
000,000,000,000,000,000
72 000 000000,000,000,000,000,000,
000,000,000,000,000,000
74 000 000000,000,000,000,000,000,
000,000,000,000,000,000
76 000 000000,000,000,000,000,000,
000,000,000,000,000,000
78 000 000000,000,000,000,000,000,
000,000,000,000,000,000
80 000 000000,000,000,000,000,000,
000,000,000,000,000,000
82 000 000000,000,000,000,000,000,
000,000,000,000,000,000
84 000 000000,000,000,000,000,000,
000,000,000,000,000,000
86 000 000000,000,000,000,000,000,
000,000,000,000,000,000
88 000 000000,000,000,000,000,000,
000,000,000,000,000,000
90 000 000000,000,000,000,000,000,
000,000,000,000,000,000
92 000 000000,000,000,000,000,000,
000,000,000,000,000,000
94 000 000000,000,000,000,000,000,
000,000,000,000,000,000
96 000 000000,000,000,000,000,000,
000,000,000,000,000,000
98 000 000000,000,000,000,000,000,
000,000,000,000,000,000

```

MAKING MUSIC



BITING 4018



FILENAME: ROM TYPWRITER

```

10          ORG 4018
20          ;
30          CLC          EQU 00344
40          PRINTSTR   EQU 400E0
50          PRINTED   EQU 000C0
60          PLOT      EQU 00FF0
70          GETIN    EQU 00FF4
80          PRINT    EQU 00710
90          ;
100         ;CLEAR THE SCREEN.
110         ;
120         ;      JRN CLC
130         ;
140         ;PRINT HEADER.
150         ;
160         ;      LDA #<TEXT
170         ;      LDY #>TEXT

```

```

180          ;      JRN PRINTSTR
190          ;
200          ;      SWITCH CG4 INTO BUSINESS MODE.
210          ;
220          ;      LDA #14
230          ;      JRN PRINT
240          ;
250          ;      SET PRINT POSITION TO BEGINNING.
260          ;
270          ;      CLC
280          ;      LDY #0
290          ;      LDY #0
300          ;      JRN PLOT
310          ;
320          ;      TURN CURSOR ON.
330          ;
340          ;      LDA #0
350          ;      STA #0
360          ;
370          ;
380          ;
390          ;      *** MAIN LOOP ***
400          ;
410          ;      GET KEYPRESS.
420          ;
430          ;      TESTKEY   JRN GETIN
440          ;      BRQ TESTKEY
450          ;
460          ;      IF F1. EXIT FROM PROGRAM.
470          ;
480          ;      CMP #103
490          ;      BRQ EXIT
500          ;
510          ;      IF COL#8 40 ADJUST TO NEXT LINE

```

520	:		1020	CLC	
530	:	LDD #02	1030	LDD #2	
540	:	CFR #30	1040	LDD #10	
550	:	BNR ROADJUST	1050	JSR PLOT	
560	:		1060		
570	:	PLA	1070	LDA #02	
580	:		1080	JSR PRINT	
590	:	CLC	1090	JSR PRINT	
600	:	LDD #02	1100	:	
610	:	INH	1110	:	PRINT CURRENT COLUMN NUMBER.
620	:	LDD #0	1120	:	
630	:	JSR PLOT	1130	NOESPACES	CLC
640	:		1140		LDD #2
650	:	PLA	1150		LDD #10
660	:		1160		JSR PLOT
670	:	PRINT ASCII CHARACTER OR EXECUTE	1170		
680	:	NO-PRINTABLE CHARACTER, SUCH AS	1180	LDD #02	
690	:	DELETE.	1190	LDA #0	
700	:		1200	JSR PRINTING	
710	:	ROADJUST	1210		
720	:	JSR PRINT	1220	:	RE-PLOT PRINT POSITION.
730	:		1230		
740	:	SAVE CURRENT PRINT POSITION.	1240		
750	:		1250	CLC	
760	:	LDA #0	1260	LDD #01	
770	:	STA #01	1270	LDD #02	
780	:	LDA #00	1280	JSR PLOT	
790	:	STA #02	1290		
800	:		1300	:	BACK TO TEXTKEY.
810	:	PRINT CURRENT LINE NO IN HEADER.	1310		
820	:	(CURRENT LINE - 0)	1320		JMP TEXTKEY
830	:		1330	:	
840	:	CLC	1340	:	
850	:	LDD #2	1350	:	*** MAIN EXIT ***
860	:	LDD #0	1360	:	
870	:	JSR PLOT	1370	:	RETURN TO BASIC.
880	:		1380	:	
890	:	SEC	1390	EXIT	RTS
900	:	LDA #01	1400	:	
910	:	SEC #1	1410	:	
920	:	TAR	1420	:	
930	:	LDA #0	1430	TEXT	BYT "
940	:	JSR PRINTING	1440		ROW "
950	:		1450	BYT "TYPERBITES	
960	:	IF BEGINNING OF NEW LINE, BLANK	1460	BYT "	,10
970	:	OUT COLUMN NUMBER.	1470	BYT "LINE: 1	COLUMN:
980	:		1480	BYT "	1
990	:	LDA #02	1490	BYT "	,10
1000	:	CFR #1	1500	BYT "	-----
1010	:	BNR NOESPACES	1510	BYT "	-----
	:			BYT "	-----,0

SHORT INTERLUDE



PROGRAM: SHORT INTERLUDE

AC	100	040000,040	80	100	040000,141,00,100,170,00,0
70	010	0000-0000,0000A,0-0-0-0	80	000	040010,07,100,00,001,00,0
		0000-1,0-0000		000	040010,00,100,00,001,00,0
74	000	0000-0000,0000,0000A	80	010	040010,00,100,00,001,00,0
		0000-1,0-0000		010	040010,100,000,00,001,00,0
78	000	0000-0000,0000-0000	80	020	040010,00,000,00,001,00,0
		0000-1,0-0000		020	040010,00,000,00,001,00,0
7C	000	040010,00,100,00,001,00,0	80	030	040010,00,000,00,001,00,0
		-100		030	040010,00,000,00,001,00,0
80	070	040010,00,141,01,0,100	80	040	040010,00,000,00,001,00,0
		-100		040	040010,00,000,00,001,00,0
84	100	040010,00,000,00,0,100	80	050	040010,00,000,00,001,00,0
		-100		050	040010,00,000,00,001,00,0
88	070	040010,00,141,01,0,100	80	060	040010,00,000,00,001,00,0
		-100		060	040010,00,000,00,001,00,0
8C	100	040010,00,000,00,0,100	80	070	040010,00,000,00,001,00,0
		-100		070	040010,00,000,00,001,00,0


```

10 1000 DATA
11 1100 DATA
12 1200 DATA
13 1300 DATA
14 1400 DATA
15 1500 DATA
16 1600 DATA
17 1700 DATA
18 1800 DATA
19 1900 DATA
20 2000 DATA
21 2100 DATA
22 2200 DATA
23 2300 DATA
24 2400 DATA
25 2500 DATA
26 2600 DATA
27 2700 DATA
28 2800 DATA
29 2900 DATA
30 3000 DATA
31 3100 DATA
32 3200 DATA
33 3300 DATA
34 3400 DATA
35 3500 DATA
36 3600 DATA
37 3700 DATA
38 3800 DATA
39 3900 DATA
40 4000 DATA

```

```

410 INPUT PROMPT "ENTER NAME OF FILE TO OPEN: ";
420 IF NOT AT END THEN RETURN
430 PRINT "NO FILES FOUND"
440 GOTO 410
450 PRINT "ENTER FILE NAME: ";
460 INPUT PROMPT "FILE NAME: ";
470 IF NOT AT END THEN RETURN
480 PRINT "NO FILES FOUND"
490 GOTO 410
500 PRINT "ENTER FILE NAME: ";
510 INPUT PROMPT "FILE NAME: ";
520 IF NOT AT END THEN RETURN
530 PRINT "NO FILES FOUND"
540 GOTO 410
550 PRINT "ENTER FILE NAME: ";
560 INPUT PROMPT "FILE NAME: ";
570 IF NOT AT END THEN RETURN
580 PRINT "NO FILES FOUND"
590 GOTO 410
600 PRINT "ENTER FILE NAME: ";
610 INPUT PROMPT "FILE NAME: ";
620 IF NOT AT END THEN RETURN
630 PRINT "NO FILES FOUND"
640 GOTO 410
650 PRINT "ENTER FILE NAME: ";
660 INPUT PROMPT "FILE NAME: ";
670 IF NOT AT END THEN RETURN
680 PRINT "NO FILES FOUND"
690 GOTO 410
700 PRINT "ENTER FILE NAME: ";
710 INPUT PROMPT "FILE NAME: ";
720 IF NOT AT END THEN RETURN
730 PRINT "NO FILES FOUND"
740 GOTO 410
750 PRINT "ENTER FILE NAME: ";
760 INPUT PROMPT "FILE NAME: ";
770 IF NOT AT END THEN RETURN
780 PRINT "NO FILES FOUND"
790 GOTO 410
800 PRINT "ENTER FILE NAME: ";
810 INPUT PROMPT "FILE NAME: ";
820 IF NOT AT END THEN RETURN
830 PRINT "NO FILES FOUND"
840 GOTO 410
850 PRINT "ENTER FILE NAME: ";
860 INPUT PROMPT "FILE NAME: ";
870 IF NOT AT END THEN RETURN
880 PRINT "NO FILES FOUND"
890 GOTO 410
900 PRINT "ENTER FILE NAME: ";
910 INPUT PROMPT "FILE NAME: ";
920 IF NOT AT END THEN RETURN
930 PRINT "NO FILES FOUND"
940 GOTO 410
950 PRINT "ENTER FILE NAME: ";
960 INPUT PROMPT "FILE NAME: ";
970 IF NOT AT END THEN RETURN
980 PRINT "NO FILES FOUND"
990 GOTO 410

```

```

990 PRINT "NO FILES FOUND"
1000 PRINT "ENTER FILE NAME: ";
1010 INPUT PROMPT "FILE NAME: ";
1020 IF NOT AT END THEN RETURN
1030 PRINT "NO FILES FOUND"
1040 GOTO 410
1050 PRINT "ENTER FILE NAME: ";
1060 INPUT PROMPT "FILE NAME: ";
1070 IF NOT AT END THEN RETURN
1080 PRINT "NO FILES FOUND"
1090 GOTO 410
1100 PRINT "ENTER FILE NAME: ";
1110 INPUT PROMPT "FILE NAME: ";
1120 IF NOT AT END THEN RETURN
1130 PRINT "NO FILES FOUND"
1140 GOTO 410
1150 PRINT "ENTER FILE NAME: ";
1160 INPUT PROMPT "FILE NAME: ";
1170 IF NOT AT END THEN RETURN
1180 PRINT "NO FILES FOUND"
1190 GOTO 410
1200 PRINT "ENTER FILE NAME: ";
1210 INPUT PROMPT "FILE NAME: ";
1220 IF NOT AT END THEN RETURN
1230 PRINT "NO FILES FOUND"
1240 GOTO 410
1250 PRINT "ENTER FILE NAME: ";
1260 INPUT PROMPT "FILE NAME: ";
1270 IF NOT AT END THEN RETURN
1280 PRINT "NO FILES FOUND"
1290 GOTO 410
1300 PRINT "ENTER FILE NAME: ";
1310 INPUT PROMPT "FILE NAME: ";
1320 IF NOT AT END THEN RETURN
1330 PRINT "NO FILES FOUND"
1340 GOTO 410
1350 PRINT "ENTER FILE NAME: ";
1360 INPUT PROMPT "FILE NAME: ";
1370 IF NOT AT END THEN RETURN
1380 PRINT "NO FILES FOUND"
1390 GOTO 410
1400 PRINT "ENTER FILE NAME: ";
1410 INPUT PROMPT "FILE NAME: ";
1420 IF NOT AT END THEN RETURN
1430 PRINT "NO FILES FOUND"
1440 GOTO 410
1450 PRINT "ENTER FILE NAME: ";
1460 INPUT PROMPT "FILE NAME: ";
1470 IF NOT AT END THEN RETURN
1480 PRINT "NO FILES FOUND"
1490 GOTO 410
1500 PRINT "ENTER FILE NAME: ";
1510 INPUT PROMPT "FILE NAME: ";
1520 IF NOT AT END THEN RETURN
1530 PRINT "NO FILES FOUND"
1540 GOTO 410
1550 PRINT "ENTER FILE NAME: ";
1560 INPUT PROMPT "FILE NAME: ";
1570 IF NOT AT END THEN RETURN
1580 PRINT "NO FILES FOUND"
1590 GOTO 410
1600 PRINT "ENTER FILE NAME: ";
1610 INPUT PROMPT "FILE NAME: ";
1620 IF NOT AT END THEN RETURN
1630 PRINT "NO FILES FOUND"
1640 GOTO 410
1650 PRINT "ENTER FILE NAME: ";
1660 INPUT PROMPT "FILE NAME: ";
1670 IF NOT AT END THEN RETURN
1680 PRINT "NO FILES FOUND"
1690 GOTO 410
1700 PRINT "ENTER FILE NAME: ";
1710 INPUT PROMPT "FILE NAME: ";
1720 IF NOT AT END THEN RETURN
1730 PRINT "NO FILES FOUND"
1740 GOTO 410
1750 PRINT "ENTER FILE NAME: ";
1760 INPUT PROMPT "FILE NAME: ";
1770 IF NOT AT END THEN RETURN
1780 PRINT "NO FILES FOUND"
1790 GOTO 410
1800 PRINT "ENTER FILE NAME: ";
1810 INPUT PROMPT "FILE NAME: ";
1820 IF NOT AT END THEN RETURN
1830 PRINT "NO FILES FOUND"
1840 GOTO 410
1850 PRINT "ENTER FILE NAME: ";
1860 INPUT PROMPT "FILE NAME: ";
1870 IF NOT AT END THEN RETURN
1880 PRINT "NO FILES FOUND"
1890 GOTO 410
1900 PRINT "ENTER FILE NAME: ";
1910 INPUT PROMPT "FILE NAME: ";
1920 IF NOT AT END THEN RETURN
1930 PRINT "NO FILES FOUND"
1940 GOTO 410
1950 PRINT "ENTER FILE NAME: ";
1960 INPUT PROMPT "FILE NAME: ";
1970 IF NOT AT END THEN RETURN
1980 PRINT "NO FILES FOUND"
1990 GOTO 410

```

SPLIT BAUD-RATE TERMINAL



PROGRAM, CBN DETROIT

```

10 FOR SPLIT BAUD RATE TERMINAL
20 PRINT "SPLIT BAUD RATE TERMINAL"
30 PRINT "SPLIT BAUD RATE TERMINAL"
40 PRINT "SPLIT BAUD RATE TERMINAL"
50 PRINT "SPLIT BAUD RATE TERMINAL"
60 PRINT "SPLIT BAUD RATE TERMINAL"
70 PRINT "SPLIT BAUD RATE TERMINAL"
80 PRINT "SPLIT BAUD RATE TERMINAL"
90 PRINT "SPLIT BAUD RATE TERMINAL"
100 PRINT "SPLIT BAUD RATE TERMINAL"
110 PRINT "SPLIT BAUD RATE TERMINAL"
120 PRINT "SPLIT BAUD RATE TERMINAL"
130 PRINT "SPLIT BAUD RATE TERMINAL"
140 PRINT "SPLIT BAUD RATE TERMINAL"
150 PRINT "SPLIT BAUD RATE TERMINAL"
160 PRINT "SPLIT BAUD RATE TERMINAL"
170 PRINT "SPLIT BAUD RATE TERMINAL"
180 PRINT "SPLIT BAUD RATE TERMINAL"
190 PRINT "SPLIT BAUD RATE TERMINAL"
200 PRINT "SPLIT BAUD RATE TERMINAL"
210 PRINT "SPLIT BAUD RATE TERMINAL"
220 PRINT "SPLIT BAUD RATE TERMINAL"
230 PRINT "SPLIT BAUD RATE TERMINAL"
240 PRINT "SPLIT BAUD RATE TERMINAL"
250 PRINT "SPLIT BAUD RATE TERMINAL"
260 PRINT "SPLIT BAUD RATE TERMINAL"
270 PRINT "SPLIT BAUD RATE TERMINAL"
280 PRINT "SPLIT BAUD RATE TERMINAL"
290 PRINT "SPLIT BAUD RATE TERMINAL"
300 PRINT "SPLIT BAUD RATE TERMINAL"
310 PRINT "SPLIT BAUD RATE TERMINAL"
320 PRINT "SPLIT BAUD RATE TERMINAL"
330 PRINT "SPLIT BAUD RATE TERMINAL"
340 PRINT "SPLIT BAUD RATE TERMINAL"
350 PRINT "SPLIT BAUD RATE TERMINAL"
360 PRINT "SPLIT BAUD RATE TERMINAL"
370 PRINT "SPLIT BAUD RATE TERMINAL"
380 PRINT "SPLIT BAUD RATE TERMINAL"
390 PRINT "SPLIT BAUD RATE TERMINAL"
400 PRINT "SPLIT BAUD RATE TERMINAL"

```

1350 DATAS27, 86, 40, 127, 201, 127, 8
92, 5, 128
1360 DATAS88, 20, 28, 201, 60, 176, 1,
68, 254
1370 DATAS91, 31, 178, 3, 105, 166, 88
102, 1021
1380 DATAS97, 178, 1, 68, 66, 332, 30, 5,
6, 187
1390 DATAS98, 2, 30, 289, 250, 32, 260
1, 103, 1189
1400 DATAS130, 20, 201, 268, 104, 59, 75
1, 242, 257
1410 DATAS136, 28, 201, 266, 104, 130, 32
8, 32, 80, 7
1420 DATAS139, 102, 20, 201, 269, 95, 3
8, 188, 1388
1430 DATAS158, 178, 104, 2, 202, 204, 1
87, 8, 1287
1440 DATAS160, 279, 199, 158, 2, 195, 1
65, 158, 2023
1450 DATAS168, 242, 181, 12, 74, 17, 67
8, 30, 1000
1460 DATAS226, 28, 242, 74, 82, 179, 1
8, 288, 394
1470 DATAS241, 1, 201, 173, 10, 102, 74
3, 3, 669
1480 DATAS246, 189, 199, 28, 38, 102, 3
8, 6, 887
1490 DATAS250, 158, 17, 141, 24, 201, 3
8, 103, 1008
1500 DATAS2, 141, 818, 188, 141, 208, 1
68, 166, 1850
1510 DATAS11, 141, 827, 128, 38, 208, 28
8, 207, 1880
1520 DATAS171, 274, 201, 173, 181, 8, 7
3, 178, 1129
1530 DATAS23, 28, 87, 189, 38, 282, 18
3, 178, 2128
1540 DATAS27, 174, 200, 199, 202, 17,
201, 21, 1880
1550 DATAS28, 888, 38, 108, 289, 170,
74, 168, 1987
1560 DATAS1, 28, 100, 108, 173, 207, 18
8, 32, 888
1570 DATAS20, 188, 173, 207, 189, 73,
252, 26, 1277
1580 DATAS26, 100, 38, 208, 283, 173,
208, 108, 1297
1590 DATAS2, 100, 100, 38, 208, 208, 8
01, 171, 1882
1600 DATAS24, 66, 173, 181, 8, 74, 178
1, 213, 1327
1610 DATAS37, 289, 38, 180, 183, 17
8, 46, 218
1620 DATAS174, 208, 199, 201, 14, 201,
21, 218, 1282
1630 DATAS2, 201, 8, 208, 289, 208, 80
7, 288, 1907
1640 DATAS20, 10, 110, 228, 208, 288, 7
8, 78, 1001
1650 DATAS83, 188, 7, 28, 180, 188, 38
1, 189, 328
1660 DATAS23, 178, 3, 174, 208, 288, 8
40, 84, 1303
1670 DATAS21, 5, 200, 227, 86, 173, 21
8, 128, 1328
1680 DATAS24, 2, 68, 66, 168, 8, 38, 18
8, 784
1690 DATAS25, 180, 188, 38, 207, 208,
287, 274, 1420
1700 DATAS28, 28, 188, 268, 71, 64, 28
8, 8, 288
1710 DATAS21, 208, 242, 28, 204, 288,
24, 88, 1281
1720 DATAS189, 1, 178, 218, 288, 188, 8
6, 206, 1113
1730 DATAS28, 242, 187, 204, 188, 76,
181, 183, 1438
1740 DATAS1188, 108, 174, 200, 188, 188
10, 21, 1182
1750 DATAS204, 288, 38, 128, 288, 78, 3
8, 183, 1478

1760 DATAS85, 281, 0, 274, 280, 174, 28
1, 208, 1088
1770 DATAS88, 201, 171, 240, 18, 208,
204, 188, 1687
1780 DATAS89, 238, 200, 208, 208, 208
1, 207, 24, 1303
1790 DATAS104, 24, 88, 88, 88, 88, 188,
0, 871
1800 DATAS111, 208, 199, 108, 108, 130
1, 201, 188, 1388
1810 DATAS118, 189, 214, 198, 78, 28, 1
58, 188, 1200
1820 DATAS129, 24, 108, 208, 188, 140,
208, 108, 1182
1830 DATAS158, 201, 208, 208, 88, 168,
81, 141, 1212
1840 DATAS123, 188, 188, 2, 141, 208, 1
88, 188, 1280
1850 DATAS11, 141, 827, 188, 28, 87, 188
1, 173, 1021
1860 DATAS13, 188, 38, 158, 188, 28, 8
88, 188, 1187
1870 DATAS21, 213, 174, 208, 188, 242
1, 27, 81, 1921
1880 DATAS2, 202, 7, 201, 0, 208, 108, 7
8, 888
1890 DATAS1, 189, 38, 188, 183, 178, 88
1, 174, 1698
1900 DATAS20, 188, 240, 78, 141, 208,
188, 28, 1893
1910 DATAS190, 188, 178, 70, 178, 208,
188, 208, 1688
1920 DATAS2, 242, 810, 188, 38, 188, 1
64, 178, 1788
1930 DATAS8, 274, 208, 189, 240, 201, 3
8, 182, 2188
1940 DATAS20, 178, 88, 174, 208, 188,
274, 70, 1878
1950 DATAS121, 201, 108, 173, 207, 188,
188, 208, 1838
1960 DATAS199, 208, 20, 73, 255, 282, 2
20, 102, 1282
1970 DATAS28, 88, 173, 208, 188, 208,
213, 188, 1418
1980 DATAS28, 24, 28, 188, 189, 208, 8
07, 188, 1288
1990 DATAS28, 8, 141, 218, 188, 78, 28,
8, 128, 1282
2000 DATAS28, 208, 188, 188, 81, 141,
213, 188, 1388
2010 DATAS18, 202, 108, 28, 8, 38, 188
1, 188, 1000
2020 DATAS28, 188, 0, 141, 208, 288, 18
8, 288, 1104
2030 DATAS20, 201, 38, 188, 183, 178,
23, 171, 1172
2040 DATAS28, 288, 242, 18, 188, 281,
287, 204, 1494
2050 DATAS28, 24, 108, 208, 188, 141,
208, 188, 1871
2060 DATAS28, 201, 208, 208, 24, 88, 8,
8, 98, 1188
2070 DATAS28, 8, 28, 207, 208, 188, 18
8, 188, 1327
2080 DATAS14, 188, 38, 210, 288, 208,
208, 277, 1583
2090 DATAS2, 204, 208, 38, 188, 0, 141
213, 1188
2100 DATAS199, 78, 201, 189, 188, 1, 28
1, 201, 1388
2110 DATAS188, 188, 0, 140, 218, 188, 1
78, 207, 1288
2120 DATAS188, 208, 207, 188, 38, 204,
288, 28, 1388
2130 DATAS171, 188, 108, 8, 38, 108, 28
8, 188, 2188
2140 DATAS2, 28, 207, 288, 187, 214, 28
8, 274, 1688
2150 DATAS2, 283, 255, 71, 84, 88, 8,
188, 208
2160 DATAS1, 141, 218, 288, 78, 201, 28
7, 207, 1688
2170 DATAS28, 208, 208, 240, 288, 188

2180 DATAS1, 171
2190 DATAS28, 188, 0, 28, 201, 208, 28
2, 0, 1274
2200 DATAS89, 214, 188, 208, 173, 201
1, 208, 208, 1818
2210 DATAS2, 38, 108, 188, 201, 10, 84
8, 8, 888
2220 DATAS28, 28, 108, 28, 71, 188, 38,
213, 838
2230 DATAS28, 208, 208, 188, 208, 208
1, 173, 213, 1711
2240 DATAS28, 240, 188, 28, 204, 255,
88, 78, 1882
2250 DATAS1, 15, 30, 88, 188, 188, 170
124, 172
2260 DATAS1, 240, 74, 74, 74, 74, 822,
10, 788
2270 DATAS2, 4, 108, 98, 188, 288, 24,
128, 788
2280 DATAS7, 158, 88, 28, 17, 189, 158
1, 38, 828
2290 DATAS20, 188, 138, 38, 213, 208,
78, 188, 1988
2300 DATAS188, 108, 188, 38, 188, 288,
173, 207, 1873
2310 DATAS188, 20, 78, 188, 188, 148, 3
68, 198, 1288
2320 DATAS28, 283, 188, 173, 208, 188,
38, 78, 1028
2330 DATAS188, 98, 187, 188, 288, 188,
38, 183, 1288
2340 DATAS188, 178, 207, 188, 38, 78, 1
91, 98, 1188
2350 DATAS12, 152, 27, 28, 207, 181, 13
8, 38, 1882
2360 DATAS12, 217, 188, 98, 181, 124,
28, 81, 174
2370 DATAS2, 0, 13, 13, 13, 100, 114, 1
04, 902
2380 DATAS13, 114, 28, 98, 118, 117, 1
08, 98, 1182
2390 DATAS21, 114, 38, 81, 38, 0, 18, 8
8, 187
2400 DATAS28, 121, 201, 187, 28, 128, 1
17, 208, 173
2410 DATAS28, 201, 114, 28, 81, 38, 0, 2
81, 124
2420 DATAS21, 128, 288, 180, 0, 177, 2
81, 242, 1482
2430 DATAS18, 28, 218, 288, 208, 208, 8
18, 103, 1281
2440 DATAS28, 208, 240, 38, 0, 0, 0, 0,
788
2450 DATAS2, 8, 8, 0, 0, 0, 0, 0, 0, 1

SPRITE LIBRARY

	PROGRAM	AIRCRAFT
A#	10	REP*****
B#	20	REP77 SPRITE LIBRARY
C#	30	REP*****
D#	40	REP77 AIRCRAFT SPLITES

LISTINGS

CE 80 80174 ***** 80 80175 ***** 80 80176 ***** 80 80177 ***** 80 80178 ***** 80 80179 ***** 80 80180 ***** 80 80181 ***** 80 80182 ***** 80 80183 ***** 80 80184 ***** 80 80185 ***** 80 80186 ***** 80 80187 ***** 80 80188 ***** 80 80189 ***** 80 80190 ***** 80 80191 ***** 80 80192 ***** 80 80193 ***** 80 80194 ***** 80 80195 ***** 80 80196 ***** 80 80197 ***** 80 80198 ***** 80 80199 *****	80 80200 ***** 80 80201 ***** 80 80202 ***** 80 80203 ***** 80 80204 ***** 80 80205 ***** 80 80206 ***** 80 80207 ***** 80 80208 ***** 80 80209 ***** 80 80210 ***** 80 80211 ***** 80 80212 ***** 80 80213 ***** 80 80214 ***** 80 80215 ***** 80 80216 ***** 80 80217 ***** 80 80218 ***** 80 80219 ***** 80 80220 ***** 80 80221 ***** 80 80222 ***** 80 80223 ***** 80 80224 ***** 80 80225 ***** 80 80226 ***** 80 80227 ***** 80 80228 ***** 80 80229 ***** 80 80230 ***** 80 80231 ***** 80 80232 ***** 80 80233 ***** 80 80234 ***** 80 80235 ***** 80 80236 ***** 80 80237 ***** 80 80238 ***** 80 80239 ***** 80 80240 ***** 80 80241 ***** 80 80242 ***** 80 80243 ***** 80 80244 ***** 80 80245 ***** 80 80246 ***** 80 80247 ***** 80 80248 ***** 80 80249 ***** 80 80250 *****	80 80251 ***** 80 80252 ***** 80 80253 ***** 80 80254 ***** 80 80255 ***** 80 80256 ***** 80 80257 ***** 80 80258 ***** 80 80259 ***** 80 80260 ***** 80 80261 ***** 80 80262 ***** 80 80263 ***** 80 80264 ***** 80 80265 ***** 80 80266 ***** 80 80267 ***** 80 80268 ***** 80 80269 ***** 80 80270 ***** 80 80271 ***** 80 80272 ***** 80 80273 ***** 80 80274 ***** 80 80275 ***** 80 80276 ***** 80 80277 ***** 80 80278 ***** 80 80279 ***** 80 80280 ***** 80 80281 ***** 80 80282 ***** 80 80283 ***** 80 80284 ***** 80 80285 ***** 80 80286 ***** 80 80287 ***** 80 80288 ***** 80 80289 ***** 80 80290 ***** 80 80291 ***** 80 80292 ***** 80 80293 ***** 80 80294 ***** 80 80295 ***** 80 80296 ***** 80 80297 ***** 80 80298 ***** 80 80299 ***** 80 80300 *****	80 80301 ***** 80 80302 ***** 80 80303 ***** 80 80304 ***** 80 80305 ***** 80 80306 ***** 80 80307 ***** 80 80308 ***** 80 80309 ***** 80 80310 ***** 80 80311 ***** 80 80312 ***** 80 80313 ***** 80 80314 ***** 80 80315 ***** 80 80316 ***** 80 80317 ***** 80 80318 ***** 80 80319 ***** 80 80320 ***** 80 80321 ***** 80 80322 ***** 80 80323 ***** 80 80324 ***** 80 80325 ***** 80 80326 ***** 80 80327 ***** 80 80328 ***** 80 80329 ***** 80 80330 ***** 80 80331 ***** 80 80332 ***** 80 80333 ***** 80 80334 ***** 80 80335 ***** 80 80336 ***** 80 80337 ***** 80 80338 ***** 80 80339 ***** 80 80340 ***** 80 80341 ***** 80 80342 ***** 80 80343 ***** 80 80344 ***** 80 80345 ***** 80 80346 ***** 80 80347 ***** 80 80348 ***** 80 80349 ***** 80 80350 *****
---	--	--	--


```

71 8000 DATA 5.2705,0.0,0.04,0.0,0.0
   5.0,0.107,0.0,0.000,0.0,0.000
72 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
73 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
74 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
75 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
76 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
77 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
78 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
79 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
80 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
81 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
82 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
83 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
84 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
85 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
86 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
87 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
88 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
89 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
90 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
91 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
92 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
93 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
94 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
95 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
96 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
97 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
98 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
99 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
100 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0

```

```

31 1000
32 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
33 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
34 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
35 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
36 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
37 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
38 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
39 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0
40 8000 DATA 0.0,0.0,0.0,0.0,0.0,0.0
   0.0,0.0,0.0,0.0,0.0,0.0

```

SPRITE LIBRARY



AIRCRAFT DISPLAY

```

85 10 00000000000000000000000000000000
86
87 30 0000 SPRITE LIBRARY DISPLAY
   00
88 30 0000 AIRCRAFT
   00
89 40 00000000000000000000000000000000
90
91 80 00000,0,0,00000,00,000,0,0,0,0,0,0
   170-170,000,000,000,0,0,0,0,0,0,0,0
92 80 00000000000000000000000000000000
93 80 00000000000000000000000000000000
94 80 00000000000000000000000000000000
95 80 00000000000000000000000000000000
96 80 00000000000000000000000000000000
97 80 00000000000000000000000000000000
98 80 00000000000000000000000000000000
99 80 00000000000000000000000000000000
100 80 00000000000000000000000000000000

```

EDUCATING SIDNEY



PROGRAM - SID 1000

```

40 5 000 000 000 000 000 000 000 000
41 10 00000000000000000000000000000000
42 20 00000000000000000000000000000000
43 30 00000000000000000000000000000000
44 40 00000000000000000000000000000000
45 50 00000000000000000000000000000000
46 60 00000000000000000000000000000000
47 70 00000000000000000000000000000000
48 80 00000000000000000000000000000000
49 90 00000000000000000000000000000000
50 100 00000000000000000000000000000000
51 110 00000000000000000000000000000000
52 120 00000000000000000000000000000000
53 130 00000000000000000000000000000000
54 140 00000000000000000000000000000000
55 150 00000000000000000000000000000000

```

EDUCATING SYDNEY



PROGRAM - SID1000

```

57 30 000 0000000000000000000000000000
58 40 000 0000000000000000000000000000
59 50 000 0000000000000000000000000000
60 60 000 0000000000000000000000000000
61 70 000 0000000000000000000000000000
62 80 000 0000000000000000000000000000
63 90 000 0000000000000000000000000000
64 100 000 0000000000000000000000000000
65 110 000 0000000000000000000000000000
66 120 000 0000000000000000000000000000
67 130 000 0000000000000000000000000000
68 140 000 0000000000000000000000000000
69 150 000 0000000000000000000000000000
70 160 000 0000000000000000000000000000
71 170 000 0000000000000000000000000000
72 180 000 0000000000000000000000000000
73 190 000 0000000000000000000000000000
74 200 000 0000000000000000000000000000
75 210 000 0000000000000000000000000000
76 220 000 0000000000000000000000000000
77 230 000 0000000000000000000000000000
78 240 000 0000000000000000000000000000
79 250 000 0000000000000000000000000000
80 260 000 0000000000000000000000000000
81 270 000 0000000000000000000000000000
82 280 000 0000000000000000000000000000
83 290 000 0000000000000000000000000000
84 300 000 0000000000000000000000000000
85 310 000 0000000000000000000000000000
86 320 000 0000000000000000000000000000
87 330 000 0000000000000000000000000000
88 340 000 0000000000000000000000000000
89 350 000 0000000000000000000000000000

```

LISTINGS

```

77 360 OPEN:"T:TWINDO
88 400 INPUT TILEDRAW:ITAX:IS
  CDEF:CF
89 400 PRINT:PRINT:G:KEY:WHEN:R
  END
90 400 SETPOS:POS:"TWINDO
  IS:300:OPEN:R:L:FR
91 400 PRINT:CLOSED:OPEN:CL
92
93 400 PRINT:PRINT:CLOSED:OPEN:CL
94 400 PRINT:PRINT:CLOSED:OPEN:CL
95 400 PRINT:PRINT:CLOSED:OPEN:CL
96 400 PRINT:PRINT:CLOSED:OPEN:CL
97 400 PRINT:PRINT:CLOSED:OPEN:CL
98 400 PRINT:PRINT:CLOSED:OPEN:CL
99 400 PRINT:PRINT:CLOSED:OPEN:CL
100 400 PRINT:PRINT:CLOSED:OPEN:CL
101 400 PRINT:PRINT:CLOSED:OPEN:CL
102 400 PRINT:PRINT:CLOSED:OPEN:CL
103 400 PRINT:PRINT:CLOSED:OPEN:CL
104 400 PRINT:PRINT:CLOSED:OPEN:CL
105 400 PRINT:PRINT:CLOSED:OPEN:CL
106 400 PRINT:PRINT:CLOSED:OPEN:CL
107 400 PRINT:PRINT:CLOSED:OPEN:CL
108 400 PRINT:PRINT:CLOSED:OPEN:CL
109 400 PRINT:PRINT:CLOSED:OPEN:CL
110 400 PRINT:PRINT:CLOSED:OPEN:CL
111 400 PRINT:PRINT:CLOSED:OPEN:CL
112 400 PRINT:PRINT:CLOSED:OPEN:CL
113 400 PRINT:PRINT:CLOSED:OPEN:CL
114 400 PRINT:PRINT:CLOSED:OPEN:CL
115 400 PRINT:PRINT:CLOSED:OPEN:CL
116 400 PRINT:PRINT:CLOSED:OPEN:CL
117 400 PRINT:PRINT:CLOSED:OPEN:CL
118 400 PRINT:PRINT:CLOSED:OPEN:CL
119 400 PRINT:PRINT:CLOSED:OPEN:CL
120 400 PRINT:PRINT:CLOSED:OPEN:CL

```

```

370 PRINT:PRINT:CLOSED:OPEN:CL
380 PRINT:PRINT:CLOSED:OPEN:CL
390 PRINT:PRINT:CLOSED:OPEN:CL
400 PRINT:PRINT:CLOSED:OPEN:CL
410 PRINT:PRINT:CLOSED:OPEN:CL
420 PRINT:PRINT:CLOSED:OPEN:CL
430 PRINT:PRINT:CLOSED:OPEN:CL
440 PRINT:PRINT:CLOSED:OPEN:CL
450 PRINT:PRINT:CLOSED:OPEN:CL
460 PRINT:PRINT:CLOSED:OPEN:CL
470 PRINT:PRINT:CLOSED:OPEN:CL
480 PRINT:PRINT:CLOSED:OPEN:CL
490 PRINT:PRINT:CLOSED:OPEN:CL
500 PRINT:PRINT:CLOSED:OPEN:CL
510 PRINT:PRINT:CLOSED:OPEN:CL
520 PRINT:PRINT:CLOSED:OPEN:CL
530 PRINT:PRINT:CLOSED:OPEN:CL
540 PRINT:PRINT:CLOSED:OPEN:CL
550 PRINT:PRINT:CLOSED:OPEN:CL
560 PRINT:PRINT:CLOSED:OPEN:CL
570 PRINT:PRINT:CLOSED:OPEN:CL
580 PRINT:PRINT:CLOSED:OPEN:CL
590 PRINT:PRINT:CLOSED:OPEN:CL
600 PRINT:PRINT:CLOSED:OPEN:CL
610 PRINT:PRINT:CLOSED:OPEN:CL
620 PRINT:PRINT:CLOSED:OPEN:CL
630 PRINT:PRINT:CLOSED:OPEN:CL
640 PRINT:PRINT:CLOSED:OPEN:CL
650 PRINT:PRINT:CLOSED:OPEN:CL
660 PRINT:PRINT:CLOSED:OPEN:CL
670 PRINT:PRINT:CLOSED:OPEN:CL
680 PRINT:PRINT:CLOSED:OPEN:CL
690 PRINT:PRINT:CLOSED:OPEN:CL
700 PRINT:PRINT:CLOSED:OPEN:CL
710 PRINT:PRINT:CLOSED:OPEN:CL
720 PRINT:PRINT:CLOSED:OPEN:CL
730 PRINT:PRINT:CLOSED:OPEN:CL
740 PRINT:PRINT:CLOSED:OPEN:CL
750 PRINT:PRINT:CLOSED:OPEN:CL
760 PRINT:PRINT:CLOSED:OPEN:CL
770 PRINT:PRINT:CLOSED:OPEN:CL
780 PRINT:PRINT:CLOSED:OPEN:CL
790 PRINT:PRINT:CLOSED:OPEN:CL
800 PRINT:PRINT:CLOSED:OPEN:CL
810 PRINT:PRINT:CLOSED:OPEN:CL
820 PRINT:PRINT:CLOSED:OPEN:CL
830 PRINT:PRINT:CLOSED:OPEN:CL
840 PRINT:PRINT:CLOSED:OPEN:CL
850 PRINT:PRINT:CLOSED:OPEN:CL
860 PRINT:PRINT:CLOSED:OPEN:CL
870 PRINT:PRINT:CLOSED:OPEN:CL
880 PRINT:PRINT:CLOSED:OPEN:CL
890 PRINT:PRINT:CLOSED:OPEN:CL
900 PRINT:PRINT:CLOSED:OPEN:CL
910 PRINT:PRINT:CLOSED:OPEN:CL
920 PRINT:PRINT:CLOSED:OPEN:CL
930 PRINT:PRINT:CLOSED:OPEN:CL
940 PRINT:PRINT:CLOSED:OPEN:CL
950 PRINT:PRINT:CLOSED:OPEN:CL
960 PRINT:PRINT:CLOSED:OPEN:CL
970 PRINT:PRINT:CLOSED:OPEN:CL
980 PRINT:PRINT:CLOSED:OPEN:CL
990 PRINT:PRINT:CLOSED:OPEN:CL
1000 PRINT:PRINT:CLOSED:OPEN:CL

```

```

1010 PRINT:PRINT:CLOSED:OPEN:CL
1020 PRINT:PRINT:CLOSED:OPEN:CL
1030 PRINT:PRINT:CLOSED:OPEN:CL
1040 PRINT:PRINT:CLOSED:OPEN:CL
1050 PRINT:PRINT:CLOSED:OPEN:CL
1060 PRINT:PRINT:CLOSED:OPEN:CL
1070 PRINT:PRINT:CLOSED:OPEN:CL
1080 PRINT:PRINT:CLOSED:OPEN:CL
1090 PRINT:PRINT:CLOSED:OPEN:CL
1100 PRINT:PRINT:CLOSED:OPEN:CL
1110 PRINT:PRINT:CLOSED:OPEN:CL
1120 PRINT:PRINT:CLOSED:OPEN:CL
1130 PRINT:PRINT:CLOSED:OPEN:CL
1140 PRINT:PRINT:CLOSED:OPEN:CL
1150 PRINT:PRINT:CLOSED:OPEN:CL
1160 PRINT:PRINT:CLOSED:OPEN:CL
1170 PRINT:PRINT:CLOSED:OPEN:CL
1180 PRINT:PRINT:CLOSED:OPEN:CL
1190 PRINT:PRINT:CLOSED:OPEN:CL
1200 PRINT:PRINT:CLOSED:OPEN:CL
1210 PRINT:PRINT:CLOSED:OPEN:CL
1220 PRINT:PRINT:CLOSED:OPEN:CL
1230 PRINT:PRINT:CLOSED:OPEN:CL
1240 PRINT:PRINT:CLOSED:OPEN:CL
1250 PRINT:PRINT:CLOSED:OPEN:CL
1260 PRINT:PRINT:CLOSED:OPEN:CL
1270 PRINT:PRINT:CLOSED:OPEN:CL
1280 PRINT:PRINT:CLOSED:OPEN:CL
1290 PRINT:PRINT:CLOSED:OPEN:CL
1300 PRINT:PRINT:CLOSED:OPEN:CL
1310 PRINT:PRINT:CLOSED:OPEN:CL
1320 PRINT:PRINT:CLOSED:OPEN:CL
1330 PRINT:PRINT:CLOSED:OPEN:CL
1340 PRINT:PRINT:CLOSED:OPEN:CL
1350 PRINT:PRINT:CLOSED:OPEN:CL
1360 PRINT:PRINT:CLOSED:OPEN:CL
1370 PRINT:PRINT:CLOSED:OPEN:CL
1380 PRINT:PRINT:CLOSED:OPEN:CL
1390 PRINT:PRINT:CLOSED:OPEN:CL
1400 PRINT:PRINT:CLOSED:OPEN:CL
1410 PRINT:PRINT:CLOSED:OPEN:CL
1420 PRINT:PRINT:CLOSED:OPEN:CL
1430 PRINT:PRINT:CLOSED:OPEN:CL
1440 PRINT:PRINT:CLOSED:OPEN:CL
1450 PRINT:PRINT:CLOSED:OPEN:CL
1460 PRINT:PRINT:CLOSED:OPEN:CL
1470 PRINT:PRINT:CLOSED:OPEN:CL
1480 PRINT:PRINT:CLOSED:OPEN:CL
1490 PRINT:PRINT:CLOSED:OPEN:CL
1500 PRINT:PRINT:CLOSED:OPEN:CL

```

SPLIT BAUD RATE TERMINAL



PROGRAM: CIBS CONTROL

```

10 360 SPLIT BAUD RATE TERMINAL
  PROGRAM
20 360 BY M-I SELLERS (CIBS)
  30 360
  40 360
  50 360
  60 360
  70 360
  80 360
  90 360
  100 360
  110 360
  120 360
  130 360
  140 360
  150 360
  160 360
  170 360
  180 360
  190 360
  200 360
  210 360
  220 360
  230 360
  240 360
  250 360
  260 360
  270 360
  280 360
  290 360
  300 360
  310 360
  320 360
  330 360
  340 360
  350 360
  360 360
  370 360
  380 360
  390 360
  400 360
  410 360
  420 360
  430 360
  440 360
  450 360
  460 360
  470 360
  480 360
  490 360
  500 360
  510 360
  520 360
  530 360
  540 360
  550 360
  560 360
  570 360
  580 360
  590 360
  600 360
  610 360
  620 360
  630 360
  640 360
  650 360
  660 360
  670 360
  680 360
  690 360
  700 360
  710 360
  720 360
  730 360
  740 360
  750 360
  760 360
  770 360
  780 360
  790 360
  800 360
  810 360
  820 360
  830 360
  840 360
  850 360
  860 360
  870 360
  880 360
  890 360
  900 360
  910 360
  920 360
  930 360
  940 360
  950 360
  960 360
  970 360
  980 360
  990 360
  1000 360

```


LISTINGS

PROGRAM: SIMPLE LOADER

```

10 SCROLL IF PRINTING=0 THEN P
20 PRINT "PROGRAM - SIMPLE PROGRAM, A
30 ENTER ADDRESS - STOP
40 FOR ADDRESS TO 1000: READ S:FOR
5 A,D,C,D,C,REST A
60 PRINT "SECTION 1 ", IF C=0 THEN
PRINT "REST A" ELSEPRINT "REST B"
70 PRINT "1000-0000"
80 FOR A=0 TO 1:FOR B=0 TO 1:FOR C=0 TO 1
9 A,D,C,D=C,REST A
100 PRINT "SECTION 2 ", IF C=0 THEN
PRINT "REST B" ELSEPRINT "REST A"
110 PRINT "0000-0000"
120 FOR A=0 TO 1:FOR B=0 TO 1:FOR C=0 TO 1
13 A,D,C,D=C,REST A
140 PRINT "SECTION 3 ", IF C=0 THEN
PRINT "REST A" ELSEPRINT "REST B"
150 PRINT "0000-0000"
160 FOR A=0 TO 1:FOR B=0 TO 1:FOR C=0 TO 1
17 A,D,C,D=C,REST A
180 PRINT "SECTION 4 "
190 PRINT "0000-0000"
200 PRINT "SECTION 5 "
210 PRINT "0000-0000"
220 PRINT "0000-0000"
230 PRINT "0000-0000"
240 PRINT "0000-0000"
250 PRINT "0000-0000"
260 PRINT "0000-0000"
270 PRINT "0000-0000"
280 PRINT "0000-0000"
290 PRINT "0000-0000"
300 PRINT "0000-0000"
310 PRINT "0000-0000"
320 PRINT "0000-0000"
330 PRINT "0000-0000"
340 PRINT "0000-0000"
350 PRINT "0000-0000"
360 PRINT "0000-0000"
370 PRINT "0000-0000"
380 PRINT "0000-0000"
390 PRINT "0000-0000"
400 PRINT "0000-0000"
410 PRINT "0000-0000"
420 PRINT "0000-0000"
430 PRINT "0000-0000"
440 PRINT "0000-0000"
450 PRINT "0000-0000"
460 PRINT "0000-0000"
470 PRINT "0000-0000"
480 PRINT "0000-0000"
490 PRINT "0000-0000"
500 PRINT "0000-0000"
510 PRINT "0000-0000"
520 PRINT "0000-0000"
530 PRINT "0000-0000"
540 PRINT "0000-0000"
550 PRINT "0000-0000"
560 PRINT "0000-0000"
570 PRINT "0000-0000"
580 PRINT "0000-0000"
590 PRINT "0000-0000"
600 PRINT "0000-0000"
610 PRINT "0000-0000"
620 PRINT "0000-0000"
630 PRINT "0000-0000"
640 PRINT "0000-0000"
650 PRINT "0000-0000"
660 PRINT "0000-0000"
670 PRINT "0000-0000"
680 PRINT "0000-0000"
690 PRINT "0000-0000"
700 PRINT "0000-0000"
710 PRINT "0000-0000"
720 PRINT "0000-0000"
730 PRINT "0000-0000"
740 PRINT "0000-0000"
750 PRINT "0000-0000"
760 PRINT "0000-0000"
770 PRINT "0000-0000"
780 PRINT "0000-0000"
790 PRINT "0000-0000"
800 PRINT "0000-0000"
810 PRINT "0000-0000"
820 PRINT "0000-0000"
830 PRINT "0000-0000"
840 PRINT "0000-0000"
850 PRINT "0000-0000"
860 PRINT "0000-0000"
870 PRINT "0000-0000"
880 PRINT "0000-0000"
890 PRINT "0000-0000"
900 PRINT "0000-0000"
910 PRINT "0000-0000"
920 PRINT "0000-0000"
930 PRINT "0000-0000"
940 PRINT "0000-0000"
950 PRINT "0000-0000"
960 PRINT "0000-0000"
970 PRINT "0000-0000"
980 PRINT "0000-0000"
990 PRINT "0000-0000"
1000 PRINT "0000-0000"

```

```

3600 DATA 24,138,101,808,804,84,1
3700 DATA 184,808,188,808,80,870,83
3800 DATA 808,101,808,100,808,144
3900 DATA 188,80,188,8,188,800,
188
4000 DATA 100,800,810,188,800,148
4100 DATA 188,188,88,148,800,138,888,
84,188
4200 DATA 810,808,808,18,188,0,18
4300 DATA 148,800,184,818,188,189,14,
808
4400 DATA 81,188,800,138,80,138,0
4500 DATA 188,800,808,808,800,180,
0
4600 DATA 188,800,808,808,184,810
4700 DATA 188,800,138,188,809,148,
180,138
4800 DATA 808,800,800,808,188,808
4900 DATA 7,84,100,810,88,800,8,
91
5000 DATA 808,7,808,808,188,808,1
5100 DATA 7,101,81,18,808,8,141,88
8
5200 DATA 7,808,187,78,810,808,38
487,807,188,884,80,184,8,188,14,
808,188,18,134,184,170,188,88,8,
78,188,187,188,808,8,178,870,13
3,817
5300 DATA 88,84,184,808,188,817,8
5400 DATA 84,188,808,188,808,188,
808
5500 DATA 184,818,808,807,88,808,
808
5600 DATA 141,808,808,84,188,808,10
1,808
5700 DATA 138,808,808,808,800,180,1
808,0,173,188,847,148,808,80,80,
178
5800 DATA 188,807,178,808,178,88,
808,88,88,88,188,187,848,188,88,
88
5900 DATA 88,181,80,88,188,808,38
808
6000 DATA 8,88,188,808,804,88,808
6100 DATA 18,188,818,808,18,88,808,
188
6200 DATA 818,78,87,88,188,808,80
1,84,808,14,188,818,801,88,840,8,
808
6300 DATA 18,78,88,808,818,78,87,
88,188,808,801,808,18,817,87
6400 DATA 808,801,808,808,188,817
6500 DATA 808,187,78,88,80,188,81
6600 DATA 80,188,808,801,88,808,17
8
6700 DATA 188,807,804,88,848,178,
88,78,88,808,817,78,87,88,88,818
6800 DATA 80,188,8,88,808,808,808,
188,138,808,880,88,84,147,13
8
6900 DATA 808,88,188,187,184,808,
808,4,744,8,188,14,78,131,184,88
7000 DATA 188,187,188,808,88,178,
817,188,808,137,81,87,880,88,808,
811
7100 DATA 188,84,138,818,808,808,
818,18,87,188,811,808,7,138,811,
174
7200 DATA 8,808,818,188,808,78,84
7300 DATA 808,188,808,0,148,81,138,1
74
7500 DATA 84,148,811,808,808,848,
177,84,148,804,188,8,188,88,188,
6300 DATA 188,88,138,80,188,0,133
7600 DATA 188,88,138,80,188,0,133
7700 DATA 188,88,138,80,188,0,133
7800 DATA 188,88,138,80,188,0,133
7900 DATA 188,88,138,80,188,0,133
8000 DATA 188,88,138,80,188,0,133
8100 DATA 188,88,138,80,188,0,133
8200 DATA 188,88,138,80,188,0,133
8300 DATA 188,88,138,80,188,0,133
8400 DATA 188,88,138,80,188,0,133
8500 DATA 188,88,138,80,188,0,133
8600 DATA 188,88,138,80,188,0,133
8700 DATA 188,88,138,80,188,0,133
8800 DATA 188,88,138,80,188,0,133
8900 DATA 188,88,138,80,188,0,133
9000 DATA 188,88,138,80,188,0,133
9100 DATA 188,88,138,80,188,0,133
9200 DATA 188,88,138,80,188,0,133
9300 DATA 188,88,138,80,188,0,133
9400 DATA 188,88,138,80,188,0,133
9500 DATA 188,88,138,80,188,0,133
9600 DATA 188,88,138,80,188,0,133
9700 DATA 188,88,138,80,188,0,133
9800 DATA 188,88,138,80,188,0,133
9900 DATA 188,88,138,80,188,0,133
10000 DATA 188,88,138,80,188,0,133

```

```

6300 DATA 188,88,138,80,188,0,133
7600 DATA 188,88,138,80,188,0,133
7700 DATA 188,88,138,80,188,0,133
7800 DATA 188,88,138,80,188,0,133
7900 DATA 188,88,138,80,188,0,133
8000 DATA 188,88,138,80,188,0,133
8100 DATA 188,88,138,80,188,0,133
8200 DATA 188,88,138,80,188,0,133
8300 DATA 188,88,138,80,188,0,133
8400 DATA 188,88,138,80,188,0,133
8500 DATA 188,88,138,80,188,0,133
8600 DATA 188,88,138,80,188,0,133
8700 DATA 188,88,138,80,188,0,133
8800 DATA 188,88,138,80,188,0,133
8900 DATA 188,88,138,80,188,0,133
9000 DATA 188,88,138,80,188,0,133
9100 DATA 188,88,138,80,188,0,133
9200 DATA 188,88,138,80,188,0,133
9300 DATA 188,88,138,80,188,0,133
9400 DATA 188,88,138,80,188,0,133
9500 DATA 188,88,138,80,188,0,133
9600 DATA 188,88,138,80,188,0,133
9700 DATA 188,88,138,80,188,0,133
9800 DATA 188,88,138,80,188,0,133
9900 DATA 188,88,138,80,188,0,133
10000 DATA 188,88,138,80,188,0,133

```

LISTINGS

0, 18, 19, 206, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

880 PRINTSPC(15) (ROWS) Ca POE
 810a 810b 811a 812 CRUSPPT(1) PRC
 813 CRUSPPT(1) PRC Ca 814 815
 816 817

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 RETURN

890 PRINTSPC(15) (ROWS) Ca POE
 810a 810b 811 CRUSPPT(1) PRINTED
 CRUSPPT(1) PRC Ca 812a 813a 814
 815 816 817 818 819 820 821 822
 823 824 825 826 827 828 829 830
 831 832 833 834 835 836 837 838
 839 840 841 842 843 844 845 846
 847 848 849 850 851 852 853 854
 855 856 857 858 859 860 861 862
 863 864 865 866 867 868 869 870
 871 872 873 874 875 876 877 878
 879 880 881 882 883 884 885 886
 887 888 889 890 891 892 893 894
 895 896 897 898 899 900 901 902
 903 904 905 906 907 908 909 910
 911 912 913 914 915 916 917 918
 919 920 921 922 923 924 925 926
 927 928 929 930 931 932 933 934
 935 936 937 938 939 940 941 942
 943 944 945 946 947 948 949 950
 951 952 953 954 955 956 957 958
 959 960 961 962 963 964 965 966
 967 968 969 970 971 972 973 974
 975 976 977 978 979 980 981 982
 983 984 985 986 987 988 989 990
 991 992 993 994 995 996 997 998
 999 1000

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 RETURN

890 PRINTSPC(15) (ROWS) Ca POE
 810a 810b 811 CRUSPPT(1) PRINTED
 CRUSPPT(1) PRC Ca 812a 813a 814
 815 816 817 818 819 820 821 822
 823 824 825 826 827 828 829 830
 831 832 833 834 835 836 837 838
 839 840 841 842 843 844 845 846
 847 848 849 850 851 852 853 854
 855 856 857 858 859 860 861 862
 863 864 865 866 867 868 869 870
 871 872 873 874 875 876 877 878
 879 880 881 882 883 884 885 886
 887 888 889 890 891 892 893 894
 895 896 897 898 899 900 901 902
 903 904 905 906 907 908 909 910
 911 912 913 914 915 916 917 918
 919 920 921 922 923 924 925 926
 927 928 929 930 931 932 933 934
 935 936 937 938 939 940 941 942
 943 944 945 946 947 948 949 950
 951 952 953 954 955 956 957 958
 959 960 961 962 963 964 965 966
 967 968 969 970 971 972 973 974
 975 976 977 978 979 980 981 982
 983 984 985 986 987 988 989 990
 991 992 993 994 995 996 997 998
 999 1000

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

890 PRINTSPC(15) (NO) COPIES TO A
 CRUSPPT

890 PRINT (ADDRESS) FUNCTION: Ca
 890A SCREEN FROM Ca 810a 811
 Ca 812

YOUR COMMODORE

Lineage: 50p per word
 (including VAT)

Serial display: £16.50 plus VAT per single column
 centimetre minimum 30cm. Ring for information on
 series bookings/discounts.

All advertisements in this section must be prepaid.
 Advertisements are accepted subject to the terms and conditions
 printed on the advertisement rate card (available on request).



01-437 0699

Send your requirements to
ALAN COLE
 ASP LTD, 1 GOLDEN SQUARE,
 LONDON W1

DISCS

SPECIAL OFFERS TO CELEBRATE OPENING OUR NEW SHOP!

3.5 in Discs (Double sided, double density) 80p
 (Min QTY 10) incl VAT & P&P
 5.25 in Discs (Double sided, Double density) 48 SP1 40p
 (min QTY 20) incl VAT & P&P

ATHENE CONSULTANTS (MEDIA CENTRE),
 16 Stoke Road, Gosport, Hampshire PO12 1JB

TEL: 0329 262063 (24 hr Credit Card Hotline)
 OR 0705 511 439 (Office Hours)

MIDI PRODUCTS



INNOVATIVE 'MIDI' SOFTWARE PRODUCTS

See April '88 Review, Page 68

Take advantage of our low prices... send SMI for free
 leaflets to:

**SONUS INTL LTD, PO Box 16,
 Wokingham RG11 4BP Berks
 Tel: (0734) 793599**

Repairs Guide

CALL
ALAN COLE

ON

01-437 0699

PROGRESS WITH YOUR COMMODORE ?
FOR FAST, RELIABLE AND PROFESSIONAL
REPAIRS AT COMPETITIVE PRICES

P M ENGINEERING

W&A

UNIT 4, NEW ROAD, ST. PEE, CAMBRIDGESHIRE, PTT7 - 8ND

ST. PEE (0450) 6266

We can also supply Hardware, Software, Blank Disks, Accessories and Spare parts.

COMMODORE SPARES & REPAIRS

COMMODORE 400	£85.00
COMMODORE 402	£95.00
COMMODORE 404	£105.00
COMMODORE 406	£115.00
COMMODORE 408	£125.00
COMMODORE 410	£135.00
COMMODORE 412	£145.00
COMMODORE 414	£155.00
COMMODORE 416	£165.00
COMMODORE 418	£175.00
COMMODORE 420	£185.00
COMMODORE 422	£195.00
COMMODORE 424	£205.00
COMMODORE 426	£215.00
COMMODORE 428	£225.00
COMMODORE 430	£235.00
COMMODORE 432	£245.00
COMMODORE 434	£255.00
COMMODORE 436	£265.00
COMMODORE 438	£275.00
COMMODORE 440	£285.00
COMMODORE 442	£295.00
COMMODORE 444	£305.00
COMMODORE 446	£315.00
COMMODORE 448	£325.00
COMMODORE 450	£335.00
COMMODORE 452	£345.00
COMMODORE 454	£355.00
COMMODORE 456	£365.00
COMMODORE 458	£375.00
COMMODORE 460	£385.00
COMMODORE 462	£395.00
COMMODORE 464	£405.00
COMMODORE 466	£415.00
COMMODORE 468	£425.00
COMMODORE 470	£435.00
COMMODORE 472	£445.00
COMMODORE 474	£455.00
COMMODORE 476	£465.00
COMMODORE 478	£475.00
COMMODORE 480	£485.00
COMMODORE 482	£495.00
COMMODORE 484	£505.00
COMMODORE 486	£515.00
COMMODORE 488	£525.00
COMMODORE 490	£535.00
COMMODORE 492	£545.00
COMMODORE 494	£555.00
COMMODORE 496	£565.00
COMMODORE 498	£575.00
COMMODORE 500	£585.00



Dept. 66, 14 BISHOP WAY ROAD, SALISBURY, WILTSHIRE, SP1 3EQ. TEL: (0450) 66667

SPECIALIST COMMODORE

G-TEN
CALL

COMPUTER REPAIRS

Commodore 40	£34.95	Commodore Model	£18.00
Commodore 128	£79.95	Commodore 128	£39.95
Commodore 200	£89.95	Commodore 200	£49.95
Commodore 300	£109.95	Commodore 300/128	£59.95
(CALL 404948 FOR THE PRICE OF THE UNIT)			

Repairs carried out by experienced technicians. A full test and service including a 4 month warranty with every repair (Commodore spares available on request).

Main charge/Field Call payable to G-TEN.
Please include O.V.O. post.
UNIT 6, 1, LAMAR COURT YARD, MILTON INDUSTRIAL ESTATE, COLBY SOUTHANTS WEST SUSSEX. TEL: (0293) 8984.

COMMODORE SERVICE CENTRE

For repair services tel:
* VIC 014 0 0144 4-COMMODORE ONLY
* LAMAR MK 4 08884 4-DISC DRIVE &
* all other models as well as

All units repaired in compliance with fully qualified staff, full spare parts stocks & dedicated support for all computers & Mod ems and phone modems, software & virus protection.

Personal contact always welcome.

* On the premises repairs, with full assurance & GUARANTEE (SEE TERMS) SERVICES.
11 Old London, Crawley, G11 7NY. Tel: 0293 6888
or 0293 8399

VTS COMPUTER REPAIRS

01 - 807 9839
Simply send your unit to the address below securely packed with full description and max fee.
COMMODORE, AMSTRAD, SBC, Sinclair
7 20200 029 88 98

VTS COMPUTER REPAIRS

12 Lily House, Gt. Dover Street, London, W1T 4BQ.
FAST, RELIABLE FORWARDING

SPARES REPAIRS SALES

all makes of computers & peripherals
Fast Reliable Guaranteed Service
Contact Dave Better T.Sq., 88888

88 ELECTRONIC SERVICES
306 Maudslayi Avenue, Gt. Baddow, Chelmsford, Essex, CM2 7JL
Tel: 0201 - 388881

CROYDON COMPUTER CENTRE

18 Maudslayi Road, Telford House, Surrey, CR8 4JL Tel: 01 871 3211
COMPUTER SERVICES
(SURREY) 1975
We repair - on the premises - quick & friendly

• Commodore & Spectrum
• BBC & Acorn (Approved Service Centre)
• Amstrad & Truarc (Approved Service Centre)
• All Local, British, Foreign, Modems
Mail Order by phone Access & Visa accepted

COMPUTER REPAIR SERVICE

SBC AMSTRAD COMMODORE SYSTEM
• Personal, Acorn and British authorized service centres
• Expert advice
• 100% repair fully guaranteed
• 24 hour service on free computers
• Free telephone advice 24 hours
• Free postage included

FORWARD YOUR UNIT TO AT
BLAIR COMP COMPUTERS LIMITED
1 MILLERS LANE, SOUTH BRITAIN
COMMODORE, Amstrad, Truarc
021-214 6056

Commodore instantly repairers

offer scopes and repairs of the full range of Commodore CBM products, including all Amstrad. Phone or write for free estimates for D.P.C.E. Valence Repair Centre
Wilkinson House, Telford St, Stafford Park, Telford, Tel: 0952 620 458
Ask for Pauline

COMPUTER REPAIRS

Fixed Super Low Prices!
Inclusive of parts, labour and VAT



FAST & 100% TECHNOLOGY

I work forward - Collect/Delivery available for local area

* SUPER OFFERS *

SPECTRUM	£24 Inc. - Price Same	ELECTRON	£24
CM	£22 Inc. - Price Same	COMMODORE 128	£24
CM	£19 Inc.	DISC DRIVE	£28 Inc.
VIC20, C-4	£19 Inc.	SPECTRUM 128	£28 Inc.
SBC	£19 Inc.	CM P.I.E. for 688	£28 Inc.

(Please contact computer store for details on repair/forwarding subject with repair)

W.T.S. ELECTRONICS

T.C. 1-4 Portland Road, Luton, Beds LU1 5BT. Tel: 0582 68875. Telex: 266671
©1978, W.T.S.

COMMODORE REPAIRS

We specialize in repairs of Commodore Computers and Peripherals and many others:
• FAST FORWARDING •
• 90 DAYS WARRANTY •
ALL INCLUSIVE PRICES START FROM £15
For a fast, reliable and professional service send your Commodore repair unit back to us for a FREE QUOTE!
ANALYTICAL ENGINEERING LTD,
85A/85B FRODO, UNIT 15A, QUINCY ROAD, HAY, WILTSHIRE, SN4 6JH.
PHONE: 1292 8200
Phone 02922 82022 for further information.
24 Hour answering service ••
•• Software also available ••

B

R

E

R

H

Bug Finder

We'd like to remind our readers that we run a Bug Finder service.

If you have typed in one of our programs and despite much checking, you still can't get it to run, then send us the following:

Two copies of your program on tape or disk.

A description of your problem.

If possible a listing of your work (you may omit this).

A stamped, self-addressed envelope for return of the program to you.

Should any of the above be missing then we will not be able to deal with your query.

We will try to point out where you have made errors and please corrected copy of the program back on to your tape or disk before we return it to you.

Do not send a program to us as soon as it stops working, please check a several times first.

We do get a large number of queries and so it may take a while for us to deal with yours personally.

Note we can only deal with problems relating to programs published in *Your Commodore*.

Commodore Where Are You?

At the *Your 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 Corby.

The Commodore plant at Corby was closed down some time ago. Reproduced here you will find the correct address for Commodore U.K.

We suggest that you write this correct address in the front of your computer manual for future reference.

Commodore Business Machines, U.K.L
Commodore House,
The Switchback,
Gardiner Road,
Maidenhead,
Berks SL6 7NA.

Oops

Correction to Easy Basic Toolkit (June '88).

Saver Program:

The last data area in line 80 should be 184 NOT 137.

Tape users should type POKE48715,1 after running.

All users should then type SYS48704 to start the save.

Code program: The following lines were mis-printed.

line 4340 DATA
32, 40, 88, 250, 253, 75, 884, 896, 896, 2, 224,
4, 288, 11, 31, 99, 1800
line 4430 DATA
188, 210, 32, 285, 188, 160, 52, 160,
165, 31, 30, 171, 185, 21, 32, 135, 1776

At the *Your Commodore* office we receive hundreds of letters from readers every month. We do try and answer each individually but sometimes this is impossible due to pressure of work. If you have written to us and not received a personal reply, we apologise for this but we cannot promise to reply to every item of mail we receive. If you feel that your question or letter really needs an answer, then inclusion of an a.s.c. will guarantee a reply, although this may still take time to arrive.

Puzzle Corner

Complete either of our cryptic puzzles
and win a *Your Commodore* binder!

- A** Complete the following phrases. For example, 26 L of the A becomes 26 letters of the alphabet.
- 12 M in a Y
 - 52 C in a P
 - 88 K on a P
 - 9 P in the SS
 - 92 T in the FL
 - 4 H of the A
 - 12 D of C
 - 11 P in a CT
 - 366 D in a LY
 - 12 M of a J
 - 50 S on the AF
 - 2 N in a B

- B** What is the next letter in the following series:
O, T, T, F, F, S, S ?

Mark clearly on envelope whether puzzle A or B and send to your Commodore, ASP Ltd, 1 Golden Square, London W1R 8BB.

MICRONET. THE FIRST TRULY INTERACTIVE MAGAZINE.



**WITH 250,000 PAGES,
150,000 CONTACTS, AND MORE
EXCITING SERVICES, YOU NEVER
KNOW WHAT'S GOING TO POP UP.**

Micronet. The interactive magazine. Modernized by phone to your computer. Original, creative, exciting, 24 hours a day, 365 days a year. Communications including interactive chatlines, telefax, telex, your own electronic mailbox, and download file programmes.

Information with more than 1/2 million frames. Entertainment from price-guides to Multi-User Games... and all for a low-cost local call. Complete the coupon, pop it in the post, and receive full details, or contact the Sales Desk on 01-887 1873.



TELEMAP GROUP LTD

TELEMAP GROUP LTD, DURBAN HOUSE
8 HERBAL HILL, LONDON EC2H 8JL



LOW COST
INTERNET
MAILBOX
TELEFAX
TELEX AND FILE SERVICE

NAME _____

ADDRESS _____

TEL. NO. _____



MACHINE TYPE
VCOMM 2000

MICRONET

FREE
MODEM
WITH 1st YEAR
SUBSCRIPTION -
WORLD STOCKS LAST!

AGE _____



If you already have a modem and computer software dial 021-818-1111 (password 4484) or 0800-0444004 for a free demonstration.

"£229 for all that?"

"Yes, it's the very latest, the industry compatible Multi-font LC-80 from Star, there are features included as standard not found on printers some times twice the price..."

"Well, there's the Star front control panel - so you don't have to fiddle with DIP switches - a built-in push tractor for low forms tear-off.

And you don't even have to take the tractor paper out to send out single sheets with the LC-80's paper parking. There's a large 48 leaflet - it's even got six NLQ fonts as standard. It prints at a realistic 30cps and has a really fast NLQ at 30cps".

"Hey, that sounds good, but what about quality? You'd have to go a long way to beat the print quality on that NL-80 you've got".

"True, but they've even improved on that and you can change all the resident fonts at the touch of a button".

"Hmm, that's really easy - £229 for all that? I wonder if my desk will have any left..."

"What's this - a new printer?"

"Oh really - like what?"

THE
star
PRINTERS FOR BUSINESS

Star Micros UK Ltd,
Crown House, 49 Leabridge Road,
Easing, London W5 2PP.
Telephone 01-848 1888

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

Price real P.M.T.

Fill in and send to your dealer (see page 48). Please send no an LC-80/80/80/80/80
Please include details of the Star range of dot matrix printers. Or you can call direct
on 01-848 1827.

Name _____ Company _____

Address _____

Telephone _____ Postcode _____

Telephone _____

12/100

or all that?"

...and fast
...and not

...it's better to
...with a
...push tractor
...Gears tear off,
...you don't even
...to take the tractor
...or not to unload single
...with the LC-80's paper
...up. There's a large ink
...it's even got six NLQ fonts
...and. It prints at a resolution
...and has a really fast NLQ

"Why, that sounds good, but what
...quality? You'd have to go a long
...to find the price quality on that
...NL-80 you're 'talking'."

...y, but they're even
...revel on that
...you can change
...the resident fonts
...the touch of a
...button."

"Wow, that's
...really easy—**\$\$\$**
...for all that? I
...wonder if my
...dealer will
...have any
...left..."



"What's this—a new printer?"

"Oh really—the what?"

Year Commencement August 1988

YOUR COMM

AUGUST 1988

GEOS-B

Wordprocessors
Which one's for you?

Expert V3.2
As fast as a parallel
DOS

Writing
Structured
Programs

UNBEATABLE
PROGRAMS:

Plus/4 SIMPLE
C64/C128 Terminal
Tape Organiser
Educating SID
A Short Interlude



Volume 3 Number 14

I'd like to find out if my dealer has any left. Please send me an LC-80 booklet.
Please send me details of the fine range of other superb printers. Call justcall@heath.com
001-660-6678

Name _____ Company _____
Address _____
Postcode _____
Telephone _____

Games Reviewed: Kar
Wizard Wars ▲ I