THE Magazine for experienced TANDY Colour Computer Users!

Jack. Rae.

$4.50

AUSTRALIAN COCO MAGAZINE

VOL4NO1SEPT87

Registered by Australia Post: Reg. No. QBG 4007

WHAT'S ON THE BEST OF CoCoOz

Best of CoCoOz #1. EDUCATION

ROADQUIZ	ROB WEBB
SHARE MARKET	ALEPH DELTA
HANGMAN	ALEPH DELTA
AUSTQUIZ	P. THOMAS
ALPHABET	RON WEBB
SPELLING TUTOR	IAN LOBLEY
TANK ADDITION	DEAN HODGESON
FRACTION TUTOR	ROBBIE DALZELL
TABLES	BARRIE GERRAND
ICOSA	BOB WALTERS
KIDSTUFF	JOHANNA VAGG
TAXMAN	TONY PARFITT
FLAGQUIZ	ROB WEBB

Best of CoCoOz #2 part 1 16K GAMES

PYTHON	W.ARMSTRONG
COCOMIND	STEVE COLEMAN
POKERMCH	GRAHAM & MATTHEWS
OILSLICK	JEREMY GANS
SPEEDMATHS	DEAN HODGESON
CCMETEOR	BOB THOMSON
BATTACK	JEREMY GANS
SKIING	JOSHUA GANS
PROBDICE	BOB DELBOURGO
RALLY	TONY PARFITT
CHECKERS	J & J GANS
FOURDRAW	JOHANNA VAGG

Best of CoCoOz #2 part 2 32K GAMES

TREASURE	DAVIDSON & GANS
SHOOTING GALLERY	TOM DYKEMA
MASTERMIND	GRAHAM JORDAN
GARDEN OF EDEN	DAVE BLUHDORN
ANESTHESIA	MIKE MARTYN
YAHTZEE	KEVIN GOWAN
OREGON TRAIL	DEAN HODGSON
BATTLESHIP	CHRIS SIMPSON
ADVENTURE +	STUART RAYNER
ANDROMEDIA	MAX BETTRIDGE
LANDATTACK	ALDO DEBERNADIS

Best of CoCoOz #3 UTILITIES

SCREEN PRINT	TOM DYKEMA
RAMTEST	TOM DYKEMA
PRINT SORT	PAUL HUMPRIES
BEAUTY	BOB THOMPSON
DATAGEN	ROBIN BROWN
PCOPY	BRIAN DOUGAN
FASTEXT	OZ-WIZ
MONITOR +	BRIAN FERGUSON
COPYDIR	THOMAS SZULCHA
LABELLER	FRED BISSELING
SPEED CONTROL	PAUL HUMPRIES
2BC	WARREN WARNE
CREAT-A-TITLE	BRIAN FERGUSON
DISKFILE	BRIAN DOUGAN
BIG REMARKS	BOB THOMSON
LABELLER	GORDON BENTZEN
DIR	MORRIE SINGER
HI	ALEX.HARTMANN

Best of CoCoOz #4 Business

HI (disk; Disk Directory Manager)	ALEX. HARTMANN
PERSMAN (Personal Finance Management)	PAUL HUMPHREYS
BANKSTAT (Annual & Store Statement)	BARRY HATTAM
CC5 (tape; Sales invoicing)	GRAHAM MORPHETT
INSURE (Analyse Home Contents)	ROY VANDERSTEEN
COCOFILE (tape; database)	BRIAN DOUGAN
DPMS (disk; Disk Program Management Sys)	PAUL HUMPREYS
DATABASE (tape; THE tape database)	PAUL HUMPREYS
RESTACC (tape; Restaurant Accounts)	DUNG LY
SPDSHEET (disk; 22 column spreadsheet)	GRAHAM MORPHETT
PRSPDSHT (disk; prints out "SPDSHEET")	GRAHAM MORPHETT
ACS3 (disk; Multi disk database)	GREG WILSON

Best of CoCoOz #5 ADVENTURES

ADV 32K	S. RAYNER
QUEST	TONY PARFITT
LABYRINTH	JAMES REDMOND
ADV +	SEAN LOWE
CRYSTAL	C & K SPRINGETT
PRISON	TIM ALTON
OPALTON	IAN CLARKE
WIZARD	DARRELL BERRY
TREASURE	C. DAVIDSON
LOST	ALEX. HARTMANN

Best of CoCoOz #6 PRESCHOOL

ALPHABET	STUART DAWSON
HATDANCE	JOHANNA VAGG
AUSTSONG	McDERMOTT FAMILY
ADVANCE	McDERMOTT FAMILY
WALTZING	McDERMOTT FAMILY
TIMEKANG	McDERMOTT FAMILY
BAND	McDERMOTT FAMILY
KIDSTUFF	JOHANNA VAGG
MATCHER	?
LETTERS	JACK FINNEN
BABYSIT	JOHANNA VAGG
SPELLING	JOHANNA VAGG
SPEEDTAB	DEAN HODGSON
10 FACES	JOHANNA VAGG

Please Note : Some of the programs on Best of Cocooz # 3 and #4 will not work on the Coco 3.

Best of CoCoOz #7 GRAPHICS

LIL'COCO	ANDREW WHITE
THE ROOM	HERMANN FREDRIKSON
BACK STREET	JOY WALLACE
LOCO	MIKE D'ESTERRE
COCO ART	SANDY McGREGOR
KANGA	JOHANNA VAGG
THE BOAT	SANDY McGREGOR
SAD COCO	P. BOLLE
TOWER	C.A.SYMS
WINDY DAY	SARAH LAW
SAILING	STEVE YOUNGBERRY
OUTHOUSE	STEVE YOUNGBERRY
SMURF	JOHANNA VAGG
SUNSTATE	STEVE YOUNGBERRY
HELICOPTER	ANDREW WHITE
MARTHA	ANDREW WHITE
BAD MOON	STEVE YOUNGBERRY
MCC	JOY WALLACE
EAGLE	?
BLASTER	PAUL YOULD
FOGHORN	PAUL STEVENSON

Best of CoCoOz #8 16K GAMES

ALIEN	STUART SANDERS
QWERL	DARRELL BERRY
SHOOTOUT	CRAIG STEWART
SHUTTLE	CRAIG STEWART
FROG	DARREN OTTERY
FROGRACE	TOM LEHANE
KIMMAT	TOM LEHANE
GRANDPRI	DOUG GREY
WATER WARS	JUSTIN LIPTON
CATERPILLER	JUSTIN LIPTON
DETECTIVE	VAL STEPHEN
BREAKOUT	WHY/BILT

Best of CoCoOz #9 32K GAMES

TRIOMINO	BOB DELBOURGO
MATCHEM	CHARLES BARTLETT
GO	BOB DELBOURGO
NARZOD	MAX BETTRIDGE
CHOMPER	MAX BETTRIDGE
POPBALL	MAX BETTRIDGE
LUDO	WHY/BILT
SABRE	ANDREW SIMPSON
MOVEABOUT	KEVIN GOWAN
JIGSAW	JAMES REDMOND
LABYRINTH	JAMES REDMOND
TANK	CRAIG STEWART

Best of CoCoOz #10 Education II

METEOR	DEAN HODGSON
DRIVERS TEST	ANDREW SIMPSON
SALE	JUSTIN LIPTON
TABLES	PAT KERMODE
OPALTON	IAN CLARKE
CAPITAL LETTERS	BOB HORNE
TEST MATCH	JEFF SHEEN
SENTENCE ENDINGS	BOB HORNE
ESCAPE	DEAN HODGSON
RAILMATH	BOB HORNE
COUNTDOWN	DEAN HODGSON
WHATZIT	BOB HORNE
HOMOPHONES	BOB HORNE
COMPOUND WORDS	BOB HORNE

Best of CoCoOz #11 Education III
This is a DISK only issue!!

CHATWIN MANOR	BOB HORNE

TAPE $16 each

DISK $16 each

Telecom

VIATEL®

AUSTRALIA'S NATIONAL VIDEOTEX SERVICE

DATE OF APPLICATION / /

APPLICATION FORM

(BEFORE COMPLETING THIS APPLICATION, PLEASE READ REVERSE SIDE CAREFULLY)

section 1

PLEASE TICK APPROPRIATE BOX TO INDICATE SERVICE REQUIRED

BUSINESS SERVICE □ NON-BUSINESS SERVICE □

(CHARGES INCURRED ON BUSINESS SERVICES ARE USUALLY TAX DEDUCTIBLE)

SURNAME (OR BUSINESS NAME IF BUSINESS SERVICE) GIVEN NAMES

POSTAL ADDRESS NUMBER/STREET

SUBURB/CITY STATE POSTCODE

TELEPHONE NUMBER ON WHICH SERVICE IS REQUIRED (INCLUDING STD CODE)

section 2

CONTACT NAME (IF BUSINESS SERVICE) GIVEN NAMES

POSTAL ADDRESS FOR BILLING IF DIFFERENT FROM SECTION 1 ABOVE
NUMBER/STREET

SUBURB/CITY STATE POSTCODE

CONTACT TELEPHONE NUMBER (INCLUDING STD CODE)

section 3

PLEASE DESCRIBE NATURE OF BUSINESS (OR OCCUPATION IF NOT A BUSINESS SERVICE)

PLEASE INDICATE TYPE OF EQUIPMENT USED TO ACCESS VIATEL

special instructions

THIS FORM SHOULD BE HANDED IN AT ANY TELECOM BUSINESS OFFICE OR MAY BE MAILED WITHOUT A STAMP TO FREEPOST 20, VIATEL BOX 188C, GPO MELBOURNE, VICTORIA 3001

PLEASE ALLOW TEN WORKING DAYS FOR PROCESSING OF APPLICATION AND RETURN MAIL ADVICE.

telecom use only

DTE PP VN

BG SC CI

GOLDLINK: The place to be on VIATEL

REF

SPAN OVERLAY NO. 140

REGISTRATION AND SUBSCRIPTIONS

Customers must register as a Business Service if the telephone number nominated for the use of the VIATEL Service is a Business Service and/or VIATEL is to be used wholly or mainly for Business, Commercial, Industrial, Professional or Government purposes. (Charges incurred on Business Services are usually tax deductible.)

Where a Business Telephone Service is nominated for the use of VIATEL, but the use of VIATEL is wholly or mainly for Non-Business purposes, the Customer may be registered as a Non-Business VIATEL subscriber, providing the registration is taken out in the Customer's personal name and address and not a Business name.

Telecom Australia will register the Business or Individual named under Section 1 as a Customer of its VIATEL Service and will provide the Customer with a confidential Customer Identity Number and Personal Password by mail.

Where billing address is indicated, bills and bill related correspondence ONLY will be forwarded to that address. All other correspondence will be forwarded to address under Section 1.

Customers should advise VIATEL of any change of address as soon as possible.

If you lose your Customer Identity Number and/or Personal Password, you must advise VIATEL in writing before new numbers are issued. Our postal address is: Freepost 20, Box 188C, GPO Melbourne, Vic. 3001. FOR SECURITY REASONS REPLACEMENT NUMBERS AND PASSWORDS CANNOT BE PROVIDED OVER THE TELEPHONE.

Customers of VIATEL acknowledge that their name and registered VIATEL Number will appear on the VIATEL Mailbox Directory and that Service Providers and/or other registered VIATEL users may send messages to their VIATEL number.

Telecom Australia undertakes no responsibility in relation to the accuracy of the information or service provided by Service Providers on VIATEL. Telecom Australia will not be responsible for any loss or damage arising out of or in any way connected with the use of this information or service.

Attention is also drawn to the terms and conditions governing the provision of information and services by some Service Providers. These terms and conditions may, in some cases, include a disclaimer absolving the Service Provider from liability regarding information and services supplied on VIATEL. The means of accessing these terms and conditions is set out on the Service Provider's Index Page on VIATEL.

Should you require any changes to your existing telephone equipment (e.g. new exchange line, additional socket), please contact your local District Telecom Office.

In a small number of cases VIATEL reception may be unsatisfactory. Correction may incur an additional charge.

inside coco

Utilities

24 TAPE UTILITIES
Emulating the convenience of a disk system.

28 FASTBACKUP
Save time switching disks.

34 LLIST32
32 column width printout.

35 COMMAND CHANGER
Change CB and ECB command

37 DISKDUMP
ALTER data stored on disk.

46 DMDATGEN
Let CoCo do all the hard data work.

48 DIRSQR
Finding square roots of floating numbers.

51 WORKSHEET 80
Nothing fancy, but it does the job.

59 CURSOR
Change the cursor to suit your own preferences.

Articles

7 CONF 87
A report on this year's festivities.

16 RE.VIEWS
Arthur Slade does his thing on OS9 PASCAL and COOKING.

42 DATA STRUCTURES IN FORTH
John Redmond's continuing series.

Games

17 NUTHACKER
A 3-D Graphic Adventure

40 QUIZ MACHINE
Add some excitment to your next quiz.

44 GOLDGRABBER
May the Gold Fever be

53 ADVENTURE + II
One for the 10 to 16yo's.
with you!

Hardware Mod.

37 DOUBLES FROM SINGLES
Getting the most out of disks.

Founder: Greg Wilson.
Managing Editor: Graham Morphett.
Editor: Alex Hartman.
Accounts: Karen Court
Production: Sheryl Bentick
Advertising: Graham Morphett.

Subeditors: John Redmond, Fred Bisselling, Jack Fricker.

Special thanks to: Geoff Fiala, Martha Gritwhistle, Mike Turk, Brian Dougon.
Deadlines: 7th of preceeding month.
Phone: 075 396177 or contact on Viatel *64213#
Registered Publication QBG 4009.

IN A NUT SHELL

We have cause to celebrate! It's been a year now since Tandy released the CoCo 3! One year down the track and look where we are!

Mind you, like any new computer to come on the market, it takes time for software and/or hardware to arrive.

As an example, 2 months ago they released ColourMax 3 (a CoCoMax for the CoCo 3); a few months before that OS-9 level II arrived! There are heaps more software/hardware releases, but they're the type of software that everyone talks about.

And if we look at games, well there are a good 30-50 games out for the CoCo 3, and that's not including games one finds in the pages of Softgold or CoCo!

So far, amazing things are available for the CoCo 3; wait until NEXT year - what will we see then?

As for the CoCo 1 & 2? The story is that Tandy & Goldsoft will still support those computers for a long time to come. Proof of this point, Tandy stopped producing the MC-10 computer two years back, and here we are, still producing articles and software for it!!

As for software for the CoCo 1 & 2: unfortunately for those people who have those computers, most of the REAL keen programmers have gone to the CoCo 3, and we don't really mind that. But we'd still like to see software coming in for the CoCo 1 & 2!

When I read letters saying, "This magazine is dominated by CoCo 3 programs - I only own a CoCo 2! This month is of no particular interest to me.!", what can I say?

We don't pick the programs to go into the magazine - the programs that go into the magazine are largely determined by you, the readers - what you send we print, and contrary to statements we've heard 3rd hand, we don't have a huge backlog of material.

Like Graham said in his "Clubroom", it's your magazine - you make it happen! I thoroughly support that statement! It IS a great magazine - made great only by those who support it, through programs, articles, etc.

What I'd like to see in this magazine is programs and articles for the CoCo 1/2 AND the CoCo 3, for all of us to share!

So let's see that article or that program come in - if you want something in the magazine or if you want something out of it - then go out and do something about it!

Conf '87

Conf'87 went down like New Years Day! Something new and refreshing came out of Conf'87! One met new friends, had new ideas, had new concepts, saw/bought new products ... the list goes on!

Graham has the complete story about Conf '87 in the next few pages of the magazine, including the winners of the competitions that we ran from August 1986 to July 1987!

Software Nabbing!

Again, this subject has to be talked about. I don't find it very funny to fall victem of this!

Recently a program from a trusted author arrived which had been typed from another source.

The only changes he made to the program were to change the copyright notice to his own name!

This sort of thing can lead to severe consequences, not only through the law and all that, but also making it harder for software houses to get software from overseas to sell here!

The situation was really bad last year already ... if it continues, there won't be any software available in Australia - then where does that leave us?

Survey '87

Like the title says, it's survey time again. The objective is to find out how you are going, what we need to do to meet your needs this year and what we can do to make better magazines for you.

Just fill the survey in, answer the competition question to win 100 disks (yes, one hundred!), and send it to us.

There is another survey to be answered, and that's on the Conference, past and future. When you have expressed your thoughts on that subject, send that to Johanna who will then give us the results.

That's it for this month - until next month!

BLAXLAND COMPUTER SERVICES PTY. LTD.

SERVICING MODERN TECHNOLOGY

Tandy ELECTRONICS DEALER

NO 9254

SOFTWARE-SOFTWARE-SOFTWARE-SOFTWARE-SOFTWAR -SOFTWARE-SOFTWARE-SOFTWARE

OS-9 LEVEL 2 - Includes Basic 09 (Direct from U.S.)	CC3 - $180.00
DESKMATE 3 - For your Coco 3	CC3 - P.O.A.
I.M.S.- 4th Generation Language Database (Single User)	CC2/3 - $299.00
SCULPTOR - 4GL Database Multiuser (OS-9 level 2 req.)	CC3 - $1200.00
THE WIZ - Communications with Windows (L2 & RS232 req.)	CC3 - $160.00
SCREEN STAR - OS-9 Screen Editor with Smart Speller	CC3 - $99.95
SCREEN STAR TEXT FORMATTER - use with Screen Star	CC3 - $69.95
RAMDISK & 512K DIAGNOSTIC - plus utilities	CC3 - $39.95
COLOR SCRIBE - RS/DOS Wordprocessor	CC3 - $99.95
THE WILD WEST - by Tom Mix (Disk)	CC3 - $51.95
NUKE THE LOVE BOAT - Adventure using 512k as ramdisk	CC3 - $69.95
MAGIC OF ZANTH - Adventure with top graphics	CC3 - $69.95
RETURN OF JUNIOR'S REVENGE - All time favourite	CC3 - $69.95
PRO GOLF - 36 Holes, 32K required (Disk)	CC2 - $59.95
MR. DIG - Tape or Disk	CC2 - P.O.A.
ESCAPE: 2012 - 64K Adventure (137 rooms)	CC2 - P.O.A.
TREASURE OF THE AZTECS - 4 voice music, 50 hires screens	CC2 - P.O.A.
ROBOT ODYSSEY - 64K Adventure (Disk)	CC2 - P.O.A.
COMPLETE RAINBOW GUIDE TO OS-9 - 2 Disks, no book	CC2 - $59.95
FIRST RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $32.00
SECOND RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $56.00
THIRD RAINBOW BOOK OF ADVENTURES - Book and Tape	CC2 - $44.00
" " " " " - Book and Disk	CC2 - $54.00
RAINBOW BOOK OF SIMULATIONS - Book and Tape	CC2 - $40.00
SECOND RAINBOW BOOK OF SIMULATIONS - Book and Tape	CC2 - $40.00
INTRODUCTORY GUIDE TO STATISTICS - Book and Tape or Disk	CC2 - $26.00
DISK TUTORIAL - for serious disk basic/ml prog. (2 disk)	CC2/3 - $74.00
COCO GRAPHICS DESIGNER - print your cards, signs,banners	CC2/3 - $59.95
" " " - 100 pre-drawn pictures	CC2/3 - $29.95
3D-GRAPHIMATOR - 3 Dimensional Drawing Design	CC2 - P.O.A.
PROGRAMMERS UTILITY - Commands & Utilities by keystroke	CC2 - P.O.A.
PRINTMASTER	IBM - $96.35
ART GALLERY 1	IBM - $56.65
ART GALLERY 2	IBM - $56.65
NEWSROOM	IBM - $101.95
CLIP ART 1	IBM - $50.95
CLIP ART 2	IBM - $62.35
CLIP ART 3	IBM - $62.35
NEWSMASTER (DESKTOP PUBLISHING)	IBM - $170.00
PRODESIGN (CAD SYSTEM WITH HI-RES PRINTOUTS)	IBM - $630.00
CLIPPER (DBASE III COMPILER)	IBM - $599.00
DAC-EASY ACCOUNTING AUSTRALIAN VERSION	IBM - $250.00
DAC-EASY WORD PROCESSOR	IBM - $129.00
FBS ACCOUNTING SUITF (Inventory,Debtors,Creditors,G/L)	IBM - $830.00
FBS PAYROLL	IBM - $630.00
FBS BILL OF MATERIALS	IBM - $630.00
FBS JOB COSTING	IBM - $630.00
A1 TYPIST WORD PROCESSOR	IBM - $159.00
PEEKS N' POKES FOR IBM	IBM - $66.00
INSIDE TRACK FOR IBM	IBM - $85.00
SPEEDIT PROGRAM EDITOR	IBM - $79.00
PROKEY	IBM - $195.00
MUSICA II DISK OR TAPE	CC2 - $49.00
MUSICA LIBRARY 100/800 EACH SET	CC2 - $49.00
GEOGRAPHY (Africa,South America,Asia,Europe,U.S.A.)	CC2 - $99.95
BOUNCING BOULDERS	CC2/3 - $49.95
GAUNTLET	CC2/3 - $49.95
KARATE	CC2/3 - $49.95
KNOCKOUT	CC2/3 - $49.95
F-16 MISSION ASSAULT	CC2/3 - $49.95
SHOCK TROOPER	CC2 - $33.45
PAPER CHASE	CC2/3 - $49.95
MARBLE MAZE	CC2/3 - $49.95
DECATHLON	CC2 - $33.45
MONEYPOLY	CC2 - $33.45
SAILOR MAN	CC2 - $33.45
COCOTEX VIATEL SOFTWARE	CC2/3 - $99.95
BUSINESS ACCOUNTING SYSTEM	CC2 - $99.95
FAMILY TREE GENEOLOGY	CC2 - $33.45
SUPER BACKUP UTILITY	CC2 - $59.95
TELEWRITER 64 DISK + TAPE	CC2 - $84.45
VIP-WRITER DISK + TAPE	CC2 - $98.95
VIP-DATABASE DISK ONLY	CC2 - $84.45
VIP-CALC DISK + TAPE	CC2 - $84.45
VIP-SPELLER DISK ONLY	CC2 - $77.45
VIP-TERMINAL DISK + TAPE	CC2 - $84.45

BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS-BOOKS

AMERICAN RAINBOW - Direct from USA, month of publication	P.O.A.
BACK ISSUE ORDERS - place your order in July and August	P.O.A.
P.C.M. - Portable Computer Monthly, month of publication	P.O.A.
INSIDE OS-9 LEVEL II - A must for Level II	$79.00
500 PEEKS and POKES - for your Coco	$33.95
200 ADDITIONAL PEEKS and POKES - supplement to 500	$19.95
ASSEMBLY LANGUAGE PROGRAMMING - by Barden	$11.95
COMPLETE RAINBOW GUIDE TO OS-9	$39.95
NANOS REFERENCE CARD FOR COCO	$5.95
6809 ASSEMBLY LANGUAGE PROGRAMMING	$41.95
VISICALC APPLICATIONS	$19.95
MULTIPLAN APPLICATIONS	$19.95
UNDERSTANDING COMPUTER SCIENCE	$5.49
DBASE III TIPS AND TRAPS	$37.95
GRAPHICS PRIMER FOR YOUR IBM PC	$45.95
YOUR IBM MADE EASY	$31.99
PC SECRETS	$35.95
PC DOS TIPS & TRAPS	$35.95
YOUR IBM PC	$39.95
GUIDE TO USING LOTUS 1-2-3	$36.95
C LIBRARY	$36.95
USING DBASE III	$37.95
THE 8086 BOOK	$39.95
THE SHAREWARE BOOK	$31.95
MBASIC HANDBOOK	$39.95

HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE-HARDWARE

10MEGABYTE HARD DISK DRIVE - Speed and Mass Storage	CC2/3 - $1299.00
20MEGABYTE HARD DISK DRIVE - Large Business Storage	CC2/3 - $1599.00
512k COCO 3 UPGRADE Includes P.D Software	CC3 - $220.00
TEAC DSDD DRIVE 40T PLUS CONTROLLER - Double Height	CC2/3 - $680.00
TEACx2 DSDD DRIVES 40T PLUS CONTR. - Equal to 4 Tandy.	CC2/3 - $950.00
BARE TEAC DSDD 40T DRIVES - Hi-tech reliability	CC2/3 - $300.00
BARE TEAC DSDD 80T DRIVES - 1.4MB Storage per drive	CC2/3 - $350.00
BARE TANDY SSDD DRIVES - not $519 (subject to stock)	CC2/3 - $250.00
T1000 20MB INTERNAL HD - 1 year warranty, formatted	T1000 - $1399.00
T1000 COMBI RS232/640K/CLOCK/CALENDAR	T1000 - $750.00
THOMSON AMBER/GREEN COMPOSITE MONITORS WITH SOUND	ALL - $350.00
TANDY 1000 SX - 384K, 2 DRIVE	T1000 - $1999.00
TANDY 1000 EX	T1000 - $1299.00
TANDY COLOUR COMPUTER 3 - 128K	CC3 - $449.00
TANDY 102 PORTABLE	T102 - $999.00
TANDY 200 PORTABLE	T200 - $1399.00
MULTIPAK INTERFACE - SWITCHABLE COCO 2/3, LED ON/OFF	CC2/3 - $199.95
VIDEO DRIVER WITH SOUND	CC2 - $38.00
VIDEO DRIVER WITHOUT SOUND (if your monitor has sound)	CC2 - $32.00
MODIFIED ARCHER JOYSTICKS - SUIT COCO 2/3	CC2/3 - $27.00
CITIZEN 120D DOT MATRIX PRINTER - EXTRA TYPEFONTS	ALL - $599.00
DMP-106 DOT MATRIX PRINTER	ALL - $399.00
DMP-130 DOT MATRIX PRINTER	ALL - $599.00
AVTEK MINI MODEM - 300 & 1200/75 baud, economy plus	ALL - $250.00
TANDY INTERNAL MODEM - 300 & 1200/75 + VTEX2 software	T1000 - $699.00
SUPER AUTO MODEM - 300,1200/75,1200 auto ans./dial/con.	T1000 - $820.00
WYSE 30 TERMINALS	CC3 - $999.00

-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-MISCELLANEOUS-

PRINTER PAPER 9-1/16" x 11" PLAIN per 1000 sheets		$17.00
PRINTER PAPER " x " PLAIN per box 4000 sheets		$60.00
PRINTER PAPER 9-1/2" x 11" PLAIN per 1000 sheets clean edge		$25.00
PRINTER PAPER 9-1/2" x 11-2/3" PLAIN A4 per 1000 sheets		$29.00
PRINTER PAPER 11" x 15" PLAIN per 1000		$30.00
PRINTER PAPER 11" x 15" BLUE RULE per 1000 (also SCREEN)		$27.00
NCR 2 PART 9-1/2" x 11" (no carbon required) per 1000		$74.00
MAILING LABELS 11mm x 50mm (4 up) per 1000		$6.00
MAILING LABELS 23mm x 89mm (1 up) per 1000		$16.00
MAILING LABELS 36mm x 102mm (1 up) per 1000		$18.00
CASSETTE LABELS (orange) per100 (10 per sheet)$5.00	(white)	$4.00
BLANK CASSETTES C-10's (10 min.) $1.10 each.	C-30's	$1.30
STORAGE CASE, HINGED, STACKABLE 5-1/4" holds 100 $25; holds 60		$22.00
STORAGE CASE, HINGED 3-1/2" holds 60		$15.00
3-1/2" DSDD DISKETTES box 10		$55.00
5-1/4" DSDD DISKETTES box 10		$25.00
DISK HEAD CLEANING KIT 3-1/2"		$11.95
DISK HEAD CLEANING KIT 5-1/4"		$11.95
LARGE RANGE PRINTER RIBBON/CARTRIDGES (Tandy/non Tandy)		P.O.A.
LARGE RANGE PRE-PRINTED INVOICE/STATEMENT FORMS AVAILABLE		P.O.A.

bankcard

VISA

AGC CREDIT LINE

BLAXLAND COMPUTER SERVICES PTY. LTD.
134 GREAT WESTERN HIGHWAY,
BLAXLAND 2774 N.S.W.
P.O. BOX 125
PHONE: (047) 39 3903

PRICES SUBJECT TO CHANGE WITHOUT NOTICE.

POSTAGE AND PACKAGING EXTRA

PARIS RADIO ELECTRONICS

161 Bunnerong Rd.Kingsford 2032 NSW
Phone 02 344 9111BBS 02 3449511
Viatel * 64268#

New Products for your CoCo 1,2 & 3

AVTEK VIATEL PACK

Includes the following;

- * Avtek Mini-Modem 11
- * RS-232 Cable
- * CoCo Tex Viatel Software

SALE PRICE.......................$279.95

AVTEK MEGA-MODEM

Smart Modem.Compatible 300/300, 1200/75 Auto Dial and Auto Answer.

PRICE AS ABOVE...........,...$499.95
PRICE WITH 1200/1200...$699.95

INSIDE OS-9 LEVEL 2

The inside story behind OS-9. L11 on the CoCo 111 written by Frank Hogg and Kevin Darling

PRICE...................................$54.95

512 K Upgrade kits

PJB inc 512K upgrade for your CoCo 111.Easy installation.No Wiring. Ram disc driver and utilities supplied.A must for OS-9 L2

SALE PRICE.......................$180.

OS-9 LEVEL 11

Operating System for CoCo 111 now in stock.Includes Basic 09

SALE PRICE.......................$180

WIZ COMMUNICATIONS PAK

New from Frank Hogg Labs. Uses windowing of OS-9 L11. Packed with features.

PRICE...................................$129.95

SCULPTOR DATA BASE

4th. Generation language + data base manager

PRICE...................................$995

BANKCARD MASTERCARD AND VISA ACCEPTED

PRODUCED WITH DESKTOP PUBLISHING AND AN ATARI ST SYSTEM

SPECIALISTS IN COMPUTER COMMUNICATIONS

CONF 87 a report

by Graham Morphett

Bundeena

What can I say?

Bundeena is just about the ideal site for a conference!

Situated just on the northern tip of the Royal National Park, and at the southern edge of Sydney's Port Hacking, Bundeena is a very beautiful spot!

We were visited by 4 different species of parrots, a possum, several species of honeyeaters, and countless other birds!

The views were just wonderful. I hope the black & white of the magazine does the shot from one of the cabins justice! Imagine waking to that view every morning!

The site was the Uniting Church's Campsite at Bundeena, and it is set up for conferences of up to 150 people.

There are cabins there set up with up to 28 beds in each.

People staying over slept in these. They are comfortable, but I think we all found the run to the showers a bit of an effort on those brisk August mornings!

The centre has two halls - one at the top of a hill, the other, at the bottom. The hill is designed to keep you fit!

This year we tried to divide the tutorials so that there were only two on at once - one in the top hall & one in the bottom hall.

The system worked well and the OS9'ers and MS DOS'ers were in the bottom hall, whilst the Basic programmers spent most of the conference in the top hall.

The food was great too! I remember places like this one from the days of my youth, and I remember only too well how bad the food could be.

But each meal at Bundeena just seemed to get better! So a big thank you goes to Frank and his staff for that!

Bundeena also offered the opportunity to provide for families - and so at the end of camp, we were able to offer a bushwalk as part of our activities - a fine way to finish conference!

Springbrook Falls – something to look forward to next year!

The view at Bundeena

Conference as a Camp

As readers will be aware, this was the first time we have run Conference as a camp.

The change in style certainly made conference far more fun.

When you live in camp with people for any length of time you get to know them far better! The friendships made at this camp will last!

In fact, based on the success of the formula this year, next year's camp.. er conference, has been already organised! But more on that soon.

The People – Who Was There?

Numbers were down on previous years.

About 75 people attended this year compared with almost twice that many last year, but despite the drop in numbers, I personally was not concerned.

The key people from groups across the east coast of Australia were there.

We had a large contingent from Melbourne, quite a few from Queensland, people from the country areas of NSW and no one from Sydney'

The Equipment

Tandy provided a heap of CoCo 3's, monitors and drives. They also had a T3000HD, a T3000HL and a T1000EX on display.

Paris Radio had several CoCo 3's and an Atari 520ST on display.

Blaxland had CoCo 3's everywhere, including one running two terminals - something to see!

And yes, there was an Amiga there. Charlie Lane owns an Amiga and a CoCo 3, and he brought the Amiga to demonstrate how much better the CoCo 3 was!

Blaxland Computer Services stole the show with their display of CoCo 3 software, much of it on show for the very first time in Australia.

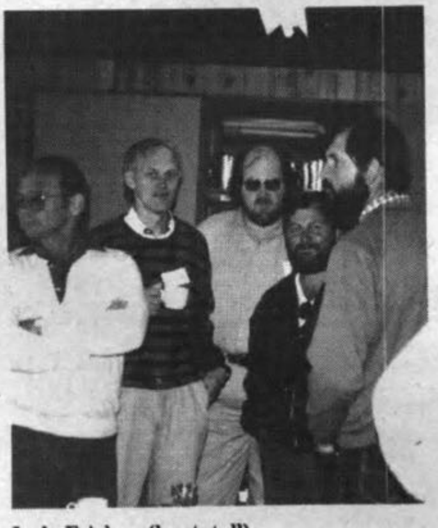
It's hard to decide which package was the most impressive.

In the games category, Magic of Zanth would take some beating.

Magic of Zanth ran hot most of the weekend, and we've been getting enquiries on how to play it or achieve the next phase on Viatel ever since!

In the Utilities area, Blaxland had the ColourMax program and the new Hires joystick adaptors.

These really add to the usability of the joysticks on the 3 and ColourMax is brilliant.



Jack Fricker (least tall), with Ross Pratt and friends discussing OS 9.

Johanna Vagg, discussing Basic with a rapt group

In the business side of things, undoubtedly, the availability of Sculptor and the fact that Blaxland have already written some programs for the 3 in this new environment had to be the big news of conference.

Sculptor is a new 'language' which transfers across many computer types right up to main frames.

Now what you program at home on your CoCo 3 or MS DOS computer can be transferred directly to the mainframe at work in the morning!

Paris Radio, not to be outdone, concentrated on communications.

Paris has one of the first on line databases in Australia - known as InfoCentre. They were showing that of course, and they also had Ron Wright's CoCoTex Viatel program, version 3.1 on show.

3.1 allows you to make up messages off line and send them when you get back on line. It has 30 pages of storage and it has a lot of other nice features including a stepping rate selector for disk users.

MPD Australia, distributors of Sculptor were themselves there, and they demonstrated the power of Sculptor very adequately!

In Softgold this month, there is an article about Sculptor. And before I hear cries that this article should have been in Australian CoCo, it's in Softgold because it effects both the MS DOS users and CoCo 3 users!

Tutorials

This year we were honoured by the quality and quantity of tutorials available.

We ran the conference a little more casually too, to allow lots of time for discussion and enquiry.

On Saturday morning, Johanna Vagg ran a well attended Basic Basic course. I think the attendees of this course would have been happy to sit and listen to Johanna for the whole of conference!

Each year she gets more confident, each year Johanna has a very valuable contribution to make.

Running in parallel with Johanna's course, Blaxland presented a showing of their new software for the CoCo 3.

Jerome Siappy was to talk first, but his car somehow mysteriously ended up balanced on a rock outside the camp and that delayed things for a while! In the meantime Bruce Sullivan kept everyone amazed!

When Jerome returned from his adventure, he showed the CoCo 3 with two terminals running Sculptor.

Such a system need not fear competition from ANY computer!

If you need a small office system with real multitasking power, then Blaxland's CoCo 3 based system is well worth looking at.

In the afternoon, Cyclops, um, that is John Redmond, showed his latest version of his Forth program.

Forth is easy to use and is far more powerful than Basic.

Poor John has somehow gained the Cyclops knick name and he looks certain to retain it to the grave!

Whatever, his avowed interest and campaigning for Forth are well founded. And quite a few of the new users there found that using Forth was not difficult at all.

Running parallel to John's

course, Mike Turk ran a course on C.

C is in use on MS DOS computers but can also be used with CoCo's running OS 9.

Not the simplest of languages, Mike made it sound easy, and I think, won some converts!

Saturday night saw our traditional dinner and presentations - more on which in a moment.

Sunday morning dawned bright & sunny - although there were a few around that morning who didn't seem to be in a position to notice!

Ron Wright and I discussed Viatel in the top hall with a group of people, whilst John Redmond did a shortened Assembly Language tutorial in the lower hall.

The fact that John's tutorial had to be shortened was the disappointment of the conference.

Hopefully John will give us a longer version next year!

Also speaking that morning was Brad McMaster from MPD (Australia) Pty Ltd - importers of Sculptor.

I've already mentioned Sculptor, but it was Brad's very capable demonstration of the Sculptor system that decided so many to give it a go.

We're certainly looking forward to seeing some work produced in Sculptor!

Finally, after lunch, Alex Hartmann discussed Advanced Basic concepts in the top hall, whilst Farley from Tandy gave an excellent tutorial on MS DOS in the lower hall.

Both these people had prepared extensively for their tutorials, and a special thanks goes to Farley because this was his first attempt at something like this. Your efforts were very much appreciated Farley.

Some of the more energetic ones left later in the afternoon for a bush walk.

The Royal National Park is Australia's oldest national park and as you walk through it, you can get a feel for how Sydney must have looked 200 years ago.

That apart, the scenery is stunning - the wild flowers were just starting to bloom, and the sandstone cliffs and the creeks that calve their way through them, make a walk in the park a real pleasure.

Prizes

This year saw the richest booty of prizes yet with well in excess of $3500 worth of prizes being handed out.

Tandy alone provided $1700 worth of equipment as minor prizes, and backed up for the big one, the Tandy Programming Award.

Paris Radio and Blaxland also gave some very generous prizes, and we at Goldsoft provided some modems too.

The Greg Wilson Award is a special award which we present to a family which has contributed greatly to the Tandy Computing world.

It was won in 1985 by the Delbourgo family, in 1986 by the Dougan family and this year went to the Wright family in Melbourne.

OS 9'ers at work!

Graham at work!!

It is a family award, because at the level of contribution we are talking about, history has shown that the whole family contributes to the endeavours of the principle recipient.

In the case of the Dougans last year, Brian is the principle recipient, in that he is the one with the driving interest in computers. But he could not have done what he has done for us all, had it not been for the help and support his family has given over the years.

In Ron Wright's case, we have a man totally dedicated to gaining and disseminating knowledge about the computer.

But his success in achieving the aim has been achieved only with the help of his lovely wife Debbie.

So the Award goes to the Wright family.

Ron started using computers with the Tandy Model 1. He purchased a 4K CoCo from the US, before they were released in Australia, and soon after was able to produce a book on the ROM calls, which would have been the first Australian book on the CoCo.

He worked with Greg Wilson on a number of projects including a disassembly of the ROMs of the MC 10 computer.

Of late, his major passion has been the provision of Viatel software for the CoCo.

This has led him down some interesting sidestreets, but nonetheless, his Viatel Terminal Emulation for the CoCo 3 is in every way, a world class program.

We were very pleased to have Ron's company - and help at conference, and we were equally glad that Debbie was able to be there too, to receive the award.

Well... receive in a way, because someone left it in Queensland didn't they!

Craig Stewart blew our minds with his programs for the CoCo 3 early this year, and it came as little surprise to anyone that he won the Tandy Programming Contest this year.

Craig is from South Australia and was unable to be at conference, but he won a cash prize of $500 plus a $1 royalty for every program of his that Tandy sells.

And the cash resultant of that could be nice, because Tandy intend bundling the program with every CoCo 3 sold at Christmas.

The program's name is "Pursuit", it is an arcade game and it demonstrates Craig's thorough understanding of the way the CoCo 3's memory works.

Jerome Siappy, Bruce Sullivan with some customers.

It is entirely written in machine language and it uses the CoCo 3 colours to advantage.

Look for it in your Tandy Store soon - oh and PLEASE buy it! Your purchase of Pursuit not only will encourage a world class programmer, it will encourage Tandy to continue with this form of encouragement for local programmers.

The Games Competition for CoCo 3's was won by Charles Bartlett from Rockhampton, with the program "3HOW?".

"3HOW?" is a pac man style program written in basic!

It plays well, and although programs were submitted with more detailed screens, this program won because of the excellent logic that Charles employs.

Charles wins a DMP 106 printer from Tandy.

James Redmond won the CoCo 2 games contest with a program called Gryphon.

Programmers of the CoCo 2 always face a problem because of the screen resolutions available to them.

Either the pictures are too small, or often, they are too chunky.

James found a nice medium between the two, and created a very playable game!

James won a CoCo 3 for his work.

The CoCo was donated by Tandy.

Scott Harvey wrote what may become the definitive CoCo2 Adventure.

Scott's adventure is called the "Drenkald Adventure".

We have a problem with it in fact because it takes up a whole disk, so we might have to issue it as a separate program one day soon.

Scott, better known on Viatel as "Zard", created the graphics for his adventure on CoCoMax, and he calls each picture as he needs them from the disk.

The program is highly imaginative and certainly deserves the award of a CoCo 3 from Tandy.

Whilst many were working on games for the CoCo 3, others have spent time creating new programs for the 3 in other interest areas.

Gordon Thurston made an Osilloscope program for the CoCo 3.

Yes, that's right, he uses the CoCo 3 as his Osilloscope, feeding in his readings through the joystick ports!

Gordon won Deskmate 3 for his CoCo 3 from Tandy for this program.

George McLintock is no stranger to the winner's circle - George was the winner of a prize last year.

This year his Machine Language "CoCo 3 Experiments" won him an OS 9 Level 2 package.

George always has much to show us, and this program certainly demonstrates some different uses for the CoCo 3 graphics system!

In the Business area, John Qrech of Melbourne was a winner with his Stock List program.

John visited us when he was on holidays here and showed this program, which is something he uses at work.

The program has a number of both novel and useful features and won John an Avtec mini modem and the FIRST of the new series of Ron Wright's CoCoTex program. This prize was donated by Paris Radio and Ron Wright.

In Softgold Magazine, we want to do more than just present programs.

We aim to entertain, to educate and to inform.

We want to reflect a little of each of our reader's other interests, and we want to publish articles which may have a socially beneficial effect.

That's a big order for a small magazine, but this year we had articles from Monique Bond, Walter Zambotti, Dean Hodgson, Allan Thompson, John Archer, John Carmichael, Mike Turk, Mark Permuter, Laurie O'Shea, and many others.

Finding a winner from the diverse subject matter presented was a real problem of course, but we finally settled on the Mike Turk articles because of their timely value to our younger readers.

Mike won a Modem plus CoCoTex software from Goldsoft for his work.

We don't get many Hardware articles.

We'd like more!

Paul Kuhn was the only one to submit one this year.

It was a good one too!

A PMODE and colour changer.

He wins a modem and CoCoTex software from Goldsoft!

Bob Horne of course took out the Education award.

Bob supplies top material for the magazine in the Educational area.

He wins software to the value of $100 from Blaxland computer Services for his program "Fraction Hunt".

In the area of OS 9, we do not receive much material, mainly because the OS 9 users group does such a good job of feeding OS 9 users with information, and because their newsletter has a wealth of their member's programs in it - and so it should be!

However, we'd like to encourage more programming in OS 9, and so Jeff Larsen wins this section with a program called "Cursor", which gives you a custom cursor in OS 9.

Jeff wins one of the new ColourMax programs from Blaxland Computer Services and Goldsoft.

In the MS DOS world, we recently received a bunch of

Basic and Pascal programs from Leigh Dawes of Taralgon in Victoria.

We have had some excellent work over the year, but it seemed to us that Leigh's work was something special, so he gets the MS DOS award which is a Busiware program from Goldsoft.

We also gave an Encouragement Award to Wayne Kely, a young fellow who has been turning in some nice work - he received the program "Cave Walker" from Tandy.

And there were two special mentions.

The first is Johanna Vagg - Johanna continued her support of new users with some lovely programs for both the CoCo 2 and the CoCo 3 this year.

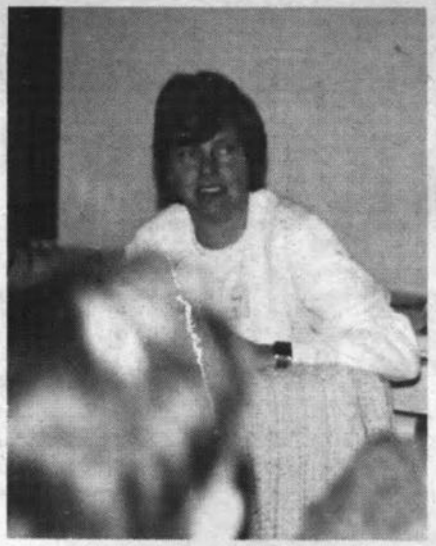
Of all our authors, by your letters, she is the one you all value most highly, and we are proud to count her as part of the team!

This year Johanna won a Tandy Multi Pack Interface from Blaxland Computer Services.

Finally, a special mention must go to Laurie O'Shea, who continued throughout the year to reflect a care and a love for the world in his "Close Encounters" series of articles.

We need more caring people and Laurie is appreciated.

So that's it for the prizes this year. If you didn't make it, sorry!



Johanna shows them how

If you did, congrats!

Either way, keep at it - we need your programs and articles for the magazine.

More importantly, your programs and articles continue to teach Australians and Canadians about their Tandy computer.

Many people tell us how eagerly they await the magazine each month. It is you authors who provide that pleasure to so many.

Next Year

How's this for organisation?

Next year's conference will be at Koonjewarre near Springbrook on the mountains behind the Gold Coast.

Date is to be confirmed, but looks like being on the first weekend of the NSW School holidays.

Springbrook is a true mountain resort. There are tremendous views; a cross section of animal life that has to be seen to be believed; and simply fantastic bushwalks!

Add to this, John Redmond, Johanna Vagg and a heap of other tutorial leaders and I KNOW we are going to have a fine time in 1988!

Of course, Expo 88 will be on at the same time, so you'll be able to take advantage of being in Qld for conference to go to that as well!

Don't miss Conf'88 - Conf'87 was the best yet - but Conf '88 will be something again!

Alex Hartmann and Michael Horne

Thanks

You don't put on an extravaganza like this without the co-operation of a very wide range of people.

Firstly, to Wilfred, Farley and the staff of Tandy, thank you everyone for your help in obtaining hardware and software for conference.

Being away from our "base" we depended on Tandy even more than we would normally, but they came through with brilliant organisation and assistance!

Next to the staff of Goldsoft, thank you for your work

Next to the Tutorial Leaders. Your work is appreciated - not just by ourselves, but also by the people who attended.

Without you, conference would be a very dull event!

Next to MPD, Paris & to Blaxland - we appreciate your part in conference, in the provision of prizes, and in the assistance you gave the attendees.

Many people came, just to see you!

Finally, thank you to those of you who came along and supported this event.

I'm absolutely certain each of you got friendship, information and perhaps even some goodies from the conference.

We enjoyed having you there!

Conf '88

At this stage it looks like we will be able to make this one cheaper than Conf'87!

Its a bit of a trip to get here for it - but that in itself is fun, and in any case - as many found this time, it is certainly a worthwhile trip!

SUBMITTING YOUR WORK

Ah! So you've finally finished that program? And you say to yourself, "What a great program that would make for CoCo Magazine/Softgold Magazine!"

And so you wonder to yourself, "How am I going to send this program in to the magazine?". Some time goes by and you suddenly realise, "Hey, there's an article in this month's magazine about submitting your work. I'll read through that and maybe that'll help me."

So you rip the magazine out of your stack of other CoCo/Softgold magazines and read the article on how to submit your program.

It reads ...

" ... we accept programs stored on both tape and disk ONLY along with a hard copy of the program(s) (optional only; we use it here as a reference to see what the program is/does) and suitable instructions.

Saving to Tape

Each program would be best saved three times with the last save being in ASCII. The tapes we recommend you use are either a C30 or less (the reason for that is that tapes longer than C30 have a tendancy to tear).

It'd be even better if you could include some instructions along with the program, either as a seperate program or in the wordprocessors listed below.

Saving to Disk

With disk, you'd be best to save it three times with the last save being in ASCII. Also, the extension name for the second and third copy should be different, so to distinguish the three copies. A simulation is given below.

' ... I have just saved 3 copies of a program called "HORSE". The directory listing would be:

```
HORSE  BAS   0 B 3
HORSE   1    0 B 3
HORSE   2    0 A 3'
```

Any instructions could be saved in the same system using either a program or in the wordprocessors listed below.

Wordprocessors we use.

Here is a list from our most preferable worprocessors to the drastic measure one could take to tell us how your program works.

1. Telewriter/Telepatch
2. Scripsit
3. PenPal
4. VIP Writer
5. Any form of data file.
6. Instructions written in a seperate program."

"Oh wow!", you think to yourself as you read it with awe and astonishment. So you go about your busy little way saving your program and instructions to tape or disk. Then you say to yourself, "Where do I send it?"

You read the article on ...

" ... any articles and programs should be sent to this address:

Submissions Editor,
Freepost 5
PO Box 1742,
Southport, Qld, 4215

All mail to this address need not be paid for.

All tapes and disks received will be returned after three months in case we need to refer to something or re-print something."

So place your tape/disk along with your hardcopy of the listing in a postpack (or suitable wrapping) and pop it in the mail.

All done!!

SURVEY 87

This is our third survey, so three years that you've had time to think about things like, "What do you hate about us, what could use improvement", and so on!

We really value your comments - even the ones we feel we can't act on! In fact past surveys have helped shape the current magazines. We feel sure this one will also contribute a great deal.

To encourage you to get the survey back to us quickly, we have a small competition with a prize.

When you've filled in all the questions, place the survey in an envelope and post it to:

Survey '87
Freepost 5
PO Box 1742,
Southport, Q. 4215

Get this survey in the mailbox BEFORE 7th October, 1987 to be elligible to win the prize which is 10 boxes of disks - ie 100 disks!

1. Name:

....

You can fill this form out anonymously if you want but you will of course be unable to receive the prize if you win it!

2. Address:

....

....

From hereon, circle the appropriate letter:

3. Where did you get this magazine?

a. I subscribe! My sub. number is: ....

b. A Newsagent - Which one? ....

c. A Tandy Store - Which one? ....

d. A Meet Group - Which one? ....

e. Elsewhere. Where: ....

4. What is your age range?

a. 0 - 10 b. 11 - 20 c. 21 - 30

d. 31 - 40 e. 41 - 60 f. 61 - 200

5. What computer(s) do you own?

a. Colour Computer, of course!

Cross out the inappropriate:

" ... I own a colour basic / extended colour basic CoCo 1 / 2 / 3 with 16k / 32k / 64k / 128k / 512k RAM."

b. Mico (or MC-10) computer.

Cross out the inappropriate:

" ... My Mico has 4k / 20k / 44k / Rom 1.1 "

c. IBM Compatible Tandy.

Cross out the inappropriate:

" ... I own a Tandy 1000EX / 1000SX / 1000 / 2000 / 3000 / IBM PC Clone / with ......k memory."

d. Some other type of computer.

What kind? ....

6. Your hardware: what do you have?

a. Cassette? Type? ....

b. Floppy drive? Number? ..... & Type? ....

c. Hard drive? Number? ..... & Type? ....

d. Printer? Number? ..... & Type? ....

e. Modem? Baud rate? ..... & Type? ....

f. Monitor? Colour / Monochrome : Type? ....

g. Television? Type? ....

h. Anything else? ....

7. What else do you intend to buy in the next twelve months?

....

8. Do you intend joining Viatel this year? Y N

a. If Yes, what about joining Goldlink? Y N

9. How do you use your computer? Please give percentage (%) of use: (it should all add up to 100%!)

Games .... %

Business .... %

Education .... %

Communication .... %

Utilities .... %

Robotics .... %

Scientific Application .... %

Commercial Software Production .... %

To gain more knowlege of the computer .... %

Other .... %

Total 100 %

10. What languages do you have a 50% greater proficiency in?

a. Basic b. Basic09 c. C

d. Forth e. Assembly/Machine Language

f. Pascal g. Cobol

h. Something else. What ? ....

11. Out of the above languages, which are you familiar with, as opposed to having a working knowledge of?

....

12. Magazine buying habits:

I subscribe to:	I purchase seperately:
a. CoCo	a. CoCo
b. Softgold	b. Softgold
c. American Rainbow	c. American Rainbow
d. 68xxx	d. 68xxx
e. 80 Micro	e. 80 Micro
f.	f. ....

13. What was the best thing about the last four months of our magazine?

....

14. What was the worst thing about our last four months of the magazine?

....

15. What would you like to see more of in the magazines?

....

16. Tandy Stores

a. Which one do you frequent? ....

b. Managers Name? ....

c. Is the shop tidy and laid out well? Y N

d. Does the store have what you want? Y N

e. Can the staff help you with computing problems? Y N

f. Have you been able to strike up a friendship with any of them there? Y N

g. To what other Tandy stores do you go to obtain the type of gear you want?

....

17. Conf'87:

a. Did you attend? Y N

b. Would you come / come again? Y N

c. How do you react to the "Camp" format of conference?

....

d. We'd appreciate hearing any other comments you may have about Conf'87:

....

18. Meet Groups: Are you a regular? Y N

19. Write here if there are any changes to be made on the list of meet groups for your group:

....

20. If you have any further comments about the magazines or our other services, please use this space:

Don't forget, this form has to be in before the 7th of October 1987 to be elligible for the prize!!

Competition:
We are about to introduce Software Downloads to our Goldlink service on Viatel. Suggest a name for this part of the service - best name wins 10 boxes of disks - value $200!

....

Thank you for filling in this form.

COM ✿ STATION 642

> Ron Wright wins the Greg Wilson award for 1987!!!!!

G

*

64290210A

Tandy Users' Board Member

378651620 SUN 26 JUL 1987 18:32

> FOR COMPUTER USERS LIVING ON OR NEAR THE MORNINGTON PENINSULA, THERE IS A COMPUTER CLUB THAT MAY SUIT YOU.

THE P.C.C.C. (PENINSULA COLOR COMP. CLUB) MEETS THE THIRD WEDNESDAY OF THE MONTH AT BRUCE PARK HALL FRANKSTON 7.30 PM.
(MELWAYS REF. MAP 102 G7)
ALL COMPUTERS WITH COLOR CAPABILTIES ARE WELCOME, NOT ONLY TANDY COCOS.
EXTRA INFO GIVE ME A MB,
MAY SEE YOU THERE !!!

*

Com Station 642 ViaTw 6429035A

OS-9 Users' Board Member

648230650 WED 12 AUG 1987 22:40

> I AM having trouble getting deskmate to boot from my hard disk ANY CLUES ?

Ross

*

Com Station 642 ViaOn 6429036A

OS-9 Users' Board Member

588316000 THU 13 AUG 1987 19:43

> TO ROSS 2
I DON'T know Deskmate but look at a dum p of program. If you see /d0, change t hat to /dd. This will enable you to boo t from the default drive.
Regards, Rick

*

Com Station 642 ViaTw 6429029A

The Tandy Users' Board

548421180 SUN 16 AUG 1987 19:10

> Can anyone please supply me with a ci rcuit diagram to power 2 DS drives using only ONE transformer .. My existing DS Mitsubishi 4851's use two separate trann ys - one for the 12 volt supply and one for the 5 volt supply.
Why can't I hang a 12 V and a 5 V regula tor on a single tranny ???

*

Com Station 642 Viate 6429037A

OS-9 Users' Board Member

755100150 FRI 14 AUG 1987 00:05

> You've not seen anything until you have seen Blaxland's CoCo3's running 2 terminals plus the main screen on OS9 L2. There is no speed lose - especially as they us hard drives. Its really incredible.

At Conference it had the Messy Dos guys very envious!

G

64290211A

Tandy Users' Board Member

756413480 TUE 28 JUL 1987 21:25

> HELP
WHEN USING COCOMAX II WITH 64K GREY CASE WITH R/BITS 1.4DOS - LOADS FILE1 OK-EDITS FILE1 OK-PRINTS OK-SAVES EDITED FILE 1 OK-WONT LOAD NEW FILE---- WHY? CAN IT BE FIXED
USING COMP SHACK ZAP PROGRAMS W ILL NOT CHECK OTHER THAN DISKO-ANY FIX F OR THAT?
KGEE

-

What colour cocomax are you using? IE Black or white? Alex

Tuesday night is computer night on Goldlink Com Station 642

64290212A

The Tandy Users' Board

755100150 TUE 28 JUL 1987 21:41

I have worked out a patch to fix Clonemaster to work with the Coco 3. It is quite simple....

```
LOADM"CLONMSTR"
POKE&HE4D,&HFE
POKE&HE4E,&HEE
SAVEM"CLONMSTR",&HE00,&H1A74,&HE12
```

That's all there is to it! What was hap pening is that the memory check routine was writing over the Vextors that the Coco3 uses at &HFEEE - &HFF00. By chan- ging the stop point from &HFF00 to &HFEEE this no longer occurs.

Peter Harry

*

Com Station 642 ViaOn 6429022A

The Tandy Users' Board

542413860 THU 06 AUG 1987 16:46

> REQUEST INFO ON MC6821 PIA CHIP
REGARDS
SEBROF

*

Com Station 642 ViaTw 6429024A

Tandy Users' Board Member

262289400 WED 12 AUG 1987 21:40

> To HAF and the any owners of a COCO 3 let me assure you that having had the priviledge to Review DESKMATE 3, it is truly a superb program. As for OS9 Level 2, which I have had for a few months now, these are only 2 programs at the moment sold by Tandy, from the inside stories I have heard that came from the 1987 COCO Conference, there are BIG things about to happen. Most of the programs I have reviewed work just fine on my 512K COCO 3, look for them in future issues of either CoCo or Softgold Magazines. ART _

Com Station 642 ViaTw 6429033A

OS-9 Users' Board Member

705471270 SUN 02 AUG 1987 10:35

Has anyone with the user group stuff tried to use the ISAM program there? The demo programs for it seem to work, as far as writing an index file, but the pointer Index.Rec is allways the same.

Jeff

*

6429031A

OS-9 Users' Board Member

726288690 MON 08 JUN 1987 19:30

OZ-WHIZZ If you are trying to get rid o f the excess modules in memory like seti me date and so on, you can't. Theye aren 't in the os9boot module at all but are part of the shell if you do an ident of /dd/cmds/shell you will see what I mean. What you need is to separate them from the shell before you boot.

U F O

*

64290312A

OS-9 Users' Board Member

648230650 SUN 19 JUL 1987 21:18

Xcopy, Xdir, xlist and Xdump do not wor k with level 2 os9. does anybody know of a patch for these programmes, or and al ternative programme(s). Also is there a patch around for Stylo III, it almost wo rks with level 2.

Ross.

*

Com Station 642 ViaOn 64290211A

The Tandy Users' Board

548421180 MON 17 AUG 1987 20:02

> MR U.F.O
THANK YOU FOR THE INFORMATION REGARDS THE DISK DRIVES AND POWER SUPPLY
REGARDS GRAEME ELLIOTT
(SORRY I FORGOT TO SAY WHO I WAS TIME !)

*

Com Station 642 ViaTw 64290210A

Tandy Users' Board Member

726288690 MON 17 AUG 1987 19:01

> Unsigned, You can indeed run 2 disk drives of 1 multitap transformer. You wi ll need 1x7805 and 1x7812 - 3 terminal regulators and a couple of electrolytic capacitors to smooth out the input and i nput. The Mitsubishi drives do not requi re a large amount of power to run. You MUST have at least 7.5 and 13.5 volts to feed the regulators.

U F O

Hi Jack, how's the Hard drives going?

REVIEWS

By Arthur Slade

OS9 PASCAL

TO QUOTE FROM THE reference manual, "Either you already know Pascal or you don't". OS-9 Pascal comes with a revamped 160pg Manual, whereas the original manual was spiral bound - this one isn't.

There are 14 chapters, 3 appendices and an index. If you already know Pascal you will be pleased to discover that OS-9 Pascal is a very thorough implementation of the language according to ISO Standard 7185.1 Level 0 with exceptions and extensions documented in Ch.14 of the manual.

Also some unnecessarily restrictive Pascal Syntax requirements have been relaxed.

If you don't already know Pascal, you have some learning to do. Pascal having been designed originally for teaching programming making it easier to learn in stages.

If you decide you would like to learn Pascal, then I suggest you obtain some reference and self study source books - try your library.

Buy the Pascal program and read the manual.

The equipment you require would be 64K (min) CoCo, 2 disk drives as well as the 2 OS-9 Pascal distribution disks that come with the program.

As always, make a BACKUP of the 2 Disks, put the originals away for safety and use the backups.

The OS-9 program package comes well presented - it will run on the 512K CoCo 3 (a lot more space), the only drawback being that you should retain your earlier version of your OS-9 disk, as OS-9 Pascal will ERROR when attempting to run a Pascal compiled program when using OS-9 Level II.

Although I certainly don't profess to be an expert on Pascal, I found no trouble in understanding the reference manual, and taking into account the cost of $179.95, I rate it a 7.5/10.

(Tandy Products. 91 Kurrajong Avenue, Mt.Druitt. N.S.W. 2770)
Reviewed by Arthur Slade. Hills District Colour Computer Club.

***

THE COMPUTER AIDED CUISINE

THE COOKBOOK PROGRAM comes well presented with a documented 32pg manual, 1 program and 1 data disk.

As always make backup copies of your original disks and use the copies. NEVER use the originals.

Not being a great cooking fan myself, I soon found out that a lot of recipes, etc, were to my taste, as well as being able to create your own recipes and save them.

Some of the features included:

* Creating an ingredient shopping list and printing it out, as well as a printout of any recipe.

* A 'HELP' screen of glossary terms as well as providing a timer to assist in meal preparation. You are able to scale a recipe either up or down to suit from 2 - 99 people.

* The program has a Database of some 320 recipes, each having visual appearance, nutritional value and flavour intensity.

* From Appetiser to Dessert as well as Beverages.

I would seriously recommend this program to anyone who has culinary interests, more so to those who want to learn more about cooking a meal and just what to do.

The Menu's are varied and interesting. You can also put your own favourite recipes, desserts, drinks or whatever on the Database for future reference.

The Program requires a min of 32K, 1 or 2 disk drives. colour (optional) T.V. set, serial printer (also optional).

I ran this program initally on my 64K CoCo 2. I then thought I would see how it went on my 512K CoCo 3 - it ran with no problems at all.

It must be realised that the reason the program seems slow, one must remember it has to Scan some 320 Recipes.

With good value priced at $69.95 I rate it 9/10.

(Tandy Products. 91 Kurrajong Avenue, Mt.Druitt.N.S.W. 2770).
Reviewed by Arthur Slade. Hills District Color Computer Club.

NUTHACKER

By Peter Fouche

3-D GRAPHIC ADVENTURE
32K ECB DISK (TAPE USE OPTIONAL)

NUTHACKER IS A 3-D graphics adventure for the CoCo 2 & 3. The programs include the following:

* NUTBOOT.BAS: boot program;
* TITLE.NUT: title picture;
* JCSS.NUT: title music;
* NUTHACK.NUT: main program;
* SETUP.NUT: string setup

Once RUN (or CLOADed), you will be presented with a title page and then played a little song (written on MUSIC+), then the main program loads. There is a short pause before starting.

There are two commands that need explaining: "TUNNEL" and "MAP".

"Tunnel" blasts away most holographic walls (created by the mutant CoCo 20), but this can only be done fifteen times, after which the power increase solidifies the walls.

You can play faster, by typing your command before your status is printed. (Not seeing your status is fine when you are just moving around looking, because you see everything your status tells you.)

You can move around the maze even faster by using the shifted arrow keys.

Up moves you in the direction your are looking. Down turns you 180 degrees. Right turns you 90 degrees to the right, and left turns you left.

The scenario:

" ... you are an expert hacker, and are hired by TANDY to save their prototype CoCo 20 which shed it's conscience circuits (which kept it stable), and 'mutated'.

It then set up a holographic maze to protect itself. With equipment left by fleeing technicians you must repair the mutant CoCo and escape - a feat only an expert hacker known as a Nuthacker can possibly hope to achieve!!"

LOADING INSTRUCTIONS

There are two sets of programs provided: one set for the tape user and one set for the disk user. Just follow these instructions and you should be right.

Disk Users who are
typing this program in:

* Format a blank disk.
* Type in listing one; this is your boot program. Delete line 60 and 70 ('DEL60-70'). Save it as "NUTBOOT.BAS".
* Type in listing five; this is the basic version of an ML file, or the Music+ title music. RUN it. This program will automatically save itself under the title of "JCSS.NUT".
* Type in listing two; this is the main program. Save it as "NUTHACK.NUT".
* Type in listing four; this sets the strings up. RUNning it will save a data file for the main program to use, called "NUTDAT.NUT".
* Finished. All you need to do to run it is type in RUN"NUTBOOT".

Tape users who are
typing in this program

* Get a blank tape.
* Type in listing three; this is your boot program. Delete line 60 and 70 ('DEL60-70'). Save it as "NUTBOOT".
* Type in listing five; this is the basic version of an ML file, or the Music+ title music. RUN it. This program will automatically save itself under the title of "JCSS".
* Type in listing two; this is the main program. Replace lines 1000-1050 and lines 30100-30140 with these lines:

```
1000 'SETUP LETTER DRAW STRINGS
1002 POKE65494,0:OPEN"I",#-1,"N
UTDAT"
1005 FORT=65TO75:INPUT#-1,A$(T)
:NEXT:FORT=76TO87:INPUT#-1,A$(T
)
1010 NEXT
1015 INPUT#-1,A$(32)
1020 FORT=48TO56:INPUT#-1,A$(T)
1025 NEXT:INPUT#-1,A$(57)
1030 INPUT#-1,A$(33)
1035 INPUT#-1,A$(40):INPUT#-1,A
$(41)
1040 INPUT#-1,A$(42):INPUT#-1,A
$(44):INPUT#-1,A$(46)
1045 INPUT#-1,A$(8)
1049 INPUT#-1,A$(45):FORT=88TO9
0:INPUT#-1,A$(T):NEXT
1050 RETURN

30100 FORT=1TO12:IF T=5OR T=10O
R T=9THENNEXTT
30110 FORY=1TO4:INPUT#-1,D$(T,Y
):NEXTY
30115 NEXT T
30120 FORT=1TO4:INPUT#-1,D$(5,T
):NEXT:FORT=1TO3:INPUT#-1,D$(9,
T):NEXT:INPUT#-1,D$(9,4)
30130 INPUT#-1,D1$:INPUT#-1,D2$
:INPUT#-1,D$(10,1)
30135 CLOSE:IFPEEK(150)=2THENPO
KE65494,0
30140 RETURN
```

Save it as "NUTHACK".

* Type in listing six; this sets the strings up. RUNning it will save a data file for the main program to use, called "NUTDAT".
* All you need to do to run it is type in CLOAD"NUTBOOT".

Hope you have fun with "Nuthack"!

The Listing:

```
0 GOTO 10
1 '**NUT HACKER LOADER**
2 '*******PROGRAM********
3 '*******8/4/87*********
4 SAVE"NUTBOOT:0":END
10 PCLEAR5
20 CLS:INPUT"CAN YOUR COMPUTER H
ANDLE THE     SPEED UP POKE? IE P
OKE 65495,0      (Y,N)";A$
30 IF A$="Y" THEN POKE 150,2:GOT
O 50
```

```
40 IF A$<>"N" THEN 20
50 PMODE4,1:PCLS
60 LOADM"TITLE.NUT"
70 SCREEN1,1
80 LOADM"JCSS.NUT"
90 POKE &HFF42,0
100 EXEC
110 RUN "NUTHACK.NUT"
```

The Listing:

```
0 GOTO 20
1 ' NUTHACKER
2 ' PETER FOUCHE
3 SAVE"NUTHACK:0"
4 END
20 X=RND(-TIMER):POKE 113,0
21 COLOR 0,0:CLEAR 3700
22 X=10:VIE=1
23 A1=194:B1=58:A4=192:B4=28
24 CO$="N   S   E   W   NORTSOUT
EASTWESTSHOOQUITWEARREADTAKEREPA
DIE ACTILOOKGET EXAMINVESUICPRES
GO  DROPINSEBYE MAP TUNN"
25 DATA "CLOSED     DOOR","OPEN D
OOR","COMPUTER  MANUAL","LAZER
   PISTOL","REFLECTIVEARMOUR","B
UTTON","KEY CARD","COMPUTER  PAR
T","TRANSMAT  DEVICE","BROKEN
 COMPUTER","-","LOCKED      DOOR"

26 DATA"REPAIRED  COMPUTER"
27 NO$="DOORDOORMANUPISTARMOBUTT
CARDPARTTRANCOMPSHIPDOORCOMPNORT
WESTSOUTEAST"
50 DIM A$,B$,L(100,4),A$(122),A(
5),O$(13),D$(15,4),O(17)
55 GOSUB 200
60 GOSUB 1000
65 GOSUB 30000
70 GOSUB 30100
75 GOSUB 400
80 GOTO 20000
100 'INPUT
103 C3=0:C$=INKEY$:IFC$=""THENC3
=1
104 DRAW"S4BM194,0":CU=0:A$=""
105 IFC3 THENB$="YOU ARE INLOCAT
ION":GOSUB300
106 B$=STR$(X):GOSUB300
110 IFC3 THENGOSUB31000
113 IFC3 THENB$="WHAT NEXT OH MA
STER":GOSUB 300
114 B1=168:A1=194
115 A1$=STR$(A1):B1$=STR$(B1):DR
AW"BM"+A1$+","+B1$+"BD6R5"
117 IFC3=0THENB$=C$:C3=1:GOTO140
120 B$=INKEY$:IF B$=""THEN120
140 DRAW"C1L5C0BU6"
145 A=ASC(B$):IF A<>95 AND A<>93
 AND A<>91 AND A<>21 AND A<> 8 A
ND A<>13 THEN 169
150 IF B$=CHR$(13) THEN RETURN
151 IF B$=CHR$(95) THEN IF VIE=1
 THEN A$="N":RETURN ELSE IF VIE=
2 THEN A$="W":RETURN ELSE IF VIE
=3 THEN A$="S":RETURN ELSE A$="E
":RETURN
152 IF B$="[" THEN VIE=VIE-2*(VI
E<3)+2*(VIE>2):A$="LOOK":RETURN
153 IF B$=CHR$(21) THEN VIE=VIE+
1+4*(VIE=4):A$="LOOK":RETURN
154 IF B$="]" THEN VIE=VIE-1-4*(
VIE=1):A$="LOOK":RETURN
155 IF LEN(A$)>0 THEN A$=LEFT$(A
$,LEN(A$)-1) ELSE SOUND 100,1:GO
TO 115
165 IF B$=CHR$(8)  AND CU=>0 THE
N CU=CU-1:A1=A1-6:IF CU=-1 THEN
A1=248:B1=B1-8:CU=9:DRAW"BM"+STR
$(A1+5)+","+STR$(B1):GOSUB 1300:
GOTO 115
166 IF B$=CHR$(8) THENGOSUB1300:
GOTO 115
169 IF LEN(A$)=29 THEN SOUND100,
1:GOTO115
170 GOSUB 1300
171 A$=A$+B$
172 CU=CU+1:A1=A1+6:IF CU=10 THE
N A1=194:B1=B1+8:CU=0
175 GOTO 115
200 'MAZE
203 IF PEEK(150)=2 THEN POKE 654
95,0
205 FORT=1 TO 180:A=RND(100):B=R
ND(4)
210 IF B=1 ANDA>10 THEN L(A,1)=A
-10:L(A-10,3)=A
215 IF B=2 AND (A+9)/10 <>INT((A
+9)/10) THEN L(A,2)=A-1:L(A-1,4)
=A
220 IF B=4 AND A/10<>INT(A/10) T
HEN L(A,4)=A+1:L(A+1,2)=A
225 IF B=3 AND A<91THEN L(A,3)=A
+10:L(A+10,1)=A
230 NEXT
233 L(55,1)=0:L(55,2)=0:L(55,3)=
0:L(55,4)=0:L(65,2)=0:L(65,4)=0:
L(65,3)=75:L(75,3)=85:L(75,2)=74
:L(85,4)=86
234 L(10,2)=9:L(9,4)=10:L(75,1)=
65:L(45,3)=0:L(56,2)=0:L(54,4)=0
:L(86,2)=85:L(85,1)=75:L(64,4)=0
:L(66,2)=0:L(65,1)=0
235 RETURN
300 'WRITE
310 DRAW"BR255BM-60,+8;S4"
330 IF LEN(B$)=<10 THEN GOSUB 13
00:RETURN
335 IF LEN(B$)/10=INT(LEN(B$)/10
) THEN 345
340 B$=B$+" ":GOTO 335
345 C2$=B$:FORT9=1 TO LEN(C2$) S
TEP 10
350 B$=MID$(C2$,T9,10):GOSUB 130
0
353 B1=B1+8
355 DRAW"BR255BM-60,+8":NEXT T9:
RETURN
400 'MAP
405 PMODE0,5:PCLS1
407 COLOR 0,1
410 LINE(40,20)-(200,180),PSET,B
F
415 DRAW"BM40,0S8":B$="MAP OF TR
AVELS":GOSUB 1300
420 FOR T=56 TO 200 STEP 16:LINE
(T,20)-(T,180),PRESET:NEXT
425 FOR T=36 TO 180 STEP 16:LINE
(40,T)-(200,T),PRESET:NEXT
427 B$="PRESS A KEY":DRAW"BM 235
,22A1":GOSUB 1300:DRAW"S4A0"
430 PMODE4,1:SCREEN1,1:COLOR 0,1
:RETURN
450 '*** UPDATE MAP ***'
455 PMODE0,5:DRAW"C0BM"+STR$(A4)
+","+STR$(B4)+"BU8NR7NL6BD8":PMO
DE4,1:B4=B4-16:RETURN'****NORTH*
***'
460 PMODE0,5:DRAW"C0BM"+STR$(A4)
+","+STR$(B4)+"BD8NR7NL6BU8":PMO
DE4,1:B4=B4+16:RETURN'****SOUTH*
***'
465 PMODE0,5:DRAW"C0BM"+STR$(A4)
+","+STR$(B4)+"BL8NU6ND7BR8":PMO
DE4,1:A4=A4-16:RETURN'**WEST**'
470 PMODE0,5:DRAW"C0BM"+STR$(A4)
+","+STR$(B4)+"BR8NU6ND7BL8":PMO
DE4,1:A4=A4+16:RETURN'**EAST**
500 'VIEW
505 PCLS1
507 COLOR0,0:DRAW"C0"
510 DRAW"BM96,5":IF VIE=1 THENB$
="N":GOSUB 1300 ELSE IF VIE=2 TH
EN B$="W" :GOSUB 1300 ELSE IF VI
E=3 THEN B$="S":GOSUB 1300ELSE B
$="E":GOSUB 1300
515 LINE (192,0)-(192,192),PSET:
LINE (0,0)-(192,0),PSET:LINE (0,
0)-(0,192),PSET:LINE (0,192)-(19
2,192),PSET
520 IF L(X,VIE)=0 THEN C=1:GOTO
555
525 A=L(X,VIE):C=2:A(1)=X:A(2)=A
530 A=L(A,VIE):IF A=0 THEN 540 E
LSE C=C+1:IF C<5 THEN A(C)=A:GOT
O 530 ELSE 530
540 IF C=4 THEN DRAW"BM93,99;M98
,99;U5L6D5"
545 IF C=3 THEN DRAW"BM86,106;M1
06,106;U20L20D20"
550 IF C=2 THENDRAW"BM75,117;M11
7,117;U42L42D42"
555 IF C=1 THEN DRAW"BM143,142;M
49,142U92R94D92":A(1)=10
560 IF C>4 THEN DRAW"BM97,98;H4D
4E4D4
565 TIE=VIE+1:IF TIE>4 THEN TIE=
1
570 IF C>3 THEN IF L(A(4),TIE)<>
0 THEN DRAW"BM 93,99;L8ND7U5NU8R
7"ELSE DRAW"BM 93,98;G8U20F8"
575 ' DRAW FAR SIDE/EXIT
580 IF C>2 THEN IF L(A(3),TIE)<>
0 THEN DRAW"BM85,106;L11ND11U20N
U11R11" ELSEDRAW"BM 84,107;G10U4
2F10"
585 IF C>1 THEN IF L(A(2),TIE)<>
0 THEN DRAW"BM49,142;U25NR24U42N
R24U25" ELSE DRAW"BM49,142;NE25U
92NF25"
590 IF C<>0 THEN IF L(X,TIE)<>0
THEN DRAW"BM0,190;U48NR49U92NR49
U49"ELSE DRAW"BM0,192;NE49U190F4
9"
595 TIE=VIE+3:IF TIE>4 THEN TIE=
TIE-4
600 IF C>3 THEN IF L(A(4),TIE)<>
```

There are two sets of programs: one set for disk, the other for tape. Follow the instructions and you should be OK.

```
0 THEN DRAW"BM 98,99;R8ND7U5NU8L
7"ELSE DRAW"BM 99,98;F8U20G8"
605 IF C>2 THEN IF L(A(3),TIE)<>
0 THEN DRAW"BM107,106;R11ND11U20
NU11L11"ELSEDRAW"BM108,107;F10U4
2G10"
610 IF C>1THEN IF L(A(2),TIE)<>0
 THEN DRAW"BM143,142;U25NL24U42N
L24U25" ELSE DRAW"BM143,142;NH24
U92NG25"
615 IF C<>0 THEN IF L(X,TIE)<>0
THEN DRAW"BM192,190;U48NL49U92NL
49U49"ELSEDRAW"BM192,192;NH49U19
0G49"
620 RETURN
1000 'LOAD$'S
1002 OPEN"D",#1,"NUTDAT.NUT"
1005 GET#1,1:FORT=65 TO 75:INPUT
#1,A$(T):NEXT:GET #1,2:FORT=76 T
O 87:INPUT #1,A$(T)
1010 NEXT
1015 GET#1,3:INPUT#1, A$(32)
1020 FORT=48 TO 56:INPUT#1,A$(T)
1025 NEXT:GET#1,4:INPUT#1,A$(57)
1030 INPUT#1, A$(33)
1035 INPUT#1, A$(40):INPUT#1,A$(
41)
1040 INPUT#1,A$(42):INPUT#1,A$(4
4):INPUT#1,A$(46)
1045 INPUT#1,A$(8)
1049 INPUT#1,A$(45):GET#1,9:FORT
=88 TO 90:INPUT#1, A$(T):NEXT
1050 RETURN
1300 FORZ7=1 TO LEN(B$)
1305 P$=MID$(B$,Z7,1)
1310 P=ASC(P$)
1315 DRAW"XA$(P);"
1320 NEXT:RETURN
1400 '*CLS*
```

```
1045 INPUT#1,A$(8)
1049 INPUT#1,A$(45):GET#1,9:FORT
=88 TO 90:INPUT#1, A$(T):NEXT
1050 RETURN
1300 FORZ7=1 TO LEN(B$)
1305 P$=MID$(B$,Z7,1)
1310 P=ASC(P$)
1315 DRAW"XA$(P);"
1320 NEXT:RETURN
1400 '*CLS*
```

```
1405 COLORO,1:LINE(193,0)-(255,1
92),PRESET,BF:DRAW"BU255"
1406 CS=1:RETURN
1450 'FAIL
1455 PLAY"L10T3EECFL5EC":FORT=1T
O2000:NEXT:PCLS1:B$="YOU HAVE BO
TCHED UP THE JOB.":DRAW "BM10,10
":GOSUB 1300
1460 B$="DO YOU WANT ANOTHER ATT
EMPT":DRAW"BM10,20":GOSUB 1300
1470 B$="PLEASE TYPE (Y) OR (N)"
:DRAW"BM80,70":GOSUB 1300
1475 DRAW D$(10,1)
1480 B$=INKEY$:IF B$="" THEN 148
0
1485 IF B$="N" THEN PCLS:CLS:POK
E 113,0:EXEC40999
1490 IF B$="Y" THEN B$="PLEASE
  WAIT WHILEA NEW MAZEIS SET UP.
":GOSUB 300:X=10:VIE=1:A1=194:B1
=58:A4=192:B4=28:WEA=0:REA=0:INS
=0:FORT=1TO4:I(T)=0:NEXTT:GOSUB
200:GOSUB 400:RESTORE:GOSUB30000
:GOTO 20000
1495 GOTO 1480
1499 GOTO 1499
1500 'T=COMND
1505 A$=A$+"    ":C=0
1507 FOR T=1 TO LEN(A$):IF MID$(
A$,T,1)<>" " THEN NEXT T
1510 C$=LEFT$(A$,T):C$=C$+"     "
:C$=LEFT$(C$,4)
1515 FOR T=1 TO LEN(CO$) STEP 4
1520 IF MID$(CO$,T,4)=C$ THEN 15
30
1525 NEXT :B$="WHAT":GOSUB 1400:
GOSUB 300:WA=1:RETURN
1530 C=(T+3)/4:RETURN
1550 '*** PRESS A KEY ***'
1552 IF CS=0 THEN GOSUB 1400:CS=
1:'CLS
1555 B$="  PLEASE    PRESS A
 KEY":GOSUB 300
1556 IF INKEY$<>"" THEN 1556
1557 IF INKEY$="" THEN 1557
1560 RETURN
1601 '*#*COMMANDS*#*
1605 '*** NORTH ***'
1610 IF L(X,1)=0 THEN VIE=1:PLAY
"T100CC#C":RETURN
1615 GOSUB 455
1640 VIE=1:X=L(X,1):RETURN
1650 '***SOUTH***'
1655 IF L(X,3)=0 THEN VIE=3:PLAY
"T100CC#C":RETURN
1660 GOSUB 460
1690 VIE=3:X=L(X,3):RETURN
1700 '***EAST***'
1705 IF L(X,4)=0 THEN PLAY"T100C
C#C":VIE=4:RETURN
1710 GOSUB 470
1740 VIE=4:X=L(X,4):RETURN
1750 '***WEST***'
1755 IF L(X,2)=0 THEN PLAY"T100C
C#C":VIE=2:RETURN
1760 GOSUB 465
1790 VIE=2:X=L(X,2):RETURN
1800 '*** SHOOT ***'
1805 GOSUB 5000
1810 GOSUB 1400
1812 WA=1
1815 IF ER=1 THEN RETURN
1816 FORR=1 TO 4:IF I(R)<> 4 THE
N NEXT R:B$="YOU HAVE  NO GUN!":
GOSUB 300:RETURN
1817 IF O(T)<>X THEN B$="I DONT
SEE ONE HERE":GOSUB 300:RETURN
1818 PLAY"V10T255L255CC#CV15C#DC
#V20D#DV30D#ED#"
1820 IF T<3 OR T=5 OR T=6 OR T>1
0 THEN B$="IT HAS NO  EFFECT":GO
SUB 300:RETURN
1825 IF T=3 OR T=7 OR T=8 OR T=9
 THEN B$="YOU LL BE  SORRY!":GOS
UB 300:O(T)=0:RETURN
1830 IF T=10 THEN B$="YOU HIT IT
BUT  YOU   TRIGGER   HALON GAS W
HICH      SMOTHERS  YOU. YOU  ARE
 DEAD.":GOSUB 300:GOTO 1450
1900 '*** QUIT ***'
1905 GOSUB 1400:'CLS
1910 B$="QUIT! WHATARE YOU!  STI
LL ITS   YOUR    CHOISE":GOSUB 3
00:B$="QUIT-(Y,N)":GOSUB 300
1915 A$=INKEY$:IF A$="" THEN 191
5
1920 IF A$="N" THEN RETURN
1925 IF A$<>"Y" THEN 1915
1930 POKE 113,0:EXEC40999
1935 END
2050 '*** WEAR ***'
2053 GOSUB 1400
2055 GOSUB 5000
2060 IF ER=1 THEN RETURN
2061 WA=1
2065 IF T<>5 THEN B$="YOU CANT
WEAR THAT":GOSUB 300:RETURN
2066 IF WEA=1 THEN B$="YOU WEAR
 IT ALREADY":GOSUB 300:RETURN
2067 FORT=1 TO 4:IF I(T)<>5 THEN
 NEXT:B$="YOU DO NOTCARRY IT!":G
OSUB 300:RETURN
2070 B$="YOU PUT ITON. NOW   YOU
 CAN   CARRY ONE MORE THING":GOS
UB 300
2080 FORT=1 TO 4:IF I(T)<>5 THEN
 NEXT T ELSE I(T)=0
2085 WEA=1
2090 RETURN
2150 '*** READ ***'
2155 GOSUB 1400
2160 GOSUB 5000
2163 WA=1
2165 IF ER=1 THEN RETURN
2167 IF T<>3 THEN B$="YOU CANNOT
READ THAT":GOSUB 300:RETURN
2170 FORT=1 TO 4:IF I(T)<> 3 THE
N NEXT T:B$="YOU DONT  HAVE IT":
GOSUB 300:RETURN
2175 B$=" IT SAYS..REPAIR    COM
PUTER  ..IT ALSO GIVES SOMEOTHER
 DATAON REPAIR TECHNIQUES":GOSUB
 300
2180 REA=1:RETURN
2250 '*** GET ***'
2255 GOSUB 5000
2260 IF ER=1 THEN RETURN
2263 GOSUB 1400
```

```
2264 IFT=1ORT=2ORT=6ORT>8 THEN B
$="YOU CANT   GET THAT!":GOSUB 3
00:WA=1:RETURN
2265 FORR=1 TO 4 :IF I(R)<>0 THE
N NEXT:B$="I CANNOT   CARRY IT!"
:GOSUB 300:WA=1:RETURN
2270 I(R)=T:B$="YOU NOW   CARRY
THE":GOSUB 300:B$=O$(T):GOSUB 30
0
2272 O(T)=0
2275 WA=1:RETURN
2350 '*** REPAIR ***'
2355 GOSUB 5000
2360 IF ER=1 THEN RETURN
2365 WA=1:GOSUB 1400
2370 IF T<>10 THEN B$="IT LOOKS
 PERFECTLY GOOD TO ME":GOSUB 300
:RETURN
2375 IF O(10)<>55 THEN T=9:GOTO2
370
2380 IF REA=0 THEN B$="I DONT
 KNOW HOW!":GOSUB 300:RETURN
2385 FORT=1 TO 4:IF I(T)<>8 THEN
 NEXT :B$="WHAT WITH?":GOSUB 300
:RETURN
2390 B$="O.K.!!    ITS FIXED!":G
OSUB 300
2395 PLAY"O2V30L4T8EEEL1C"
2400 O(10)=0:O(13)=55:I(T)=0
2405 RETURN
2450 '*** ACTIVATE ***'
2455 GOSUB5000:IF  ER=1 THEN RET
URN
2457 WA=1
2460 GOSUB 1400
2465 IF T<>9 THEN B$="WHAT FOR":
GOSUB 300:RETURN
2470 IF INS=0 THEN B$="INSERT KE
YCARD FIRST":GOSUB 300:RETURN
2475 IF O(9)<>X THEN B$="ACTIVAT
E!!I CANT EVEN SEE IT!":GOSUB 30
0:RETURN
2480 B$="TRANSMAT  OPERABLE":GOS
UB 300:TEL=1:RETURN
2500 '***LOOK***'
2503 DRAW"C1"
2504 NE=1
2505 GOSUB 5000
2507 IF ER=1 THEN ER=0:WA=0:RETU
RN
2509 IF T<14 THEN ER=1:RETURN
2510 VIE=T-13
2515 RETURN
2700 '*** EXAM ***'
2705 GOSUB5000:IF ER=1THENRETURN
2707 GOSUB1400:WA=1
2710 IF O(T)=X THEN 2720
2715 FORR=1TO4:IFI(R)<>T THEN NE
XT:B$="ITS NOT   HERE":GOSUB300:
RETURN
2720 IFT<>7THENB$="LOOKS LIKE A"
:GOSUB 300:B$=O$(T):GOSUB300:RET
URN
2725 B$="ITS MULTI-PURPOSE":GOSU
B 300:RETURN
2800 '*** INVENTORY ***'
2810 GOSUB 1400
2815 A=0
2820 B$="YOU NOW    CARRY":GOSUB
 300
2830 FORT=1 TO 4:B$=O$(I(T)):IF
B$="" THEN NEXT T:GOTO 2845
2840 GOSUB 300:A=1:NEXT T
2845 WA=1:IF A=0 THEN B$="NOTHIN
G":GOSUB 300:RETURN ELSE RETURN
2900 '*** PRESS ***'
2905 GOSUB 1400:WA=1:GOSUB 5000:
IFER=1 THEN RETURN
2910 IF T=4 THEN B$="ITS LIKELYT
O GO OFF":GOSUB 300:RETURN
2915 IF T=10 THEN B$="CAREFULL!!
YOU COULD BREAK IT":GOSUB 300:RE
TURN
2920 IF T<>6 THEN B$="THERE IS
LITTLE    EFFECT":GOSUB 300:RETU
RN
2935 IF X<>65 AND X<>55 THEN B$=
"I CANT SEEONE":GOSUB300:RETURN
2937 IF O(13)=65 THEN B$="NOTHIN
G   HAPPENS!":GOSUB 300:RETURN
2940 IFO(1)=65THEN O(1)=0:O(2)=6
5:B$="O.K. THE  DOOR OPENS":GOSU
B300:RETURN
2941 IF O(12)=65 THEN B$="NOTHIN
G   HAPPENS!":GOSUB 300:RETURN
2942 O(2)=0::O(1)=65:B$="THE DOO
R  CLOSES!":GOSUB 300:RETURN
3000 '*** GO ***'
3010 GOSUB 5000:IF ER=1 THEN RET
URN
3020 IF T<>1 THEN 3070
3025 WA=1:GOSUB 1400
3030 IF X<>65 AND X<>55 THENB$="
I CANT SEEONE!":GOSUB300:RETURN
3040 IFO(2)<>65THEN B$="YOU CANT
  GO THROUGHSHUT DOORS":GOSUB300
:RETURN
3050 IF O(2)=65 THEN IF X=65 THE
N X=55:VIE=1 ELSE IF X=55 THEN X
=65:VIE=3
3060 PLAY"V10T255L255CC#CV15C#DC
#V20D#DV30D#ED#":B$="THE MUTANTC
OCO 20   SHOOTS YOUWITH ITS  DEF
ENCE   LASER":GOSUB300
3065 IF WEA=0THEN 1450 ELSEB$="T
HE BEAM  BOUNCES OFTHE ARMOUR":G
OSUB300:RETURN
3070 IF T<>9 THEN 3100
3075 GOSUB1400:WA=1
3080 IF TEL<>1 THEN B$="IT IS NO
T OPERABLE":GOSUB300:RETURN
3090 IFX<>O(9)THENB$="IT IS NOT
HERE":GOSUB 300:RETURN
3095 B$="YOU GET ONAND ARE   ZAP
PED     AWAY":GOSUB300:GOTO 40000
3100 IF T> 13 THEN ON (T-13) GOS
UB 1605,1750,1650,1700
3110 RETURN
3150 '*** DROP ***'
3155 GOSUB 5000:  IF ER=1THEN RET
URN
3160 GOSUB1400:WA=1
3165 IF T=5 ANDWEA=1THEN WEA=0:O
(T)=X:B$="YOU TAKE  IT OFF ANDDR
OP IT":GOSUB300:RETURN
3170 FORY=1 TO 4:IF I(Y)<>T THEN
 NEXT:B$="I SEEM TO HAVE LOST IT
":GOSUB300:RETURN
3175 B$="O.K. YOU  DROP IT":GOSU
B 300:I(Y)=0:O(T)=X:RETURN
3250 '*** INSERT ***'
3255 GOSUB 5000:GOSUB1400:IF ER=
1 THEN RETURN
3257 WA=1
3260 IF T<>7 THEN B$="IT WONT
FIT!":GOSUB 300:RETURN
3265 FORT=1 TO 4:  IF I(T)<>7 THE
N NEXT:B$="YOU HAVENTGOT IT!!":G
OSUB 300:RETURN
3266 IF X=O(9)THEN3270
3267 IFX=65ORX=55 THEN 3280
3268 B$="INTO WHAT!":GOSUB 300:R
ETURN
3270 INS=1:B$="ACCESS    ACCEPTE
D. YOU GET    THE CARD":GOSUB 300
:RETURN
3280 B$="AFTER THE TUMBLERS  CLI
CK YOU REMOVE THECARD":GOSUB 300
3285 IF O(12)=65 THEN O(12)=0:O(
1)=65:B$="THE DOOR  UNLOCKED":GO
SUB 300:RETURN
3290 O(12)=65:O(1)=0:O(2)=0:B$="
YOU LOCKEDTHE DOOR":GOSUB300:RET
URN
3300 '*** MAP ***'
3305 PMODE0,5:SCREEN1,1
3307 DRAW"BM"+STR$(A4)+","+STR$(
B4)+"C1L2U2R2D2"
3310 IF INKEY$<>"" THEN 3310
3315 IF INKEY$="" THEN 3315
3317 DRAW"C0U2L2D2R2"
3320 PMODE4,1:SCREEN1:RETURN
3350 '*** TUNNEL ***'
3351 IFX=55ORX=65THEN 3380
3352 TN1=TN1+1:IFTN1>14THENB$="A
LL WALLS ARE NOW   SOLID":GOSUB1
400:GOSUB300:WA=1:RETURN
3353 IFL(X,VIE)<>0 THENGOSUB1400
:B$="NO WALL":GOSUB300:WA=1:RETU
RN
3355 IF VIE=1 THEN IF X<11THEN 3
380 ELSE L(X,1)=X-10:L(X-10,3)=X
3360 IF VIE=2 THEN IF (X+9)/10 =
INT((X+9)/10)ORX=56ORX=66THEN338
0 ELSE L(X,2)=X-1:L(X-1,4)=X
3365 IF VIE=3THENIFX>90 ORX=45 T
HEN3380 ELSE L(X,3)=X+10:L(X+10,
1)=X
3370 IF VIE=4 THEN IFX/10=INT(X/
10)ORX=54 ORX=64THEN 3380 ELSE L
(X,4)=X+1:L(X+1,2)=X
3375 GOSUB1400:B$="OK!":GOSUB300
:WA=1:GOSUB 300:WA=1
3377 RETURN
3380 GOSUB 1400:B$="THE WALLS AR
E SOLID   -SORRY-":GOSUB 300:WA=
1
3385 RETURN
5000 ' DETERMINE NOUN '
5003 ER=0
5005 FOR T=1 TO LEN(A$)
5006 IF MID$(A$,T,1)<>" " THENNE
XT
5015 C$=MID$(A$,T+1,4)
5020 FORT=1TO LEN(NO$)STEP4
5025 IF C$<>MID$(NO$,T,4) THEN N
EXT T ELSE 5035
5030 ER=1:IF NE=0 THEN GOSUB 140
0:B$="I DONT SEEONE HERE!!":WA=1
:GOSUB 300:RETURN ELSE NE=0:RETU
```

```
RN:' NE=0 - REPORT ERR
5035 T=(T+3)/4
5040 RETURN
20000 ' MIAN CONTROL LOOP
20003 IF WA=1 THEN GOSUB 1550
20004 WA=0' RESET WAIT
20005 GOSUB 500' DRAW VIEW
20007 GOSUB 31031'DRAW OBJECTS
20008 CS=0
20010 GOSUB 100' INPUT COMMAND
20015 GOSUB 1500' DESIPHER INPUT
20020 ON C GOSUB 1605,1650,1700,
1750,1605,1650,1700,1750,1800,19
00,2050,2150,2250,2350,1900,2450
,2500,2250,2700,2800,1900,2900,3
000,3150,3250,1900,3300,3350
20025 GOTO 20000
30000 '*** SET UP OBJECTS ***'
30020 FOR R=1 TO 13
30030 O(R)=RND(100):NEXT:O(1)=-1
:O(2)=-1:O(10)=55:O(12)=65:O(6)=
65:O(11)=0:O(13)=0
30040 FORT=1 TO 13:READ O$(T):NE
XT T
30098 RETURN
30099 '***LOAD DRAW$***'
30100 FORT=1 TO 12:IF T=5 OR T=1
0 OR T=9 THENNEXTT
30110 GET#1,4+T:FORY=1 TO 4:INPU
T#1, D$(T,Y):NEXTY
30115 NEXT T
30120 FORT=1 TO 4:GET#1,16+T:INP
UT#1,D$(5,T):NEXT:GET#1,21:FORT=
1 TO 3:INPUT#1,D$(9,T):NEXT:GET
#1,22:INPUT#1,D$(9,4)
30125 GET#1,23:INPUT#1,A$
30130 GET#1,23:INPUT#1,D1$:GET#1
,24:INPUT#1,D2$:GET#1,25:INPUT#1
,D$(10,1)
30135 CLOSE:RETURN
30140 OPEN"D",#1,"NUTDAT.NUT"
30145 GET#1,23
30150 INPUT#1,A$:CLOSE
30155 END
30199 RETURN
31000 'OBJECTS
31010 B$="IN THIS   ROOM YOU   S
EE:":GOSUB 300:CC=0
31020 FOR R=1 TO 13:IF O(R)=X TH
EN B$=O$(R):GOSUB 300:CC=1
31030 NEXT:IF UD<>0 THEN B$=O$(U
D):GOSUB 300:B$=O$(6):GOSUB 300:
RETURN ELSE IF CC=0 THEN B$="NOT
HING":GOSUB 300:RETURN ELSE RETU
RN
31031 '***DRAW OBLECTS***'
31032 IF C>4THEN C=4
31033 A(1)=X:UD=0
31034 FORR=1 TO 13:FORT=0 TO C
31035 IF X<>65 AND X<>55 THEN 31
045
31036 IF X=65 THEN IF VIE<>1 THE
N IF R=6 OR R=2 OR R=1 OR R=12 T
HEN R=R+1
31038 IF X=55 AND VIE=3 THEN DRA
W D$(6,1):IF O(12)=65 THEN DRAW
D$(12,1) :UD=12ELSE IF O(2)=65 T
HEN DRAW D$(2,1) :UD=2 ELSE IF O
(1)=65 THEN DRAW D$(1,1):UD=1
31039 IFX=55ANDVIE<>3 THENIFR=13
 THENDRAWD$(10,1)
31040 IFX=55 ANDVIE=3 THENIFR=10
 OR R=13THEN R=R+1
31045 IF O(R)=A(T) THEN DRAW D$(
R,T)
31047 NEXTT,R
31049 RETURN
40000 '*** WINNER ***
40005 IFO(10)=55THEN B$="YOU LEF
T WITHOUT   REPAIRING THE COCO2
0":GOSUB300:GOTO1450
40010 PLAY "T30L1FL2CP5CL1C#L1CC
P5L1EF"
40015 FORT=1TO2000:NEXT
40020 PCLS1:DRAW"BM20,20S8":B$="
CONGRATULATIONS!":GOSUB 1300:DRA
W"BM20,30":B$=STRING$(19,"-"):GO
SUB1300
40025 DRAW"BM20,50S4":B$="YOU SU
CCEEDED! WHAT CAN I SAY!":GOSUB
1300:DRAW"BM20,60":B$="VERY FEW
DO!":GOSUB 1300
40030 DRAW"BM20,75":B$=" DO YOU
WANT TO DO IT AGAIN":GOSUB 1300:
B$="(Y) OR (N)":DRAW"BM130,94":G
OSUB 1300
40035 DRAWD$(10,1)
40040 A$=INKEY$:IF A$="" THEN400
40
40045 IF A$="Y" THEN1490
40050 IFA$="N"THEN POKE 113,0:EX
EC40999
40060 GOTO 40040
```

You can move around the maze even faster by using the shifted arrow keys.

The Listing:

```
0 GOTO 10
1 '**NUT HACKER LOADER**
2 '********PROGRAM**********
3 '********8/4/87***********
4 CSAVE"NUTBOOT":END
10 PCLEAR5:CLEAR3700,32707
20 CLS:INPUT"WILL YOUR COMPUTER
HANDLE THE   SPEED UP POKE? IE P
OKE 65495,0       (Y,N)";A$
30 IF A$="Y" THEN POKE 150,2:GOT
O 50
40 IF A$<>"N" THEN 20
50 PMODE4,1:PCLS
60 CLOADM"TITLE"
70 SCREEN1,1
80 CLOADM"JCSS"
90 EXEC
100 EXEC 32708
```

The Listing:

```
0 GOTO 10
1 '****** STRING SETUP *********
2 '****** PETER FOUCHE *********
3 '********** 7/6/87 **********
4 SAVE "NUTSETUP:0":END
10 'SET UP LETTER DRAW STRINGS'
15 CLEAR2000:DIM A$(122),O$(13),
D$(13,4)
17 OPEN "D",#1,"NUTDAT/NUT"
20 FORT=65 TO 75:READ A$(T):WRIT
E #1,A$(T):NEXT:PUT#1,1:FORT=76
TO 87:READA$(T):WRITE #1,A$(T):N
EXT:PUT#1,2
30 FORT= 88 TO 90:READ A$(T):NEX
T T
40 READ A$(32):WRITE #1,A$(32)
50 FORT=48 TO 56:READA$(T):WRITE
#1,A$(T):NEXT T:PUT#1,3:READA$(5
7):WRITE#1, A$(57)
70 READ A$(33):WRITE #1,A$(33)
80 READ A$(40):READ A$(41):WRITE
#1,A$(40):WRITE #1,A$(41)
90 READ A$(42):READ A$(44):READ
A$(46):WRITE #1,A$(42):WRITE#1,A
$(44):WRITE#1,A$(46)
100 READ A$(8):WRITE#1,A$(8)
110 READ A$(45):WRITE#1,A$(45):P
UT #1,4
115 FORT=88 TO 90:WRITE#1,A$(T):
NEXT:PUT#1,9
130 DATA"BD2ND4D1R4ND3U1H2G2BR4B
E2"
140 DATA"D6R3E1U1H1NL3E1U1H1NL3B
R3"
150 DATA "BD1D4F1R2E1BU4H1L2BR5"
160 DATA "R1D6NL1R2E1U4H1NL3BR3"
170 DATA "D3NR2D3R4BU6NL4BR2"
180 DATA "D3NR2D3BE4BU2NL4BR2"
190 DATA "BD1D4F1R2U2NL1BU3U1NL2
BR3"
200 DATA "D6U3R4ND3U3BR2"
210 DATA "BR1R1D6NL1NR1U6R1BR3"
220 DATA "BD5F1R1E1U5BR3"
230 DATA "D6U3R1NF3E3BR2"
240 DATA "D6R4BU6BR2"
250 DATA "ND6F2E2ND6BR2"
260 DATA "ND6F4D2U6BR2"
270 DATA "BD1D4F1R2E1U4H1NL2BR3"
280 DATA "ND6D3R3E1U1H1NL3BR3"
290 DATA "BD1D4F1R1E1NH1NF1E1U3H
1NL2BR3"
300 DATA "D6U3R1NF3R2E1U1H1NL3BR
3"
310 DATA "BD5F1R2E1H4E1R2F1BU1BR
2"
320 DATA "R2ND6R2BR2"
330 DATA "D5F1R2E1U5BR2"
340 DATA "D2F1D1F1ND1E1U1E1U2BR2
"
350 DATA "D6E2F2U6BR2"
360 DATA "D1F4ND1H2G2ND1E4U1BR2"
370 DATA "D1F2ND3E2U1BR2"
380 DATA "R4D1G4D1R4BU6BR2
390 DATA "BR6"
400 DATA "BD1D4F1R2E1U4NG4H1L2BR
4"
410 DATA "BD1BR1E1D6NR1NL1BU6BR4
"
```

```
420 DATA "BD1E1R2F1D1G1L2G1D2R4B
U6BR2"
430 DATA "BD1E1R2F1D1G1NL1F1D1G1
L2H1BR6BU5"
440 DATA "BR3ND6G3R4BU3BR2"
450 DATA "NR4D2R3F1D2G1L2H1BR6BU
5"
460 DATA "BR4L2G1D4F1R2E1U1H1L3B
R6BU3"
470 DATA "R4D1G3D2BR6BU6"
480 DATA "BR1G1D1F1G1D1F1R2E1U1H
1NL2E1U1H1NL2BR3"
490 DATA "BR1G1D1F1R3D2G1L2R2E1U
4H1NL2BR3"
500 DATA "BR1D3BD2D1BU6BR3"
510 DATA "BR4G2D2F2BR3BU6"
520 DATA "F2D2G2BR4BU6"
530 DATA "BR3D5U3NF2NG2D1NH2E2BR
3BU1"
540 DATA "BD4BR2ND1R1D1BD1G1E1U1
BU5BR3"
550 DATA "BD4BR2D1R1U1BR4BU4"
560 DATA"C1L5D1R4D1L4D1R4D1L4D1R
4DL5U6C0"
570 DATA "BD3R4BU3BR"
1000 '*** SET UP OBJECTS ***'
1010 D$="U70L30D70BE25BU20U4GDFE
UHS4":D$(1,1)="BM112,142"+D$
1020 D$(1,2)="BM104,117S2"+D$:'S
2U70L30D70BE25BU20U2ND4BH2ND8G2D
4S4
1030 D$(1,3)="BM100,105;S1U65L30
D65BE25BU20U2ND4BH2ND8G2D4S4"
1040 D$(1,4)="BM96,96ND4L1D3"
1050 D$(2,1)="BM112,142;U70L30D7
0M+11,-8U22BL1U2HGD2FEBR1U33M-11
,-5"
1060 D$(2,2)="BM104,117;S2U70L30
D70M+11,-8U22BL2U2HGD2FEBR2U33M-
11,-5"
1070 D$(2,3)="BM100,105;S1U65L30
D65M+11,-8U19BL2U1BR2U30M-11,-8
1080 D$(2,4)=D$(1,4)
1090 D$(12,1)=D$(1,1):D$(12,2)=D
$(1,2):D$(12,3)=D$(1,3):D$(12,4)
=D$(1,4)
1100 D$="NM+8,-4;D15R10U15L10R4N
M+5,-4R3E2S4":D$(3,1)="BM131,172
;S4"+D$
1110 D$(3,2)="BM108,124;S3"+D$
1120 D$(3,3)="BM99,110;S2"+D$
1130 D$(3,4)="BM98,102;S1"+D$
1140 D$="R12DL6U3NLNRD3NL4FNL3GB
L2UD4L2RU2S4":D$(4,1)="BM116,172
;S4"+D$
1150 D$(4,2)="BM100,124;S3"+D$
1160 D$(4,3)="BM98,110;S2"+D$
1170 D$(4,4)="BM97,102;S1"+D$
1180 D$=";M-11,+9;M+3,+8;M+10,-5
;M+5,-5;GU5L2GLU2L4BF4LGLDL6DLD2
LBR3NM+5,-3DLG3BM+9,+1R3FGNL2NGD
RNG2D2GL2H2U3BE5URU3RUR3S4":D$(5
,1)="BM100,157;S6"+D$
1190 D$(5,2)="BM100,120;S4;"+D$
1200 D$(5,3)="BM98,107 ;S2"+D$
1210 D$(5,4)="BM97,100;S1"+D$
1220 D$="U4ER4FD4GL4BE2ULDS4"
1230 D$(6,1)="BM116,96S4"+D$
1240 D$(6,2)="BM107,96S3"+D$
1250 D$(6,3)="BM102,96S2"+D$
1260 D$(6,4)="BM97,97L"
1270 D$="L7ER7EL7ER7EL7S4"
1280 D$(7,1)="BM60,170"+D$
1290 D$(7,2)="BM75,140E3LG3LE3LG
3LE3"
1300 D$(7,3)="BM88,115ELGLE"
1310 D$(7,4)="BM92,105L
1320 D$="U2E4R7ND2G4ND2L3ND2L4E2
BR3RS4":D$(8,1)="BM75,170"+D$
1330 D$(8,2)="BM84,140S3"+D$
1340 D$(8,3)="BM89,115S2"+D$
1350 D$(8,4)="BM93,105S1"+D$
1360 D$="R80H20L40G20R5E20L5BD2B
LBRD2UBG5G2BG1G7BR30R15E5U5H5L15
G5D5F5S4":D$(9,1)="BM15,191S8"+D
$
1370 D$(9,2)="BM56,143S4"+D$
1380 D$(9,3)="BM75,120S2"+D$
1390 D$(9,4)="BM85,106S1"+D$
1400 D1$="BM123,110;XD2$;F17NM-5
6,+28D34M-56,+28L2NH31RU33H32ND3
4UM+24,-12;NEHL3H2L8G2DF2RE2UHBD
4R9URUR4D9HNU9NH2F18D3GLNF3GNF3G
NF3RNELGLG2LNE8LNE9H19DNF19U2NRE
8DNG8R2NF18HBD3F18G6H12RNF5GNF5U
H5E6G1; F2BF2FBF2FBF2FBF2FDHBH2H
BH2HBH2HBH2HGNLNDBF2NURDB
1410 D$(10,1)="XD1$;NURDBF2NUBF2
NURDBF2DBL2NDLUBH2NDLUBH2NDLUBH2
NDLUBH2NDLUBR12E4FNG4FNG4FNG4F9R
FREU20H21DNF21DNF21LNU2ND21NULD2
0NF8RNF20RF20U24F2;RERER3ER2E2RE
2RE3D8G3LDG2DG2NRDGDLD2G2NRDGDGD
L2DL2G2BM+2,-31;BR18UHH9L2GL3DL8
HL5HL5HL4":D$(13,1)=D$(10,1)
1420 D2$="C1LD32NL70NR18U32RC0"
1430 FORT=1 TO 12:FORY=1 TO 4:IF
 T=10 OR T=9 OR T=5 THEN NEXT T
1440 WRITE #1, D$(T,Y):NEXT Y:PU
T#1,4+T:NEXT T
1445 FORY=1 TO 4:WRITE#1,D$(5,Y)
:PUT #1,16+Y:NEXT Y
1447 WRITE #1,D$(9,1):WRITE #1,D
$(9,2):WRITE #1,D$(9,3):PUT#1,21
:WRITE #1,D$(9,4):PUT#1,22
1450 WRITE #1,D1$:PUT#1,23:WRITE
 #1,D2$:PUT#1,24:WRITE #1,D$(10,
1):PUT#1,25
1460 CLOSE#1
```

The Listing:

```
20 FOR I= 19712TO 20529
30 READ Q$:POKEI,VAL("&H"+Q$):NE
XTI
40 IF PEEK(49152)=68 THEN SAVEM"
JCSS.NUT",19172,20529,20100:END
41 CSAVEM"JCSS",19172,20529,2010
0:END
50 DATA D,E,F,11,12,14,15,16,18,
19,1B,1C,1D,1F,20,22,23,26,27,29
,2A,2B,2D,2E,30,31,32,34,35,37,3
8,38,39,3B,3B,3C,3C,3E,3E,3E,3F,
3F,3F,3F,3F,3F,3F,3E,3E,3E,3C,3C
,3B,3B,39,38,37,37,35,34,32,31,3
0,2E,2E,2D,2B,2A,29,27,26,24,23,
22,20
60 DATA 1F,1F,1D,1C,1B,1B,19,19,
18,18,16,16,15,15,15,15,15,15,15
,15,15,15,15,15,15,16,16,16,18,1
8,18,19,19,1A,1A,1B,1B,1B,1C,1C,
1E,1F,1F,20,20,22,22,22,22,22,23
,23,23,23,23,23,23,22,22,22,22,2
0,20,20,1F,1F,1D,1D,1C,1C,1B,19,
19,18,16
70 DATA 15,15,14,12,12,11,F,E,E,
D,B,B,A,8,8,7,7,6,6,4,4,3,3,3,1,
1,1,1,1,1,0,0,0,0,0,1,1,1,1,1,1,
1,3,3,3,3,3,4,4,4,4,4,6,6,6,6,6,
6,6,6,7,7,7,7,7,7,7,6,6,6,6,6,6,
6,6
80 DATA 6,4,4,4,4,4,4,4,4,4,4,3,
3,3,4,4,4,4,4,4,6,6,6,7,7,7,8,A,
A,B,D,0,0,2,6F,2,94,2,BB,2,E4,3,
10,3,3F,3,71,3,A5,3,DC,4,17,4,56
,4,98,4,DD,5,28,5,76,5,C9,6,21,6
,7F,6,E2,7,4A,7,B9
90 DATA 8,2F,8,AC,9,30,9,BB,A,50
,A,ED,B,93,C,43,C,FE,D,C4,E,95,F
,73,10,5F,11,58,12,60,13,77,14,A
0,15,DA,17,26,18,87,19,FC,1B,88,
1D,2B,1E,D3,20,BE,22,B0,24,C0,26
,EF,29,40,2B,B4,2E,4D,31,E,33,F9
,37,10,3A,57,3D,CF,41,7C,45
100 DATA 60,49,81,4D,DF,52,81,57
,69,5C,9B,62,1D,16,0,88,86,4C,1F
,8B,86,3F,B7,FF,23,1A,50,30,8D,F
E,6A,9F,C3,9F,C6,9F,C9,9F,CC,30,
8D,0,80,9F,C0,39,9E,C0,A6,80,97,
C2,39,31,8D,FF,50,A6,80,EE,A6,DF
,CF,A6,80,EE,A6,DF,D1,A6,80,EE,A
6,DF,D3
110 DATA A6,80,EE,A6,DF,D5,9F,C0
,39,10,8E,0,AB,A6,9F,4C,C3,AB,9F
,4C,C6,A9,9F,4C,C9,A9,9F,4C,CC,B
7,FF,20,DC,C4,D3,CF,DD,C4,DC,C7,
D3,D1,DD,C7,DC,CA,D3,D3,DD,CA,DC
,CD,D3,D5,DD,CD,31,A2,26,6,A,C2,
27,C,20,C7,A6,80,20,0,20,0,20,0,
20
120 DATA C1,39,17,FF,75,8D,91,81
,0,26,4,4F,1F,8B,39,8D,8E,8D,AB,
20,F0,C,32,0,0,0,C,36,0,0,0,30,3
A,32,28,0,C,3C,32,28,0,C,3A,32,2
8,0,C,36,32,28,0,C,32,0,28,0,30,
44,3C,2C,C,30,40,3A,32,28,18,0,2
4,32,0,C,3C,32,24
130 DATA 0,C,3A,32,24,0,18,36,32
,24,0,C,3A,32,24,0,C,3C,32,24,0,
30,3A,28,3A,0,18,36,28,30,0,C,30
,0,0,0,C,2C,0,0,0,30,28,0,0,0,C,
28,0,0,0,C,3A,0,0,0,C,3A,0,0,0,C
,36,0,0,0,30,28,0,0,0,C,0,0,0
140 DATA 0,C,3A,32,28,0,C,3C,32,
28,0,20,40,32,22,0,C,40,32,22,0,
18,32,32,28,0,C,36,32,28,0,C,32,
32,28,0,C,0,32,22,0,C,40,32,22,0
,C,44,32,22,0,C,40,32,22,0,18,32
,32,28,0,C,32,32,28,0,C,3C,2C,24
,0,C,3C,2C,24
150 DATA 0,C,3A,2C,24,0,C,3A,2C,
24,0,C,36,2C,24,0,C,36,2C,24,0,C
,32,2C,24,0,C,32,2C,24,0,30,36,3
0,28,0,C,0,1E,24,0,C,32,1E,24,0,
C,32,1E,24,0,C,32,1E,24,0,30,32,
22,28,0,0,0,0,0,0,0,0
```

■

The Listing:

```
0 GOTO 10
1 '****** STRING SETUP ********
2 '****** PETER FOUCHE ********
3 '********** 7/6/87 **********
4 SAVE "NUTSETUP:0":END
10 'SET UP LETTER DRAW STRINGS'
15 CLEAR2000:DIM A$(122),O$(13),
D$(13,4)
17 OPEN "O",#-1,"NUTDAT"
20 FORT=65 TO 75:READ A$(T):PRIN
T #-1,A$(T):NEXT:FORT=76 TO 87:R
EADA$(T):PRINT #-1,A$(T):NEXT
30 FORT= 88 TO 90:READ A$(T):NEX
T T
40 READ A$(32):PRINT #-1,A$(32)
50 FORT=48 TO 56:READA$(T):PRINT
#-1,A$(T):NEXT T:READA$(57):PRIN
T#-1, A$(57)
70 READ A$(33):PRINT #-1,A$(33)
80 READ A$(40):READ A$(41):PRINT
#-1,A$(40):PRINT #-1,A$(41)
90 READ A$(42):READ A$(44):READ
A$(46):PRINT #-1,A$(42):PRINT#-1
,A$(44):PRINT#-1,A$(46)
100 READ A$(8):PRINT#-1,A$(8)
110 READ A$(45):PRINT#-1,A$(45)
115 FORT=88 TO 90:PRINT#-1,A$(T)
:NEXT
130 DATA"BD2ND4D1R4ND3U1H2G2BR4B
E2"
140 DATA"D6R3E1U1H1NL3E1U1H1NL3B
R3"
150 DATA "BD1D4F1R2E1BU4H1L2BR5"
160 DATA "R1D6NL1R2E1U4H1NL3BR3"
170 DATA "D3NR2D3R4BU6NL4BR2"
180 DATA "D3NR2D3BE4BU2NL4BR2"
190 DATA "BD1D4F1R2U2NL1BU3U1NL2
BR3"
200 DATA "D6U3R4ND3U3BR2"
210 DATA "BR1R1D6NL1NR1U6R1BR3"
220 DATA "BD5F1R1E1U5BR3"
230 DATA "D6U3R1NF3E3BR2"
240 DATA "D6R4BU6BR2"
250 DATA "ND6F2E2ND6BR2"
260 DATA "ND6F4D2U6BR2"
270 DATA "BD1D4F1R2E1U4H1NL2BR3"
280 DATA "ND6D3R3E1U1H1NL3BR3"
290 DATA "BD1D4F1R1E1NH1NF1E1U3H
1NL2BR3"
300 DATA "D6U3R1NF3R2E1U1H1NL3BR
3"
310 DATA "BD5F1R2E1H4E1R2F1BU1BR
2"
320 DATA "R2ND6R2BR2"
330 DATA "D5F1R2E1U5BR2"
340 DATA "D2F1D1F1ND1E1U1E1U2BR2
"
350 DATA "D6E2F2U6BR2"
360 DATA "D1F4ND1H2G2ND1E4U1BR2"
370 DATA "D1F2ND3E2U1BR2"
380 DATA "R4D1G4D1R4BU6BR2
390 DATA "BR6"
400 DATA "BD1D4F1R2E1U4NG4H1L2BR
4"
410 DATA "BD1BR1E1D6NR1NL1BU6BR4
"
420 DATA "BD1E1R2F1D1G1L2G1D2R4B
U6BR2"
430 DATA "BD1E1R2F1D1G1NL1F1D1G1
L2H1BR6BU5"
440 DATA "BR3ND6G3R4BU3BR2"
450 DATA "NR4D2R3F1D2G1L2H1BR6BU
5"
460 DATA "BR4L2G1D4F1R2E1U1H1L3B
R6BU3"
470 DATA "R4D1G3D2BR6BU6"
480 DATA "BR1G1D1F1G1D1F1R2E1U1H
1NL2E1U1H1NL2BR3"
490 DATA "BR1G1D1F1R3D2G1L2R2E1U
4H1NL2BR3"
500 DATA "BR1D3BD2D1BU6BR3"
510 DATA "BR4G2D2F2BR3BU6"
520 DATA "F2D2G2BR4BU6"
530 DATA "BR3D5U3NF2NG2D1NH2E2BR
3BU1"
```

This program comes with a pictured title page on this month's CoCoOz

```
540 DATA "BD4BR2ND1R1D1BD1G1E1U1
BU5BR3"
550 DATA "BD4BR2D1R1U1BR4BU4"
560 DATA"C1L5D1R4D1L4D1R4D1L4D1R
4DL5U6C0"
570 DATA "BD3R4BU3BR"
1000 '*** SET UP OBJECTS ***'
1010 D$="U70L30D70BE25BU20U4GDFE
UHS4":D$(1,1)="BM112,142"+D$
1020 D$(1,2)="BM104,117S2"+D$:'S
2U70L30D70BE25BU20U2ND4BH2ND8G2D
4S4
1030 D$(1,3)="BM100,105;S1U65L30
D65BE25BU20U2ND4BH2ND8G2D4S4"
1040 D$(1,4)="BM96,96ND4L1D3"
1050 D$(2,1)="BM112,142;U70L30D7
0M+11,-8U22BL1U2HGD2FEBR1U33M-11
,-5"
1060 D$(2,2)="BM104,117;S2U70L30
D70M+11,-8U22BL2U2HGD2FEBR2U33M-
11,-5"
1070 D$(2,3)="BM100,105;S1U65L30
D65M+11,-8U19BL2U1BR2U30M-11,-8
1080 D$(2,4)=D$(1,4)
1090 D$(12,1)=D$(1,1):D$(12,2)=D
$(1,2):D$(12,3)=D$(1,3):D$(12,4)
=D$(1,4)
1100 D$="NM+8,-4;D15R10U15L10R4N
M+5,-4R3E2S4":D$(3,1)="BM131,172
;S4"+D$
1110 D$(3,2)="BM108,124;S3"+D$
1120 D$(3,3)="BM99,110;S2"+D$
1130 D$(3,4)="BM98,102;S1"+D$
1140 D$="R12DL6U3NLNRD3NL4FNL3GB
L2UD4L2RU2S4":D$(4,1)="BM116,172
;S4"+D$
1150 D$(4,2)="BM100,124;S3"+D$
1160 D$(4,3)="BM98,110;S2"+D$
1170 D$(4,4)="BM97,102;S1"+D$
1180 D$=";M-11,+9;M+3,+8;M+10,-5
;M+5,-5;GU5L2GLU2L4BF4LGLDL6DLD2
LBR3NM+5,-3DLG3BM+9,+1R3FGNL2NGD
RNG2D2GL2H2U3BE5URU3RUR3S4":D$(5
,1)="BM100,157;S6"+D$
1190 D$(5,2)="BM100,120;S4;"+D$
1200 D$(5,3)="BM98,107 ;S2"+D$
1210 D$(5,4)="BM97,100;S1"+D$
1220 D$="U4ER4FD4GL4BE2ULDS4"
1230 D$(6,1)="BM116,96S4"+D$
1240 D$(6,2)="BM107,96S3"+D$
1250 D$(6,3)="BM102,96S2"+D$
1260 D$(6,4)="BM97,97L"
1270 D$="L7ER7EL7ER7EL7S4"
1280 D$(7,1)="BM60,170"+D$
1290 D$(7,2)="BM75,140E3LG3LE3LG
3LE3"
1300 D$(7,3)="BM88,115ELGLE"
1310 D$(7,4)="BM92,105L
1320 D$="U2E4R7ND2G4ND2L3ND2L4E2
BR3RS4":D$(8,1)="BM75,170"+D$
1330 D$(8,2)="BM84,140S3"+D$
1340 D$(8,3)="BM89,115S2"+D$
1350 D$(8,4)="BM93,105S1"+D$
1360 D$="R80H20L40G20R5E20L5BD2B
LBRD2UBG5G2BG1G7BR30R15E5U5H5L15
G5D5F5S4":D$(9,1)="BM15,191S8"+D
$
1370 D$(9,2)="BM56,143S4"+D$
1380 D$(9,3)="BM75,120S2"+D$
1390 D$(9,4)="BM85,106S1"+D$
1400 D1$="BM123,110;XD2$;F17NM-5
6,+28D34M-56,+28L2NH31RU33H32ND3
4UM+24,-12;NEHL3H2L8G2DF2RE2UHBD
4R9URUR4D9HNU9NH2F18D3GLNF3GNF3G
NF3RNELGLG2LNE8LNE9H19DNF19U2NRE
8DNG8R2NF18HBD3F18G6H12RNF5GNF5U
H5E6G1; F2BF2FBF2FBF2FBF2FDHBH2H
BH2HBH2HBH2HGNLNDBF2NURDB
1410 D$(10,1)="XD1$;NURDBF2NUBF2
NURDBF2DBL2NDLUBH2NDLUBH2NDLUBH2
NDLUBH2NDLUBR12E4FNG4FNG4FNG4F9R
FREU20H21DNF21DNF21LNU2ND21NULD2
0NF8RNF20RF20U24F2;RERER3ER2E2RE
2RE3D8G3LDG2DG2NRDGDLD2G2NRDGDGD
L2DL2G2BM+2,-31;BR18UHH9L2GL3DL8
HL5HL5HL4":D$(13,1)=D$(10,1)
1420 D2$="C1LD32NL70NR18U32RC0"
1430 FORT=1 TO 12:FORY=1 TO 4:IF
 T=10 OR T=9 OR T=5 THEN NEXT T
1440 PRINT #-1, D$(T,Y):NEXT Y:N
EXT T
1445 FORY=1 TO 4:PRINT#-1,D$(5,Y
):NEXT Y
1447 PRINT #-1,D$(9,1):PRINT #-1
,D$(9,2):PRINT #-1,D$(9,3):PRINT
 #-1,D$(9,4)
1450 PRINT #-1,D1$:PRINT #-1,D2$
:PRINT #-1,D$(10,1)
1460 CLOSE
```

■

TAPE UTILITIES

By Bill Holt

32K ECB CoCo 1/2/3
TAPE ONLY UTILITY

FINALLY, I AM IN a position to submit something for the magazine - a couple of linked tape utilities which I feel are most useful.

Together they give the ability for a tape system to emulate much of the convenience of a disk operating system; automatic indexing, auto-run or exec, etc.

In one of your columns you mentioned that some readers were disappointed that CoCoOz tapes purchased from you did not autoexec.

These programs will give that ability, at the expense only of running the tape through the utility. I have included a sample index of CoCoOz #42 for you to try out if you like.

Alex's plea in the June Coco stirred me to get off my backside and do something about it. I didn't know what I was letting myself in for, as I will explain later, however this is my entry to the utilities competition.

The programs just grew, and I was using them in a very raw state, always intending to clean them up for submission sometime, but then I bought a couple of disk drives, and had no incentive.

The second program, "Print Index" came first and had its origin when I typed in the excellent utility, "Tape Directory with Crun" by Harold Nicholls from the March 86 Aust. Rainbow.

I didn't like the idea of continually altering Basic Data statements so soon changed it to add a couple of routines to load and save the data on tape.

It also seemed a waste to simply store program names on tape, so I added counter numbers and extended details for use in a database. As I don't have an independant remote on my recorder, a routine to switch motor and audio on and off also went in.

Purchase of the CocoOz Utilities tape gave me a cassette label print utility ("Labeller" by J.D. Ray) which I initially converted to accept these files, but I was not satisfied with the layout so added a routine based on this to the program. Together with routines to add, delete and modify entries, I felt it was a pretty satisfactory program.

It was now pretty long, so I modularised the program so I could create a run only version by deleting lines 775, 790, and 995 on. This reduced the possibility of a loaded program overwriting sections of the Print Index program and also dramatically reduced the loading time.

All the while, I was looking for a way to reduce the effort of typing in the data. Much of the information I was looking for was directly accessable by the CoCo, if only I could get access to it. I have an excellent commercial utility (Tapeutil by Thomas P. Olk) which obviously could achieve this, but I am a rank amateur at M/L., nevertheless I decided to see if I could find out what it did.

After many hours, I found the routine through Z-Bug, and played around with it a bit. Much of it, however was incomprehensible to my limited understanding, but I fiddled around and with the insertion of several NOP's and a couple of RTS's found that I could access it from Basic, and recover the details I wanted. Next, Joy of joys, it appeared to be re-locatable. I had my first usable M/L routine (even if I didn't really understand it).

I then wrote a bare bones Basic program to access the routine, and to my surprise, it did all that I wanted. I then added the necessary routines to add the details required by "Prntindx", place a marker at the end of the tape, and save the file.

I then located the M/L routine at the top of the basic listing to allow maximum space for indexed programs, resulting in listing 1- "Autoindx".

On deciding to submit the programs I loaded them to disk in order to clean them up and add a few bells and whistles and remarks.

"Prntindx" was fairly straightforward, but "Autoindx" required me to shift the M/L location. On I went, and everything was working fine, until I loaded it back to tape and tried it out. The program ran well until I went to enter the tape details, when I had a fatal crash with a syntax error in some impossible line number.

Most of line 1600 was wiped out. I tried everything I could think of to try to overcome the problem, relocating the M/L, clearing more memory, but nothing changed. I went back and deleted most of my improvements, now it crashed in Line 5050.

Desperate, I reloaded my old version and ran it - no problems. I then printed a listing to compare with the current version, and found that one data item at the end of line 7010 was missing. Then, after many hours, the penny dropped, "my" M/L program must be writing something to a location in the basic area. Hence the peculiar entry in line 5000, I decided that it was best to let it write where it wouldn't do any damage.

Not very elegant but it seems to work.

Accordingly, I must stress the need to type this listing EXACTLY as listed. For those of you with Rainbow "Check Plus" I have included a table of values to assist you.

Therefore, this program is not quite as finished as I had intended, but I am most reticent to alter it again and am afraid it will have to suffice. If users wish to amend the program further, they are welcome, but at their own risk, as I don't know what other surprises may be in store. Someone more expert in asembler may be able to develop a more elegant solution.

I am still very much a beginner and everything in this submission incorporates something I have learnt from reading them. It includes many experiments, such as different approaches to Menus and my first "Pop-up" Menus, and I have learnt a lot in developing them.

Check Plus Values

AUTOINDX		PRNTINDX	
440	245	510	157
990	14	990	240
2500	42	1800	251
5100	168	2550	24
7190	221	10050	201

ML Checksums

AUTOINDX	PRNTINDX
35251	1570

Instructions

Listing One - "AUTOINDEX"

Automatic Tape Index

Operation of this program is fairly straightforward and is Menu driven. CLOAD and RUN the program. After a short delay, the main menu appears, then:

1. Select option 4 - "Mark Tape End".

2. Advance tape past last program and set recorder to "Record".

3. Press any key. The program will write a short file called "End".

4. Select option 1 - "Load Index".

5. You will be asked for a Tape Number, Tape Name (preferably less than 16 characters), and Side

6. On the message "Loading Tape Index"- rewind tape to start and set recorder to "Play".

7. The program will run through the tape and display program details. If you do not have a record of counter numbers take a note now, otherwise you can just let the tape run through.

Note: Occasionally, a long program will display an "Error" at the end address - this can be ignored as it will not affect the operation of the program.

8. On return to Menu, select Option 3. "Insert Counter Numbers".

9. The name of each program will be displayed and you will be asked to enter the counter number (up to three characters).

'For those users with 'Check Plus' I have included a table of values to assist you.'

10. You will then be asked for a single number entry for "Category". A Pop-up Menu is provided as an aide-memoir. (Line 1600). Press "Space" to repeat previous entry. After confirmation the next item will be displayed.

11. On return to main menu select Option 2. "Save Index".

12. Position tape and record index as per instructions.

13. Repeat procedure for next tape or select option 5.

Warning: Due to bug in M/L Routine you cannot re-run the program and will get a U.L. error in line 20.

Listing 2 - "PRINTINDEX"

Autoexecute Directory, Modify, or Print Cassette Label

Similarly, this program is Menu driven, and reasonably self explanatory.

1. On running the program you will be asked if you wish to load an index from tape. If you answer "Y" you will be asked to input the tape number and side.

Place target tape in recorder and load index.

2. If there are more than 26 items loaded, the program will accept up to 30 and display those that cannot be displayed on the screen. This is to allow you to delete any items that cannot be executed (such as data files) if you so desire, otherwise the directory screen will be displayed.

This displays the program names and counter numbers beside an alpha key. You now have the option to run a program or move into the utility section of the program.

3. To "run" a Basic program or "execute" an M/L program - press Space to turn on motor and audio - advance tape to counter number and press appropriate alpha key. The program will then load and (usually) run.

Note: Occasionally a program will overwrite this program and you will get either a syntax error, or the program will hang up. Placement of the execute routines in lines 2 and 3 should minimise this occurrence, but if it should happen, <reset> and "Run" or "Exec" will work in most cases.

4. To exit the directory screen press <SHIFT><CLEAR>. The utility menu will then be displayed.

5. Press "R" to return to directory screen. You will be given the option of returning to the previous index or loading a new index.

6. Press "E" to end program. These are the only options given in the execute only version.

7. Press "A" to Amend directory. You will be guided by a menu to "Add", "Delete" or "Change" any records, you can also use "Change" to view all the file details of a program. The directory screen is used in these sub-routines. Select items as above.

8. Press "S" to save modified directory to tape.

9. Press "P" to print a Cassette label. Only the first 24 items will be listed, but you can use "A" and "D" to customise your listing before selecting this item if you so desire.

You are given the option of printing an alignment pattern to assist in lining up your labels.

Note: Printer Codes are for the R/S DMP 105 and are annaotated in lines 2000-2400. Modify these lines to suit your printer.

The Listing:

```
1 '***********AUTOINDX************
**************W.J.HOLT************
*************7 JULY 87***********
*
2 GOTO5100
3 CLS:PRINT"READY TAPE FOR SAVE"
:MOTORON:AUDIOON:PRINT"PRESS ANY
 KEY WHEN READY":EXEC44539:MOTOR
OFF:AUDIOOFF
4 INPUT"VERSION NUMBER";VN$:VN$=
"AUTINDX"+VN$:CSAVE VN$
5 END
6 SAVE"299:3":END'9
10 ' ** SET UP ROUTINE **
20 CLS:GOSUB5050:PRINT@170,;"by
  w.j. holt";:PRINT@263,"LOADING
 M/L PROGRAM";
30 POKE65495,0'**H/S**
35 IX=0
40 FORX=&H2791 TO &H291C
50 READ ZZ$:IX=IX+VAL(ZZ$):POKE
X,VAL(ZZ$):NEXT X
60 POKE65494,0'**N/S**
65 IF IX<>35251THENSOUND200,10:C
LS:PRINT:PRINT" ERROR IN LINES 7
000-7190:END
70 CLEAR1000,&H2789
80 DIM MO$(30),T$(30),CO$(30),CA
(30)
90 ' ** INITIAL SCREEN **
100 GOSUB5050
110 PRINT@96,"<1>   load index":
PRINT@160,"<2>   save index"
120 PRINT@224,"<3>   insert coun
ter numbers":PRINT@288,"<4>   ma
rk end of tape":PRINT@352,"<5>
 end program"
140 GOSUB 5030:K=VAL(K$):IF K<1
OR K>5 THEN 100 ELSE ON K GOTO40
0,1000,1500,2500,2000
390 '  **  ENTER TAPE DATA  **
400 GOSUB5050:PRINT@97,"";:LINEI
NPUT"TAPE NUMBER?";TZ$:LINEINPUT
"TAPE NAME?";NN$
410 INPUT"SIDE <A/B>";SI$:IF SI$
<>"A" AND SI$<>"B"THEN410
420 TN$="INDEX"+TZ$+SI$
440 I=26
490 ' ** LOAD TAPE ROUTINE **
500 GOSUB5050:PRINT@97,"loading
tape index   "TN$:FORZ=1TO I:T$(
Z)="          ":NEXTZ
510 I=1
520 EXEC&H2791
530 MOTOROFF
540 A=PEEK(&H2916)
550 IFA=0THEN MO$(I)="B"ELSE IFA
=1 THEN MO$(I)="D"ELSE MO$(I)="M
"
560 T$(I)=CHR$(PEEK(&H290E))
570 FOR X= &H290F TO &H2915
590 T$(I)=T$(I) + CHR$(PEEK(X))
600 NEXT X
610 IFT$(I)="END     "THEN880
810 I=I+1:IFI<30THEN520
820 SOUND 160,5:CLS:PRINT:PRINT"
RECORD CAPACITY EXCEEDED":PRINT"
<ANY KEY> TO RETURN TO MENU":GOS
UB5030
880 GOTO100
890 ' ** SAVE INDEX FILE **
970 PRINT@449,"TO CONTINUE ";G$;
"[SP/BAR]"
980 Q$=INKEY$:IFQ$=""THEN980 ELS
EIFQ$<>" "THEN 980
990 RETURN
1000 GOSUB5050:PRINT@65,"*make";
CHR$(128);"newfile*",NN$;" side
";SI$:PRINTTN$
1010 PRINT"PLACE DATA TAPE IN RE
CORDER NOW":PRINT:PRINT"setup";C
HR$(128);"tape NOW":MOTORON:AUDI
OON:GOSUB970:PRINT" NOW PRESS <r
ecord> KEY!"CHR$(13):GOSUB970:AU
DIOOFF
1030 PRINT"csave(ing).";TN$:FORD
L=1 TO100:NEXT
1050 OPEN"O",#-1,TN$
1055 PRINT#-1,TZ$,SI$,NN$
1060 FORZ=1TOI
1080 PRINT #-1,CO$(Z),T$(Z),MO$(
Z),CA(Z)
1090 PRINTZ"  "CO$(Z)"  "T$(Z)"
 "MO$(Z)"  "CA(Z)"  "TS$
1100 NEXT
1110 CLOSE #-1
1120 Z=0:GOTO100
1490 '  ** INSERT COUNTER NUMBER
S
1500 FORZ=1TOI
1510 GOSUB5050:PRINT@64,"   inse
rt counter numbers"
1520 PRINT@96,Z;T$(Z):PRINT@110,
"counter";CHR$(128);:LINEINPUT"n
umber? ";CO$(Z)
1530 IF LEN(CO$(Z))<3 THEN CO$(Z
)="0"+CO$(Z):GOTO1530
1535 PRINT@125,CO$(Z)
1600 PRINT@288,"<1> application"
,"<6> game","<2> business","<7>
adventure","<3> education","<8>
simulation","<4> utility","<9> m
usic","<5> telewriter","<0> misc
","<SPACE-BAR>to"CHR$(128)"repea
t"CHR$(128)"entry
1700 PRINT@173,"category? ";:GOS
UB5030:IFK$=" "THENCA(Z)=CA(Z-1)
ELSEK=VAL(K$):CA(Z)=K
1705 PRINTCA(Z)
1710 PRINT"IS ENTRY CORRECT? ":G
OSUB5030:IFK$<>"Y"THEN1510
1970 NEXT
1980 GOTO100
1990 '** FINISH **
2000 END

2490 ' ** TAPE END MARKER **
2500 GOSUB5050:PRINT@128,"  mark
 end of tape":PRINT:PRINT"positi
on tape past last program":MOTOR
ON:AUDIOON:PRINT:PRINT"<any key>
 WHEN READY <record>":GOSUB5030:
AUDIOOFF:OPEN"O",#-1,"END":PRINT
"WRITING MARKER":CLOSE#-1:GOTO10
0
4980 '  *** SUB- ROUTINES  ***
5030 K$=INKEY$:IFK$=""THEN5030EL
SE RETURN
5040 ' ** PAGE HEADER ** THIS LI
NE IS EXTENDED TO DEFEAT M/L BUG
**
5050 CLS(RND(8)):PRINT"        aut
omatic tape index":RETURN
5090 ' ** ROM BUG FIX **
5100 PCLEAR1:GOTO20
6990 ' ** M/L DATA **
7000 DATA127,23,5,127,255,64,13,
37,111,141,1,112,48,141,1,109,15
9,126,189,167
7010 DATA1,39,10,23,1,49,13,124,
38,244,22,0,191,23,1,39,13,124,3
8,234
7020 DATA189,167,233,159,126,48,
141,1,76,198,8,189,185,162,109,1
41,1,75,38,22
7030 DATA142,0,0,175,141,1,71,17
5,141,1,65,23,0,135,32,66,65,83,
32,37
7040 DATA32,8,141,125,32,77,76,3
2,32,37,109,141,1,40,43,9,141,11
1,66,73
7050 DATA78,32,37,32,18,142,0,0,
175,141,1,26,175,141,1,20,141,91
,65,83
7060 DATA67,32,37,236,141,1,11,2
3,0,169,236,141,1,2,23,0,162,189
,167,124
7070 DATA189,167,11,16,38,0,138,
159,126,13,124,43,12,23,0,175,16
6,141,0,230
7080 DATA76,38,233,32,228,189,16
7,233,48,31,31,16,48,141,0,220,5
2,16,163,225
7090 DATA227,141,0,210,237,141,0
,186,141,109,23,0,138,13,111,39,
3,189,185,88
7100 DATA109,141,0,175,57,18,18,
18,57,53,16,166,128,129,37,39,5,
189,162,130
7110 DATA32,245,110,132,108,141,
0,152,109,141,0,147,39,3,126,185
,88,141,226,69
7120 DATA82,82,79,82,32,73,78,32
,72,69,65,68,69,82,46,32,83,75,7
3,80
7130 DATA80,73,78,71,32,70,73,76
,69,13,37,141,58,189,167,1,38,24
9,13,124
7140 DATA38,245,22,255,11,108,14
1,0,91,141,174,69,82,82,79,82,37
,32,150,141
7150 DATA7,31,152,141,3,126,185,
172,52,2,68,68,68,68,141,4,53,2,
132,15
7160 DATA139,48,129,57,35,2,139,
7,126,162,130,127,255,2,182,255,
0,72,39,29
7170 DATA189,161,193,129,3,38,7,
50,98,108,141,0,27,57,129,32,38,
11,189,167
7180 DATA233,189,161,177,189,167
,202,32,230,57,255,255,0,2,68,15
4,166,128,47,0
7190 DATA5,70,79,82,77,65,84,77,
32,2,0,255,84,9,68,69
```

■

The Listing:

```
1 '**********PRNTINDX***********
*************V.J.HOLT***********
*************1 JULY 87**********
2 GOTO610
3 '**M/L LOAD**
4 CLOADM T$(PN):EXEC:END
5 '**BASIC LOAD**
6 EXEC 32666 T$(PN):END
9 '**TAPE SAVE**
10 CLS:PRINT"READY TAPE FOR SAVE
":MOTORON:AUDIOON:PRINT"PRESS AN
Y KEY WHEN READY":EXEC44539:MOTO
ROFF:AUDIOOFF
11 INPUT"VERSION NUMBER";VN$:VN$
="RUNINDX"+VN$:CSAVE VN$
12 END
19 '**DISK SAVE**
20 N$="RUNINDX":KILL N$+"/BAK":R
ENAME N$+"/BAS" TO N$+"/BAK"
21 N$="RUNINDX":VERIFYON:SAVE N$
: PRINT N$"  SAVED":PRINT:DIR:PR
INTFREE(0);" GRANS FREE":END
22 END
23 SAVE"299B:3":END'9
100 '**INSTALL CRUN**
105 POKE65495,0'**HIGH SPEED POK
E**
110 DATA 0F,78,32,62,BD,A5,C5,BD
,A6,48,7D,01,E4,26,05,B6,01,E2,2
7,03,7E,A6,16,BD,AD,19,BD,A7,7C,
9E,19,9F,7E,DC,7E,4C,BD,AC,37,BD
,A7,0B,26,34,96,7C,27,30,2A,ED
120 DATA 9F,1B,BD,A7,E9,8E,AB,EC
,BD,B9,9C,BD,AD,21,BD,AC,EF,BD,0
1,82,8E,02,DD,86,52,A7,80,86,55,
A7,80,86,4E,A7,80,6F,84,C6,04,8E
,02,DC,4F,7E,AC,7F
130 DATA BD,AD,19,7E,A6,19
140 FOR P=A TO A+101
150 READ D$:POKE P,VAL("&H"+D$)
160 NEXT P
200 '**INITIALIZE DIRECTORY TABL
E**
220 CA$(0)="MISC.   ":CA$(1)="AP
PLIC'N":CA$(2)="BUSINESS":CA$(3)
="EDUCAT'N":CA$(4)="UTILITY":CA$
(5)="TELEWRTR":CA$(6)="GAME    "
:CA$(7)="ADVNTURE":CA$(8)="SIMUL
'TN":CA$(9)="MUSIC   "
225 POKE65494,0'**NORMAL SPEED**
230 CLS:PRINT@9,"TAPE INDEX":PRI
NT:PRINT"LOAD INDEX FROM TAPE? <
y/n?>":GOSUB710:IFD$="N"THEN305
ELSE IFD$<>"Y"THEN230
235 FORZ=1 TO 26:T$(Z)="
":CO$(Z)="    ":MO$(Z)=" ":CA(Z)=
0:TZ$(Z)=".":NEXTZ:Z=0'**CLEAR V
ARIABLE TABLE**
240 PRINT:LINEINPUT"TAPE NUMBER?
 (1-99) <enter>";TZ$:PRINT"SIDE
A/B?":GOSUB710:TS$=D$:TN$="INDEX
"+TZ$+TS$
245 POKE65494,0'**NORMAL SPEED**
250 CLS:PRINT:PRINT" READY TAPE
<play>":MOTORON:AUDIOON:PRINT:PR
INT:PRINT:PRINT" < any key >WHEN
 READY":EXEC44539:AUDIOOFF:GOSUB
730:PRINT:PRINT"file loading: ";
TN$;"    side ";TS$:GOSUB810:GOS
UB730:GOSUB750:PRINT:PRINT" ";TN
$;" LOADED":GOSUB720
300 '**DISPLAY TAPE DIRECTORY**
305 POKE65495,0'**HIGH SPEED POK
E**
310 GOSUB730:GOSUB750
320 FORI=1 TO 13
330 PRINTCHR$(64+I);"- ";T$(I);"
 ";CO$(I),CHR$(77+I);"- ";T$(I+1
3);" ";CO$(I+13)
340 NEXTI
344 POKE65494,0'**NORMAL SPEED**
345 IFDR THENRETURN
350 MM$="ON ":GOSUB760
360 PRINT"   USE (shift)(clear)
TO EXIT";
370 SCREEN 0,1
```

Note: Printer Codes are for the RS/ DMP 105 and are annotated in lines 2000–2400. Modify these lines to suit your printer.

```
400 '**PROGRAM SELECTION**
410 GOSUB710:IF D$=" " THEN MOTO
RON:AUDIOON:MM$="OFF":GOSUB760:G
OSUB710:IF D$=" "THEN MOTOROFF:M
M$="ON ":GOSUB760:GOTO410
420 AUDIOOFF:PN=ASC(D$)-64
430 IF ASC(D$)=92 THEN DR=0:GOTO
770
440 IF ASC(D$)<65 OR ASC(D$)>90
THEN GOTO 410
450 IF T$(PN)="          " THEN SO
UND 1,3:GOTO 410
455 IFDR THENRETURN
460 IF MO$(PN)<>"BAS"THEN480
470 CLS:SCREEN 0,0:PRINT:PRINT:P
RINT"     LOADING basic PROGRAM"
:PRINT@237,T$(PN):GOTO 6
480 IF MO$(PN)="DAT"THEN SOUND4,
4:CLS:PRINT:PRINT:PRINT"  data t
ape -- CANNOT EXECUTE-":FOR TT=1
TO500:NEXT:CLS:GOTO300
490 CLS:PRINT:PRINT:PRINT"LOADIN
G machine language PROGRAM":PRIN
T:PRINT@237,T$(PN):PRINT:PRINT"
IF ERROR OCCURS AFTER LOADING":P
RINT:PRINT@331,"TYPE  <exec>":GO
TO4
500 GOTO4
510 END
600 '** INITIALISE PROGRAM **
610 PCLEAR1:CLEAR800,32666:A=326
66
620 CLS:PRINT@9,"TAPE DIRECTORY"
:PRINT@77,"loading":PRINT@322,"w
arning- BASIC PROGRAMS WILL    B
E LOADED AND RUN IN pclear1    *
** PROGRAMS must INCLUDE***    *
** CORRECT pclear INSTRN***
625 PRINT@192,"**********BY BILL
 HOLT************************198
7**************";
630 DL=1000:GOSUB720
640 DIMCO$(50),T$(50),MO$(50),CA
(50),TZ$(50):I=1:GOTO105
700 '*GENERAL SUB ROUTINES*
710 D$=INKEY$:IFD$=""THEN710ELSE
RETURN
720 FOR TT=1TO DL:NEXT:RETURN
730 CLS
740 PRINT@9,"tape directory":RET
URN
750 PRINT@0,NN$:GOSUB740:PRINT@2
5,"SIDE ";TS$,:RETURN
760 PRINT@450," spacebar MOTOR "
MM$:SCREEN 0,1:RETURN
770 CLS:PRINT:PRINT"   RETURN TO
 DIRTRY <r>"
780 PRINT:PRINT"   END PROGRAM
    <e>"
785 SCREEN 0,1:GOSUB710:IFD$="R"
THEN230
795 IFD$="E"THEN PCLEAR4:CLEAR20
0:CLS0:PRINT@239,"bye";:PRINT@0,
"";:END ELSE GOTO770
800 ' *LOAD INDEX SUB-ROUTINE*
810 OPEN"I",#-1,TN$:I=1
815 INPUT#-1,TZ$,SI$,NN$
820 IF EOF(-1) THEN895
830 INPUT#-1,CO$(I),T$(I),MO$(I)
,CA(I)
840 IF LEN(T$(I))<8THEN T$(I)=T$
(I)+" ":GOTO840
845 IFT$(I)="          "THENCO$(I)
="":GOTO820
860 IFMO$(I)="B"THEN MO$(I)="BAS
"ELSEIFMO$(I)="M"THEN MO$(I)="M/
L"ELSEIFMO$(I)="D"THEN MO$(I)="D
AT"
865 IFI>26THENPRINTT$(I)"   "MO$
(I),CO$(I)"   "CA$(CA(I))
870 I=I+1
880 IFI=27THENPRINT"MORE THAN 26
 ITEMS-PASSING REST":FORTT=1 TO5
00:NEXT
890 GOTO820
895 IF I>25THENGOSUB970
900 Z=I-1:I=1
910 CLOSE #-1
920 RETURN
930 END
970 PRINT@449,"TO CONT:NUE ";G$;
"[SP/BAR]"'**CONTROL ROUTINE**
980 Q$=INKEY$:IFQ$=""THEN980 ELS
EIFQ$<>" "THEN 980
990 RETURN
```

■

CoCo 3 Program

FASTBACKUP

by Gordon Thurston

UTILITY

AFTER BACKING UP many disks with a single drive, I decided there must be a better way. When the CoCo 3 arrived with 128k, it seemed a shame to have to spend so much time switching disks.

The following program will back up a disk in just 2 goes. It is prompted, and beeps just like the routine in disk Basic.

I have included both a Basic loader, and a commented source code for those who want to understand and possibly modify it for their own needs.

If the CoCo has 512k, it would be simple to make it back up in one fast go with a few slight mods.

Type in the basic program, and save it to disk before running it. Using a fresh disk, or one you can afford to loose, run it, and if all goes well, it should make a machine language program, and save it to disk.

DSKINI a disk to back up to, and insert a source disk.

Execute the program and follow the prompts. If all goes well, you should have a duplicate disk. If not, reload the original basic program, find the mistake, and re-save it.

After backing up a disk, it instructs to press enter to back up another disk. Any other entry will do a cold start.

The Listing:

```
0 GOTO10
1 '********* FASTBACK **********
2 '******* GORDON THURSTON *****
3 SAVE"294:3":END'8
10 FORA=&HE01 TO &H1001:READP:PO
KEA,P:NEXT:SAVEM"FASTBACK/BIN",&
HE01,&H1001,&HE01
20 DATA 183,255,222,16,206,17,0,
142
30 DATA 255,34,166,1,132,251,167
,1
40 DATA 230,132,202,2,231,132,13
8,4
50 DATA 132,247,167,1,23,1,47,23
60 DATA 0,56,23,0,80,23,1,55
70 DATA 23,0,47,23,0,75,23,1
80 DATA 29,23,0,41,23,0,62,23
90 DATA 1,37,23,0,32,23,0,57
100 DATA 23,1,167,48,141,1,132,2
3
110 DATA 1,112,173,159,160,0,39,
250
120 DATA 129,13,39,172,15,113,11
0,159
130 DATA 255,254,79,32,2,134,18,
190
140 DATA 192,6,167,2,111,1,134,1
150 DATA 167,3,204,96,0,237,4,13
4
160 DATA 48,183,255,163,57,134,2
,32
170 DATA 2,134,3,167,0,173,159,1
92
180 DATA 4,166,6,38,69,166,3,76
190 DATA 167,3,129,19,38,17,134,
1
200 DATA 167,3,166,2,76,129,18,3
9
210 DATA 48,129,35,39,44,167,2,3
2
220 DATA 2,167,3,16,174,4,49,169
230 DATA 1,0,16,175,4,16,140,128
240 DATA 0,38,20,16,142,96,0,16
250 DATA 175,4,182,255,163,76,12
9,56
260 DATA 38,2,134,57,183,255,163
,32
270 DATA 180,57,68,68,198,255,68
,92
280 DATA 36,252,88,48,141,0,14,1
74
290 DATA 133,23,0,222,173,159,16
0,0
300 DATA 39,250,22,255,27,14,242
,14
310 DATA 252,15,6,15,37,15,49,15
320 DATA 63,76,79,83,84,32,68,65
330 DATA 84,65,141,67,82,67,32,6
9
340 DATA 82,82,79,82,141,83,69,6
9
350 DATA 75,32,69,82,82,79,82,32
360 DATA 79,82,32,82,69,67,79,82
370 DATA 68,32,78,79,84,32,70,79
380 DATA 85,78,68,141,87,82,73,8
4
390 DATA 69,32,70,65,85,76,84,14
1
400 DATA 87,82,73,84,69,32,80,82
410 DATA 79,84,69,67,84,141,68,8
2
420 DATA 73,86,69,32,78,79,84,32
430 DATA 82,69,65,68,89,141,23,0
440 DATA 153,48,141,0,26,23,0,98
450 DATA 173,159,160,0,39,250,57
,23
460 DATA 0,136,48,141,0,44,141,8
2
470 DATA 173,159,160,0,39,250,57
,73
480 DATA 78,83,69,82,84,32,83,79
490 DATA 85,82,67,69,32,68,73,83
500 DATA 75,32,65,78,68,32,80,82
510 DATA 69,83,83,32,69,78,84,69
520 DATA 82,141,73,78,83,69,82,8
4
530 DATA 32,68,69,83,84,73,78,65
540 DATA 84,73,79,78,32,68,73,83
550 DATA 75,32,65,78,68,32,80,82
560 DATA 69,83,83,32,69,78,84,69
570 DATA 82,141,166,128,43,6,173
,159
580 DATA 160,2,32,246,128,128,17
3,159
590 DATA 160,2,57,80,82,69,83,83
600 DATA 32,69,78,84,69,82,32,70
610 DATA 79,82,32,65,78,79,84,72
620 DATA 69,82,32,66,65,67,75,85
630 DATA 80,141,198,2,134,200,24
6,255
640 DATA 34,200,2,247,255,34,142
,0
650 DATA 200,48,31,38,252,74,38,
238
660 DATA 57,175,238,170,170,170,
234,171
```

```
 00100 *************************
 00110 ******* FASTBAC **********
 00120 *************************
 00130 *** BY G. THURSTON *******
 00140 *** 22/5/87 *************
 00150 ORG $E00 PROGRAM STARTS HERE
 00160 DCOPC EQU 0  OFFSETS FOR DSKCON
 00170 DCDRV EQU 1
 00180 DCTRK EQU 2
 00190 DCSEC EQU 3
 00200 DCBPT EQU 4
 00210 DCSTA EQU 6
 00220 LSTRK RMB 1           LAST TRACK POINTER
 00230 START STA $FFDE       CHANGE TO 32K FORMAT
 00240  LDS #$1100           STORE STACK SAFELY
 00250  LDX #$FF22           POINT TO PIA1
 00260  LDA 1,X              GET CONTROL REG
 00270  ANDA #$FB            MASK OFF BIT 2
 00280  STA 1,X              ENABLE DATA DIRECTION REG
 00290  LDB ,X               GET DATA DIRECTION STATUS
 00300  ORB #2               FORCE BIT 1 HIGH
 00310  STB ,X                STORE IN DATA DIRECTION REG NOW BIT
1 IS
                              OUTPUT
 00320  ORA #4               FORCE BIT 2 HIGH
 00325  ANDA #$FF-8          TURN OFF 6 BIT SOUND
 00330  STA 1,X              ENABLE PERIPHERAL REGISTER
 00340  LBSR SOURCE          PRINT MESSAGE AND WAIT
 00350  LBSR INIT1           INITIALIZE VARIABLES
 00360  LBSR READGO          READ IN HALF A DISK
 00370  LBSR DEST            PRINT MESS AND WAIT
 00380  LBSR INIT1           INITIALIZE VARIABLES
 00390  LBSR WRITE           PUT ON SECOND DISK
 00400  LBSR SOURCE          PRINT MESS
 00410  LBSR INIT2           INITIALIZE FOR SECOND HALF
 00420  LBSR READ            GO READ IN HALF DISK
 00430  LBSR DEST            PRINT MESS
 00440  LBSR INIT2           INIT SECOND HALF
 00450  LBSR WRITE           GO WRITE IT
 00455  LBSR BEEP
 00460  LEAX AGAIN,PCR       GET ADR OF MESS
 00470  LBSR PRINT           GO PRINT IT
 00480  WAIT JSR [$A000]     WAIT FOR KEY ENTRY
 00490  BEQ WAIT
 00500  CMPA #13             ENTER KEY?
 00510  BEQ START            DO AGAIN IF ENTER PUSHED
 00520 OUT CLR $71           CLEAR WARM START FLAG
 00530  JMP [$FFFE]          DO COLD START
 00540 INIT1 CLR A           START AT TRACK 0
 00550  BRA INIT
 00560 INIT2 LDA #18         SECOND HALF OF DISK
 00570 INIT LDX $C006        POINT TO DSKCON VARIABLE AREA
 00580  STA DCTRK ,X          INITIALIZE TRACK #
 00590  CLR DCDRV,X          SELECT DRIVE 0
 00600  LDA #1
 00610  STA DCSEC,X          INIT SECTORS
```

```
00620  LDD #$6000          START OF MEMORY BLOCK
00630  STD DCBPT,X         START BUFFER AT BEGINNING
00640  LDA #$30            BOTTOM OF MEMORY
00650  STA $FFA3           GET FIRST BANK OF MEM
00660  RTS
00670 READ LDA #2          DSKCON OPCODE FOR READ
00680  BRA MOVEIT
00690 WRITE LDA #3         DSKCON OPCODE FOR WRITE
00700 MOVEIT STA DCOPC,X   PASS OPCODE ON
00710 MOVE1 JSR [$C004]    CALL DSKCON
00720  LDA DCSTA,X         TEST STATUS
00730  BNE ERROR           BRANCH IF ERROR
00740  LDA DCSEC,X
00750  INC A               NEXT SECTOR
00760  STA DCSEC,X         CHANGE IT
00770  CMPA #19            OUT OF RANGE?
00780  BNE MOVE2           BRANCH IF OUT OF RANGE
00790  LDA #11             ST SECTOR
00800  STA DCSEC,X         CHANGE IT
00810  LDA DCTRK,X         GET TRACK
00820  INC A               INCREMENT IT
00830  CMPA #18            END OF FIRST RUN?
00840  BEQ BACK            RETURN IF SO
00850  CMPA #35            END OF SECOND RUN?
00860  BEQ BACK
00870  STA DCTRK,X         CHANGE IT
00880 MOVE2 BRA MOVE3
00890  STA DCSEC,X         CHANGE IT
00900 MOVE3 LDY DCBPT,X    GET BUFFER POINTER
00910  LEAY $100,Y         ADD 256 TO IT
00920  STY DCBPT,X         CHANGE IT
00930  CMPY #$8000         END OF BANK OF MEM?
00940  BNE MOVE4           BRANCH IF NOT
00950  LDY #$6000          START OF NEXT BANK
00960  STY DCBPT,X         CHANGE IT
00970  LDA $FFA3           GET LAST BANK
00980  INC A               INC IT
00990  CMPA #$38           END OF RUN?
01000  BNE MOVE5           BRANCH IF NOT
01010  LDA #$39            START OF NEXT RUN
01020 MOVE5 STA $FFA3      CHANGE IT
01030 MOVE4 BRA MOVE1      GO DO SOME MORE
01040  BACK RTS
01050 ERROR LSRA           ERROR ROUTINE
01060  LSRA                DISCARD FIRST TWO BITS
01070  LDB #$FF            ADJUST FOR FIRST INC
01080 ERR1 LSRA            SHIFT A BIT
01090  INC B               COUNT IT
01100  BCC ERR1            BRANCH IF NOT FOUND
01110  LSLB                MULT BY 2
01120  LEAX MESTBL,PCR     POINT TO TABLE OF ADRESSES
01130  LDX B,X             GET ADR OF MESS
01140  LBSR PRINT          GO PRINT IT
01150 ERRW JSR [$A000]     INKEY$
01160  BEQ ERRW            WAIT FOR KEYPRESS
01170  LBRA START          TRY AGAIN
```

```
01180 MESTBL FDB LOST       MESSAGE ADDRESS LIST
01190  FDB CRC
01200  FDB SEEK
01210  FDB WRIT
01220  FDB PROTEC
01230  FDB DRIV
01240 LOST FCC /LOST DATA/   MESSAGES
01250  FCB $8D  LINE FEED + $80      MARK END OF MESS
01260 CRC FCC /CRC ERROR/
01270  FCB $8D
01280 SEEK FCC /SEEK ERROR OR RECORD NOT FOUND/
01290  FCB $8D
01300 WRIT FCC /WRITE FAULT/
01310  FCB $8D
01320 PROTEC FCC /WRITE PROTECT/
01330  FCB $8D
01340 DRIV FCC /DRIVE NOT READY/
01350  FCB $8D
01360 SOURCEL BSR BEEP
01370  LEAX SORMES,PCR         GET ADRESS OF MESS
01380  LBSR PRINT             GO PRINT IT
01390 SOR2 JSR [$A000]        WAIT FOR KEYPRESS
01400  BEQ SOR2
01410  RTS
01420 DEST LBSR BEEP
01430  LEAX DESMES,PCR        GET ADR OF MESS
01440  BSR PRINT              PRINTIT
01450 DEST2 JSR [$A000]       WAIT FOR KEYPRESS
01460  BEQ DEST2
01470  RTS
01480 SORMES FCC /INSERT SOURCE DISK AND PRESS ENTER/
01490  FCB $8D
01500 DESMES FCC /INSERT DESTINATION DISK AND PRESS ENTER/
01510  FCB $8D
01520 PRINT LDA,X+             GET A BYTE OF MESSAGE
01530  BMI LAST                BRANCH IF LAST BYTE
01540  JSR [$A002]             GO PRINT TO SCREEN
01550  BRA PRINT               GO DO MORE
01560 LAST SUBA #$80           MAKE NORMAL
01570  JSR [$A002]             GO PRINT IT
01580  RTS RETURN
01590 AGAIN FCC /PRESS ENTER FOR ANOTHER BACKUP/
01600  FCB $8D
01610 BEEP LDB #2
01620  LDA #200
01630 BEEP2 LDB $FF22
01632  EOR B#2
01635  STB $FF22
01640  LDX #200
01650 BEEP3 LEAX-1,X
01660  BNE BEEP3
01670  DECA
01680  BNE BEEP2
01690  RTS
01700 END START
```

■

Tandy ELECTRONICS

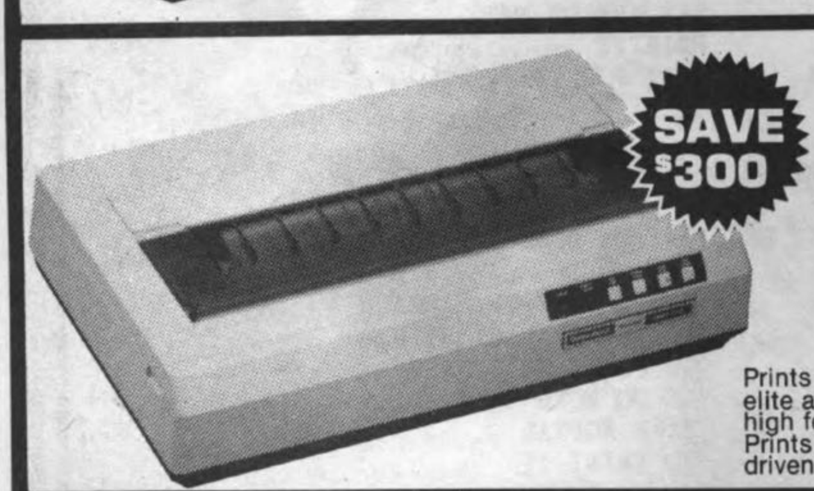
LAY-BY NOW FOR CHRISTMAS!

TRP100 PRINTER

$199

Reg **499.00**

TRP 100. Thermal Ribbon Printer. Prints text (50 cps) and graphics on both regular and thermal paper. Parallel and Color Computer compatible serial interfaces. Uses AC adapter or 5 "C" batteries (not included). Save an incredible $300 on this fast, efficient printer which will save your business time and money. Tandy has printers for every need. 26-1275



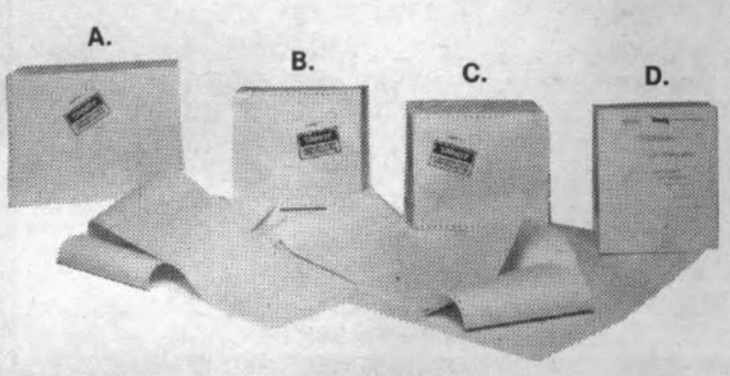
DMP-430 Printer

$999

Reg **1299.00**

Prints 10, 12 or 16.7 cpi, plus elongated standard, elite and condensed. Get micro, italic and double-high fonts, too. Also produces bit-image graphics. Prints up to 180 characters per second. True pin-driven tractor. IBM® compatible. 26-1277

Fan Fold and Wordprocessing Paper



from 19^{95}

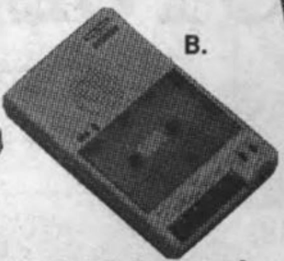
A. Fanfold Paper. Size 241 x 279mm, design blue H/S. Comes in a quantity of quantity 1000 sheets so you'll have a big supply. 26-9317...**39.95**

B. Fanfold Paper. Size 381 x 279. Design blue HS, quantity 1000 sheets. 26-9315...**39.95**

C. Wordprocessing Paper. Size 241 x 279mm. Design blank white, 1000 sheets. 26-9316...**39.95**

D. Wordprocessing Paper. Size 241 x 279. Design quartz. 250 sheets. 26-9318...**19.95**

Computer Cassette Recorders



CCR-81 Recorder

89^95

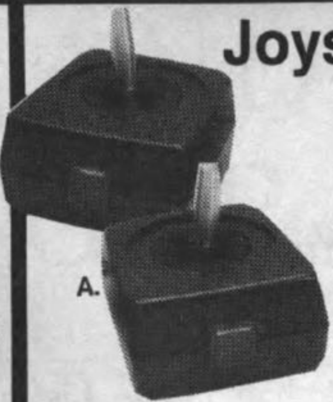
A. Load and save your programs and date on cassette. Battery/Record LED indicator. With cable. 240V AC operation or 4 "C'" batteries (not included). 26-1208

CCR-82 Recorder

79^95

B. Features volume control with preset marker. Includes cable. Requires 4 "AA" batteries or AC adapter (not included). 26-1209

Joysticks



Fun Action for Two Players

34^95 Pair

A. Our lowest-priced joystick. Fast 360° movement to add speed and accuracy to your game. Single shot button. 26-3008

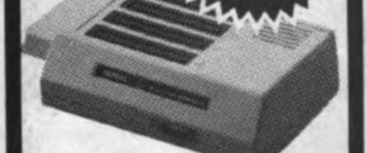
Deluxe Joysticks

49^95 Each

B. You can feel the difference. Get accurate cursor control and quicker response. Dual axis trim control. 26-3012

Multi-Pak Interface

Save $30



149^95

Reg **179.95**

Connects up to four Program Pak™ cartridges. Change between slots with selector switch or under program control. 240V AC, 50 Hz. 26-3124

Color Mouse

79^95

Our Best! Adds speed and ease to games and graphics creation. Simply "roll" the mouse across table top to accurately position cursor. 26-3025

RS-232 Program Pak

SAVE $50

129^95

Reg **179.95**

Communicate with major information services and with other computers over telephone lines. Requires modem. Frees Serial Port. 26-2226

Color Thinline Disk #0

SAVE UP TO $220

FD-501 Color Thinline Disk #0. Turn any color computer with Extended BASIC into a complete disk system and store over 156,000 characters of data. 26-3129 Was **599.00**
Save $200. . . . . . . Sale! 399.95

Second Drive for FD-501. Adds 156K of storage. Requires installation. (not included). 26-3130 Was **519.95**
Save $220. . . . . . . Sale! 299.95

WE SERVICE WHAT WE SELL!

Tandy ELECTRONICS

A DIVISION OF TANDY
AUSTRALIA LIMITED
INC IN N S W

Nearly
350 Stores
Australia-
Wide

Available From 350 Stores Australiawide Including Tandy Computer Centres

or Order On VIATEL *642614#

Independent Tandy Dealers may not be participating in this ad or have every item advertised.
Prices may also vary at individual Dealer Stores

LLIST32

CoCo1/2/3 + PRINTER

UTILITY

by Grahame Pollock

WHEN YOU'RE TRYING to debug a BASIC program, you really need to work with a printout (hardcopy) of the LISTing. This is achieved on the CoCo by typing LLIST.

If you have a TP-10 then you're printout will be 32 columns across.

If you have any other sort of printer, then the listing will be the full width of the printer (usually 80 columns or more).

The 32 column listing is an advantage because it's the same as the screen width. It's also the same as the magazine listings, (which may be done by loading into a wordprocessor).

This 32 column listing really comes in handy when you're trying to find a typographical error.

With "LLIST32" you'll be able to get a 32 column printout on ANY printer. Simply (C)LOAD"LLIST32" and RUN.

Any BASIC program that is (C)LOADed into memory from that point on will LLIST with 32 columns.

This program should work with ALL ROMS:- Colour Basic, Extended Colour Basic, or Disk Extended Colour Basic.

In the BASIC loader, lines 10 and 20 find the end of memory.

Lines 30-50 POKE in the m.l. into memory.

Line 55 moves the output hook information to the end of the m.l. This is important because each type of BASIC has different instructions in this hook.

Line 60 replaces the output hook with a jump to the start of the m.l. program.

The assembly listing for the m.l. is included here for those who want to dissect it.

The Listing:

```
1 CLS:PRINT"32 COLUMN LLIST FOR
THE COCO    BY GRAHAME POLLOCK"
2 GOTO10
3 SAVE"290:3":END'8
10 PK=116
20 X=256*PEEK(PK)+PEEK(PK+1):X=X
-32
30 FORZ=X TO X+30
40 READ A:POKE Z, A
50 NEXT Z
55 POKEX+31,PEEK(359):POKEX+32,P
EEK(360):POKEX+33,PEEK(361)
60 POKE359,126:POKE360,INT(X/256
):POKE361,X-INT(X/256)*256
70 PRINT"ANY BASIC PROGRAM WILL
NOW LLISTWITH 32 COLUMNS TO ANY
PRINTER"
75 CLEAR100,X
80 DATA 52,18,134,254,145,111,39
,4,53,18,32,19,150,156,129,32,39
,2,32,244,52,36,134,13,189,162,1
91,53,36,32,233
```

For those of you with 64K, I've included this program on the RANDISK file that's free with every copy of RANDOS64.

Also included on the file is:-

```
CANON    - a projectile game
BORDERS  - for a DMP-105
SUMS     - for the kids
RECOVER  - to recover programs
              with IO ERRORS
TYPTCH   - to learn typing
LOWRCASE-for true lower case
```

For RANDOS64 and free disk file send $20 to the address below:-

GRAHAME POLLOCK,
24 KENT ST,
MINTO,2566.
AUST.

The Listing:

```
7FDE 34 12    00210 ENTRY  PSHS X,A     PUSH REGISTER A AND X
7FE0 86 FE    00220        LDA  #$FE    "PRINTER ON" CODE
7FE2 91 6F    00230        CMPA $6F     IS PRINTER ON?
7FE4 27 04    00240        BEQ  LENGTH  IF YES GO CHECK LENGTH
7FE6 35 12    00250 RET    PULS A,X     PULL REGISTERS
7FE8 20 13    00260        BRA  RETURN
7FEA 96 9C    00270 LENGTH LDA  $9C     PRINTER HEAD POSITION
7FEC 81 20    00280        CMPA #$20    32 CPL?
7FEE 27 02    00290        BEQ  LFEED   IF YES GO LINE FEED
7FF0 20 F4    00300        BRA  RET     RETURN
7FF2 34 24    00310 LFEED  PSHS Y,B     PUSH Y AND B ONTO THE STACK
7FF4 86 0D    00320        LDA  #$0D    LOAD LINE FEED
7FF6 BD A2BF  00330        JSR  $A2BF   PRINT IT
7FF9 35 24    00340        PULS B,Y
7FFB 20 E9    00350        BRA  RET     RETURN
7FFD 12       00360 RETURN NOP           REPLACED BY
7FFE 12       00370        NOP           OUTPUT HOOK DURING
7FFF 12       00380        NOP           BASIC LOADER
              00390        END
```

■

COMMAND CHANGER

64K ECB
UTILITY

by Martin Eade

COMMAND CHANGER IS AN entry into the utilities competition but is mainly for fun. It is for 64kecb (and possibly decb) systems and is a program that allows you to change allmost any Colour or Extended Colour Basic command to anything you wish, as long as it has the same number of characters as the original.

This program can be used as a fun program, or to make an easier to undestand Basic (for young children, beginners, etc).

When RUN, a menu is displayed, prompting you for a selection.

Type in the required number and an appropriate screen will appear.

Options ...

1 - This allows you to change most commands supported by the computer (I do not have Disk Basic, but it would be possible to incorporate them into the program).

Also, some commands couldn't be changed (eg, RUN).
The computer will ask for a command to be change. If the inputted command is acccepted, it will ask for a replacement.
This must have the same amount of characters as the original.
It will keep asking for commands to change until nothing is inputted at the first prompt.
Then it will return to the menu.

2 - This exits to Basic and allows you to switch between command sets and to program normally (when you are returned to Basic, the new command set is in operation. To get to the original, press reset or type...

```
POKE 65502,127
```

It is possible to get to the new set by typing ...

```
POKE 65503,127
```

- the new set will of course be forgotten if the computer is turned off!

3 - This will save your changes as a basic program and 3 m/c files.
To load in the command set, the Basic program has to be loaded and RUN.
The program will automatically load the m/c and return you to Basic.

Note!!!

'MOTOR ON/OFF" and "AUDIO ON/OFF" cannot be changed entirely. Only the "motor" and "audio" parts can.

The Listing:

```
1 GOTO5
2 CSAVE"CCHANGE"
3 SAVE"305C:3":END'9
5 'COMMAND CHANGING PROGRAM FOR
 64K ECB COCO'S.BY MARTIN EADE
 AGE13  17/7/87
90 DIM C(95),C$(95)
100 CLS:PRINT"      COMMAND CHANG
E PROGRAM"
110 FORA=3072TO3096:READ B:POKEA
,B:NEXT
120 DATA26,80,142,128,0,166,132,
183,255,223,167,128,140,224,0,39
,5,183,255,222,32,239,28,175,57
125 EXEC3072:POKE65503,127
130 PRINT:PRINT"READING DATA-PLE
ASE WAIT......."
140 FORA=1TO95:READ C(A),C$(A):N
EXT
145 CLS:PRINT"COMMAND CHANGING P
ROGRAM"
150 PRINT:PRINT"               ME
NU"
155 PRINT"  1.CHANGE COMMANDS":P
RINT"  2.EXIT TO BASIC":PRINT"
3.SAVE CHANGES TO TAPE"
160 PRINT:PRINT:INPUT"TYPE 1,2 O
R 3";CH:IF CH<1THEN160:IF CH>1TH
EN160
170 ON CH GOTO 200,280,300
180 GOTO145
200 CLS:PRINT"          CHANGE COM
MANDS"
210 PRINT:PRINT:INPUT"ENTER COMM
AND TO CHANGE";CC$:IFCC$=""THEN1
45ELSE220
220 FORZ=1TO95:IF CC$=C$(Z)THEN
GOTO240 ELSE NEXT Z
230 PRINT"COMMAND DOES NOT EXIST
 OR CANNOTBE CHANGED-TRY AGAIN
<INKEY>":EXEC44539:GOTO200
240 PRINT"ENTER NEW COMMAND(";LE
N(CC$);"CHARS.)":INPUT NC$
245 POKE65503,127
250 IF LEN(NC$)>LEN(CC$)THENPRIN
T"TOO LONG":GOTO240ELSE255
255 IF LEN(NC$)<LEN(CC$)THENPRIN
T"TOO SHORT":GOTO240ELSE260
260 FOR L=1TO LEN(NC$)-1:POKE C(
Z)+(L-1),ASC(MID$(NC$,L,1)):NEXT
 L
265 A$=RIGHT$(NC$,1):AA=ASC(A$):
AA=AA+128
270 POKE C(Z)+(LEN(CC$)-1),AA
275 GOTO200
280 CLS:PRINT"TO USE NORMAL COMM
AND SET PRESS RESET OR USE POKE6
5502,127.       TO USE NEW COMMAND
 SET USE POKE 65503,127.THE CURR
ENT COMMAND   SET IS THE NEW ONE
":END
300 IN$(1)="10 CLS:FOR A=3072 TO
 3096:READ B:POKE A,B:NEXT A:EXE
C3072"
305 IN$(2)="20 DATA26,80,142,128
,0,166,132,183,255,223,167,128,1
40,224,0,39,5,183,255,222,32,239
,28,175,57"
310 IN$(3)="30 POKE 65503,127"
315 IN$(4)="40 CLOADM:CLOADM:CLO
ADM"
320 IN$(5)="50 PRINT"+CHR$(34)+"
THE NEW COMMAND SET IS NOW IN
ACTION"+CHR$(34)
400 CLS:PRINT"TAPE SAVE"
410 PRINT:PRINT"INPUT FILENAME":
LINEINPUTF$
420 IF LEN(F$)>8THENPRINT"(MAX.8
 CHARS.)":GOTO310ELSE430
430 PRINT"PRESS RECORD AND PLAY
ON TAPE   AND THEN PRESS A KEY":
EXEC44539
440 OPEN"O",#-1,F$
450 FORPR=1TO5
```

continued on p52

DISKDUMP

32K DECB
UTILITY

by Brendon Pudney

THIS PROGRAM IS FOR the utilities competition. It is called "DISK DUMP". It is used to alter the data stored on the disk. The controls are:

* (right arrow) - go to the second half of the sector,
* (left arrow) - go to the first half of the sector,
* SHIFT (right arrow) - go to the next sector,
* SHIFT (left arrow) - go to the previous sector,
* (up arrow) - go to the next track,
* (down arrow)- go to the previous track,
* E - go into edit mode, and
* J - jump to a specific track and sector.

IN EDIT MODE

While you are in edit mode, simply move the pointer (to the right of the current byte) to the byte that you would like to change.

Then, enter a hexidecimal number from 00 to FF. This is then stored in memory.

To re-save the sector to disk, press (ENTER). The computer will prompt you to make sure that you want it saved.

If you reply "NO", the program re-runs. If you reply "YES" then the sector is saved to the disk.

I hope that you will find this program useful, as I have found that most "industrial" software is not worth the magnetic-media that it is stored on!

In the future I hope to include a printing routine that prints the entire sector, ASCII and all to the printer for later reference.

The Listing:

```
0 GOTO10
3 SAVE"306:3":END'9
10 ' ***************************
20 ' *        DISK  DUMP        *
30 ' *   BY  BRENTON PUDNEY     *
40 ' *      110 BOWER ROAD      *
50 ' *      SEMAPHORE PARK      *
60 ' *  SOUTH AUSTRALIA 5019    *
70 ' * PHONE (08) 491741 A.H.   *
80 ' *      (C) MCMLXXXVII      *
90 ' *      COPYRIGHT 1987      *
100 '*   BY S.B.PRODUCTIONS     *
110 '***************************
120 CLEAR1000
130 DIM O1$(128),O2$(128)
140 CLS3
150 PRINT@203,"DISK  DUMP";
160 PRINT@224,"         BY S.B.PROD
UCTIONS       ";
170 PRINT@288,STRING$(32,32);:PR
INT@288,"";
180 INPUT"TRACK";TN
190 PRINT@320,STRING$(32,32);:PR
INT@320,"";
200 INPUT"SECTOR";SN
210 IF TN<0 OR TN>39 THEN170
220 IF SN<1 OR SN>18 THEN190
230 FS=1
240 CLS0
250 DSKI$0,TN,SN,I1$,I2$
260 FORI=1TO128
270 O1$(I)=MID$(I1$,I,1)
280 O2$(I)=MID$(I2$,I,1)
290 NEXTI
300 V=1048
310 P=0
320 FORA=1TO128 STEP8
330 FORB=A TO A+7
340 IF FS=1 THENOT$=RIGHT$("00"+
HEX$(ASC(O1$(B))),2)
350 IF FS=2 THENOT$=RIGHT$("00"+
HEX$(ASC(O2$(B))),2)
360 PRINT@P,OT$;:P=P+3
370 POKE V,VAL("&H"+OT$)
380 V=V+1
390 NEXTB
400 P=P+8
410 V=V+24
420 NEXTA
430 SOUND1,1
440 A$=INKEY$:IFA$=""THEN440
450 IF A$=CHR$(9)THEN FS=2:CLS0:
GOTO300
460 IF A$=CHR$(8)THEN FS=1:CLS0:
GOTO300
470 IF A$=CHR$(93)THEN FS=1:SN=S
N+1:IF SN>18THEN SN=18:CLS0:GOTO
240:ELSE CLS0:GOTO240
480 IF A$=CHR$(21)THEN FS=2:SN=S
N-1:IF SN<1THEN SN=1:CLS0:GOTO24
0:ELSE CLS0:GOTO240
490 IF A$=CHR$(94)THEN TN=TN+1:F
S=1:IF TN>39 THEN TN=39:CLS0:GOT
O240:ELSE CLS0:GOTO240
500 IF A$=CHR$(10)THEN TN=TN-1:F
S=1:IF TN<0 THEN TN=0:CLS0:GOTO2
40:ELSE CLS0:GOTO240
510 IF A$=CHR$(74) THEN GOTO550
520 IF A$=CHR$(81)THENCLS0:RUN"B
OOT/DOS":END
530 IF A$=CHR$(69)THEN 630
540 FORA=1TO5:SOUND1;1:SOUND10,1
:NEXTA:GOTO440
550 CLS3
560 PRINT@160,STRING$(32,32);:PR
INT@160,"";
570 INPUT"TRACK";TN
580 IF TN<0 OR TN>39 THEN 560
590 PRINT@352,STRING$(32,32);:PR
INT@352,"";
600 INPUT"SECTOR";SN
610 IF SN<1 OR SN>18 THEN 590
620 GOTO240
630 FORA=1TO5:FORB=1TO100:NEXTB:
SOUND150,1:NEXTA
640 PV=1026
650 LL=1026:LR=1047
660 POKE PV,95
670 A$=INKEY$
680 POKE PV,240
690 B$=INKEY$
700 IF A$="" AND B$="" THEN 660
710 IF A$="" THEN A$=B$
720 IF A$=CHR$(9)THEN PV=PV+3:IF
 PV>LR THEN PV=LR:GOTO660:ELSE M
X=MX+1:GOTO 660
730 IF A$=CHR$(8)THEN PV=PV-3:IF
 PV<LL THEN PV=LL:GOTO660 ELSE M
X=MX-1GOTO 660
740 IF A$=CHR$(10)THEN LL=LL+32:
LR=LR+32:PV=PV+32:IF LL>1506 THE
N LL=1506:LR=LR-32:PV=PV-32:GOTO
660:ELSE MX=MX+8:GOTO 660
750 IF A$=CHR$(94)THEN LL=LL-32:
LR=LR-32:PV=PV-32:IF LL<1026THEN
 LL=1026:LR=LR+32:PV=PV+32:GOTO6
60:ELSE MX=MX-8:GOTO660
760 IF A$=CHR$(13)THEN 940
770 IF A$<"0"THEN 860
780 IF A$>"F"THEN 860
790 IF A$>"9"AND A$<"A"THEN 820
800 GOTO830
```

continued on p52

DOUBLES FROM SINGLES

HARDWARE MODIFICATION

by Gordon Thurston

AFTER WEEKS OF WAITING, Tandy had a sale on disk drives. Double sided drives were selling for less than $200.

This was what I had been looking forward to - the ability to use both sides of every disk, and to backup from one drive to another is every Coco owner's dream.

When I got it home, there remained the problem of whether it would work well. The CoCo is not well set up for double sided drives, and it was with some trepidation that I installed the new drive.

The main problem is, that the Coco uses the side select line to access drive 3. I got around this problem by using a single diode as a gate.

These are the instuctions for installing a double sided double density half height drive in a Tandy FD500. When finished, the drive will have drives 0, 2, and 3.

Drive 3 will be the flip side of drive 2.

First remove the case. This voids any warranty that is left, so be warned. There is a plastic insert which looks like the front of a drive. Remove this.

There is already a second connector, but the new 1000 drive has the connector upside-down.

There are two plastic hooks which lock the connector on. Pry the end of each hook gently loose, and push the end of the hook gently to release the back cover of the connector.

When free, the connector can be pulled free from the ribbon cable. There is a metal stake for each of the 34 lines. It takes a bit of pulling to remove the connector.

Remove the two plastic clips which hold the ribbon cable to the back of the drive to free the cable. Install the new drive in the space provided.

You may have to loosen the 4 screws in the lower drive to make the new drive low enough to clear the case. Two extra screws will have to be bought or found (I found mine) for there's only two provided with the new drive.

Move the link on the drive to select drive 2. Twist the cable 180 degrees, and measure where the new connector will come to. Install the connector.

I placed the conne 'or on the bench and used a hammer to gently (??!!) push the back onto the connector and pierce the cable with all 34 stakes. When the back is straight, it should be on properly.

Connect up the computer and drive, and keeping clear of the insides of the drive, check that drive 0 and 2 are operational. A DIR to each drive should confirm.

You now have 2 down and one to go. To make the third one operational, you must operate on the controller, namely the rompack part that slides into the rompack slot.

Disconnect the ribbon cable, and remove it from the computer. There's a screw under the label. Remove it, and you'll find at the more open end, two plastic hooks holding the case together.

Gently pushing on one corner will release one of these hooks. Work slowly and carefully.

When the case is apart, remove the 2 screws and lift out the board. The back is covered with a plastic and metal shield. Pry the fasteners, and remove the shield. Use a germanium diode for the gate.

Count out and mark pins 14 and 32 of the drive end of the board. Bend the leads of the diode to follow the tracks on the board. Cut 2 pieces of insulation (spagetti) to insulate the leads where they might short. Scrape the tracks clean. They are very small, and must be handled carefully.

Using a temperature controlled soldering iron with a small tip, tin the two tracks and solder the diode to the two tracks, keeping the diode clear of the pins.

This is important; the cathode goes to pin 32 (that's the end with the band.)

Put the controller back together, hook everything up, and test for drive 0, 2 and 3.

You might have to DSKINI a disk on the flip side to test it. See if all drives will save ok.

Just a few pointers. If you don't have much experience with printed circuits, or the proper equipment, see if you can find someone to help you out. The tracks are very small on the controller. If you do it yourself, have good light, take your time, don't force anything, and keep track of where everything comes from.

The 240 mains voltage is pretty well insulated, but don't get careless. We need all the computer experts we can get. Be careful of the motor under the drive. it's vulnerable. The gates in the controller are open collector type, so it's OK to use the diode as a gate if you're wondering.

It sure is nice to tell it to backup 0 to 2 or 3. Good luck! ■

A problem with 'Rockfall'?, here is some....

HELP

16/32K CoCo1/2 DEMONSTRATION PROGRAM

by Tom Lehane

THE TITLE SAYS 'HELP' and I have noticed in a few letters to the editor that some colour computer users' have been asking for help with a particular program.

The program is called "Rockfall" by T.J.Davies, Australian CoCo magazine October 1986, page 18.

"Rockfall" is one of those rare programs that uses a hidden machine language routine to achieve a special effect that is not normally available form the computer's own ROM - Read Only Memory.

The machine language routine used with "Rockfall" is appended to the Basic part of the program and follows it around hidden from any listing to the screen or printer.

When you CSAVE or CLOAD the machine language appended to the Basic part of the program is also saved or loaded.

This is achieved by moving the BASIC pointers to the end of the hidden machine language routine and when you type CSAVE the computer looks for the start pointers and CSAVEs the program to tape from the start pointers and continues on until it reaches the end pointers - in this case the pointers are 1874 bytes further on than the true Basic's ending.

How can you tell if there is a hidden machine language program tacked on to a Basic program's listing?

For Basic to call up the machine language routine the computer needs to know where this machine code is.

Some where in the listing there will be a PEEK(27)*256+PEEK(28) - this is where the end pointers of the BASIC program in memory are.

In the program "Rockfall" this PEEK is in line number 6. The BASIC word DEFUSR defines where the entry point of a machine language routine will be.

In the case of "Rockfall" ...

```
DEFUSR=PEEK(27)*256+PEEK(
28)-1874
```

The minus 1874 is the length of the hidden machine code.

The POKE 359,57 also in line number 6 is a nasty one. This POKE disables the return to a text screen, so in a running program that uses PRINT, the computer normally return to the TEXT SCREEN and prints to the screen but with POKE 359,57 nothing happens.

You can use POKE 359,57 in a Basic program but must be accompanied with SCREEN 0,0.

Why has the programmer used this POKE?

The program "Rockfall" uses the hidden machine language routine to transfer what is normally printed on the text screen to your graphic screen and the POKE 359,57 prevents the computer from jumping out of any of the PMODE areas.

To execute the machine language from BASIC we use the USR command. In "Rockfall" this is in line number 62,

```
ML=USR(FC*256+NL)
```

The other variables in side the parentheses were defined also in line number 6.

FC=0 this can be changed to a 1 as the FC variable set dark or light printing to a graphic screen.

The other variable LN=16 sets the number of text lines that can be transferred from the bottom of the text screen to the graphic screen.

You can change this value from 1 to 16, in the case of "Rockfall" all text is transferred to the graphic screen.

The machine language routine used to transfer text screen print to the HI-RES graphic screen was written by Bruce A. Brown and was available on November 1982 CHROMASETTE monthly tapes.

The program is called GRAFTEX.

The author provided only a Basic demonstration with the hidden machine language appended. He suggested deleting the Basic listing and then typing your own Basic program into the computer.

As the end pointers are already set to the end of the machine code nothing would be lost. (The hidden GRAFTEX ...)

The same can be done with "Rockfall" if you have that issue of CoCo Oz on tape (number 43).

The author of GRAFTEX openly invited CoCo users' to use the machine language routine for their own Basic programs.

With this open invitation I have assembled the machine language code in to a Basic program that POKEs the code from DATA statements into a protected area of memory depending on computer size.

Type in the listing and save a copy before you run the program. There is a small demonstration program from line 360 to 440.

To use this with "Rockfall" delete lines 50, 80, 90, 100 and 360-460.

Run the program - when the code has been poked into memory load "Rockfall" and change the following line:

Change line 6 to

```
PMODE3:COLOR3,2:PCLS:DEFUSR=
30886:POKE359,57:NL=16:FC=0
:SCREEN1,0
```

The DEFUSR=30886 is for 32K to 64K computers. If you have a 16K computer change DEFUSR=14506.

Note: If you press the BREAK key the program stops but will not return to the text screen due to the POKE359,57. If you hit the reset button then the computer resets this location to 126 and returns you to the text screen.

Some new CoCo 3 users may ask what is all this fuss about putting text onto a graphic screen? The early Color Computer and CoCo 2 could only put text on to a graphic screen using a fancy machine language program or have each letter drawn using the DRAW statement.

The Listing:

```
0 GOTO10
1 '********* GRAFTEXT ************
   ****** TOM LEHANE ***********
3 SAVE"288:3":END'8
10 CLS:PRINT"ONE MOMENT"
20 M=PEEK(116)
30 IF M=127 THEN CLEAR200,30885:
A=30886:B=32760
40 IF M=63 THEN CLEAR200,14505:A
=14506:B=16380
50 IF PEEK(A)=52 THEN 80
60 FOR I= A TO B
70 READ Q$:POKEI,VAL("&H"+Q$):NE
XTI
80 DEFUSR=A
90 NL=16:FC=0
100 GOTO 360
110 DATA 34,77,DC,B7,93,BA,10,83
,18,0,10,2D,1,20,BD,B3,ED,ED,8D,
1,32,5D,10,27,1,14,4D,26,3,43,20
,1,4F,A7,8D,1,22,31,8D,1,23,10,A
F,8D,1,15,96,B6,81,4,26,E,86,1,A
7,8D,1,11,C1,10,23,16,C6,10,20,1
2,81,3,10,26,0,E6,86,2,A7
120 DATA 8D,0,FD,C1,8,23,2,C6,8,
E7,8D,0,F0,86,20,3D,ED,8D,0,EA,C
C,6,0,A3,8D,0,E3,ED,8D,0,DF,86,C
,E6,8D,0,D8,3D,A6,8D,0,D6,3D,86,
20,3D,ED,8D,0,C4,DC,B7,A3,8D,0,B
E,1F,2,10,AF,8D,0,B7,AE,8D,0,BB,
96,B6,81,4,27,5C,C6,10
130 DATA E7,8D,0,A6,10,AF,8D,0,A
2,A6,80,AF,8D,0,A0,30,8D,0,87,84
,7F,C6,C,3D,E3,8D,0,91,1F,3,C6,C
,A6,C0,34,2,44,44,44,44,A6,86,A8
,20,A7,20,35,2,84,F,A6,86,A8,21,
A7,21,31,A8,20,5A,26,E2,AE,8D,0,
6D,10,AE,8D,0,64,31,22,6A,8D
140 DATA 0,5D,26,B5,31,A9,1,60,8
C,6,0,2D,A6,20,3E,C6,20,E7,8D,0,
4A,10,AF,8D,0,46,A6,80,84,7F,C6,
C,3D,E3,8D,0,3D,1F,3,C6,C,A6,C0,
A8,8D,0,37,A7,20,31,A8,20,5A,26,
F2,10,AE,8D,0,24,31,21,6A,8D,0,1
D,26,D1,31,A9,1,60,8C,6,0
150 DATA 2D,C2,35,F7,0,3,C,F,30,
33,3C,3F,C0,C3,CC,CF,F0,F3,FC,FF
,0,24,9F,3D,DF,6,0,0,10,4,0,1,0,
10,8,0,0,0,0,0,0,0,0,0,0,0,0,3C,
2,3E,42,42,3C,0,0,0,40,40,40,7C,
42,42,42,42,7C,0,0,0,0,0,0,3C,42
,40,40
160 DATA 42,3C,0,0,0,2,2,2,3E,42
,42,42,42,3E,0,0,0,0,0,0,3C,42,7
E,40,42,3C,0,0,0,8,14,10,10,7C,1
0,10,10,10,0,0,0,0,0,0,3C,42,42,
42,42,3E,2,42,3C,40,40,40,7C,42,
42,42,42,42,0,0,0,0,10,0,30,10,1
0,10,10,7C,0
170 DATA 0,0,0,0,2,0,6,2,2,2,2,2
,42,3C,40,40,44,48,70,48,44,42,4
2,0,0,0,10,10,10,10,10,10,10,10,
30,0,0,0,0,0,0,77,49,49,41,41,41
,0,0,0,0,0,0,5C,62,42,42,42,42,0
,0,0,0,0,0,3C,42,42,42,42,3C,0,0
,0,0
180 DATA 0,0,7C,42,42,42,42,7C,4
0,40,40,0,0,0,3C,42,42,42,42,3E,
2,2,3,0,0,0,4C,52,60,40,40,40,0,
0,0,0,0,0,3E,40,3C,2,2,7C,0,0,0,
8,8,8,3E,8,8,8,A,4,0,0,0,0,0,4
2,42,42,42,46,3A,0,0,0,0,0,0,42
190 DATA 42,42,42,24,18,0,0,0,0,
0,0,41,41,41,49,49,77,0,0,0,0,0,
0,42,24,18,18,24,42,0,0,0,0,0,0,
42,42,42,42,42,3E,2,42,3C,0,0,0,
7E,2,4,18,20,7E,0,0,0,6,8,8,8,10
,8,8,8,6,0,0,0,8,8,8,8,8,8,8
20 DATA 8,8,0,0,0,30,8,8,8,4,8,
8,8,30,0,0,0,0,0,0,30,49,6,0,0
,0,0,0,55,AA,55,AA,55,AA,55,AA,5
5,AA,55,AA,FF,81,81,81,81,81,81,
81,81,81,81,FF,0,0,8,1C,3E,3E,3E
,3E,3E,7F,49,41,0,1C,22,22,1C,0,
2A,2A,49,49
210 DATA 8,8,3,F,3C,60,C1,C3,C1,
60,3C,F,3,0,C0,F0,3C,6,83,C3,83,
6,3C,F0,C0,0,1,2,2,6,4,8,8,10,30
,40,40,80,80,40,40,20,20,10,10,8
,6,2,1,1,1F,3,5,9,10,20,40,80,0,
0,0,0,F8,C0,A0,90,8,4,2,1,0,0,0,
0,0

220 DATA 0,0,0,80,40,20,10,9,5,3
,1F,0,0,0,0,1,2,4,8,90,A0,C0,F8,
18,24,42,81,42,24,18,3C,3C,42,42
,0,FF,80,9E,B0,E0,C0,E0,B0,9E,80
,FF,0,0,81,42,24,18,FF,18,FF,18,
FF,18,FF,FF,80,8A,80,8A,80,8A,80
,8A,80,80,FF,FF,1,51,1
230 DATA 51,1,51,1,51,1,1,FF,FF,
81,BD,A5,A5,A5,A5,A5,A5,A5,A5,A5
,A5,A5,A5,A5,A5,A5,A5,A5,A5,BD,8
1,FF,3C,7E,3C,18,7E,BD,BD,99,3C,
42,42,C3,99,BD,BD,99,5A,3C,3C,18
,3C,42,42,42,1,3,7,5,7,D,1F,3B,7
F,5E,7C,78,80,C0,E0,A0,E0,B0,F8
240 DATA DC,FE,7A,3E,1E,0,0,3C,3
C,18,DB,FF,DB,18,18,0,0,FF,81,99
,81,99,81,FF,0,0,0,0,0,0,0,0,0,0
,FF,81,99,81,99,81,FF,0,0,8,14,2
2,41,22,14,8,0,0,0,0,3C,42,5A,24
,18,18,24,5A,42,3C,0,F0,90,FF,9,
FF,90,FF,9,FF,90
250 DATA FF,9,8,4,2,1,2,4,8,10,2
0,10,8,4,18,0,F,0,18,0,F0,0,18,0
,F,0,0,80,40,20,10,1,F3,1,10,20,
40,80,0,1,2,4,8,80,CF,80,8,4,2,1
,3E,41,40,40,4E,51,51,49,3E,0,0,
0,8,14,22,41,7F,41,41,41,41,0,0,
0,7E
260 DATA 21,21,21,3E,21,21,21,7E
,0,0,0,3E,41,40,40,40,40,40,41,3
E,0,0,0,7E,21,21,21,21,21,21,21,
7E,0,0,0,7F,40,40,40,7E,40,40,40
,7F,0,0,0,7F,40,40,40,7E,40,40,4
0,40,0,0,0,3E,41,40,40,40,47,41,
41,3E,0,0,0,41,41,41,41
270 DATA 7F,41,41,41,41,0,0,0,3E
,8,8,8,8,8,8,8,3E,0,0,0,1F,4,4,4
,4,4,4,44,38,0,0,0,41,42,44,48,7
0,48,44,42,41,0,0,0,40,40,40,40,
40,40,40,40,7F,0,0,0,41,63,55,49
,41,41,41,41,41,0,0,0,41,41,61,5
1,49,45,43
280 DATA 41,41,0,0,0,3E,41,41,41
,41,41,41,41,3E,0,0,0,7E,41,41,4
1,7E,40,40,40,40,0,0,0,3E,41,41,
41,41,49,45,42,3D,0,0,0,7E,41,41
,41,7E,48,44,42,41,0,0,0,3E,41,4
0,40,3E,1,1,41,3E,0,0,0,7F,8,8,8
,8,8,8,8,8,0
290 DATA 0,0,41,41,41,41,41,41,4
1,41,3E,0,0,0,41,41,41,41,41,41,
22,14,8,0,0,0,41,41,41,41,41,49,
55,63,41,0,0,0,41,41,22,14,8,14,
22,41,41,0,0,0,41,41,41,22,14,8,
8,8,8,0,0,0,7F,1,2,4,8,10,20,40,
7F,0,0,0,3C
300 DATA 20,20,20,20,20,20,20,3C
,0,0,0,40,40,20,10,8,4,2,1,1,0,0
,0,3C,4,4,4,4,4,4,4,3C,0,0,0,8,1
C,2A,49,8,8,8,8,8,8,0,0,0,0,8,10
,20,7F,20,10,8,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,8,8,8,8
310 DATA 8,8,8,0,8,0,0,0,24,24,0
,0,0,0,0,0,0,0,0,0,24,0,FF,24,
24,FF,24,24,0,0,0,8,3E,49,28,1C,
A,9,49,3E,8,0,0,70,51,72,4,8,10,
27,45,7,0,0,0,C,12,12,C,18,25,42
,46,39,0,0,0,18,10,0,0,0,0,0
320 DATA 0,0,0,0,0,4,8,10,10,10,
10,0,8,4,0,0,0,10,8,4,4,4,4,4,8,
10,0,0,0,0,0,41,22,14,7F,14,22,4
1,0,0,0,0,0,0,8,8,7F,8,8,8,0,0,0
,0,0,0,0,0,0,0,18,18,8,10,0,0,0,
0,0,0,7F,0,0,0,0
330 DATA 0,0,0,0,0,0,0,0,0,18,18
,0,0,0,1,1,2,4,8,10,20,40,40,0,0
,0,3E,41,43,45,49,51,61,41,3E,0,
0,0,8,18,8,8,8,8,8,8,3E,0,0,0,3E
,41,1,2,4,8,10,20,7F,0,0,0,3E,41
,1,1,E,1,1,41,3E,0,0,0,2
340 DATA 6,A,12,22,7F,2,2,2,0,0,
0,7F,40,40,40,7E,1,1,41,3E,0,0,0
,3E,41,40,40,7E,41,41,41,3E,0,0,
0,7F,1,1,2,4,8,10,20,20,0,0,0,3E
,41,41,41,3E,41,41,41,3E,0,0,0,3
E,41,41,41,3F,1,1,41,3E,0,0,0,0,
0,0,0
350 DATA 18,18,0,18,18,0,0,0,0,0
,0,0,18,18,0,18,18,8,10,0,0,2,4,
8,10,20,10,8,4,2,0,0,0,0,0,0,7F,
0,7F,0,0,0,0,0,0,20,10,8,4,2,4,8
,10,20,0,0,3E,41,1,2,4,8,8,0,8,0
,0,0,16,F8,B1,0,0,0,45
360 POKE359,57:PMODE4
370 SCREEN1,0:PCLS:CLS
380 FOR X=32 TO 122
390 PRINTCHR$(X);:NEXT
400 PRINTSTRING$(32,32)
410 FOR X=155 TO 191
420 PRINTCHR$(X);:NEXT
430 GOSUB 450
440 GOTO 440
450 NL=USR(FC*256+NL)
460 RETURN
```

■

QUIZ MACHINE

32K ECB + HARDWARE
GAME

by Nathan Gibson

THE IDEA FOR THIS program came when my mother asked if it would be possible to build a device with three buttons, three lights and a buzzer. The device to be used for a quiz contest with the juniors at the local church.

It seemed that a software solution to this problem would be the easiest way out and also provide a means of scoring.

A simple bit of wiring, three push buttons two 5 pin Din connectors and a 4.7K ohm resistor is still necessary to get the most out of this program.

The program is written for three players, each player having a button.

I used plasic pill bottles from the local chemist to house pushbutton switches. These buttons are connected to the left and right joystick firebutton pins on 5 pin din conectors, pin 4.

The third button uses the joystick (0) pin on one of these conectors, pin 1. It is also necessary to connect a resistor about 4.7K ohms from pin 1 to earth, pin 3.

Using the joystick position was necessary as at the time the program was written.

The second wire from each switch is connected to pin 5, +5 volts on the respective connectors.

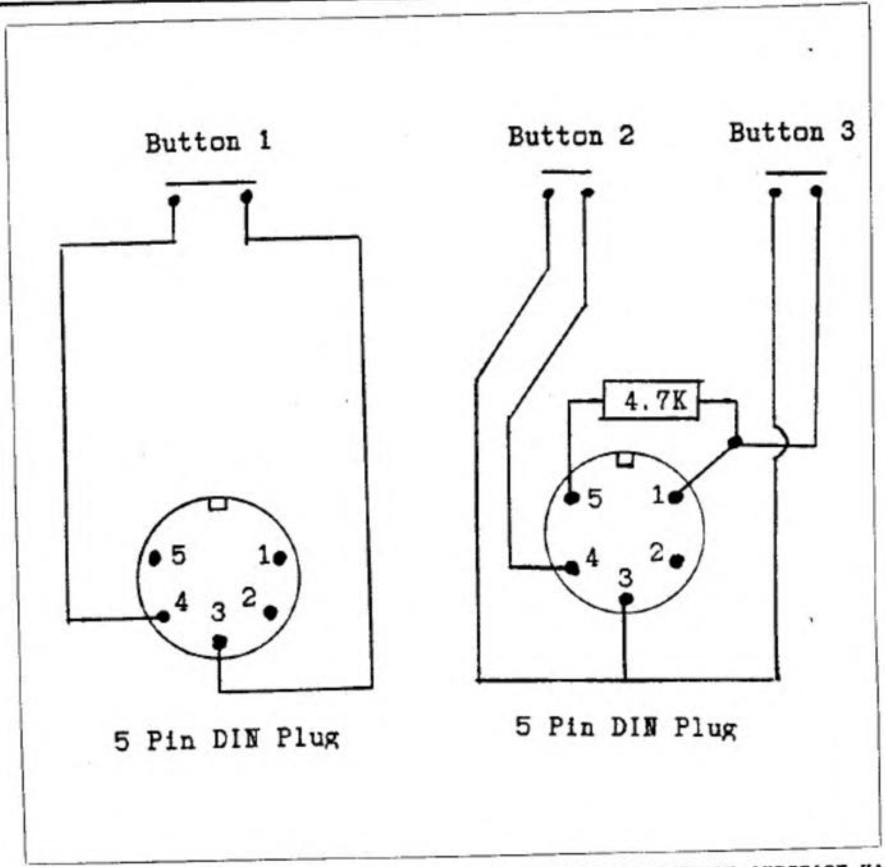
It is possible to use the program for two players without any hardware and simply use the two fire buttons on your joysticks. The third players button is now joystick (0) so pushing the joystick to the right will activate player three's response.

I am twelve years old and have only picked up my programing skills from magazines so this program may not be as efficient as it could be, but it did the job it was intened for, and my Mother has used it with great success on a number of occasions.

The Listing:

```
0 '****************************
1 '*        QUIZ MACHINE      *
2 '*             BY           *
3 '*        NATHAN GIBSON     *
4 '*     (C) COPYRIGHT 1987   *
5 '****************************
6 GOTO10
8 SAVE"281:3":END'1
10 GOSUB49:GOSUB54:PCLS:N$(1)="G
3E3D10":N$(2)="R5D5L5D5R5":N$(3)
="R5D5L5R5D5L5":N$(4)="G5R8L3U5D
10":N$(5)="L5D4R5D6L5":N$(6)="L5
D10R5U5L5":N$(7)="L5R5G1D1G1D1G1
D1G1D1G1D1":N$(8)="L5D10R5U5L5R5
U5":N$(9)="L5D5R5U5D10L5":N$(0)=
"L5D10R5U10
11 N$(1)=N$(1)+"R3L5":V$="D10H3G
3U10":R$="L5D10U5F5H5R5U5":T$="L
5R3D10":I=0:I$="":AA=20:BB=20:CC
=20:LINE(30,30)-(20,40),PSET:LIN
E(20,90)-(40,92),PSET,BF
12 LINE(30,30)-(32,90),PSET,BF:L
INE(222,30)-(192,32),PSET,BF:LIN
E(222,30)-(224,90),PSET,BF:LINE(
222,90)-(192,92),PSET,BF:LINE(22
2,60)-(192,62),PSET,BF:LINE(141,
30)-(111,32),PSET,BF:LINE(141,30
)-(143,60),PSET,BF
13 LINE(141,60)-(111,62),PSET,BF
:LINE(111,60)-(113,90),PSET,BF:L
INE(111,90)-(141,92),PSET,BF:J$=
"L5R2D10L2R5":G$="L5D10R5U5L2":H
$="D10U5L5U5D10":T$="L5R2D10":M$
="D10U2H5U3D10":MM$="D10U10G3H3D
10":E$="L5D5R5L5D5R5":U$="D10L5U
10"
14 FRED=0:AA$=STR$(AA):BB$=STR$(
BB):CC$=STR$(CC):LINE(0,140)-(25
5,150),PRESET,BF:LINE(23,9)-(233
,14),PSET,BF:GOSUB35:GOSUB15:GOT
O32
15 Q=236:T=0:P=PEEK(65280):IFP=1
26ORP=254THEN22ELSEIFP=125ORP=25
3THEN21ELSEIFJOYSTK(0)>60THEN23
16 I$=INKEY$:IFI$=""THEN15ELSEIF
I$="T"THENGOTO48ELSEPLAY"T10GA":
LINE(72,175)-(177,185),PRESET,BF
:RETURN
17 IFFRED<>12 THENLINE(67,175)-(
177,185),PRESET,BF ELSERETURN
18 I$=INKEY$:IFI$="V"THEN24ELSEI
FI$="R"THEN27
19 Q=Q-4:IFQ=24 THEN46ELSELINE(Q
,10)-(Q-4,13),PRESET,BF
20 FORA=1TO10:NEXT:GOTO18
21 GOSUB30:V=1:FORI=1TO7:PAINT(3
0,30),0,0:LINE(30,30)-(32,90),PS
ET,BF:LINE(30,30)-(20,40),PSET:L
INE(20,90)-(40,92),PSET,BF:FORA=
1TO30:NEXTA,I:GOTO17
22 GOSUB30:V=2:FORI=1TO7:PAINT(1
41,30),0,0:LINE(141,30)-(111,32)
,PSET,BF:LINE(141,30)-(143,60),P
SET,BF:LINE(141,60)-(111,62),PSE
T,BF:LINE(111,60)-(113,90),PSET,
BF:LINE(111,90)-(141,92),PSET,BF
:FORA=1TO40:NEXTA,I:GOTO17
23 GOSUB30:V=3:FORI=1TO7:PAINT(2
22,30),0,0:LINE(222,30)-(192,32)
,PSET,BF:LINE(222,30)-(224,90),P
SET,BF:LINE(222,60)-(192,62),PSE
T,BF:LINE(222,90)-(192,92),PSET,
BF:FORA=1TO40:NEXTA,I:GOTO17
24 IFFRED<>12THENPLAY"T255GEGEGE
GEGE":LINE(67,175)-(177,185),PRE
SET,BF:DRAW"BM102,175"+V$+"BM117
,175"+R$+"BM132,175"+N$(0):DRAW"
BM147,175"+M$+"BM162,175"+G$ELSE
RETURN
25 IFV=1THENAA=AA-5ELSEIFV=2THEN
BB=BB-5ELSEIFV=3THENCC=CC-5
26 GOTO14
27 IFFRED<>12THENPLAY"T1A":LINE(
67,175)-(177,185),PRESET,BF:DRAW
"BM102,175"+R$+"BM117,175"+J$+"B
M132,175"+G$:DRAW"BM147,175"+H$+
"BM162,175"+T$ELSERETURN
28 IFV=1THENAA=AA+5ELSEIFV=2THEN
BB=BB+5ELSEIFV=3THENCC=CC+5
29 GOTO14
30 PLAY"T100AAACCCAAAD":RETURN
31 PLAY"T20ACDECDEA":GOTO14
32 T=T+1:IFT=70THEN31
33 P=PEEK(65280):IFP=126ORP=254T
HEN22ELSEIFP=125ORP=254THEN21ELS
EIFJOYSTK(0)>60THEN23
34 FORI=1TO10:NEXTI:GOTO32
35 IFAA<0THENAA$=STR$(0):AA=0ELS
EIFBB<0THENBB$=STR$(0):BB=0ELSEI
FCC<0THENCC$=STR$(0):CC=0
36 IFFRED<>12THENIFAA>5ANDAA<100
THENDRAW"BM30,140"+N$(VAL(LEFT$(
AA$,2)))
37 IFAA<100 THENDRAW"BM45,140"+N
$(VAL(RIGHT$(AA$,1)))
38 IFBB>5ANDBB<100THENDRAW"BM115
,140"+N$(VAL(LEFT$(BB$,2)))
39 IFBB<100 THENDRAW"BM130,140"+
N$(VAL(RIGHT$(BB$,1)))
40 IFCC>5ANDCC<100THENDRAW"BM205
,140"+N$(VAL(LEFT$(CC$,2)))
41 IFCC<100 THENDRAW"BM220,140"+
N$(VAL(RIGHT$(CC$,1)))
42 IFAA>99 THENDRAW"BM30,140"+N$
```

```
(VAL(LEFT$(AA$,2))):DRAW"BM45,14
0"+N$(VAL(MID$(AA$,3,1))):DRAW"B
M60,140"+N$(VAL(RIGHT$(AA$,1)))
43 IFBB>99 THENDRAW"BM115,140"+N
$(VAL(LEFT$(BB$,2))):DRAW"BM130,
140"+N$(VAL(MID$(BB$,3,1))):DRAW
"BM145,140"+N$(VAL(RIGHT$(BB$,1)
))
44 IFCC>99 THENDRAW"BM205,140"+N
$(VAL(LEFT$(CC$,2))):DRAW"BM220,
140"+N$(VAL(MID$(CC$,3,1))):DRAW
"BM235,140"+N$(VAL(RIGHT$(CC$,1)
))
45 RETURN
46 PLAY"BB":DRAW"BM72,175"+T$+"B
M87,175"+J$+"BM102,175"+MM$:DRAW
"BM117,175"+E$+"BM147,175"+N$(0)
+"BM162,175"+U$:DRAW"BM177,175"+
T$
47 I$=INKEY$:IFI$="V"THEN24ELSEI
FI$="R"THEN27ELSE47
48 FRED=12:GOSUB15:GOTO14
49 CLS:PRINT@480,"
QUIZ                           M
ACHINE":PRINT:SOUND150,2:PRINT:S
OUND150,2:PRINT:SOUND150,2:PRINT
:SOUND150,2:PRINT:SOUND150,2:PRI
NT:SOUND150,2:PRINT:SOUND50,2
50 PRINT@204,"* quiz  *":PRINT@1
72,"*********":PRINT@268,"******
***":PRINT@236,"*";:PRINT@244,"*
":FORI=1TO8:PRINT@206,"QUIZ";:PR
INT@237,"machine";:SOUND55,2:PRI
NT@206,"quiz";:PRINT@237,"MACHIN
E";:SOUND150,2:NEXT:PRINT@329,"B
Y NATHAN GIBSON"
51 PRINT@360,"(C) COPYRIGHT 1987
"
52 FORI=1TO2000:NEXT:FORI=1TO15:
PRINT:NEXT
53 RETURN
54 CLS:PRINT"DO YOU NEED INSTRUC
TIONS"
55 I$=INKEY$:IFI$=""THEN55ELSEIF
I$="N"THENPMODE4,1:SCREEN1,1:RET
URNELSE56
```



```
56 PRINT" TO TEST BUTTONS PRESS
<T> AND  ASK CONTESTANT 1 TO PRE
SS HIS   BUTTON. REPEAT THIS FOR
 EACH    CONTESTANT":PRINT" AFTE
R ASKING EACH QUIZ QUESTIONPRESS
 ANY KEY TO START THE TIMER.":PR
INT" IF THE QUESTION IS ANSWERED
    CORECTLY PRESS <R>";
57 PRINT" AND THE SCOREWILL INCR
EASE.":PRINT" IF QUESTION IS ANS
WERED            INCORECTLY PRESS <
W>":PRINT"IF TIME-OUT OCURES, AD
JUDICATOR DECIDES IF QUESTION HA
S BEEN    ANSWERED CORECTLY":PRI
NT"          PRESS <ENTER>";
58 I$=INKEY$:IFI$<>CHR$(13)THEN5
8
59 CLS:PRINT" IF NO CONTESTANT P
RESSES HIS    BUTTON AND THE TONE
 SOUNDS       CONTINUE TO NEXT QU
ESTION.":PRINT"            PRESS <E
NTER>"
60 I$=INKEY$:IFI$<>CHR$(13)THEN6
0ELSEPMODE4,1:SCREEN1,1:RETURN
```

DATA STRUCTURES in FORTH

by John Redmond

LAST MONTH WE started to look at how we might use Forth's special flair for data structures in constructing a full-screen editor which uses high resolution character display. This editor, and the special words it uses are part of the standard E*FORTH library, and available as an upgrade to all legal A*FORTH users.

The high-res screen was a must for a serious editor, in order to get around the 32-column problem and the lack of lower case on the text screen. And, as was confidently expected, Forth turned out to be ideal for the job.

To define the obvious, the graphics screen we will use is 256 by 192. The video memory is organized as a 192 by 32 byte array and, if we are to display 24 lines of text, each line will require 192/24 = 8 of the 32-byte rows of bytes.

As we define our character set, it is convenient (and fairly efficient) to use one byte for each of the rows of pixels so that, for each printable character, we require these eight definition bytes. For convenience, these will be stored in consecutive memory locations in the definition array but, when displayed on the screen, they must be introduced into video memory at locations 32 bytes apart.

To keep the concepts simple for a start, consider that we will display only 32 characters across each screen line. This means that each character, in video memory, will be one byte wide. All we need to know is 1. The address of the first byte in video memory where we will introduce the character (we will call this 'CURSOR); and 2. The address of the group of eight bytes in the character definition array.

Next month, the Forth screen text file as a data structure.

Let us take the second of these first. Assume that we want to display 'A', which has an ASCII value of 65 (41 hex). For simplicity, we have introduced no space economies; so the definition array (called FONTS) will contain (127 x 8) bytes for the 127 ASCII characters. We will define it by

```
CREATE FONTS 127 8 * ALLOT.
```

Now the 65th character ('A') will be defined by the 8 bytes starting (65 x 8) bytes after the base address of the array. Therefore, the word to display the character (we will call it .CHAR) needs to do this arithmetic. The bones of its definition will be

```
: .CHAR  ( char is on stack)
  8 * ( offset to 8 bytes)
  FONTS + ( add base addr)
  'CURSOR @ (screen dest)
  (.CHAR) ; ( does the work)
```

Of course we've put off all the work by invoking the low-level slave word, (.CHAR), but this sort of code factoring keeps the program simple, limits the bugs and makes it possible to introduce optimization at a later stage. Poor old (.CHAR) is given the source and destination address on the stack, and has the job of transferring the character bytes in the correct way.

```
: (.CHAR)  ( srce add, dest)
  8 0  ( moving 8 bytes)
  DO
   OVER C@ ( fetch a byte)
   OVER C! ( store it)
   SWAP 1+ ( incr srce add)
   SWAP 32 + ( and dest add)
  LOOP
  2DROP ; ( drop addresses)
```

Isn't it simple? Provided you have all the appropriate bytes set up in FONTS, it's trivially easy to display a character. In truth, it's not quite as easy as that because, as it stands, .CHAR will display all characters in the same position on the screen. There really is quite a lot of housekeeping to do as well: The destination pointer must be updated by 1 'CURSOR +! and we must update a counter of the number of characters so far displayed on the line. This must be compared with the maximum allowed (presently 32) and, if we've gone too far, we must go to the start of the next screen line.

But what if we were already on the bottom line? Then we must scroll!. And what if the character we are passing to .CHAR is a control char, like carriage return or backspace? Obviously they have to be given special treatment. Yes I've told you about only a part of the whole process, but all these complications can be handled very nicely, and compactly, in Forth. It's all on the E*FORTH library disk.

To complete this point, we've taken a very easy line by choosing to display only 32 characters across the screen. We really want 51. To do this, we must reassess the nature of video memory.

Previously, we saw it as 192 rows of 32 bytes: now we must see it as 192 rows of 256 bits.

Forth can handle it, but life is more complicated. To display more than 32 characters across, we are allowed to use only PART of each video byte. For 51 characters, we are entitled to only 5 bits of each byte. To use these, we have to left-justify each of the definition bytes to fit neatly alongside the previous character displayed. This requires us to keep track of the appropriate number of left shifts. (Starting to sound like assembly programming, isn't it? But Forth is not afraid to roll up its sleeves at the low level.)

BITS is the variable to keep track of this left-shifting. Each time a character is displayed, the character width (contained in the variable /CHAR) is SUBTRACTED from the value in BITS and, if the value becomes negative, 8 is added and the address in 'CURSOR is increased. In fact, .CHAR will not have the right to play with the value in 'CURSOR. This has been given to +BITS:

```
: +BITS  BITS @ /CHAR @ -
  DUP 0< ( is it negative?)
  IF 8 + ( adjust it)
   1 'CURSOR +! ( next add)
  THEN
  BITS ! ; ( update shifts)
```

There are further considerations, about whether the new character is simply stored on the screen, or whether it is ORed (merged) on the screen, but these are not relevant to the present discussion. What emerges from what I've said, however, is that several parameters have to be identified in order to fully define the position of a character on the screen. For full and convenient manipulation of the screen, five variables must be defined:

```
(LINE) : current screen line

(COLUMN) : current char position

'LINE : add of start of line

'CURSOR : add of next char

BITS : left shifts to do.
```

'With a full screen editor, different things happen at different times at different places on the screen.'

Now the obvious way to handle the task is to store the information in five variables; so we would define:

```
VARIABLE (LINE)  etc
```

With a full screen editor, different things happen at different times at different places on the screen. The cursor might, for instance, be presently on character 5 of line 3, but you want to input a command to search for a string. You will be prompted to enter the string somewehere else, at the top or bottom of the display. But, as soon as this happens, the screen driver program forgets where the cursor is currently located. So we must save the above five values in other variables, like

```
BITS @ CURSOR-BITS ! etc.
```

A bit of a pain, all this, but the cursor position MUST be saved. Brodie, in 'Thinking Forth' introduces the notion of a status array. Say we have the five variables stacked away in an array, then all we need is a reserve array, called (CURSOR) to which we do a simple and efficient block copy. We will retain the name of the first 'variable', (LINE), as the name of the status array and the code will be ...

```
CREATE (LINE) 10 ALLOT

(LINE) 2+   CONSTANT (COLUMN)

(LINE) 4 +  CONSTANT 'LINE

(LINE) 6 +  CONSTANT 'CURSOR

(LINE) 10 + CONSTANT BITS

CREATE (CURSOR) 10 ALLOT

: SAVECURSOR (LINE) (CURSOR)

  10 CMOVE ;

: FINDCURSOR  (CURSOR) (LINE)

  10 CMOVE ;
```

This simple device makes the source code for SAVECURSOR AND FINDCURSOR so brief and clear - and they work faster!

Again we are out of space. Next month, the Forth screen text file as a data structure, how we find the lines in it and print them out in the correct place on the editor screen.

There will be only one more instalment in this current series of Forth articles. By this stage there is a sizeable number of Forth users on the CoCo. I know that interesting things are being done with the language, but it is high time that the world was told about them. How about it? If anyone wants to discuss Forth, I'm available at 23 Mirool St., West Ryde, NSW 2114 or (better) (02)-85-3751 (after 7 pm). Come Forth! ■

GOLDGRABBER!

32K DISK SYSTEM
GAME

by Andrew Voutsis
Screens by Chris Voutsis

AFTER MANY GREAT hours of enjoyment playing the very challenging "Hoards of the Deep Realm" programmed by Vaughn Clarkson for the Microbee, we all wished we could play it on our own CoCo. As far as we know, there was one similar game made for the CoCo, called "Gold Runner" by Dave Dies, that ran off a tape based system. What we wanted was a game in which we could run off a disk based system, and that we could build our own screens on.

So, after several unsuccessful attempts, we finally got it right. It is called "Goldgrabber" and it is divided into two parts. The first called "Builder", is a program to build the screens.

The second, is called "Goldgrab" and is THE program you need to play the game.

HOW TO BUILD A SCREEN

You are a cursor that you manipulate around a 20 X 11 pmode 3, screen 1,0 'block' graphics screen. The arrows move you at the direction they indicate. The following numbers produce individual items for the screens as follows:

#1 - gold bars (red),
#2 - ladders (yellow),
#3 - ropes (yellow) and
#4 - blocks (blue).

When you are happy with your screen, place the cursor in a blank area and press "S" for save.

Now follow the instructions to disk-save your product. Alernatively, "L"oad a previous saved screen and alter it if you so wish. When playing, call the desired screen by the name you gave it.

PLAYING GOLDGRAB

The ropes and ladders help you climb. The gold increases your score by 50 points each, and the blocks attempt to create a difficult environment for you to operate in.

You are the red character wandering and collecting gold (50 points) by passing under it, while being chased by two yellow, unpredictable guards. Burying each guard gives you 200 points and an extra life (and I can assure you you need it, even though you start with five men).

This is achieved by pressing either 'M' for the gap to open to diagonally down left, or "/" to right. Remember, you have to wait for the gap to fill in before you can dig again. To move, use the up and down arrows for up and down, and comma and full stop for left and right. Press "M" and "/" to bury guards and provide access to seemingly impossible places. When in a hopeless position, press "X" to commit suicide.

Remember, you always start at a random position, and so do your hunters.

You die when the guards catch up with you, or fall into the your own hole or pressing "X". When all the gold is collected, the program restarts by asking you for the next screen. Your score and lives carry on to the next screen.

Good luck, and we hope you like the program.

However, if you are too lazy and want ready made screens, send a blank disk and $5 to A. Voutsis 23 Curragundi Rd, Jindalee, 4074, and we will send you 20 screens.

Anything we have forgotten, you just have to figure out for yourselves. May the Gold Fever be with you.

The Listing:

```
0 GOTO5
1 '******* GOLDGRABBER *********
   * ANDREW & CHRIS VOUTSIS ****
3 SAVE"274:3":END'1
5 CLEAR 500
7 Q$="O3T255L255ABC
10 REM goldgrabber by andrew  v
outsis
12 B$="T2P4V15L16O4CEGL8O5CL16O4
AL4O5C
13 C$="O1V25T25L2CCC
14 DD$="V25T5O3P4L2CL3CL8CL2CE-L
8DL3DL8CL3CO2L8BO3L2CP2
15 E$="T5V15L4O4ED#ED#EO3BO4DCL4
O3AP4
16 F$="O3T41;9;2;8;3;7;4;6;5;6
19 MEN=5:SC=0
20 REM january 1986
30 CLS RND(7)+1
35 PLAY E$
40 PRINT@230,"FILENAME OF SCREEN
 ";:INPUT A$
50 PMODE 3,1:SCREEN 1,0:PCLS
55 POKE 65494,0
60 LOADM A$
70 POKE 65495,0
80 L=8:TR=0:DIG=0:XX=0:YY=0:AM=0
:MA=0:GID=0:EM=1
81 C=RND(12)+7:D=RND(10):C=C*12:
D=D*16:C=C+8:D=D+8:IF PPOINT(C,D
)=3 THEN 81 ELSE 82
82 CC=RND(12)+7:DD=RND(10):CC=CC
*12:DD=DD*16:CC=CC+8:DD=DD+8:IF
PPOINT(CC,DD)=3 THEN 82 ELSE 90
90 FOR T=14 TO 242 STEP 12
100 IF PPOINT(T,L)=4 THEN TR=TR+
1:NEXT ELSE NEXT
110 IF L=168 THEN 120 ELSE L=L+1
6:GOTO 90
120 BT$="C1R4D4R4U4L6D4R8U4L2
130 US$="BD7BR4F2NU2NE2D2NF2NG2
135 D$="R10D14L10U14D3NR10D3NR10
D4NR10D4R4NU14R4NU14
140 X=RND(10):Y=RND(5):X=X*12:Y=
Y*16:X=X+8:Y=Y+8
150 IF PPOINT(X,Y)=3 THEN GOTO 1
40 ELSE 155
155 PLAY B$
159 GOSUB 1010:GOSUB 1020
160 GOSUB 1050
161 IF DIG=1 THEN AM=AM+1:IF AM=
>20 THEN DRAW"BM"+STR$(XX)+","+S
TR$(YY)+"C3"+D$:DIG=0:IF X=XX AN
D Y=YY THEN PLAY DD$:GOTO 205 EL
SE 1060
162 IF GID=1 THEN MA=MA+1:IF MA=
>20 THEN DRAW"BM"+STR$(AA)+","+S
```

```
TR$(BB)+"C3"+D$:GID=0:IF X=AA AN
D Y=BB THEN PLAY DD$:GOTO 205 EL
SE 1060
163 REM taking treasures
164 IF PPOINT(X+6,Y)=4 THEN PLAY
 Q$:DRAW"BM"+STR$(X)+","+STR$(Y)
+BT$:SC=SC+50:TR=TR-1:IF TR=<0 T
HEN 1130
165 IF PPOINT(X+4,Y+28)<>3 THEN
IF PPOINT(X+2,Y+8)=2 OR PPOINT(X
+4,Y+4)=2 THEN GOTO 170 ELSE GOS
UB 1000:Y=Y+16:GOTO 160
167 IF EM=1 THEN EM=2 ELSE IF EM
=2 THEN EM=1
170 REM digging holes
175 REM DIGGING TO THE LEFT
180 IF PEEK(343)=253 THEN IF DIG
=1 OR Y=168 THEN 190 ELSE IF PPO
INT(X-6,Y+16)<>3 THEN 190 ELSE D
RAW"BM"+STR$(X-12)+","+STR$(Y+16
)+"C1"+D$:AM=0:DIG=1:XX=X-12:YY=
Y+16:PLAY C$
185 REM DIGGING TO THE RIGHT
190 IF PEEK(345)=223 THEN IF GID
=1 OR Y=168 THEN 195 ELSE IF PPO
INT(X+18,Y+16)<>3 THEN 195 ELSE
DRAW"BM"+STR$(X+12)+","+STR$(Y+1
6)+"C1"+D$:MA=0:GID=1:AA=X+12:BB
=Y+16:PLAY C$
195 REM when the guards catch
you
200 IF X=C AND Y=D OR X=CC AND Y
=DD THEN 203 ELSE 240
203 PLAY DD$:GOSUB 1000,1030,104
0
205 MEN=MEN-1:CLS
210 IF MEN=0 THEN PRINT@201,"gam
e over"; ELSE 220
215 PLAY F$:PRINT@265,"FINAL SCO
RE ";SC;
217 PRINT @480,"HIT ANY KEY TO P
LAY AGAIN ";:INPUT B$:IF B$="N"
THEN END ELSE CLEAR:RUN
220 IF MEN=>1 THEN PRINT@234,"SC
ORE ";SC;
225 PRINT@294,"YOU HAVE ";MEN;"
MEN LEFT";
230 PRINT@480,"HIT ANY KEY TO CO
NTINUE ";:INPUT E$:PMODE 3,1:SCR
EEN 1,0:GOSUB 1100:GOSUB 1050:GO
SUB 1010:GOSUB 1020:GOTO 161
240 '
250 IF PEEK(341)=247 THEN IF PPO
INT(X+6,Y-2)=3 THEN 260 ELSE IF
PPOINT(X+2,Y+8)=2 THEN GOSUB 100
0:Y=Y-16:GOSUB 1050
260 IF PEEK(342)=247 THEN IF PPO
INT(X+6,Y+16)=3 OR PPOINT(X+4,Y+
28)=3 THEN 270 ELSE GOSUB 1000:Y
=Y+16:GOSUB 1050
270 IF PEEK(342)=223 THEN IF PPO
INT(X-1,Y+8)=3 THEN 280 ELSE GOS
UB 1000:X=X-12:GOSUB 1050
280 IF PEEK(344)=223 THEN IF PPO
INT(X+12,Y+8)=3 THEN 250 ELSE GO
SUB 1000:X=X+12:GOSUB 1050
285 IF PEEK(338)=247 THEN GOTO 2
03
290 REM making guards chase you
295 IF EM=2 THEN 999 ELSE IF EM=
1 THEN 310
310 IF PPOINT(C+4,D+28)<>3 THEN
IF PPOINT(C+2,D+8)=2 OR PPOINT(C
+4,D+4)=2 THEN GOTO 315 ELSE GOS
UB 1030:D=D+16:GOSUB 1010:GOTO 3
10 ELSE 315
315 IF PPOINT(CC+4,DD+28)<>3THEN
 IF PPOINT(CC+2,DD+8)=2OR PPOINT
(CC+4,DD+4)=2 THEN GOTO320ELSE G
OSUB 1040:DD=DD+16:GOSUB 1020
320 IF C<X THEN IF PPOINT(C+12,D
+8)=3 THEN 330 ELSE GOSUB 1030:C
=C+12:GOSUB 1010
330 IF C>X THEN IF PPOINT(C-1,D+
8)=3 THEN 340 ELSE GOSUB 1030:C=
C-12:GOSUB 1010
340 IF D<Y THEN IF PPOINT(C+6,D+
16)=3 THEN 350 ELSE GOSUB 1030:D
=D+16:GOSUB 1010
350 IF D>Y THEN IF PPOINT(C+6,D-
2)=3 THEN 360 ELSE IF PPOINT(C+2
,D+8)=2 THEN GOSUB 1030:D=D-16:G
OSUB 1010 ELSE 360
360 REM moving the blue guard
370 IF CC<X THEN IF PPOINT(CC+12
,DD+8)=3 THEN 380 ELSE GOSUB 104
0:CC=CC+12:GOSUB 1020
380 IF CC>X THEN IF PPOINT(CC-1,
DD+8)=3 THEN 390 ELSE GOSUB1040:
CC=CC-12:GOSUB 1020
390 IF DD<Y THEN IF PPOINT(CC+6,
DD+16)=3 THEN 400 ELSE GOSUB 104
0:DD=DD+16:GOSUB 1020
400 IF DD>Y THEN IF PPOINT(CC+6,
DD-2)=3 THEN 999 ELSE IF PPOINT(
CC+2,DD+8)=2 THEN GOSUB 1040:DD=
DD-16:GOSUB 1020 ELSE 999
999 GOTO 161
1000 DRAW"BM"+STR$(X)+","+STR$(Y
)+"C1"+US$:RETURN
1010 DRAW"BM"+STR$(C)+","+STR$(D
)+"C2"+US$:RETURN
1020 DRAW"BM"+STR$(CC)+","+STR$(
DD)+"C2"+US$:RETURN
1030 DRAW"BM"+STR$(C)+","+STR$(D
)+"C1"+US$:RETURN
1040 DRAW"BM"+STR$(CC)+","+STR$(
DD)+"C1"+US$:RETURN
1050 DRAW"BM"+STR$(X)+","+STR$(Y
)+"C4"+US$:RETURN
1060 IF C=AA AND D=BB OR C=XX AN
D D=YY THEN SOUND 100,1:SC=SC+20
0:MEN=MEN+1:GOTO 1065 ELSE 1070
1065 C=RND(12)+7:D=RND(10):C=C*1
2:D=D*16:C=C+8:D=D+8:IF PPOINT(C
,D)=3 OR C=X AND D=Y THEN 1065 E
LSE GOSUB 1010:GOTO 1070
1070 IF CC=AA AND  DD=BB OR CC=X
X AND DD=YY THEN SOUND 100,1:SC=
SC+200:MEN=MEN+1:GOTO 1075 ELSE
GOTO 163
1075 CC=RND(12)+7:DD=RND(10):CC=
CC*12:DD=DD*16:CC=CC+8:DD=DD+8:I
F PPOINT(CC,DD)=3 OR CC=X AND DD
=Y THEN 1075 ELSE GOSUB 1020:GOT
O 163
1100 X=RND(10):X=X*12:X=X+8:Y=RN
D(5):Y=Y*16:Y=Y+8:IF PPOINT(X,Y)
=3 THEN X=0:Y=0:GOTO 1100 ELSE 1
102
1102 REM
1104 REM
1110 RETURN
1130 ' loading the next screen
1140 CLS:PRINT@234,"SCORE ";SC;
1150 PRINT@320,"ENTER FILENAME O
F NEXT SCREEN ";:INPUT A$:GOTO 5
0
```

■

The Listing:

```
0 GOTO10
3 SAVE"274A:3":END'1
10 REM BUILDER FOR GOLDGRAB
20 REM (C) A. VOUTSIS, SEPT '86
22 X=20:Y=24
30 DIM CU(14)
40 REM DEFINING PIECES VARIABLES
50 REM TREASURE
60 A$="C4BR4R4D4L4U4L2D4R8U4L2
70 REM LADDER
80 B$="C2BR2D14BR8U14
90 REM ROPES
100 C$="C2BD4R10
110 REM BLOCKS
120 D$="C3R10D14L10U14D3NR10D3NR
10D4NR10D4R4NU14R4NU14
121 CLSRND(8):PRINT@0,"LOAD OLD
SCREEN?(Y/N)":INPUT FL$:IF FL$="
Y" THEN PRINT"FILENAME":INPUT LD
$ ELSE 122
122 PMODE 3,1:SCREEN 1,0:PCLS
123 IF LD$="" THEN 125 ELSE LOAD
M LD$
125 COLOR 3:LINE(7,7)-(249,185),
PSET,B:PAINT(1,1),3,3
130 LINE(X,Y)-(X+10,Y+14),PSET,B
F
140 GET(X,Y)-(X+10,Y+14),CU,G
150 REM the loop
160 IF PEEK(341)=247 THEN IF Y=8
 THEN 170 ELSE Y=Y-16:E$=Z$
170 IF PEEK(342)=247 THEN IF Y=1
68 THEN 180 ELSE Y=Y+16:E$=Z$
180 IF PEEK(343)=247 THEN IF X=8
 THEN 190 ELSE X=X-12:E$=Z$
190 IF PEEK(344)=247 THEN IF X=2
36 THEN 160 ELSE X=X+12:E$=Z$
200 PUT(X,Y)-(X+10,Y+14),CU,OR
206 LINE(X,Y)-(X+10,Y+14),PRESET
,BF
220 IF PEEK(339)=239 THEN E$=A$
230 IF PEEK(340)=239 THEN E$=B$
240 IF PEEK(341)=239 THEN E$=C$
250 IF PEEK(342)=239 THEN E$=D$
255 V$=INKEY$
257 IF V$="S" THEN LINE(X,Y)-(X+
10,Y+14),PRESET,BF:CLS:PRINT"SUR
E";:INPUT Q$:IF Q$<>"Y" THEN SCR
EEN 1,0:GOTO 260 ELSE PRINT"FILE
NAME";:INPUT X$:SAVEM X$+"",3584
,9728,3584:END
260 DRAW"BM"+STR$(X)+","+STR$(Y)
+";"+E$
400 GOTO 150
410 SCREEN1,1:GOTO410
```

■

A helping hand for your Dot Matrix.

DMDATGEN

32K ECB + DMP–110
UTILITY

by Alan Bridges

BEING AN ENTHUSIASTIC reader of CoCo/Softgold magazines, I look forward each month to picking up my copies at the local Tandy store (Sringwood, Q).

Some of the articles in the magazines have lead to further investigation on my part, and have definitely given me a better understnding of my colour computer.

Upon reading through the June edition of Softgold, my imagination was fired up by one of Johanna Vagg's articles - 'Crumbling Mental Block'.

After reading the article I too, took another look at "GARFIELD". The printer I use is a DMP-110 and the program works well.

From there I tried a few creations of my own. PROBLEM - heaps of DATA statements for a very small result! ANSWER - Why not get CoCo to calculate all the data from a picture drawn by the user?

After a few experiments to improve my knowledge of binery numbers and graphics printing on the DMP-110, I came up with my Utility program 'DOT MATRIX DATA GENERATOR'.

The program is very simple to operate, a title / instruction screen is displayed for which I used the character set from 'COMSAT' (by Max Bettridge Softgold April '87). I hope the author accepts the compliment.

The directional controls for drawing are the same as CoCo's DRAW statement.

e.g.
U - up
E - diagonally up & right
D - down
F - diagonally down & right
L - left
G - diagonally down & left
R - right
H - diagonally up & left

Pressing <ENTER> displays the drawing screen which has a 62 X 49 matrix. Cursor movement is by the previously defined keys.

If help is required to remember the keys, pressing 'I' will return to the instruction screen.

Pressing <ENTER> again returns to the drawing screen.

Pressing 'C' toggles the colour on & off.

When your masterpiece is complete, pressing CLEAR & 'P' together will initiate the data processing and file creation.

Each data line is displayed on the screen as it is created. When this process is complete you are asked to press the RECORD & PLAY buttons on the cassette player and enter the filename.

The newly created file complete with the necessary printer control lines is written to tape and instructions for using the file are displayed.

Using the file is very simple; LOAD the program file, turn on the printer and RUN the program.

Although the finished print is still fairly small (approx. 50cm x 25cm) this utility does alleviate the odious task of mapping out all that data.

Up to 434 items of data are producted by 'DMDATGEN' when covering the full matrix. It does not however produce all that data if the picture drawn is smaller than the matrix.

Lines 1000-1100 check the matrix to find the smallest area that may be used from the top left hand corner.

I hope everyone has fun with this, my entry for the utilities competition - even the kids may be able to find a use for it creating ornate labels for their school book.

By the way I think we should all congratulate JOHANNA VAGG on her well written and most informative articles - WELL DONE JOHANNA.

The Listing:

```
0 '****************************
   * DOT MATRIX DATA GENERATOR *
   *       BY ALAN BRIDGES      *
   ****************************
   *  CREATED - JUNE 1987       *
   ****************************
   *  UPDATED -                 *
1 '****************************
   THIS PROGRAM GENERATES DATA
   STATEMENTS FROM A PICTURE
   DRAW ON THE HI-RES SCREEN
   THESE ARE THEN USE TO CREATE
   A PICTURE IN DOT ADDRESSABLE
   GRAPHICS ON THE TANDY DMP 110
   PRINTER.
2 GOTO 9
3 SAVE"271:3":END'8
9 CLEAR 2000:PCLEAR8:DIM L$(57),
D$(62),C$(7):GOSUB 5000
10 PMODE4,1:PCLS1:SCREEN1,0:COLO
R0
20 FORX=5 TO 249 STEP 4
25 LINE(X,8)-(X,155),PSET
26 NEXT
30 FOR Y=8 TO 156 STEP 3
35 LINE(5,Y)-(249,Y),PSET
36 NEXT
40 FOR Y=8 TO 156 STEP 21
45 PSET(1,Y):PSET(2,Y)
46 PSET(252,Y):PSET(253,Y)
48 NEXTY
50 FOR X=5 TO 249 STEP12
51 PSET(X,5):PSET(X,6)
52 NEXT
54 X1=40:Y1=170:V$="PRESS I FOR
INSTUCTIONS"
55 FOR L=1TO 23:P$=MID$(V$,L,1):
IF P$=" " THEN X1=X1+7 ELSE DRAW
"BM"+STR$(X1)+","+STR$(Y1)+L$(AS
C(P$)-33)
56 X1=X1+7:NEXT
100 X=6:Y=9:C=1
110 SOUND200,2
115 A$=INKEY$:IF PEEK(339)=191 A
ND PEEK(338)=251 THEN 1000
116 IF A$="C" THEN GOSUB 300
117 IF A$="I" THEN GOSUB 250
120 COLOR C
125 PSET(X+1,Y+1):PSET(X,Y):PSET
(X+2,Y+1)
140 IF A$<>"" THEN GOSUB 500
148 PRESET(X+1,Y+1):PRESET(X,Y):
PRESET(X+2,Y+1):PRESET(X+2,Y+2)
150 GOTO 115
```

```
200 REM INKEY ROUTINE
210 IF INKEY$<>CHR$(13) THEN 210
ELSE RETURN
250 REM DISPLAY HELP SCREEN
255 PMODE4,5:SCREEN1,0
260 GOSUB 200
265 PMODE4,1:SCREEN1,0
300 REMRUB OUT
310 IF C=0 THEN C=1 ELSE C=0
315 LINE(X,Y)-(X+2,Y+1),PRESET,B
F
370 RETURN
500 IFA$="U" THEN Y=Y-3
505 IF A$="E" THEN X=X+4:Y=Y-3
510 IFA$="D" THEN Y=Y+3
515 IF A$="F" THEN X=X+4:Y=Y+3
520 IFA$="L"THEN X=X-4
525 IF A$="G" THEN X=X-4:Y=Y+3
530 IFA$="R"THEN X=X+4
535 IF A$="H" THEN X=X-4:Y=Y-3
540 IF Y<9 THEN Y=9 ELSE IF Y>15
3 THEN Y=153
544 IF X<6 THEN X=6 ELSE IF X>24
6 THEN X=246
546 LINE(X,Y)-(X+2,Y+1),PSET,BF
550 RETURN
1000 REM PRECESS DATA*******
1001 REM TEST FOR SMALLEST AREA
1002 LINE(X,Y)-(X+2,Y+1),PSET,BF
1010 FOR YY=153 TO 9 STEP-3
1020 FOR XX=7 TO 247 STEP 4
1030 IF PPOINT(XX,YY)=0 THEN 105
0
1040 NEXTXX,YY
1050 Z2=YY+3
1060 FOR X=247 TO 7 STEP-4
1070 FOR Y=9 TO 153 STEP 3
1080 IF PPOINT(X,Y)=0 THEN GOTO
1095
1090 NEXT Y,X
1095 Z1=X+4
1100 IF Z1<11 AND Z2<12 THEN 100
 ELSE Z3=INT((Z2-12)/21)+1
1200 REM CREATE DATA FILE
1205 N=0:Y=10:LN=10
1206 CLS:PRINT"  THESE ARE THE L
INES OF DATA":PRINTSTRING$(32,"#
")
1210 FOR X=7 TO Z1 STEP 4
1222 IF PPOINT(X,Y)=0 THEN D=D+1
1223 IF PPOINT(X,Y+3)=0 THEN D=D
+2
1224 IF PPOINT(X,Y+6)=0 THEN D=D
+4
1225 IF PPOINT(X,Y+9)=0 THEN D=D
+8
1226 IF PPOINT(X,Y+12)=0 THEN D=
D+16
1227 IF PPOINT(X,Y+15)=0 THEN D=
D+32
1228 IF PPOINT(X,Y+18)=0 THEN D=
D+64
1230 N=N+1
1240 D$(N)=STR$(D)
1245 D=0
1250 NEXT X
1300 REM PRINT OUT DATA LINES
1305 LN$=STR$(LN+100)
1310 C$(LN/10)=RIGHT$(LN$,LEN(LN
$)-1)+" DATA ":N=1
1320 IF D$(N)="" THEN 1340 ELSE
C$(LN/10)=C$(LN/10)+RIGHT$(D$(N)
,LEN(D$(N))-1)+","
1330 N=N+1:IF N*4=>Z1 THEN1340 E
LSE 1320
1340 CL=LEN(C$(LN/10))
1342 C$(LN/10)=LEFT$(C$(LN/10),C
L-3)
1350 PRINT C$(LN/10)
1355 SOUND 240,2
1360 LN=LN+10
1365 IF LN/10 >Z3 THEN 1550
1380 N=0:Y=Y+21
1385 IF Y>Z2 THEN 1550
1390 GOTO 1210
1500 REM CLEAR ARRAY
1510 FOR R=1TO62:D$(R)="":NEXT
1520 RETURN
1550 REM END THE PROGRAM
1555 LN$=STR$(LN+100)
1565 POKE65494,0
1570 GOSUB 1900
1600 OPEN "O",#-1,NM$
1605 GOSUB 1700
1610 FOR R=1TO Z3
1620 PRINT #-1,C$(R)
1630 NEXT R
1636 C$=RIGHT$(LN$,LEN(LN$)-1)+"
 DATA 256":PRINT#-1,C$
1640 CLOSE#-1:SOUND200,5
1650 CLS:PRINT"       INSTRUCTIONS
 FOR USE":PRINTSTRING$(32,"-")
1660 PRINT:PRINT"  1>  REWIND TH
E TAPE"
1670 PRINT:PRINT"  2>  LOAD THE
NEW PROGRAM FILE"
1680 PRINT:PRINT"  3>  TURN ON T
HE PRINTER"
1690 PRINT:PRINT"  4>  TYPE 'RUN
' - PRESS ENTER"
1699 END:'END OF PROGRAM
1700 REM CREATE PROGRAM LINES
1710 C$="10 REM '"+NM$+"' CREATE
 BY 'DMDATGEN'":PRINT#-1,C$
1720 C$="20 PRINT#-2,CHR$(18):'
GRAPHICS MODE":PRINT#-1,C$
1730 C$="30 PRINT#-2,CHR$(27);CH
R$(14);:' ELONGATION":PRINT#-1,C
$

1740 C$="40 FOR X=1 TO"+STR$(INT
((Z1-5)/4))+":READ F":PRINT#-1,C
$
1750 C$="50 IF F<128 THEN F=F+12
8":PRINT#-1,C$
1760 C$="60 IF F=256 THEN 100":P
RINT#-1,C$
1770 C$="70 PRINT#-2,CHR$(28);CH
R$(2);CHR$(F);":PRINT#-1,C$
1780 C$="80 NEXT X":PRINT#-1,C$
1790 C$="90 PRINT#-2:GOTO 40":PR
INT#-1,C$
1795 C$="100 PRINT#-2,CHR$(27);C
HR$(15);CHR$(30)":PRINT#-1,C$
1800 RETURN
1900 REM GET THE NAME OF THE PRO
GRAM
1905 PRINT:PRINT"   1>  INSERT C
ASSETTE              2>  PRESS PL
AY AND RECORD"
1910 PRINT"   3>  ENTER THE DRAW
ING NAME":PRINT
1920 INPUT NM$
1930 RETURN
5000 REM TITLE OR HELP SCREEN
5005 POKE65495,0:GOSUB 1500
5010 FOR R=15 TO 57:READ I$:L$(R
)=I$:NEXT R
5020 PMODE4,5:SCREEN1,0:COLOR1
5030 GOTO 5090
5035 REM DRAW LETTERS
5040 FOR L=1TO LEN(W$):P$=MID$(W
$,L,1):IF P$=" " THEN 5070
5050 POKE178,0:LINE(X1,Y1-6)-(X1
+6,Y1),PSET,BF:DRAW"C5;"
5060 DRAW"BM"+STR$(X1)+","+STR$(
Y1)+";":DRAWL$(ASC(P$)-33)
5070 X1=X1+7:IF X1>248 THENX1=3:
Y1=Y1+10
5080 NEXT:RETURN
5090 REM DRAW SCREEN
5100 PCLS0:LINE(2,2)-(253,189),P
SET,B
5110 LINE(40,14)-(215,14),PSET
5120 X1=40:Y1=12:W$="DOT MATRIX
DATA GENERATOR                BY
ALAN BRIDGES":GOSUB 5040
5130 X1=10:Y1=40:W$="THIS GENERA
TOR WILL CREATE ON TAPE  A BASIC
 PROGRAM FILE WHICH WILL      PRI
NT THE IMAGE CREATED BY YOU ON
 A TANDY DMP 110 PRINTER":GOSUB
5040
5140 X1=95:Y1=80:W$="COMMANDS":L
INE(95,82)-(150,82),PSET:GOSUB50
40
5150 DRAW"BM50,130;NU22ND22NR22N
L22NE20NF20NG20NH20"
5160 DRAW"BM25,107;"+L$(39):DRAW
"BM48,105;"+L$(52):DRAW"BM72,107
;"+L$(36)
5170 DRAW"BM25,160;"+L$(38):DRAW
"BM48,162;"+L$(35):DRAW"BM72,160
;"+L$(37)
5180 DRAW"BM20,132;"+L$(43):DRAW
"BM79,132;"+L$(49)
5190 X1=18:Y1=92:W$="DIRECTIONS"
:GOSUB 5040
5200 X1=125:Y1=110:W$="PRESS CLE
AR P WHEN":GOSUB 5040:X1=125:Y1=
120:W$="FINSIHED DRAWING":GOSUB
5040
5210 X1=125:Y1=140:W$="PRESS C T
O TOGGLE":GOSUB 5040:X1=125:Y1=1
50:W$="COLOR ON OR OFF":GOSUB 50
40
5220 X1=50:Y1=186:W$="PRESS ENTE
R TO CONTINUE":GOSUB 5040
5222 POKE65494,0
5225 GOSUB 200
5230 RETURN
5240 DATAU6R4D6L4,R2NR2U6LG,NR4E
4UHL2G,BUFR2EUHNL2EUHL2G,BR3U6G3
R,BUFR2EU2HL2GU3R4,BU3ER2FD2GL2H
U4ER3,E4U2L4D,BUUENR2HUER2FDGFDG
L2H,BRR2EU4HL2GD2FR2E,,,,,,BU5ER
2FDGLDBD2D,,U4E2F2D2NL4D2,R3EUHE
UHL3RD3NR2D3,BE4BUHL2GD4F1R2E,R3
EU4HL2NLD6
5242 DATA R4UBU4UL3NLD3NR2D3
5250 DATARNRU3NR2U3NLR3D,BE2RNRD
NDGLHU4ER2D,U3NU3R4NU3D3,BRRNRU6
NLR,BUNUFREU5NLR,RU6NLBD3RE2UBD5
NDH2,R4UBGBL2U6NLR,U6F2E2D6,U6F4
NU4D2,BUU4ER2FD4GL2H,RNRU6NLR2FD
```

continued on p50

An exercise with floating point numbers.

DIRSQR

32K ECB
UTILITY

by Jim Jacobs

DIRSQR IS A machine language program to find square roots of floating point numbers on the CoCo. It is nearly ten times faster than the SQR function of Extended Basic. There is no such function in non Extended Basic.

ALGORITHM

This uses the normal way of finding square roots, arranged for binary arithmetic.

The bits of the number, S, whose square root, R, is required are treated in pairs extending on either side of the fractional point.

One bit of R is found for each pair, zeros are added on the right if required to give the correct number of bits in R.

Starting with the most significant on the left, each bit of the root is chosen so that the square of the partial root is just less than or equal to the number formed by the pairs of bits so far.

Now let b be the 'i'th bit,
'Si' be the number so far
and 'Ri' the partial root.
The condition is: Si >= Ri*Ri

The difference, Di=Si-Ri*Ri,
now Ri = 2Ri-1 + b,
x2 (left shift) for each new bit. Squaring,

Ri*Ri=4Ri-1*Ri-1+4Ri-1*b+b*b

... so Di=4Di-1+ bit pair of Si
- 4Ri-1*b - b*b

... and b can only be 0 or 1 which makes the process much easier to follow than the decimal case used at school.

This is illustrated in the flow diagram:-

```
            :
            # setup, exponent,
            : initial values, etc
 .--------->:
 :          # left shift S into D
 :          :  twice, x4
 :          : (bit pair)
 :    .no-<-# D>4R ?    )
 :    :  yes:           )
 :    0:   1# D=D-4R-1 )  bit
 :    :     :           )
 :    :     # R=2R+1    )
 :    '---->:
 :          # left shift R, using
 :          : the leading 1 to
 :          : count the bits
 :          :
 '---no-<-# penultimate bit?
         yes:
            # last bit
            :
            # round, store, etc.
            :
```

STORAGE

In CoCo, Basic numbers are stored in floating point form as an exponent byte followed by four mantissa bytes. The mantissa is a fraction between half and one.

It is normalized so that the most significant bit is always one. This is done by adjusting the value of the exponent.

Four bytes are equivalent to nine or ten significant figures in decimal.

The exponent is the power of two the fraction is to be multiplied by to give the number. In two's compliment form one byte gives a range of 2**+127 to 2**-129.

Actually the exponent is in excess 128 form, the most significant bit of the exponent is complimented so that positive exponents have a leading one and negative a zero.

With this notation the number zero is not represented, so the smallest range of numbers, with zero exponent, are taken to be zero. This has the advantage that a simple one byte zero test can be used. The range of numbers is slightly restricted by this but is ample for practical purposes, about 3x10**+38 to 3x10**-38.

Since the mantissa is normalized to always have a leading one, bit 7 of its first byte, this is replaced by a sign bit, zero for positive and one for negative.

For numerical operations two six byte floating point accumulators, FPAC1 and 2 are located at $004F and $005C, these store numbers as discussed above except that the sign bit is replaced by a one and the original first mantissa byte is stored in the sixth byte as the sign byte, this is more convenient for arithmetic.

LISTINGS

Listing 2 is a Basic program "SQRT", which, when RUN, pokes the routine into protected high memory where it can be utilized via the USR function. The machine language is held in data lines.

The poking takes a few seconds. Once done the "SQRT" program can be deleted or overwritten by other programs.

Listing 1 is the machine and assembly language version.

LISTING 1

* The program does not follow the flow diagram directly. Three loops are used. The first is for the first byte of the result, the next for the second and the last for the final two. This is

for speed, avoiding unnecessary bit shifting.

* The assembly language used here is to explain the machine language, and does not follow any particular assembler.

```
 * Difference in registers X , D
 * Register Y Byte Pointer,
 * Number S in FPAC1:-

  004F...  XPNT S0   S1   S2  S3
SIGN

 * Root R in FPAC2:-

  005C...      -  R0 R1  R2  R3 -
```

listing one

```
xx19 0853: LEFT ASL S3  Number
   B 0952:      ROL S2
   D 0951:      ROL S1     x2
   F 0950:      ROL S0
  21 39  :      RTS

   2 3436:START PSH S Registers
   4 5F4F:      CLR D      )
   6 1F01:      TFR D X    )Dif'ce
   8 5C  :      INC B
   9 DD5D:       ST D R0  )
   B D75F:       ST B R2  )  First
   D C604:       LD B '4' ) result
   F D760:       ST B R3  )     x4
  31 C611:       LD B '4s' Bit pair
   3 D75C:       ST B CNT   counter
   5 C650:       LD B S0  Number
   7 1F02:      TFR D Y  byte pntr
   9 8DDE:      BSR LEFT  Bit1

   B D64F:EXPNT  LD B XPNT
   D CB7F:      ADD B '-1'  1st shft
   F 57  :      ASR B    /2
  40 46  :      ROR A   save parity
   1 CB81:      ADD B '1'  restore
   3 D74F:       ST B XPNT
   5 5F  :      CLR B
   6 48  :      ASL A      parity?
   7 2404:      BCC LOOP1
   9 8DCE:      BSR LEFT  odd exp'nt
   B 59  :      ROL B    this bit
   C 5C  :      INC B     hidden 2-1

   D 68A4:LOOP1 ASL ,Y  First byte
   F 5949:      ROL D
  51 68A4:      ASL ,Y
   3 5949:      ROL D
   5 2604:      BNE ONE1
   7 D160:      CMP B R3
   9 2308:      BLS GOON1
   B 0C60: ONE1 INC R3
   D D060:      SUB B R3
   F 0C60:      INC R3
  61 8200:      SBC A '0'  carry

   3 8D64:GOON1 BSR TOUT
   5 24E6:      BCC LOOP1

   7 8D6B:LOOP2 BSR LFT22 2nd byte
 .8C0001:       CMP X '1'
   C 2405:      BCC ONE2
 .10935F:       CMP D R2
  71 230A:      BLS GOON2
   3 0C60: ONE2 INC R3
   5 935F:      SUB D R2
   7 0C60:      INC R3
   9 2402:      BCC GOON2
   B 301F:      LEA X-1,X  carry

   D 8D4A:GOON2 BSR TOUT
   F 095F:      ROL R2
  81 24E4:      BCC LOOP2
   3 8D4F:      BSR LFT22  bit 16
   5 8C  :      CMP X      skip2

   6 8D54:LOOP3 BSR LFT32  3rd &
   8 8D5D:      BSR BIT   last bytes
   A 0860:      ASL R3
   C 095F:      ROL R2
   E 095E:      ROL R1
  90 095D:      ROL R0
   2 24F2:      BCC LOOP3
   4 8D46:      BSR LFT32 32nd bit
   6 2402:      BCC HALF
   8 8D4D:      BSR BIT

   A 1A01: HALF OR CC '1'  leading 1
   C 065D:      ROR R0   Correct
   E 065E:      ROR R1   result
  A0 065F:      ROR R2     /2
   2 0660:      ROR R3

   4 9C5D:      CMP X R0  rounding
   6 2603:      BNE SKP
  10935F:       CMP D R2
   B 230E: SKP  BLS OUT      0

   D 0C60:RNDUP INC R3       1
   F 260A:      BNE OUT
  B1 0C5F:      INC R2
   3 2606:      BNE OUT
   5 0C5E:      INC R1
   7 2602:      BNE OUT
   9 0C5D:      INC R0

   B DC5F:  OUT  LD D R2
   D DD52:       ST D S2   result
   F DC5D:       LD D R0    into
  C1 DD50:       ST D S0   mantissa
   3 847F:      AND A '+ve'
   5 9754:       ST A SIGN

   7 35B6:  END PUL S P,Registers

   9 085C: TOUT ASL CNT
   B 2404:      BCC GOON   count
   D 0C5C:      INC CNT   bit pairs
   F 3121:      LEA Y 1,Y
  D1 0860: GOON ASL R3

   3 39  :      RTS

   4 8D00:LFT22 BSR 0     repeat
   6 68A4:      ASL ,Y  number
   8 5949:      ROL D     into
   A 2004:      BRA NOUT

   C 8D00:LFT32 BSR 0     repeat
   E 5849:      ASL D     dif'ce
  E0 1E01: NOUT EXG D X
   2 5949:      ROL D
   4 1E01: XOUT EXG D X    x2
   6 39  :      RTS

   7 9C5D:  BIT CMP X R0     D<S?
   9 25FB:      BCS RTN
   B 2205:      BHI ONE
 .10935F:       CMP D R2
  F0 23F4:      BLS RTN

   2 0C60:  ONE INC R3    +1/2
   4 935F:      SUB D R2
   6 0C60:      INC R3    +1/2
   8 1E01:      EXG D X      new
   A D25E:      SBC B R1    dif'ce
   C 925D:      SBC A R0
   E 20E4:      BRA XOUT
```

The Listing:

```
10 GOSUB12:CLEAR200,A-229:GOSUB1
2:A=A+1:PRINT"POKING":B=INT((A+9
)/256):POKE275,B:POKE276,A+9-B*2
56
11 FOR A=A TO A+230:READB:POKEA,
B:NEXT:END
12 A=PEEK(39)*256+PEEK(40):RETUR
N
20 DATA8,83,9,82,9,81,9,80,57,52
,54,95,79,31,1,92,221,93,215,95,
198,4,215,96,198,17,215,92,198,8
0,31,2,141,222,214,79,203,127,87
,70,203,129,215,79,95,72,36,4,14
1,206,89,92
21 DATA104,164,89,73,104,164,89,
73,38,4,209,96,35,8,12,96,208,96
,12,96,130,0,141,100,36,230,141,
107,140,0,1,36,5,16,147,95,35,10
,12,96,147,95,12,96,36,2,48,31,1
41,74,9,95,36,228,141,79,140
22 DATA141,84,141,93,8,96,9,95,9
,94,9,93,36,242,141,70,36,2,141,
77,26,1,6,93,6,94,6,95,6,96,156,
93,38,3,16,147,95,35,14
23 DATA12,96,38,10,12,95,38,6,12
,94,38,2,12,93,220,95,221,82,220
,93,221,80,132,127,151,84,53,182
24 DATA8,92,36,4,12,92,49,33,8,9
6,57,141,0,104,164,89,73,32,4,14
1,0,88,73,30,1,89,73,32,1,57,156
,93,37,251,34,5,16,147,95,35,244
,12,96,147,95,12,96,30,1,210,94,
146,93,32,228
```

LINKING

If the address of the square root routine is held in the USR vector at address $0113-4 (275-6 decimal), a statement like Y=USR(X) in a Basic program will give the square root of the variable X as the variable Y.

This is because the routine takes the floating point number in FPAC1 and replaces it by its square root. Basic automatically transfers the numbers between the accumulators and the variable storage area.

No tests are provided for zero or negative number inputs, these are up to the user.

The routine can be located anywhere convenient in memory. The addresses given in the listing are notional only.

TIMING

I worked the routine out because I needed a square root routine for another program and the Basic routine given in the Color Computer manual seemed slow. I ran the following programs to check the timing:

```
 10 FORI=1TO10000:NEXT:END
Loop timing: 10secs - 1mS each

 15 FORI=1TO10000:Y=I:NEXT:END
Vbl Loop: 23secs - 2.3mS each

 20  K=9:FOR I=1TO10000:Y=K+I
  :NEXT:END
Addition: 36secs - 3.6mS each

 30 K=9:FOR I=1TO10000:Y=K-I
  :NEXT:END
Subtraction:  37secs  -   3.7mS
each.

 40 K=9:FORI=1TO1000:Y=K*I
  :NEXT:END
Multiply: 5secs - 5mS each

 50 K=9:FORI=1TO1000:Y=K/I
  :NEXT:END
Divide: 7secs - 7mS each

 60 FORI=1TO100:T=I/2
 65 R=T:T=(I/R+R)/2:IFT<>R
  THEN65
 70 NEXT:END
Basic SQRT: 10secs - 100mS each

 80 FORI=1TO1000:Y=USR(I):NEXT
  :END
M/L SQRT: 8secs - 8mS each
```

... and finally using Extended Colour Basic,

```
 90 FORI=1TO100:Y=SQR(I):NEXT
  :END
ECB SQRT: 7secs - 70mS each
```

Showing that the routine is almost ten times faster than the ECB routine and that it compares well with the other arithmetic routines.

REFERENCES

I had some difficulty finding out about square roots, the best reference I could find was ...

"Mathematical Microprocessor Software: a SQRT(X) Comparison" by Michael Andrews, IEEE Micro, May 1982, p63.

Others are:-

Yaohan Chu "Digital Computer Design Fundamentals", McGraw Hill, New York, 1962.

Ivan Flores "The Logic of Computer Arithmetic", Prentice-Hall, Inglewood Cliffs, 1962.

■

Hint ...

Double spacing LLIST

Now you can put an extra line between program lines when LLISTing them (cassette users only).

Type in:

```
POKE 383,126:POKE384,185
:POKE 385,88
```

To turn it on type: POKE383,126
To turn it off type: POKE383,57

John Carmichael

Hint ...

Print-out your directory to the printer

By typing in:

```
POKE111,254:DIR DR:
PRINT#-2,"Free = "
;FREE(DR)
```

... you will be able to get a hard copy of your disk directory. Replace 'DR' with the drive number you want printed.

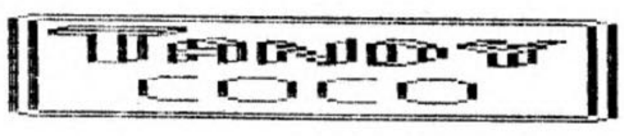
continued from p47

```
GL2,BUU4ER2FD4GDRBHL2H,U6R3FDGNL
2F2D,BUFR2EUHL2HUER2F,BR2U6NL2R2
,BUNU5FRERNDU5
5260 DATABU3NU3FDFEUEU3,NU6E2F2U
6,UE4UBL4DF4D,BRRNRU3H2UBR4DG2,B
U5UR4DG4DR4U,,,,,
```

The Listing:

```
0 GOTO10
3 SAVE"271A:3":END'8
10 REM 'TANDY' CREATE BY 'DMDATG
EN'
20 PRINT#-2,CHR$(18):' GRAPHICS
MODE
30 PRINT#-2,CHR$(27);CHR$(14);:'
 ELONGATION
40 FOR X=1 TO 61:READ F
50 IF F<128 THEN F=F+128
60 IF F=256 THEN 100
70 PRINT#-2,CHR$(28);CHR$(2);CHR
$(F);
80 NEXT X
90 PRINT#-2:GOTO 40
100 PRINT#-2,CHR$(27);CHR$(15);C
HR$(30)
110 DATA 124,2,121,5,5,69,37,37,
37,37,37,37,37,37,37,37,101,101,
5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,
5,5,5,5,5,5,5,5,121,2,124
120 DATA 127,0,127,0,7,5,5,5,125
,1,1,125,5,125,5,7,66,33,112,16,
40,72,80,96,0,96,48,80,96,0,0,0,
96,16,120,0,96,48,104,40,72,16,9
6,0,0,96,48,40,88,24,96,32,48,10
4,24,8,0,0,127,0,127
130 DATA 127,0,127,0,0,0,0,0,127
,0,0,127,0,127,0,0,127,20,115,18
,18,127,0,127,0,127,0,125,6,27,1
08,16,127,0,127,0,127,0,127,0,0,
127,32,31,0,0,1,2,124,1,124,2,12
7,0,0,0,0,0,127,0,127
140 DATA 127,0,127,0,0,0,0,0,3,2
,2,3,1,0,0,0,3,1,0,0,0,3,1,0,0,3
,1,0,0,0,0,3,3,1,0,0,3,3,2,2,1,0
,0,0,0,0,0,0,3,2,3,1,0,0,0,0,0,0
,127,0,127
150 DATA 127,0,127,0,0,0,0,0,0,0
,0,0,0,120,4,2,2,2,2,0,0,120,4,2
,2,2,2,4,120,0,0,120,4,2,2,2,2,0
,0,120,4,2,2,2,2,4,120,0,0,0,0,0
,0,0,0,0,0,0,127,0,127
160 DATA 127,0,127,0,0,0,0,0,0,0
,0,0,0,15,16,32,32,32,32,0,0,15,
16,32,32,32,32,16,15,0,0,15,16,3
2,32,32,32,0,0,15,16,32,32,32,32
,16,15,0,0,0,0,0,0,0,0,0,0,0,127
,0,127
170 DATA 31,32,79,80,80,80,80,80
,80,80,80,80,80,80,80,80,80,80,8
0,80,80,80,80,80,80,80,80,80,80,
80,80,80,80,80,80,80,80,80,80,80
,80,80,80,80,80,80,80,80,80,80,8
0,80,80,80,80,80,80,80,79,32,31
180 DATA 256
```

■



WORKSHEET 80

16K CoCo1/2/3
UTILITY

by Harry Hoffmann

TODAY I AM SUBMITTING another and last worksheet program. I found these programs rather indispensable when I am programming.

To run this program, the printer must be in the serial I/O socket, otherwise the computer will kick you if it isn't turned on.

There are no fancy menus or any other things to make this program more interesting - it works just like a kettel: you want hot water, fill it up and turn it on.

You want a worksheet, load the program and RUN it - simple, isn't it?

Please let me now expose on my past programs. You may remember the first two programs I submitted. They where about the weather - look at the February issue of CoCo magazine, page 26 and 29.

I am now working on a series of weather programs like rainfall and temperature - one will be a daily data input program for both, one a printout for the DMP-105 printer, one a display program for the CoCo 3, and one a display program for the CoCo 2, which can also be dumped to the printer by using Craig Stewart's "Screen Dump" or any other.

The last 3 (6) aforementioned programs will make use of the files created in the first 1 (2) program(s). I am also trying to work out a program for the TP-10 printer using the same files.

May I give a little hint to low memory computer users? A quick way of saving quite a lot of memory is this, type:

```
RENUM0,,1
```

... by doing it to the program I am submitting today it will save almost 200 bytes of memory. That's a lot for such a short program.

Another hint: I have about two full disks with useful subroutines. So far when I want to merge one, I had to find out what line number they start on.

This can take up a lot of programming time. My solution to this problem is this:

I am now re-saving all subroutines starting with line number 3000 incremented by 1 - so if want to merge in a subroutine, I renumber the program I am working on, renumbering from the point I wish to merge into [4000,x,1] and merge.

No more problems with line numbers.

I am sorry, but I have to leave you with this, for now.

The Listing:

```
0 GOTO10
1 *********WORKSHEET 80*********
2 ******BY HARRY HOFFMANN*******
3 **CROWS NEST CARAVAN PARK*****
4 **CROWS NEST QLD 4355*********
5 **WRITTEN ON COCO 3***********
6 SAVE"296:3":END'8
7 END
10 CLS7:GOSUB890
20 PRINT@233," CROWS NEST ";:PRI
NT@416,""
30 GOSUB50:GOTO90
40 '***************************
    ** TITLESONG BY HARRY     **
    ** HOFFMANN CROWS NEST    **
    ***************************

50 FORX=1TO3:PLAY"T6O3V30L4CV20L
8CV10CL8V30CL16V20CV10CL8V30DL16
V20DV10L8V30EL16V20EV10EL4V30CL8
V16CV10CL4V30FL8V20FV10FL4V30EL8
V20EV10EL4.V30DL8.V20DV10D":NEXT
60 FORX=1TO3:PLAY"V30L8GV20L16GV
10G":NEXT:PLAY"V30L4.GL8.V20GV10
GV30L8FV20L16FV10FV30L4EV20L8EV1
0EL4V30DL8V20DV10DL2V30GL4V20GV1
0G"
70 PLAY"V30L8GV20L16GV10GV30L4.G
V20L8.GV10GV30L8FV20L16FV10FV30L
4EV20L8EV10EL4V30DL8V20DV10DL1V3
0CL4V20CV15CL2V10CV5C"
80 RETURN
85 CLS4
90 IF PEEK(65314)/2<>INT(PEEK(65
314)/2)THEN 95:'** CHECK FOR PRI
NTER **
93 GOTO100
95 GOSUB890:PRINT@229," PRINTER
NOT ON LINE ";:GOTO90
100 A$=CHR$(27)
110 B$=A$+CHR$(20):'** CONDENSED
120 C$=A$+CHR$(21):'** SET CR=CR
130 D$=A$+CHR$(56):'** 3/4 LFEED
140 E$=A$+CHR$(22):'** CR=CR+LF
150 F$=A$+CHR$(16)+CHR$(0)+CHR$(
69):'** SET CARRIAGE
160 G$=CHR$(13):'** CR
170 H$=CHR$(10):'** LF+CR
180 I$=CHR$(18):'** GRAPHICS MOD
E
190 J$=CHR$(30):'** CHARACTER MD
200 K$=STRINGS$(8,128)+CHR$(255)
210 L$=STRINGS$(8,192)+CHR$(255)
220 M$=A$+CHR$(16)+CHR$(0)+CHR$(
64)
230 N$=A$+CHR$(28):'** HALF LINE
 FEED
240 O$=A$+CHR$(90)+CHR$(7):'** F
ULL LF
250 P$=A$+CHR$(16)+CHR$(0)+CHR$(
63)
260 Q$=CHR$(255)
270 R$=CHR$(14)'** END UNDERLINE
 OR BIT FOR START ELONGATION
280 S$=CHR$(15):'** START UNDERL
INE OR BIT FOR END ELONGATION
290 T$=A$+R$:'** START ELONGATIO
N
300 U$=A$+S$:'** END ELONGATION
310 V$=A$+CHR$(19):'** STANDARD
CHARACTER MODE
320 CLS3:GOSUB890:PRINT@227,"PLE
ASE WAIT, I'M PRINTING";:PRINT@4
16,""
330 POKE150,18:'** 2400 BAUD **
340 PRINT#-2,J$;V$;O$
350 PRINT#-2,TAB(23);"";:PRINT#-
2,T$;S$;"WORKSHEET 24 X 80";R$;U
$;V$;H$
360 PRINT#-2,TAB(24),"BY HARRY H
OFFMANN";H$;H$;H$
370 CMP
380 PRINT#-2,B$C$
390 FOR A=1TO40
400 READ X
410 PRINT#-2,TAB(10);X;
420 NEXT A
430 DATA ,,,,,1,1,1,1,1,2,2,2,2,
2,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,
6,6,6,6,6,7,7,7,7,7
440 PRINT#-2,C$;
```

```
450 PRINT#-2,G$
460 PRINT#-2,F$;
470 RESTORE
480 FOR A=1TO40
490 READ X
500 PRINT#-2,X;
510 NEXT A
520 PRINT#-2,D$;H$
530 FOR Y=1TO8
540 FOR X=0TO9STEP2
550 PRINT#-2,TAB(10);X;
560 NEXT X
570 NEXTY
580 PRINT#-2,C$;
590 PRINT#-2,G$
600 PRINT#-2,F$;
610 FOR Y=1TO8
620 FOR X=1TO9STEP2
630 PRINT#-2,X;
640 NEXT X,Y
650 PRINT#-2,M$;I$;
660 FOR X=1TO 80
670 PRINT#-2,K$;
680 NEXT
690 PRINT#-2,N$;H$;M$;
700 FOR X=1TO80
710 PRINT#-2,L$;
720 NEXT
730 FOR X=0TO23
740 PRINT#-2,J$;O$
750 PRINT#-2,TAB(5);"";:PRINT#-2
,USING"##";X;
760 PRINT#-2,I$;P$;Q$;:FORY=1TO8
0
770 PRINT#-2,K$;
780 NEXT Y
790 PRINT#-2,N$;H$;P$;Q$;
800 FORY=1TO80
810 PRINT#-2,L$;
820 NEXT Y
830 NEXTX
840 PRINT#-2,J$;E$;D$
850 CLS7:GOSUB890:PRINT@235,CHR$
(128)"finish"CHR$(128);:PRINT@41
6,"":GOSUB50
860 END
880 ***************************
    *        ATTENTION!        *
    * WATCH OUT FOR SEMICOLON *
    * OMITTING OR ADDING ANY  *
    * COULD MESS UP THE WHOLE *
    * PROGRAM ;;;;;;;;;;;;;;; *
    ***************************

890 PRINT@32," WORKSHEET 80 BY H
ARRY HOFFMANN ";
900 RETURN
1000 **************************
     * THIS PROGRAM WAS WRITTEN*
     * BY HARRY HOFFMANN       *
     * CROWS NEST CARAVAN PARK *
     * CROWS NEST QLD 4355     *
     ***************************

1010 ##########################
     # ALL COMMENT LINES WITH  #
     # OR WITHOUT <'> CAN BE   #
     # OMITTED. THIS PROGRAM   #
     # SHOULD ALSO WORK ON A   #
     # COCO 2  (+DMP 105 )     #
     ###########################
```

■

COMMAND CHANGER

continued from p35

```
460 PRINT#-1,IN$(PR):NEXT
470 CLOSE#-1:GOTO500
500 CSAVEM"1",&H8183,&H81EF.0:CS
AVEM"2",&H821E,&H8256,0:CSAVEM"3
",&HAA66,&HAB66,0
530 PRINT:PRINT"TO OPERATE COMMA
ND SET,LOAD       BASIC PROGRAM,KE
EP PLAY DOWN ANDRUN THE PROGRAM"
:PRINT:PRINT"PRESS A KEY":EXEC44
539
540 GOTO145
1000 DATA43670,RESTORE,43677,RET
URN,43683,STOP,43687,POKE,43691,
CONT,43695,LIST,43699,CLEAR,4370
4,NEW,43707,CLOAD,43712,CSAVE,43
717,OPEN,43721,CLOSE,43726,LLIST
,43731,SET,43734,RESET,43739,CLS
,43742,MOTOR,43747,SOUND,43752,A
UDIO,43757,EXEC
1010 DATA43761,SKIPF,43770,TO,43
775,THEN,43814,RND,43817,SIN,438
20,PEEK,43824,LEN,43827,STR$,438
31,VAL,43834,ASC,43837,CHR$,4384
1,EOF,43782,STEP,43794,AND,43802
,SGN,43805,INT,43808,ABS,43811,U
SR,43635,IF,43637,DATA,43641,PRI
NT,43646,ON,43648,INPUT
1020 DATA43653,END,43656,NEXT,43
660,DIM,43663,READ,33349,TIMER,3
3354,PPOINT,33360,STRING$,43622,
FOR,43625,GO,43627,REM,43630,',4
3631,ELSE,33158,EDIT,33162,TRON,
33166,TROFF,33174,LET,33177,LINE
,33181,PCLS,33185,PSET,33189,PRE
SET,33195,SCREEN,33201
1030 DATA PCLEAR,33207,COLOR,332
12,CIRCLE,33218,PAINT,33223,GET,
33226,PUT,33229,DRAW,33233,PCOPY
,33239,PMODE,43844,JOYSTK,43850,
LEFT$,43855,RIGHT$,43861,MID$,43
865,POINT,43870,INKEY$,43876,MEM
,33243,PLAY,33247,DLOAD,33252,RE
NUM,33259,USING,33310,ATN
1040 DATA33313,COS,33316,TAN,333
19,EXP,33322,FIX,33325,LOG,33328
,POS,33331,SQR,33334,HEX$,33338,
VARPTR,33344,INSTR
```

■

DISKDUMP

continued from p37

```
810 FORA=1TO5:SOUND1,1:SOUND20,1
:NEXTA
820 GOTO 630
830 NV$=A$
840 POKE PV-2,191
850 POKE PV-1,191,
860 A$=INKEY$:IFA$=""THEN860
870 IF A$<"0" OR A$>"F"THEN860
880 IF A$>"9" AND A$<"A"THEN 860
890 NV$=NV$+A$
900 IF FS=1THEN O1$(MX)=CHR$(VAL
("&H"+NV$))
910 IF FS=2THEN O2$(MX)=CHR$(VAL
("&H"+NV$))
920 PRINT@((PV-2)-1024),NV$;
930 GOTO660
940 S1$="":S2$=""
950 FORA=1TO128
960 S1$=S1$+CHR$(VAL("&H"+O1$(A)
))
970 NEXT
980 FORA=1TO128
990 S2$=S2$+CHR$(VAL("&H"+O2$(A)
))
1000 NEXT
1010 CLS4
1020 PRINT@224,"PREPARING TO WRI
TE BACK TO DISK!";:PRINT@288,"PL
EASE CONFIRM (Y/N)";
1030 FORA=1TO5:SOUND1,3:SOUND100
,3:NEXTA
1040 A$=INKEY$:IFA$=""THEN1040
1050 IF A$<>"Y"THEN GOTO120
1060 PRINT:PRINT"NOW WRITING....
."
1070 DSKO$0,TN,SN,S1$,S2$
1080 GOTO120
```

■

COMPETITIONS

Yes, we're going to run the following competitions again this year:

- Games, for CoCo 1/2 and CoCo 3
- Utility
- Music
- Adventure
- Business
- Educational
- Graphics
- Applications

Quite a lot, eh?

Like this year, the winners will be announced at Conf'88 September next year. If you haven't read what the winners got this year, then read! One could get envious!

If you would like to win something, then send in your program or article! Who knows, it could be you in the winners lis next year! You might end up with a very nice prize for your efforts!

Especially for the young one we have...

ADVENTURE + II

32K ECB
GAME

by Sean Lowe

ADVENTURE PLUS II IS the sequel to "Adventure +" which was printed during mid-1985. Again, it is directed towards the younger adventurers (aged between 10 and 16), but of course any age will enjoy playing it.

This adventure was intended for the 16k users, but I got carried away and only those with more than 16k can use it (ie, those with 32 or 64k).

Instructions

Your wealthy uncle has died and left you his mansion. It has a garden, an attic and a basement.

The object of this adventure is to retrieve 11 of his most prized possesions and return to your starting position.

Some objects can be picked up but you need a rope and a key to access the rest.

There are three transport rooms placed about the mansion. If you walk into one of these rooms it will place you close by your starting position.

More instructions are inside the program.

Happy hunting!!

The Listing:

```
0 '* ADVENTURE PLUS II *
1 '*   BY SEAN LOWE.   *
2 '*SEAL SOFTWARE INC. *
3 GOTO5
4 SAVE"297:3":END'7
5 POKE65495,0
6 CLS
10 POKE280,PEEK(275)
11 GOSUB20000
12 FORT=1TO17
13 READRD$:FOR RT=134TO151-T:PRI
NT@RT," ";RD$;:FOR RR=1TO55:NEXT
RR:NEXTRT
14 NEXTT
15 FORT=1TO1200:NEXT
25 CLSRND(9)-1:PRINT@106,"INSTRU
CTIONS";:PRINT@141,"[Y/N].";
26 I$=INKEY$:IFI$="Y"THEN30 ELSE
IFI$="N"THEN50 ELSEIFI$<>""THEN
26
30 CLS:PRINT"WELCOME TO YOUR UNC
LE'S MANSION.IT WAS A GREAT TRAD
GEDY WHEN    YOUR UNCLE DIED, BU
T BECAUSE YOUWERE HIS FAVORITE P
ERSON, THIS  MANSION WAS LEFT TO
 YOU IN HIS  WILL."
31 PRINT" HE WAS A MAN OF FEW PO
SSESIONS BUT EVERYTHING HE OWNED
 WAS OF  GREAT VALUE.  AS THE NE
W OWNER  OF THE MANSION, YOU WIL
L HAVE   TO SEEK OUT HIS PRICELE
SS       TREASURES.
32 PRINT@480,"PRESS [ENTER]";:IF
INKEY$=CHR$(13) THEN33 ELSEGOTO3
2
33 CLS:PRINT"YOU WILL START AT T
HE FRONT DOOROF THE MANSION.  YO
U CAN MOVE   ABOUT BY PRESSING T
HE            DIRECTIONAL KEYS.
34 PRINT:PRINT"F=FORWARD","B=BAC
KWARDS":PRINT"L=LEFT","R=RIGHT":
PRINT"U=UP","D=DOWN":PRINT"G=GET
 OBJECT.
35 PRINT@480,"PRESS [ENTER]";:IF
INKEY$=CHR$(13)THEN36 ELSEGOTO35
36 CLS:PRINT"THERE ARE 4 MAIN AR
EAS OF YOUR  MANSION.  THE GARDE
N, THE HOUSE,THE BASEMENT AND TH
E ATTIC.":PRINT:PRINT"ALL TOGETH
ER THERE ARE 11        TREASURES
TO BE FOUND.
37 PRINT"SOME CAN BE FOUND BY WA
LKING    AROUND AND PICKING THEM
 UP, BUT IN ORDER TO FIND THE RE
ST YOU   NEED 2 OBJECTS TO HELP
YOU FIND THEM.  ONE IS THE ROPE
IN ORDER TO CLIMB AND THE OTHER
IS A KEY IN ORDER TO OPEN LOCKED
 DOORS.
38 PRINT@480,"PRESS [ENTER]";:IF
INKEY$=CHR$(13) THEN40 ELSEGOTO3
8
40 CLS:PRINT"WHEN YOU THINK YOU
HAVE ALL THE TREASURES, GET BACK
 TO THE FRONTDOOR IN ORDER TO WI
N THE GAME.  TO MAKE LIFE EASIER
, THERE ARE  3 TRANSPORT ROOMS S
PREAD AROUND
41 PRINT"THE MANSION AND THESE W
ILL TAKE YOU TO WITHIN 2 MOVES A
WAY FROM THE FRONT DOOR.
42 PRINT:PRINT"HAPPY HUNTING.
49 PRINT@480,"PRESS [ENTER]";:IF
INKEY$=CHR$(13)THEN50 ELSEGOTO49
50 GOTO850
100 GOSUB15100
110 GOSUB12000
130 ON I GOTO400,100,350,150,100
,100,100
150 GOSUB15100
160 IFT1=1 THEN170
165 CIRCLE(82,188),4:DRAW"BM82,1
84 H2E2F2G2U4G2R4"
170 GOSUB12000
180 IFI=7 THENT1=1
195 ON I GOTO420,150,100,200,150
,150,150
200 GOSUB15100
210 GOSUB12000
230 ON I GOTO450,200,150,250,200
,200,200
250 GOSUB15100
260 GOSUB12000
280 ON I GOTO480,250,200,300,250
,250,250
300 GOSUB15100
310 GOSUB12000
330 ON I GOTO500,300,250,350,300
,300,300
350 GOSUB15100
360 GOSUB12000
380 ON I GOTO550,350,300,100,350
,350,350
400 GOSUB15100
410 GOSUB12000
415 ON I GOTO570,100,550,420,400
,400,400
420 GOSUB15100
430 GOSUB12000
440 ON I GOTO600,150,400,450,420
,420,420
450 GOSUB15100
460 GOSUB12000
462 ONI GOTO620,200,420,480,460,
460,460
480 GOSUB15100
485 GOSUB12000
495 ON I GOTO640,250,450,500,480
,480,480
500 GOSUB15100
505 IFR1=1 THEN542
510 DRAW"BM151,129 D47R2U47L2"
520 FORT=132TO172 STEP4:PSET(152
,T):NEXT
542 GOSUB12000
544 IFI=7 THENR1=1
546 ON I GOTO500,300,480,550,542
,542,500
550 GOSUB15100
560 GOSUB12000
565 ON I GOTO680,350,500,400,550
,550,550
570 GOSUB15100
580 GOSUB12000
```

```
590 ON I GOTO700,400,680,600,570
,570,570
600 GOSUB15100
605 GOSUB12000
615 ON I GOTO750,420,570,620,600
,600,600
620 GOSUB15100
625 GOSUB12000
635 ON I GOTO800,450,600,640,620
,620,620
640 GOSUB15100
645 GOSUB12000
655 ON I GOTO850,480,620,660,640
,640,640
660 GOSUB15100
665 GOSUB12000
675 ON I GOTO900,660,640,680,660
,660,660
680 GOSUB15100
685 GOSUB12000
695 ON I GOTO950,550,660,570,680
,680,680
700 PCLS:GOSUB15200
742 GOSUB12000
746 ON I GOTO742,570,950,750,742
,742,742
750 GOSUB15100
752 GOSUB15200
792 GOSUB12000
793 IFI=5 AND R1=0 THENI=1
796 ON I GOTO792,600,700,800,455
0,792,792
800 PCLS:GOSUB15200
840 GOSUB12000
845 ON I GOTO840,620,750,850,840
,840,840
850 PMODE4,1:SCREEN1,1
851 PCLS:GOSUB15200
852 DRAW"BM104,108 R40D76L40U76"
:CIRCLE(112,148),4
889 IFT1+T2+T3+T4+T5+T6+T7+T8+T9
+TQ+TZ=11 THEN25000
890 GOSUB12000
895 ON I GOTO1000,640,800,900,85
0,850,850
900 PCLS:GOSUB15200
940 GOSUB12000
945 ON I GOTO940,660,850,950,940
,940,940
950 PCLS:GOSUB15200
990 GOSUB12000
995 ON I GOTO990,680,900,700,990
,990,990
1000 GOSUB15000
1010 DRAWF$+L$+R$
1015 DRAWP$
1020 GOSUB12000
1040 ON I GOTO1650,850,1400,1050
,1020,1020,1020
1050 GOSUB15000
1060 DRAWL$+R$
1070 GOSUB12000
1090 ON I GOTO1070,1070,1000,110
0,1070,1070,1070
1100 GOSUB15000
1110 DRAWF$+L$+R$
1120 GOSUB12000
1140 ON I GOTO1750,1120,1050,115
0,1120,1120,1120
1150 GOSUB15000
1160 DRAWF$+L$+R$
1165 DRAWC$
1170 GOSUB12000
1190 ON I GOTO1800,1170,1100,120
0,1170,1170,1170
1200 GOSUB15000
1210 DRAWL$
1215 DRAWTA$
1220 DRAWC$
1230 GOSUB12000
1240 ON I GOTO1230,1230,1150,123
0,1230,1230,1230
1250 GOSUB15000
1255 DRAWF$+D$
1260 GOSUB12000
1270 ON I GOTO1450,1250,1260,126
0,1260,6900,1260
1300 GOSUB15000
1305 DRAWF$+L$+R$
1310 GOSUB12000
1330 ON I GOTO1500,1310,1300,135
0,1310,1310,1310
1350 GOSUB15000
1360 DRAWF$+L$+R$
1370 GOSUB12000
1390 ON I GOTO1550,1370,1300,140
0,1370,1370,1370
1400 GOSUB15000
1410 DRAWF$+L$+R$
1420 GOSUB12000
1440 ON I GOTO1600,1420,1350,100
0,1420,1420,1420
1450 GOSUB15000
1460 DRAWR$
1465 DRAWC$
1470 GOSUB12000
1490 ON I GOTO1470,1250,1470,150
0,1470,1470,1470
1500 GOSUB15000
1510 DRAWF$+L$
1515 DRAWTA$
1520 GOSUB12000
1540 ON I GOTO1950,1300,1450,152
0,1520,1520,1520
1550 GOSUB15000
1560 DRAWF$+R$
1570 GOSUB12000
1590 ON I GOTO2000,1350,1570,160
0,1570,1570,1570
1600 GOSUB15000
1610 DRAWL$+R$
1620 GOSUB12000
1640 ON I GOTO1620,1400,1550,165
0,1620,1620,1620
1650 GOSUB15000
1660 DRAWL$+R$
1670 GOSUB12000
1690 ON I GOTO1670,1000,1600,170
0,1670,1670,1670
1700 GOSUB15000
1710 DRAWL$+R$
1720 GOSUB12000
1740 ON I GOTO1700,1720,1650,175
0,1720,1720,1720
1750 GOSUB15000
1760 DRAWF$+L$
1765 DRAWC$
1770 GOSUB12000
1790 ON I GOTO2200,1100,1700,177
0,1770,1770,1770
1800 GOSUB15000
1810 DRAWF$+R$
1820 GOSUB12000
1840 ON I GOTO2250,1150,1820,185
0,1820,1820,1820
1850 GOSUB15000
1855 DRAWF$+L$
1860 DRAWTA$
1870 GOSUB12000
1890 ON I GOTO2300,1870,1800,187
0,1870,1870,1870
1900 GOSUB15000
1905 DRAWR$+D$
1907 CIRCLE(225,120),3
1910 GOSUB12000
1930 ON I GOTO1910,1910,1910,195
0,1910,7700,1910
1950 GOSUB15000
1955 DRAWF$+L$+R$
1960 CIRCLE(17,116),3
1970 GOSUB12000
1975 IFI=3 AND K=0 THENI=7
1990 ON I GOTO2400,1500,1900,200
0,1970,1970,1970
2000 GOSUB15000
2010 DRAWL$+R$
2015 DRAWP$
2020 GOSUB12000
2040 ON I GOTO2020,1550,1950,205
0,2020,2020,2020
2050 GOSUB15000
2060 DRAWF$+L$
2070 GOSUB12000
2090 ON I GOTO2480,2070,2000,207
0,2070,2070,2070
2100 GOSUB15000
2110 DRAWR$+U$
2120 GOSUB12000
2140 ON I GOTO2120,2120,2120,215
0,4650,2120,2120
2150 GOSUB15000
2160 DRAWF$+L$
2165 DRAWP$
2170 GOSUB12000
2190 ON I GOTO2550,2170,2100,217
0,2170,2170,2170
2200 GOSUB15000
2210 DRAWF$
2220 GOSUB12000
2240 ON I GOTO2600,1750,2220,222
0,2220,2220,2220
2250 GOSUB15000
2255 DRAWF$+R$
2260 CIRCLE(112,116),3
2270 GOSUB12000
2275 IFI=1 AND K=0 THENI=3
2290 ON I GOTO2650,1800,2270,230
0,2270,2270,2270
2300 GOSUB15000
2310 DRAWL$
2320 GOSUB12000
2340 ON I GOTO2320,1850,2250,232
0,2320,2320,2320
2350 GOSUB15000
2360 DRAWF$+R$
2370 GOSUB12000
2390 ON I GOTO2750,2370,2370,240
0,2370,2370,2370
2400 GOSUB15000
2410 DRAWF$+L$
2415 DRAWC$
2420 GOSUB12000
2440 ON I GOTO2800,1950,2350,242
0,2420,2420,2420
2450 GOSUB15000
2455 FORT=2TO96 STEP4
```

```
2460 LINE(T,T)-(256-T,192-T),PSE
T,B
2465 NEXT
2470 X=RND(3):ON X GOTO1000,1400
,1050
2480 GOSUB15000
2484 DRAWF$+L$+R$
2488 GOSUB12000
2495 ON I GOTO2900,2050,2450,250
0,2488,2488,2488
2500 GOSUB15000
2510 DRAWL$+R$
2520 GOSUB12000
2540 ON I GOTO2520,2520,2480,255
0,2520,2520,2520
2550 GOSUB15000
2560 DRAWL$+R$
2570 GOSUB12000
2590 ON I GOTO2570,2150,2500,260
0,2570,2570,2570
2600 GOSUB15000
2610 DRAWL$
2620 GOSUB12000
2640 ON I GOTO2620,2200,2550,262
0,2620,2620,2620
2650 GOSUB15000
2660 DRAWF$
2670 GOSUB12000
2690 ON I GOTO3100,2250,2670,267
0,2670,2670,2670
2700 GOSUB15000
2705 DRAWF$
2710 IFT2=1 THEN2720
2715 DRAW"BM91,183 L10E11R4G8NL4
F3E8H3;BM104,183 L10E11R4G8NL4F3
E8H3;BM117,183 L10E11R4G8NL4F3E8
H3"
2720 GOSUB12000
2735 IFI=7 THENT2=1
2740 ON I GOTO3150,2720,2720,272
0,2720,2720,2700
2750 GOSUB15000
2760 DRAWR$
2765 DRAWP$
2770 GOSUB12000
2790 ON I GOTO2770,2350,2770,280
0,2770,2770,2770
2800 GOSUB15000
2810 DRAWF$+L$+R$
2820 GOSUB12000
2840 ON I GOTO3250,2400,2750,285
0,2820,2820,2820
2850 GOSUB15000
2860 DRAWF$+L$+R$
2870 GOSUB12000
2890 ON I GOTO3300,2870,2800,290
0,2870,2870,2870
2900 GOSUB15000
2910 DRAWF$+L$
2915 DRAWTA$
2920 GOSUB12000
2940 ON I GOTO3350,2480,2850,292
0,2920,2920,2920
2950 GOSUB15000
2955 DRAWF$
2960 IFT3=1 THEN2970
2965 DRAW"BM176,160 F4G4D13F3R14
E3U13H4E4L2G4L8H4L2;BM188,170 L4
D4R4D4L4R2D2U12"
2970 GOSUB12000
2980 IFI=7 THENT3=1
2990 ON I GOTO3400,2970,2970,297
0,2970,2970,2950
3000 GOSUB15000
3010 DRAWF$
3020 GOSUB12000
3040 ON I GOTO3450,3020,3020,302
0,3020,3020,3020
3050 GOSUB15000
3060 DRAWR$+P$
3070 GOSUB12000
3090 ON I GOTO3070,3070,3070,310
0,3070,3070,3070
3100 GOSUB15000
3110 DRAWF$+L$
3120 GOSUB12000
3140 ON I GOTO3550,2650,3050,312
0,3120,3120,3120
3150 GOSUB15000
3160 DRAWF$
3165 DRAWTA$
3170 GOSUB12000
3190 ON I GOTO3600,2700,3170,317
0,3170,3170,3170
3200 GOSUB15000
3210 DRAWF$+R$
3220 GOSUB12000
3240 ON I GOTO3200,3220,3220,325
0,3220,3220,3220
3250 GOSUB15000
3260 DRAWF$+L$+R$
3270 GOSUB12000
3290 ONI GOTO3700,2800,3200,3300
,3270,3270,3270
3300 GOSUB15000
3310 DRAWF$+L$+R$
3315 DRAWP$
3320 GOSUB12000
3340 ONI GOTO3750,2850,3250,3350
,3320,3320,3320
3350 GOSUB15000
3360 DRAWL$
3370 GOSUB12000
3390 ONI GOTO3370,2900,3300,3370
,3370,3370,3370
3400 GOSUB15000
3410 DRAWF$+L$
3420 GOSUB12000
3440 ONI GOTO3850,2950,3800,3420
,3420,3420,3420
3450 GOSUB15000
3460 DRAWF$+R$
3465 DRAWC$
3470 GOSUB12000
3490 ONI GOTO3900,3000,3470,3500
,3470,3470,3470
3500 GOSUB15000
3510 DRAWF$+L$
3520 GOSUB12000
3540 ONI GOTO3950,3520,3450,3520
,3520,3520,3520
3550 GOSUB15000
3560 DRAWF$+R$
3570 GOSUB12000
3590 ONI GOTO3550,3100,3570,3600
,3570,3570,3570
3600 GOSUB15000
3610 DRAWF$+L$
3620 GOSUB12000
3640 ONI GOTO4050,3150,3550,3620
,3620,3620,3620
3650 GOSUB15000
3660 DRAWF$+R$
3670 GOSUB12000
3680 ONI GOTO4100,3670,3670,3700
,3670,3670,3670
3700 GOSUB15000
3710 DRAWF$+L$
3720 GOSUB12000
3730 ONI GOTO4150,3700,3650,3720
,3720,3720,3720
3750 GOSUB15000
3760 DRAWF$+L$
3765 DRAWTA$
3770 GOSUB12000
3780 ONI GOTO4200,3300,3650,3770
,3770,3770,3770
3800 GOSUB15000
3820 GOSUB12000
3830 ONI GOTO3820,3400,3820,3820
,3820,3820,3820
3850 GOSUB15000
3860 DRAWF$+TA$
3870 GOSUB12000
3890 ONI GOTO4300,3400,3870,3870
,3870,3870,3870
3900 GOSUB15000
3910 DRAWR$
3920 GOSUB12000
3930 ONI GOTO3920,3450,3920,3950
,3920,3920,3920
3950 GOSUB15000
3960 DRAWF$+L$+R$
3970 GOSUB12000
3990 ONI GOTO3950,3500,3900,4000
,3970,3970,3970
4000 GOSUB15000
4010 DRAWL$+R$
4020 GOSUB12000
4030 ONI GOTO4020,4020,3950,4050
,4020,4020,4020
4050 GOSUB15000
4060 DRAWF$+L$
4070 GOSUB12000
4080 ONI GOTO4490,3600,4000,4070
,4070,4070,4070
4100 GOSUB15000
4110 DRAWR$
4120 GOSUB12000
4130 ONI GOTO4120,3650,4120,4150
,4120,4120,4120
4150 GOSUB15000
4160 DRAWL$+R$
4170 GOSUB12000
4190 ONI GOTO4170,3700,4100,4200
,4170,4170,4170
4200 GOSUB15000
4210 DRAWL$+R$
4220 GOSUB12000
4230 ONI GOTO4220,3750,4150,4250
,4220,4220,4220
4250 GOSUB15000
4260 DRAWL$+R$
4270 GOSUB12000
4280 ONI GOTO4270,4270,4200,4300
,4270,4270,4270
4300 GOSUB15000
4310 DRAWL$+R$
4320 GOSUB12000
4330 ONI GOTO4320,3850,4250,4350
,4320,4320,4320
4350 GOSUB15000
4360 DRAWL$+R$
4365 DRAWP$
4370 GOSUB12000
4390 ONI GOTO4370,4370,4300,4400
```

```
,4370,4370,4370
4400 GOSUB15000
4410 DRAWL$+R$
4420 GOSUB12000
4430 ONI GOTO4420,4420,4350,4450
,4420,4420,4420
4450 GOSUB15000
4455 DRAWL$+R$
4457 DRAW"BM60,68 R24D24L24U24"
4460 IFT4=1 THEN4470
4465 DRAW"BM62,70 R20D20L20U20;B
M72,80 NR4NL4NU4D4":CIRCLE(72,80
),4
4470 GOSUB12000
4475 IFI=7 THENT4=1
4480 ONI GOTO4470,4470,4400,4450
,4470,4470,4450
4490 GOSUB15000
4495 GOSUB12000
4497 ONI GOTO4495,4050,4495,4495
,4495,4495,4495
4500 GOSUB15000
4505 DRAWF$
4510 IFK=1 THEN4520
4515 DRAW"BM60,176 U2L2ND2L2ND1L
4U2L2D1L1D2R1D1R2U2"
4520 GOSUB12000
4530 IFI=7 THENK=1
4540 ONI GOTO4900,4900,4520,4520
,4520,4520,4500
4550 GOSUB15000
4560 DRAWF$+R$+D$
4570 GOSUB12000
4580 ONI GOTO4950,4570,4570,4600
,4570,750,4570
4600 GOSUB15000
4610 DRAWF$+L$
4620 GOSUB12000
4630 ONI GOTO5000,4620,4550,4620
,4620,4620,4620
4650 GOSUB15000
4660 DRAWR$+D$
4670 GOSUB12000
4680 ONI GOTO4670,4670,4670,4700
,4670,2100,4670
4700 GOSUB15000
4710 DRAWF$+L$+R$
4720 GOSUB12000
4730 ONI GOTO5100,4720,4650,4750
,4720,4720,4720
4750 GOSUB15000
4760 DRAWL$+R$
4770 GOSUB12000
4780 ONI GOTO4770,4770,4700,4800
,4770,4770,4770
4800 GOSUB15000
4810 DRAWF$+L$+R$
4820 GOSUB12000
4830 ONI GOTO5200,4820,4750,4850
,4820,4820,4820
4850 GOSUB15000
4860 DRAWL$+R$
4870 GOSUB12000
4890 ONI GOTO4870,4870,4800,4650
,4870,4870,4870
4900 GOSUB15000
4910 DRAWF$
4920 GOSUB12000
4930 ONI GOTO5300,4920,4920,4920
,4920,4920,4920
4950 GOSUB15000
4960 DRAWF$+R$
4970 GOSUB12000
4980 ONI GOTO5350,4550,4970,5000
,4970,4970,4970
5000 GOSUB15000
5010 DRAWF$+L$+R$
5015 DRAWS$
5020 GOSUB12000
5030 ONI GOTO5400,4600,4950,5050
,5020,5020,5020
5050 GOSUB15000
5060 DRAWL$+R$
5070 GOSUB12000
5080 ONI GOTO5070,5070,5000,5050
,5070,5070,5070
5100 GOSUB15000
5110 DRAWF$
5115 DRAWS$
5120 GOSUB12000
5130 ONI GOTO5500,4700,5120,5120
,5120,5120,5120
5150 GOSUB15000
5160 DRAWF$+R$
5170 GOSUB12000
5180 ONI GOTO5550,5170,5170,5200
,5170,5170,5170
5200 GOSUB15000
5210 DRAWF$+L$+R$
5213 DRAWS$
5216 CIRCLE(17,116),3
5220 GOSUB12000
5225 IFI=3 AND K=0 THENI=5
5230 ONI GOTO5600,4800,5150,5250
,5220,5220,5220
5250 GOSUB15000
5260 DRAWF$+L$
5270 GOSUB12000
5290 ONI GOTO5650,5270,5200,5270
,5270,5270,5270
5300 GOSUB15000
5310 DRAWF$+L$+R$
5320 GOSUB12000
5330 ONI GOTO5700,4900,4900,4500
,5320,5320,5320
5350 GOSUB15000
5360 DRAWF$+R$
5365 DRAWS$
5370 GOSUB12000
5380 ONI GOTO5750,4950,5370,5400
,5370,5370,5370
5400 GOSUB15000
5410 DRAWL$+R$
5415 CIRCLE(225,120),3
5420 GOSUB12000
5425 IFI=4 AND K=0 THENI=5
5430 ONI GOTO5420,5000,5350,5450
,5420,5420,5420
5450 GOSUB15000
5455 DRAWL$
5460 IFT5=1 THEN5470
5465 CIRCLE(180,162),10,,.24
5466 DRAW"BM170,162 F3G2D10F3R12
E3U10H2E3":PAINT(180,169),5,5
5467 DRAW"BM180,158 U12;BM172,15
8 H11;BM188,158 E11"
5470 GOSUB12000
5480 IFI=7 THENT5=1
5490 ONI GOTO5470,5470,5400,5470
,5470,5470,5450
5500 GOSUB15000
5510 DRAWF$
5520 GOSUB12000
5530 ONI GOTO5900,5100,5520,5520
,5520,5520,5520
5550 GOSUB15000
5560 DRAWF$
5570 GOSUB12000
5580 ONI GOTO5950,5150,5570,5570
,5570,5570,5570
5600 GOSUB15000
5610 DRAWR$
5620 GOSUB12000
5630 ONI GOTO5620,5200,5620,5650
,5620,5620,5620
5650 GOSUB15000
5660 DRAWL$
5670 GOSUB12000
5680 ONI GOTO5670,5250,5600,5670
,5670,5670,5670
5700 GOSUB15000
5710 DRAWR$
5720 GOSUB12000
5730 ONI GOTO5720,5300,5720,5750
,5720,5720,5720
5750 GOSUB15000
5760 DRAWL$+R$
5770 GOSUB12000
5780 ONI GOTO5770,5350,5700,5800
,5770,5770,5770
5800 GOSUB15000
5810 DRAWL$
5820 GOSUB12000
5830 ONI GOTO5820,5820,5750,5820
,5820,5820,5820
5850 GOSUB15000
5860 DRAWF$+R$
5870 GOSUB12000
5880 ONI GOTO6250,5870,5870,5900
,5870,5870,5870
5900 GOSUB15000
5910 DRAWL$
5920 GOSUB12000
5930 ONI GOTO5920,5500,5850,5920
,5920,5920,5920
5950 GOSUB15000
5960 DRAWR$
5970 GOSUB12000
5980 ONI GOTO5970,5550,5970,6000
,5970,5970,5970
6000 GOSUB15000
6010 DRAWF$+L$+R$
6020 GOSUB12000
6030 ONI GOTO6350,6020,5950,6050
,6020,6020,6020
6050 GOSUB15000
6060 DRAWF$+L$+R$
6070 GOSUB12000
6080 ONI GOTO6400,6070,6000,5950
,6070,6070,6070
6100 GOSUB15000
6110 DRAWF$+R$
6120 GOSUB12000
6130 ONI GOTO6450,6120,6120,6150
,6120,6120,6120
6150 GOSUB15000
6160 DRAWL$+R$
6170 GOSUB12000
6180 ONI GOTO6170,6170,6100,6200
,6170,6170,6170
6200 GOSUB15000
6210 DRAWF$+L$+R$
6215 DRAWS$
6220 GOSUB12000
6230 ONI GOTO6550,6220,6150,6250
,6220,6220,6220
```

```
6250 GOSUB15000
6260 DRAWF$+L$+R$
6270 GOSUB12000
6280 ONI GOTO6600,5850,6200,6300
,6270,6270,6270
6300 GOSUB15000
6310 DRAWL$+R$
6320 GOSUB12000
6330 ONI GOTO6320,6320,6250,6340
,6320,6320,6320
6340 GOSUB15000
6342 FORT=2TO96 STEP4
6344 LINE(T,T)-(256-T,192-T),PSE
T,B
6346 NEXT
6348 X=RND(3):ONX GOTO1400,1000,
1050
6350 GOSUB15000
6360 DRAWF$+R$
6365 DRAWS$
6370 GOSUB12000
6380 ONI GOTO6800,6000,6370,6400
,6370,6370,6370
6400 GOSUB15000
6410 DRAWF$+L$
6420 GOSUB12000
6430 ONI GOTO6850,6050,6350,6420
,6420,6420,6420
6450 GOSUB15000
6460 DRAWR$
6470 GOSUB12000
6480 ONI GOTO6470,6100,6470,6450
,6470,6470,6470
6500 GOSUB15000
6505 DRAWR$
6510 IFT6=1 THEN6520
6515 DRAW"BM172,153 U57E4U8G4E10
F10H4D8F4D57;BM179,143;M182,116
R6D32L12U32R6;BM182,102 NU5R3":C
IRCLE(182,102),8:CIRCLE(179,145)
,2
6520 GOSUB12000
6525 IFI=7 THENT6=1
6530 ONI GOTO6520,6520,6520,6550
,6520,6520,6500
6550 GOSUB15000
6560 DRAWL$+R$
6570 GOSUB12000
6580 ONI GOTO6570,6200,6500,6600
,6570,6570,6570
6600 GOSUB15000
6610 DRAWL$
6620 GOSUB12000
6630 ONI GOTO6620,6250,6550,6650
,6620,6620,6620
6650 GOSUB15000
6660 DRAWR$
6670 GOSUB12000
6680 ONI GOTO6670,6670,6670,6750
,6670,6670,6670
6750 GOSUB15000
6760 DRAWL$+R$
6770 GOSUB12000
6780 ONI GOTO6770,6770,6650,6800
,6770,6770,6770
6800 GOSUB15000
6810 DRAWL$
6820 GOSUB12000
6830 ONI GOTO6820,6350,6750,6820
,6820,6820,6820
6850 GOSUB15000
6860 IFT7=1 THEN6870
6865 DRAW"BM81,183 E11R4NG8F3G8L
10E3R4F3"
6870 GOSUB12000
6875 IFI=7 THENT7=1
6880 ONI GOTO6870,6400,6870,6870
,6870,6870,6850
6900 GOSUB15000
6910 DRAWF$+R$+U$
6920 GOSUB12000
6930 ONI GOTO7300,6920,6920,6950
,1250,6920,6920
6950 GOSUB15000
6960 DRAWF$+L$+R$
6970 GOSUB12000
6990 ONI GOTO7350,6970,6900,7000
,6970,6970,6970
7000 GOSUB15000
7010 DRAWF$+L$+R$
7020 GOSUB12000
7030 ONI GOTO7400,7020,6950,7050
,7020,7020,7020
7050 GOSUB15000
7060 DRAWF$+L$
7070 GOSUB12000
7080 ONI GOTO7450,7070,7000,7070
,7070,7070,7070
7100 GOSUB15000
7110 DRAWR$
7120 GOSUB12000
7130 ONI GOTO7120,7120,7120,7150
,7120,7120,7120
7150 GOSUB15000
7160 DRAWF$+L$
7165 DRAWS$
7170 GOSUB12000
7180 ONI GOTO7550,7170,7100,7170
,7170,7170,7170
7200 GOSUB15000
7210 DRAWF$
7220 GOSUB12000
7230 ONI GOTO7600,7700,7220,7220
,7220,7220,7220
7250 GOSUB15000
7255 DRAWF$
7260 IFT8=1 THEN7270
7265 DRAW"BM52,168 R48H4L40G4D20
R4NU20R4NU20R14NU20R4NU20R14NU20
R4NU20R4U20":FORT=1TO60:Q=RND(40
)+56:V=RND(3)+164:PSET(Q,V):NEXT
7270 GOSUB12000
7275 IFI=7 THENT8=1
7280 ONI GOTO7650,7270,7270,7270
,7270,7270,7250
7300 GOSUB15000
7310 DRAWR$
7320 GOSUB12000
7330 ONI GOTO7320,6900,7320,7350
,7320,7320,7320
7350 GOSUB15000
7360 DRAWL$+R$
7370 GOSUB12000
7380 ONI GOTO7370,6950,7300,7400
,7370,7370,7370
7400 GOSUB15000
7410 DRAWF$+L$+R$
7420 GOSUB12000
7430 ONI GOTO7400,7000,7350,7450
,7420,7420,7420
7450 GOSUB15000
7460 DRAWF$+L$
7465 DRAWS$
7470 GOSUB12000
7480 ONI GOTO7850,7050,7400,7470
,7470,7470,7470
7500 GOSUB15000
7510 DRAWF$+S$
7520 GOSUB12000
7530 ONI GOTO7900,7520,7520,7520
,7520,7520,7520
7550 GOSUB15000
7560 DRAWF$
7570 GOSUB12000
7580 ONI GOTO7950,7150,7570,7570
,7570,7570,7570
7600 GOSUB15000
7610 DRAWR$
7620 GOSUB12000
7630 ONI GOTO7620,7200,7620,7650
,7620,7620,7620
7650 GOSUB15000
7660 DRAWF$+L$+R$
7670 GOSUB12000
7680 ONI GOTO8050,7250,7600,7700
,7670,7670,7670
7700 GOSUB15000
7710 DRAWF$+U$
7720 GOSUB12000
7730 ONI GOTO8100,7720,7720,7720
,1900,7720,7720
7750 GOSUB15000
7760 DRAWF$+L$+R$
7765 CIRCLE(225,120),3
7770 GOSUB12000
7775 IFI=4 AND K=0 THENI=7
7780 ONI GOTO8150,7770,7750,7800
,7770,7770,7770
7800 GOSUB15000
7805 DRAWL$
7810 IFT9=1 THEN7820
7815 DRAW"BM64,72 R73D16L73U16;B
M68,77 U1R45D1NL45U1;M116,78 R18
D7;M115,81D1L1H3NR1L21H2"
7820 GOSUB12000
7830 IFI=7 THENT9=1
7840 ONI GOTO7820,7820,7750,7820
,7820,7820,7800
7850 GOSUB15000
7860 DRAWF$
7870 GOSUB12000
7880 ONI GOTO8250,7450,7870,7870
,7870,7870,7870
7900 GOSUB15000
7910 DRAWR$
7920 GOSUB12000
7930 ONI GOTO7920,7500,7920,7950
,7920,7920,7920
7950 GOSUB15000
7960 DRAWF$+L$
7970 GOSUB12000
7980 ONI GOTO8350,7550,7900,7970
,7970,7970,7970
8000 GOSUB15000
8010 DRAWR$
8020 GOSUB12000
8030 ONI GOTO8020,8020,8020,8050
,8020,8020,8020
8050 GOSUB15000
8060 DRAWF$+L$+R$
8065 DRAWS$
8070 GOSUB12000
8080 ONI GOTO8450,7650,8000,8050
,8070,8070,8070
8100 GOSUB15000
8110 DRAWF$
```

```
8120 GOSUB12000
8130 ONI GOTO8500,7700,8120,8120
,8120,8120,8120
8150 GOSUB15000
8160 DRAWF$+S$
8170 GOSUB12000
8180 ONI GOTO8550,7750,8170,8170
,8170,8170,8170
8200 GOSUB15000
8205 DRAWL$+R$
8210 IFTQ=1 THEN8220
8215 DRAW"BM125,177 R11D3L20U3R9
U8R1U4D2L1R2L1D2R1D8":CIRCLE(122
,175),6,5,1,.43,.83:CIRCLE(129,1
75),6,5,1,.69,.07
8220 GOSUB12000
8225 IFI=7 THENTQ=1
8230 ONI GOTO8220,8220,8200,8250
,8220,8220,8200
8250 GOSUB15000
8260 DRAWL$
8270 GOSUB12000
8290 ONI GOTO8270,7850,8200,8270
,8270,8270,8270
8300 GOSUB15000
8310 DRAWR$
8320 GOSUB12000
8330 ONI GOTO8320,8320,8320,8350
,8320,8320,8320
8350 GOSUB15000
8360 DRAWF$+L$+R$
8365 DRAWS$
8370 GOSUB12000
8380 ONI GOTO8750,7950,8300,8400
,8370,8370,8370
8400 GOSUB15000
8410 DRAWF$+L$+R$
8420 GOSUB12000
8440 ONI GOTO8800,8420,8350,8400
,8420,8420,8420
8450 GOSUB15000
8460 DRAWF$+L$
8470 GOSUB12000
8480 ONI GOTO8850,8050,8400,8470
,8470,8470,8470
8500 GOSUB15000
8510 DRAWR$
8520 GOSUB12000
8530 ONI GOTO8520,8100,8520,8550
,8520,8520,8520
8550 GOSUB15000
8560 DRAWF$+L$
8570 GOSUB12000
8580 ONI GOTO8950,8150,8500,8570
,8570,8570,8570
8600 GOSUB15000
8610 DRAWF$+R$
8620 GOSUB12000
8625 DRAWS$
8630 ONI GOTO9000,8620,8620,8650
,8620,8620,8620
8650 GOSUB15000
8660 DRAWF$+L$
8670 GOSUB12000
8680 ONI GOTO9050,8670,8600,8670
,8670,8670,8670
8700 GOSUB15000
8710 DRAWF$+S$
8720 GOSUB12000
8730 ONI GOTO9100,8720,8720,8720
,8720,8720,8720
8750 GOSUB15000
8760 DRAWF$
8770 GOSUB12000
8780 ONI GOTO9150,8350,8770,8770
,8770,8770,8770
8800 GOSUB15000
8810 DRAWF$
8820 GOSUB12000
8830 ONI GOTO8800,8400,8820,8820
,8820,8820,8820
8850 GOSUB15000
8860 DRAWF$
8870 GOSUB12000
8880 ONI GOTO9250,8450,8870,8870
,8870,8870,8870
8900 GOSUB15000
8910 DRAWF$+R$
8911 DRAWS$
8920 GOSUB12000
8930 ONI GOTO9300,8920,8920,8950
,8920,8920,8920
8950 GOSUB15000
8960 DRAWF$+L$+R$
8965 DRAWS$
8970 GOSUB12000
8980 ONI GOTO9350,8550,8900,9000
,8970,8970,8970
9000 GOSUB15000
9010 DRAWF$+L$+R$
9020 GOSUB12000
9030 ONI GOTO9360,8600,8950,9050
,9020,9020,9020
9050 GOSUB15000
9060 DRAWF$+L$+R$
9070 GOSUB12000
9080 ONI GOTO9400,8650,9000,9100
,9070,9070,9070
9100 GOSUB15000
9110 DRAWF$+L$+R$
9120 GOSUB12000
9130 ONI GOTO9450,8700,9050,9150
,9120,9120,9120
9150 GOSUB15000
9160 DRAWF$+L$
9170 GOSUB12000
9180 ONI GOTO9500,8750,9100,9170
,9170,9170,9170
9200 GOSUB15000
9210 DRAWF$+R$
9215 DRAWS$
9220 GOSUB12000
9230 ONI GOTO9550,8800,9220,9250
,9220,9220,9220
9250 GOSUB15000
9260 DRAWL$
9270 GOSUB12000
9280 ONI GOTO9270,8850,9200,9270
,9270,9270,9270
9300 GOSUB15000
9310 IFTZ=1 THEN9320
9315 DRAW"BM72,153 L20U33R20D33R
2U38R22D3L22R4U1R16D1R2D2NL22D8L
22R2U4R18D4R2D8L22R2U4R18D4R2D17
R2U33R20D33":CIRCLE(62,126),3:CI
RCLE(108,126),3:CIRCLE(62,142),8
:CIRCLE(108,142),8
9320 GOSUB12000
9325 IFI=7 THENTZ=1
9330 ONI GOTO9320,8900,9320,9320
,9320,9320,9300
9350 GOSUB15000
9355 FORT=2TO96 STEP4
9356 LINE(T,T)-(256-T,192-T),PSE
T,B
9357 NEXT
9358 X=RND(3):ONX GOTO1400,1000,
1050
9360 GOSUB15000
9365 DRAWF$+L$+R$
9370 GOSUB12000
9380 ONI GOTO9360,9000,9350,9400
,9370,9370,9370
9400 GOSUB15000
9410 DRAWF$+L$+R$
9420 GOSUB12000
9430 ONI GOTO9400,9050,9360,9450
,9420,9420,9420
9450 GOSUB15000
9460 DRAWL$
9470 GOSUB12000
9480 ONI GOTO9470,9100,9400,9470
,9470,9470,9470
9500 GOSUB15000
9510 DRAWR$
9520 GOSUB12000
9530 ONI GOTO9520,9150,9520,9550
,9520,9520,9520
9550 GOSUB15000
9560 DRAWL$+R$
9570 GOSUB12000
9580 ONI GOTO9570,9200,9500,9600
,9570,9570,9570
9600 GOSUB15000
9610 DRAWL$
9620 GOSUB12000
9630 ONI GOTO9620,9620,9550,9620
,9620,9620,9620
12000 I$=INKEY$:IFI$="F"THENI$="
1"ELSEIFI$="B"THENI$="2"ELSEIFI$
="L"THENI$="3"ELSEIFI$="R"THENI$
="4"ELSEIFI$="U"THENI$="5"ELSEIF
I$="D"THENI$="6"ELSEIFI$="G"THEN
I$="7"ELSEIFI$<>""THENI$="0"
12010 I=VAL(I$)
12015 IFI=0 THEN12000
12020 RETURN
15000 PCLS:DRAW"BM0,0;M48,38 R16
0;N;M255,0 D115;N;M255,191 L160;
N;M0,191 U115"
15001 RETURN
15100 PCLS:CIRCLE(124,96),23:PAI
NT(124,96),5,5:CIRCLE(156,104),1
7:PAINT(156,104),5,5:CIRCLE(159,
86),17:PAINT(159,86),5,5:CIRCLE(
146,68),17:PAINT(146,68),5,5:CIR
CLE(118,68),17:PAINT(118,68),5,5
:CIRCLE(94,77),17:PAINT(94,77),5
,5:CIRCLE(94,105),17
15101 PAINT(94,105),5,5
15110 DRAW"BM116,118 D62;M113,18
6;M109,191;BM132,118 D63;M136,19
1;BM132,136 G2D1F2R10E4;M156,132
 U4;M146,132 G4L10"
15111 RETURN
15200 DRAW"BM0,40 R4U4R8D4R48U4R
8D4R48U4R8D4R48U4R8D4R48U4R8D4R2
0D4L255U4"
15201 DRAW"BM20,0 D32R32U32BR24D
32R32U32BR24D32R32U32BR24D32R32U
32BR24D32R12"
15202 DRAW"BM20,72 R32D28L32U28;
BM20,104 R32D44L32U44;BM188,72 R
32D28L32U28;BM188,104 R32D44L32U
44"
15210 RETURN
```

continued on p62

Wash your mouth out, you little...

CURSOR

OS9, WORDPAK, OS9 ASSEMBLER.
OS9 UTILITY

by Jeff Larson

BACK IN THE BAD old days, when I had an old grey case CoCo and wanted 80 column text, I had a Wordpack II.

After using it for a while, I decided to find out how it worked. By this I don't mean the electronic details of its insides (I must admit that I peeked inside and shuddered before quickly replacing the cover).

No, I wanted to know how to program it to do its tricks.

Most of what I discovered about it was in the source code for the basic driver for it, but one thing I discovered was how to alter the cursor to suit myself.

This is illustrated in the assembly language programme I wrote called "cursor".

This program is designed to be used under OS9, and will change the cursor to suit your own preferences.

I found sometimes that the blinking cursor drove me nuts, and other times that I needed it blinking so I could find it.

This was the first option I added. Then I discovered that the cursor could be made to blink at double speed. I can't imagine anyone using this option, but amongst all the strange users, there is sure to be one.

The last option is to turn the cursor off completely.

This is handy if you don't want the cursor displayed while you are setting up a screen.

A short word of explanation here about writing OS9 programs. Ho hum, I can hear from all you "experienced" people out there.

Too bad! I read all I can about programming, and manage to learn something new from the most elementary articles.

Now I've got that out of the way, OS9 assembly language is really easy. All the hard work is done for you by the operating system, leaving you to concentrate on the logic of the program.

Take parameters for example. To read these from the command line, all that is needed is ...

```
lda ,x+
```

Simple! The ascii value of the character is is register a. If (for instance) it is a carriage return ($0d) index register x points to the end of the line, and there is no (or no more) parameters.

This passing of parameters is illustrated in the first part of the program in line 35, and again in line 40.

Once the parameter is in register a, then we can check it for a value, or do something with it. The first thing I do is check if it is a "-". This happens in line 38.

If it is a dash, the program then gets the next character from the command line. Then comes the tricky bit. Line 41 is ...

```
anda $df
```

If you examine the ASCII code carefully you will see the difference between upper and lower case is bit 5. If bit 5 is set, then the character is lower case. If bit 5 is clear, the the character is upper case.

All line 41 does is clear bit 5 of the number in register a (which is the next character from the command line). This allows the user to enter the option in upper or lower case, and still have it recognized as a valid command.

The rest of the program is fairly ordinary stuff. It proceeds to check the various options, and prepare register b for storing in wordpack's cursor control register.

This is finally accomplished in line 73.

The program then exits with no errors in line 81.

The only other thing of note in the program is that any unknown option on the command line will stop the program from doing anything to wordpack.

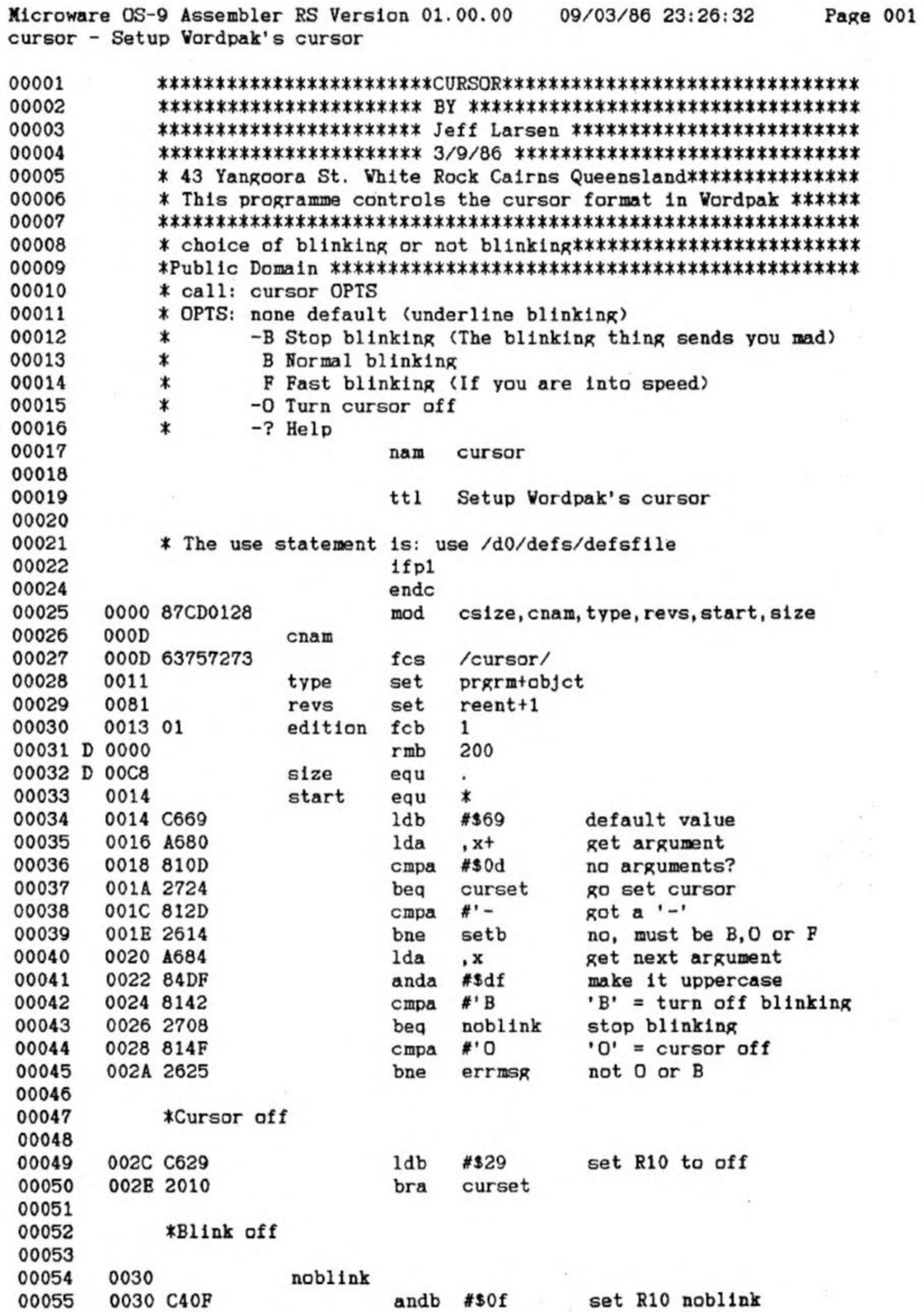
It will go onto the errmsg routine, and print to the standard error output a message to show the user what are valid options.

This is done from line 82 to 89, but all the work is done by OS9 in line 86. If you read your OS9 Technical Information, you will not only learn something, but you will see that the I$write call will send whatever you want to your screen.

In conclusion, this was an educational program, as well as being useful. I found that while I had my wordpack, I used the cursor program in my startup file, so that the cursor was initialized every time I booted into OS9.

Mainly I set the cursor to underline and blinking, which is the default as just a plain underline is sometimes hard to find on a full 80 column screen.

The Listing:



```
Microware OS-9 Assembler RS Version 01.00.00    09/03/86 23:26:32       Page 001
cursor - Setup Wordpak's cursor

       ***********************CURSOR*******************************
       *********************** BY **********************************
       *********************** Jeff Larsen *************************
       *********************** 3/9/86 ******************************
       * 43 Yangoora St. White Rock Cairns Queensland***************
       * This programme controls the cursor format in Wordpak ******
       *************************************************************
       * choice of blinking or not blinking*************************
       *Public Domain **********************************************
       * call: cursor OPTS
       * OPTS: none default (underline blinking)
       *       -B Stop blinking (The blinking thing sends you mad)
       *        B Normal blinking
       *        F Fast blinking (If you are into speed)
       *       -O Turn cursor off
       *       -? Help
                           nam   cursor

                           ttl   Setup Wordpak's cursor

       * The use statement is: use /d0/defs/defsfile
                           ifp1
                           endc
  0000 87CD0128            mod   csize,cnam,type,revs,start,size
  000D             cnam
  000D 63757273            fcs   /cursor/
  0011             type     set   prgrm+objct
  0081             revs     set   reent+1
  0013 01          edition  fcb   1
D 0000                     rmb   200
D 00C8             size     equ   .
  0014             start    equ   *
  0014 C669                ldb   #$69       default value
  0016 A680                lda   ,x+        get argument
  0018 810D                cmpa  #$0d       no arguments?
  001A 2724                beq   curset     go set cursor
  001C 812D                cmpa  #'-        got a '-'
  001E 2614                bne   setb       no, must be B,O or F
  0020 A684                lda   ,x         get next argument
  0022 84DF                anda  #$df       make it uppercase
  0024 8142                cmpa  #'B        'B' = turn off blinking
  0026 2708                beq   noblink    stop blinking
  0028 814F                cmpa  #'O        'O' = cursor off
  002A 2625                bne   errmsg     not O or B

       *Cursor off

  002C C629                ldb   #$29       set R10 to off
  002E 2010                bra   curset

       *Blink off

  0030             noblink
  0030 C40F                andb  #$0f       set R10 noblink
  0032 200C                bra   curset
```

```

     *Turn cursor or blink on

0034            setb
0034 84DF                anda  #$df       make it uppercase
0036 8142                cmpa  #'B        turn slow blink on?
0038 2706                beq   curset
003A 8146                cmpa  #'F        fast blink on?
003C 2613                bne   errmsg     anything else is wrong
003E C649                ldb   #$49       set fast blink

     *point Y at wordpak's control register

0040            curset
0040 108EFF98            ldy   #$ff98
0044 860A                lda   #$0a       setting R10
0046 EDA4                std   ,y         put value in R10

     * set cursor to stop scanning on line 9

0048 CC0B09              ldd   #$0b09
004B EDA4                std   ,y         store in R11
004D 5F                  clrb              no errors
004E            exit
004E 103F06              os9   f$exit     finished
0051            errmsg
0051 308D000E            leax  msg,pcr
0055 108E00C2            ldy   #msglen
0059 8602                lda   #2
005B 103F8A              OS9   I$write
005E 25EE                bcs   exit
0060 5F                  clrb
0061 20EB                bra   exit

0063            msg
0063 55736561            fcc   /Useage: Cursor OPTS/
0078 0D0A                fdb   $0d0a
007A 4F505453            fcc   /OPTS:  None (Underline blinking)/
009A 0D0A                fdb   $0d0a
009C 20202020            fcc   /      -B (Stop blinking)/
00B4 0D0A                fdb   $0d0a
00B6 20202020            fcc   /       B (Slow blinking)/
00CE 0D0A                fdb   $0d0a
00D0 20202020            fcc   /      -O (Turn cursor off)/
00EA 0D0A                fdb   $0d0a
00EC 20202020            fcc   /       F (Fast blinking)/
0104 0D0A                fdb   $0d0a
0106 20202020            fcc   /      -? (Print this message)/
0123 0D0A                fdb   $0d0a
00C2            msglen   equ   *-msg
0125 55613A              emod
0128            csize    equ   *
                         end

00000 error(s)
00000 warning(s)
$0128 00296 program bytes generated
$00C8 00200 data bytes allocated
$2839 10297 bytes used for symbols
```

■

ADVENTURE + II

continued from p58

```
20000 F$="BM104,153 U81R48D81"
20001 L$="BM12,182 U125;M36,72 D
91"
20002 R$="BM244,182 U125;M220,72
 D91"
20003 S$="BM123,72;M127,81;M130,
86;M137,92;M145,95 R7U5;M146,89;
M140,87;M135,83;M133,78 U6R7D4;M
142,79;M145,81;M148,82 R4U10;N;M
145,95 N;M137,92 N;M130,86;M127,
81"
20004 TA$="BM172,132 R36;M224,14
5 L36;M172,132 D21;M176,156;NU17
R2U15;BM172,136;M188,149;NU4R36U
4D21L2NU17;M218,163;U14L7D7L2NU7
;M206,153 U4L16D17L2NU17;M184,16
3 U17"
20005 D$="BM104,192 E8R32F8"
20006 U$="BM104,0 F8R32E8"
20007 P$="BM160,56 D36NE4R44NH4U
36NG4L44F4R36D28L36U28;BM176,88
H8R4F2R16E3R5D5G4;BM183,82 U17NG
15R1D5ND12F9"
20009 C$="BM66,132 U16E4R20F4D16
L28;M62,145 NR36D3NR36D18R3U18D8
R2U8R31D18L3U18L2D8R2U8R3U3;M94,
132"
20100 RETURN
25000 FORT=1TO100:NEXT:CLS:PRINT
"YOU HAVE FOUND ALL OF THE
 TREASURES.  WOULD YOU LIKE
 ANOTHER GAME  (Y/N).
25001 IK$=INKEY$:IFIK$="Y"THENRU
N ELSEIFIK$="N"THENEND ELSEIFIK$
<>""THEN25001
26000 DATA I,I, ,S,U,L,P, ,E,R,U
,T,N,E,V,D,A
```

■

GOLDSOFT

P.O. BOX 1742, SOUTHPORT. QLD. 4215 Phone (075) 39–6177

Goldsoft Price list as at March, 1987

Please tick (✓) your requirements.

HARDWARE

CoCoConnection: $206.00 ( )
Video Amp: With Sound - $35.00 ( )
Without Sound - $25.00 ( )
The Probe: $49.95 ( )

GOLDLINK

Access Goldlink *642# on Viatel with a 1200/75 baud modem. Annual subscription: $39.75 ( )

SOFTWARE

Magazines, Tapes & Disks

Australian CoCo (Advanced Programs for your CoCo):

Magazines:	Tape ( ) or Disk ( )
12 Months $39.95 ( )	12 Months $123.75 ( )
6 Months $24.95 ( )	6 Months $ 74.25 ( )
1 Month $ 4.50 ( )	1 Month $ 16.50 ( )

Softgold (Programs for your CoCo):

Magazines	Tape ( ) or Disk ( )
12 Months $39.95 ( )	12 Months $123.75 ( )
6 Months $24.95 ( )	6 Months $ 74.25 ( )
1 Month $ 4.50 ( )	1 Month $ 16.50 ( )

Gold Disk - Available Quarterly:
1 Month: $16.00 ( )

The CoCo3 Tape/Disk:

# 1	Tape: $10.00	Disk: $16.00 ( )
# 2	Tape: $10.00	Disk: $16.00 ( )
# 3	TAPE: $10.00	DISK: $16.00 ( )

"Say tue Wordz":
Two Curriculum based speller programs for your Tandy Speech/Sound Pack: $29.95
Req. 32K + Tandy Speech Pack ( )

Best of CoCoOz - $16.00
A selection of programs from Australian CoCo Magazine.

	Tape:	Disk:
# 1 - Education:	( )	( )
# 2 - Games 16K:	( )	( )
# 2 - Games 32K:	( )	( )
# 3 - Utilities:	( )	( )
# 4 - Business :	( )	( )
# 5 - Adventure:	( )	( )
# 6 - Preschool:	( )	( )
# 7 - Graphics :	( )	( )
# 8 - Games 16K:	( )	( )
# 9 - Games 32K:	( )	( )
#10 - Education:	( )	( )
#11 - Education:	-	( )

BRIC-A-BRAC

Blank Tapes:	12 @ $18.00	( )
(C-30)	1 @ $ 2.00	( )
Tape Cases:	12 @ $ 5.00	( )
Disks DSDD:	10 @ $20.00	( )
	1 @ $ 2.50	( )

BOOKS

Help (for your CoCo): $9.95 ( )
Mico Help (for your MC-10): $9.95 ( )

BACK ISSUES

Australian CoCo: Sep 84 - Dec 85: $2.00 ( )
Australian CoCo: Jan 86 - Feb 87: $3.75 ( )
Australian Mico: Aug 84 - Dec 85: $2.00 ( )

ADDITIONAL REQUIREMENTS

....

....

....

....

New Subscription: ( ) Renewal: ( )

Sub No: ....

Name: ....

Address: ....

....

.............P/Code......

Phone: (....) ....

Please find enclosed:
a. Cheque ( )
b. Money Order ( )
c. Credit Card ( )

Credit Card Type & Number:

Bankcard ( ): ...... - ...... - ...... - ......

Visa ( ): ...... - ...... - ...... - ......

Mastercard ( ): ...... - ...... - ...... - ......

Expiry Date: ...../.....

Authorised Amount: $...........

Signed: ....

user contacts

(Stop between numbers = b.h. else a.h.; but, hyphen between = both)

ACT:

CANBERRA NTH	JOHN BURGER	062 58 3924
CANBERRA STH	LES THURBON	062 88 9226

NSW:

SYDNEY:

BANKSTOWN	PAT DORSETT	02 646 3619
BLACKTOWN	KEITH GALLAGHER	02-627-4627
CARLINGFORD	ROSKO MCKAY	02 624 3353
CHATSWOOD	BILL O'DONNELL	02 419 6081
COLYTON	HERMAN FREDRIKSSON	02 623 6379
FAIRFIELD	ARTH PITTARD	02 72 2881
GLADESVILLE	MARK ROTHWELL	02 817 4627
HILLS DIST	ARTHUR SLADE	02 622 8940
HORNSBY	ATHALIE SMART	02 848 8830
INGLEBURN	STEPHEN RIDGEWAY	02 605 7382
KENTHURST	TOM STUART	02 654 2178
LEICHHARDT	STEVEN CHICOS	02 560 6207
	or GORGE ECHEGARAY	02 560 9664
LIVERPOOL	LEONIE DUGGAN	02-607-3791
MACQUARIE FIELDS		
	BARRY DARNTON	02 618 1909
MINTO	GRAHAM POLLOCK	02 603 5028
SUTHERLAND	IAN ANNABEL	02 528 3391
SYDNEY EAST	JACKY COCKINOS	02 344 9111
ALBURY	RON DUNCAN	060 43 1031
ARMIDALE	DOUG BARBER	067 72 7647
BLAXLAND	BRUCE SULLIVAN	047 39 3903
BROKEN HILL	TERRY NOONAN	080 88 2382
CAMDEN	SEAN MURDOCH	047 74 8291
COFFS HARBOUR	BOB KENNY	066 51 2205
COOMA	ROSS PRATT	064 52 3065
COORANBONG	GEORGE SAVAGE	049 77 1054
COOTAMUNDRA	CHERYL WILLIS	069 42 2264
DENILIQUIN	WAYNE PATTERSON	058 81 3014
DUBBO	GRAEME CLARKE	068 89 6549
FORBES	JOHANNA VAGG	068 52 2943
GOSFORD	PETER SEIFERT	043 32 7874
GRAFTON	PETER LINDSAY	066 42 2503
GUYRA	MICHAEL J. HARTMANN	067 79 7547
JUNEE	PAUL MALONEY	069 24 1860
KEMPSEY	RICK FULLER	065-62-7222
LEETON	BRETT WALLACE	069-53-2081
LISMORE	ROB HILLARD	066 24 3089
LITHGOW	DAVID BERGER	063 52 2282
MAITLAND	BILL SNOW	049 66 2557
MOREE	ALF BATE	067 52 2465
MUDGEE	BRIAN STONE	063-72-1958
NAMBUCCA HDS	WENDY PETERSON	065 68 6723
NARROMINE	GRAEME CLARKE	068 89 6549
NEWCASTLE	LYN DAWSON	049 49 8144
NOWRA	ROY LOPEZ	044 46 5449
PARKES	DAVID SMALL	068 62 2682
PORT MACQUARIE	RON LALOR	065 83 8223
SPRINGWOOD	JIM HOPPITT	047 54 1474
TAHMOOR	GARY SYLVESTER	046 81 9318
UPPER HUNTER	TERRY GRAVOLIN	065 45 1698
URALLA	FRANK MUDFORD	067 78 4391
WAGGA WAGGA	CES JENKINSON	069 25 2263
WYONG	JOHN WALLACE	043 90 0312

NT:

DARWIN	BRENTON PRIOR	089.81.7766

QLD:

BRISBANE:

BIRKDALE	COLIN NORTH	07 824 2128
CLAYFIELD	JACK FRICKER	07 262 8869
COLL'WOOD PK	ANDREW SIMPSON	07 288 5206
IPSWICH	MICK MURPHY	07 271 1777
PINE RIVERS	BARRY CLARKE	07 204 2806
SOUTH WEST	BOB DEVRIES	07 372 7816
SANDGATE	MARK MIGHELL	07 269 3846
SCARBOROUGH	PETER MAY	07 203 6723
WOODRIDGE	ALLAN ALLSOP	07 349 1631
AIRLIE BEACH	GLEN EVANS	079 46 1264
BIGGENDEN	ALAN MENHAM	071 27 1272
BOWEN	TERRY COTTON C/O	077 86 2220
BUNDABERG	RON SIMPKIN	071 71 5301
CAIRNS	JEFF LARSEN	070 54 7127
DALBY	MERRICK TANSKY	074.62.3228
GLADSTONE	CAROL CATHCART	079 78 3594
GOLD COAST	GRAHAM MORPHETT	075 51 0577
GYMPIE	BERT LLOYD	071 8219100
HERVEY BAY	LESLEY HORWOOD	071 22 4989
MACKAY	LEN MALONEY	079511333x782
MARYBOROUGH	JOHN EFFER	071 21 6638
MT ISA	JACK RAE	077 43 3486
MURGON	PETER ANGEL	071 68 1628
ROCKHAMPTON	KEIRAN SIMPSON	079 28 6162
TARA	DEBBIE DORFIELD	074 65 3177
TOOWOOMBA	LEN GERSEKOWSKI	076 35 8264
TOWNSVILLE	JOHN O'CALLAGHAN	077 73 2064

SA:

ADELAIDE	JOHN HAINES	08 278 3560
MORPHETT VALE	KEN RICHARDS	08 384 4503
PORT NOARLUNGA	ROB DALZELL	08 386 1647
SEACOMBE HTS	GLENN DAVIS	08 296 7477
PORT LINCOLN	BILL BOARDMAN	086 82 2385
PORT PIRIE	VIC KNAUERHASE	086 32 1230
WHYALLA	MALCOLM PATRICK	086 45 7637

TAS:

DEVONPORT	JEFF BEST	004 24 6759
HOBART	BOB DELBOURGO	002 25 3896
KINGSTON	WIM DE PUIT	002 29 4950
LAUNCESTON	BILL BOWER	003 44 1584
SMITHTON	HARRY CHRISAFIS	004-52-1590
WYNYARD	ANDREW WYLLIE	004 35 1839

VIC:

MELBOURNE:

MELBOURNE CCC	JOY WALLACE	03 277 5182
DANDENONG	DAVID HORROCKS	03 707 5870
DONCASTER	JUSTIN LIPTON	03 857 5149
FRANKSTON	BOB HAYTER	03.783.9748
NARRE WARREN	LEIGH EAMES	03 704 6680
NTH EASTERN	PETER WOOD	03 435 2018
MELTON	MARIO GERADA	03 743 1323
RINGWOOD	IVOR DAVIES	03 758 4496
SUNBURY	JACK SMIT	03.744.1355
SUNSHINE	IAN BUTTRISS	03 314 3240
UPR F'TREE GLY	RORY DOYLE	03 758 2671
BAIRNSDALE	COLIN LEHMANN	051 57 1545
BALLARAT	MARK BEVELANDER	053 32 6733
DAYLESFORD	DANNY HEDJI	054 24 8329
GEELONG	DAVID COLLEN	052 43 2128
MAFFRA	MAX HUCKERBY	051 45 4315
MOE	JOSEPH HESTER	051 27 7817
MORNINGTON	MICHAEL MONCK	03 789 7997
MORWELL	JEFF SHEEN	051 33 9904
SHEPPARTON	ROSS FARRAR	058 25 1007
SMYTHESDALE	TONY PATTERSON	053 42 8815
SWAN HILL	BARRIE GERRAND	050.32.2838
TONGALA	TONY HILLIS	058 59 2251
TRARALGON	LEIGH DAWES	051 74 5552
WONTHAGGI	LOIS O'MEARA	056 72 1593

WA:

PERTH	IAIN MACLEOD	09 448 2136
GIRRAWHEEN	HANK WILLEMSEN	09 342 7639
KALGOORLIE	TERRY BURNETT	090.21.5212

CANADA - CoCo:

Ontario	Richard Hobson	416 293 2346
Toronto	Franz Lichtenberg	416 845 2889

special interest groups

TEACHERS' INTEREST GROUP

BRISBANE	BOB HORNE	07 281 8151

BUSINESS:

BRIZBIZ	BRIAN BERE-STREETER	07 349 4696

OS9 GROUPS:

NATIONAL OS9 USERS' GROUP

	GRAEME NICHOLS	02 451 2954
NSW		
SYDNEY		
BANKSTOWN	CARL STERN	02 646 3619
CARLINGFORD	ROSKO MCKAY	02 624 3353
GLADESVILLE	MARK ROTHWELL	02 817 4627
SYDNEY EAST	JACKY COCKINOS	02.344.9111
COOMA	ROSS PRATT	064 52 3065
QLD		
BRISBANE	JACK FRICKER	07 262 8869
VIC		
LATROBE VLY	GEORGE FRANCIS	051 34 5175
WA		
KALGOORLIE	TERRY BURNETT	090.21.5212

MC-10 CONTACTS:

LISMORE	BOB HILLARD	066 24 3089
SYDNEY	GRAHAM POLLOCK	02 603 5028
WARRNAMBOOL	GARY FURR	055 62 7440

TANDY 1000 / MS DOS:

NSW:		
GLADESVILLE	MARK ROTHWELL	02 817 4627
SYDNEY WEST	ROGER RUTHEN	047.39.3903
WYONG	JOHN WALLACE	043 90 0312
QLD:		
BRISBANE		
NORTH	BRIAN DOUGAN	07 30 2072
SOUTH	BARRY CAWLEY	07 390 7946
GOLD COAST	GRAHAM MORPHETT	075 51 0577
SA		
PORT LINCOLN	BILL BOARDMAN	086 82 2385
VIC:		
LA TROBE VALLEY	PETER FOLEY	051 74 5791
MELBOURNE	TONY LLOYD	03 882 4664

FORTH:

SYDNEY	JOHN REDMOND	02 85 3751

ROBOTICS:

GOLD COAST	GRAHAM MORPHETT	075 51 0577
SYDNEY	GEOFF FIALA	02 84 3172

CHRISTIAN USERS' GROUP:

COLLIE	RAYMOND L. ISAAC	097 34 1578

MSX

FRANKSTON	ALAN HASSELL	03 786 6290

300 BAUD BULLETIN BOARDS

SYDNEY:	
INFOCENTRE	02 344 9511
TANDY ACCESS	02 625 8071
THE COCOCONNECTION	02 618 3591
HOBART	
HUB	002 49 4405

VIDEOTEX SYSTEMS

VIATEL	01955
MOUSETEX	059 42 5528
VTX 4000	03 741 3295

TANDY INFO ON VIATEL

GOLDLINK	VIATEL *642#
PARIS RADIO	VIATEL *64266#
POWER CODE	VIATEL *64265#
TANDY	VIATEL *64261#

SOME TANDY USERS ON VIATEL

ALLAN BEALE	726353300
FRED BISSELING	648232630
JACK FRICKER	726288690
JOHN GRIGSBY	945872030
STUART HALL	939765790
BOB KENNY	665122050
JEFF LARSEN	705471270
IAIN MACLEOD	944821360
CHRIS NAGLE	689523360
RICHARD PANKHURST	280717870
ROSS PRATT	648230650
RICCAY SCHMAHL	298151500
ARTHUR SLADE	262289400
DARCY O'TOOLE	755105770
RON WRIGHT	352924510

DIRectory

Insertions in this Directory cost $160.00 for six months or $300.00 for twelve months per frame. Changes to Insertions incur a further charge.

If you sell Soft or Hardware for Tandy computers, you need to be listed in this quick reference guide.

Remember! Tandy owners READ this magazine!

N.S.W.

Central Coast

Computer Wizardry
P.O. Box 979,
Gosford N.S.W. 2250

- ★ *Educational Software*
- ★ *Communicating Software Hardware*
- ★ *Agents for Computer Hut*
- ★ *Agents for Speech Systems*
- ★ *Prompt, Courteous Service*
- ★ *Phone or Write for Catalogue*

Bankcard & Visa Card Welcome

043-24-7293

N.S.W. Gunnedah

Eather's Sports & Electronics
186 Conadilly St.,
Gunnedah N.S.W. 2380
(Tandy Dealer 9223)

Agricultural Computing Specialists

For Friendly Service to the man on the land, or for fast accurate help to the town dwellers

067-42-2230

N.S.W. Lismore

Decro Electronic Services
12 Carrington St.,
Lismore, N.S.W. 2480
(Tandy Dealer 9225)

Best range of Computers and Computing Equipment in Summerland.
Whether you live on the North Coast, or are just on holidays, you can't afford not to call and see us!

066-32-1896

South Australia

Morphettvale

NATIONAL
1stTIME ADVERTISED
THE PROFIT POTENTIAL IN YOUR PERSONAL COMPUTER
$25,000 PER MONTH IS POSSIBLE
For FREE Brochure
Ph (08) 382.7281
or send SSAE to
Mal. P. Bailey,
6 Bower Crt,
(Dept.7) Morphettvale
South Australia 5162

JUST RELEASED

CoCo3

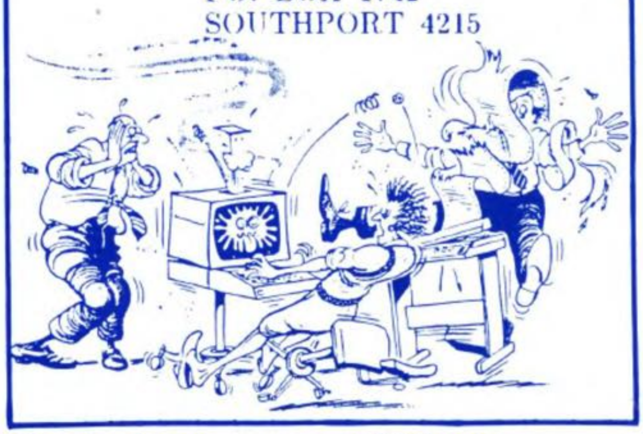
PART 3

DISK/TAPE

ARTIST	3AL	3ONO
CUBES	3HOW?	3YAHTZEE
SUNSET	3SNAKES	3MISSION

TAPE $10.00
DISK $16.00

Available from:
GOLDSOFT
P.O. BOX 1742
SOUTHPORT 4215



Victoria

Blackburn

D & L Wilson & Co Pty. Ltd.
6 Stafford St.,
Blackburn, Vic. 3130

Serving Melbourne's East
Software — Hundreds of titles
Hardware — Drives, Printers
Service — Upgrades

03-898-4521

Tasmania

Hobart

The Delbourgos.
15 Willowdene Ave.,
Sandy Bay
Hobart. Tas. 7005

- ★ *Expanded Basic — a better Basic for your CoCo*
- ★ *The Proportioner — a utility to provide equal gaps between proportional letters on Tandy's DMP200 printer*
- ★ *Mathematical function database*

002-25-3896

Queensland

Brisbane

Queensland Colour Software Supplies
P.O. Box 306
Clayfield. Qld. 4011

- ☆ *64k &128K upgrades*
- ☆ *80 column cards*
- ☆ *Y cables*
- ☆ *Games*
- ☆ *Terminal programs*

07 - 262 - 8869
A/H

CANADA

Computer Assembly Manuals
BIG BLUE SEED for IBMtm BUILDERS

- * parts list
- * placement diagrams
- * instructions

for assembling over 75 bare IBM-compat. cards.
Now includes guides for 640K, Turbo & AT MthBds!
$17.95 US

Money Order, VISA/MC
NuScope Associates, 1C
P.O. Box 742, Stn B
Willowdale, Ontario