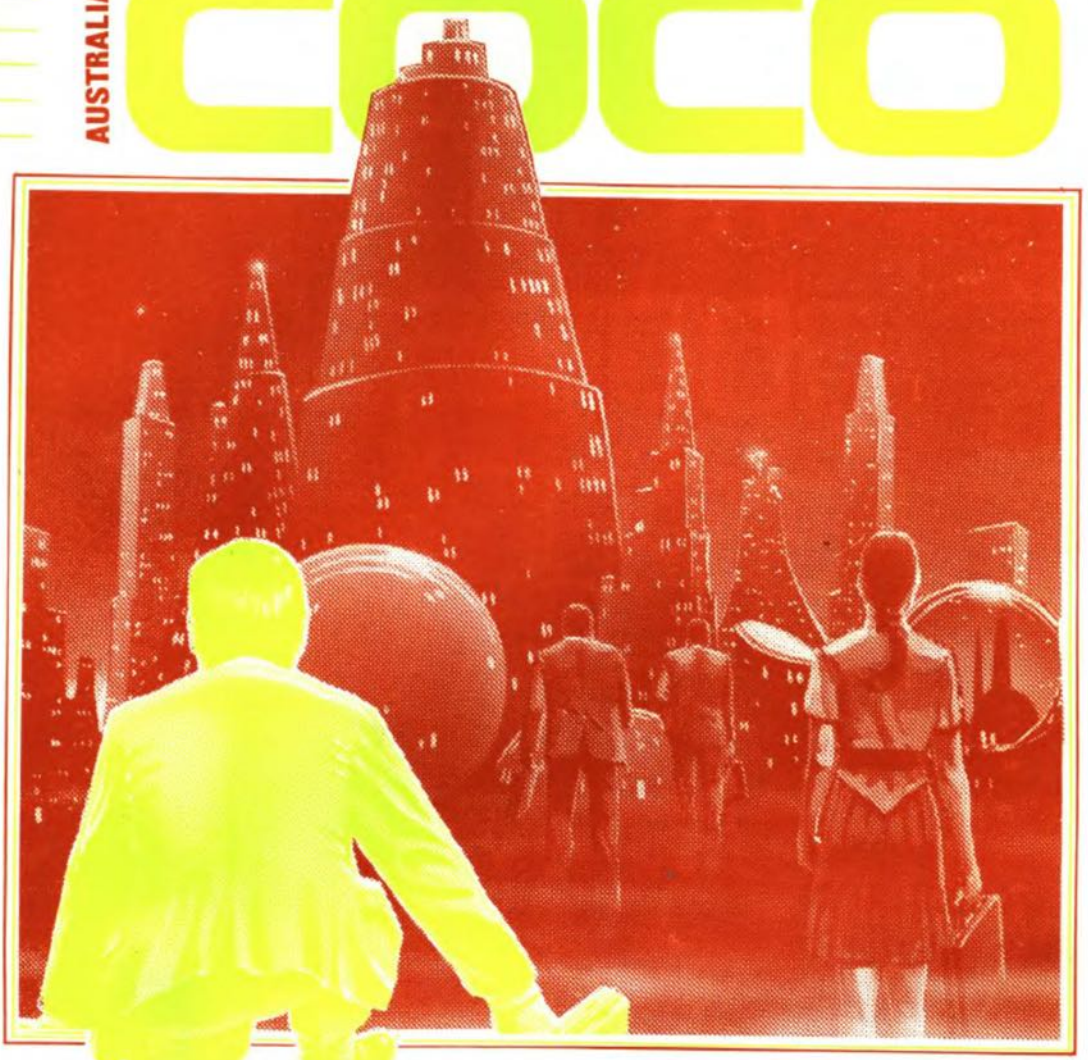
THE Magazine for experienced Tandy Color Computer Users!

$4.50

AUSTRALIAN

COCO

MAGAZINE



VOL3NO9
MAY87

This month's special features include:

- Mission Infiltrate —32KDECB Adventure
- Getting it altogether with — CoCo Merge
- Hardware — An Inverse Switch
- Orbital Prediction for the Amateur Radio User

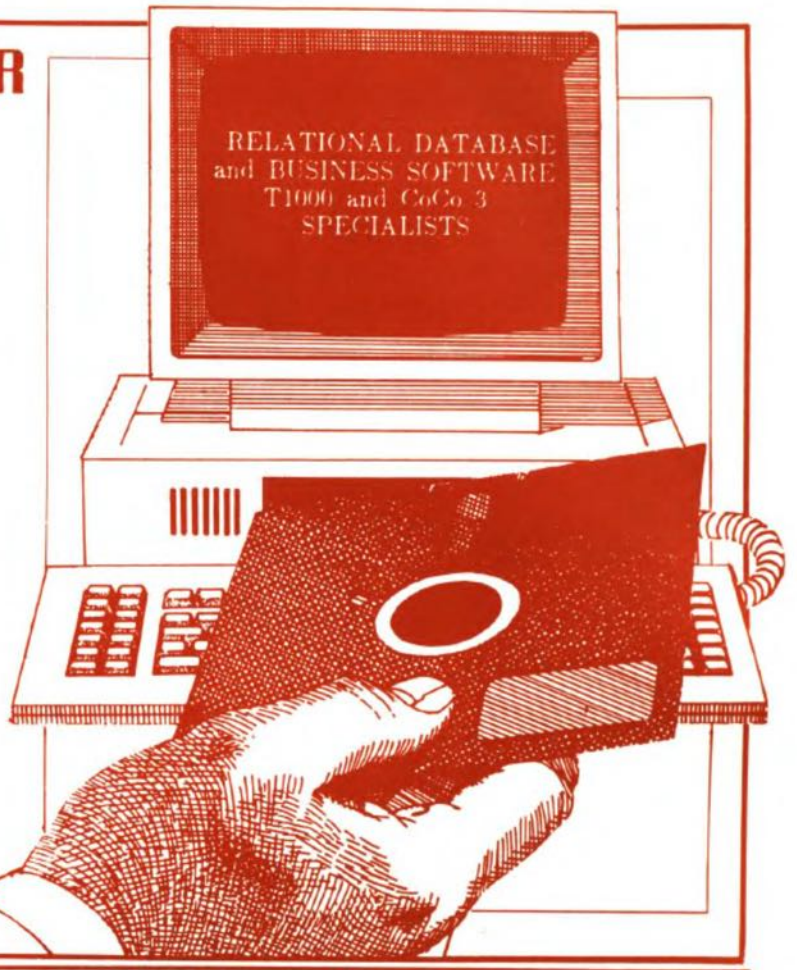
plus

- M/L Programming — Tutorial Part 3 by Mal Patrick
- Solo Scrabble

BLAXLAND COMPUTER SERVICES PTY. LTD.

134 GREAT WESTERN HIGHWAY,
BLAXLAND. N.S.W. 2774

TELEPHONE: (047) 39 3903



SUPER SALE

STOCK PURCHASED AT SPECIAL PRICES

6 only CoCo3's $400 ea

6 only Tandy Drives $450 ea

6 only RS232 Packs $160 ea
(for use with OS-9 level 2)

10 Mb Hard drives with controller for your CoCo2/3 $1299

20 Mb Hard drives $1599

512K upgrades for CoCo3 (won't work loose-premium 120 n/s 256K chips, gold edge contacts.) $199.00 inc. Disk Public Domain Software- RAM Disk Text Editor, Disassembler, Window Utilities, Coms Software & more!

SEEN GOLDLINK YET?

You can only access GOLDLINK on VIATEL if you have a videotex terminal program and a modem capable of operating at 1200/75 baud.

Now Goldsoft presents two great specials!

1 CoCo Owners: Modem, CoCotex Software and cable $289.00

2 TANDY 1000 or IBM PC Owners: Modem, VTex Software and cable $399.00

GOLDSOFT
PO BOX 1742
SOUTHPORT QLD 4215
(075) 51-0577

Telecom

VIATEL®

AUSTRALIA'S NATIONAL VIDEOTEX SERVICE

APPLICATION FORM

DATE OF APPLICATION / /

(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

16 PICTURES
A Screen Dump with some added features.

28 CoCo MERGE
Merges two or more programs from tape.

28 CoCo LIST CONTROL
Control LISTing speeds with the joystick.

29 PI–COS
Cassette Operating System.

35 CONVERSION of BASIC PROGRAMS for the CoCo 3.

39 RECOVER
continues to CLOAD after an I/O ERROR.

49 HAMSAT
A Ham Radio Utility.

61 TAPE ZAPPER
Modify M/L programs.

Music

38 PETER GUNN
For use with Orchestra 90cc.

56 ALL MY LOVING
Plus two other Beatle songs.

Graphics

7 PORSCHE
Looks good, but you can't drive it.

8 MOTORBIKE
This looks good too, but you still can't ride it.

48 CoCoLOGO
From the Melbourne Colour Computer Club.

Games

14 GUNFIGHT
A Reflex Game.

20 MISSION INFILTRATE
Become a Crack Commando Agent.

30 PYRAMINX
Rubiks Cube in the shape of a pyramid.

43 SOLO SCRABBLE
The Classic board game on the CoCo.

52 PENTOMINOES

62 SUBMARINE BATTLE

Business

BUSINESS
10 WORD–PRO
A Word Processor which make writing letters a breeze.

45 BILLS, DATA & DATES
Does everything except pay the bills.

Applications

40 PRETTY PRINTER PATTERNS
Allows you to create borders.

61 ECHO SONG
Plays songs with a touch of echo.

Hardware Mod.

34 INVERSE SWITCH
Reverses standard screen colours.

Plus: In a Nut Shell, Smarter Data Structures in Forth, Reviews, The Colourful CoCo 3.

Founder: Greg Wilson.
Managing Editor: Graham Morphett.
Editor: Alex Hartman.
Accounts: Annette Morphett.
Production: Paul Wynne
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 51 0577 or contact on Viatel *64213#
Registered Publication QBG 4009.

IN A NUT SHELL

Welcome Ken

We are very excited here at Goldsoft because Ken Allen, formerly Tandy's Computer Buyer, has just joined our company.

Ken's presence with Goldsoft will herald a new time of expansion for us into a number of new or expanded areas.

At this stage some of those areas are still being defined, but the bottom line is that we intend to deepen the services we provide for Tandy computer users, thereby continuing and expanding our commitment to you all.

Ken will no doubt have more to say on this issue at some later stage, but I believe that his move into the magazines gives you even more incentive to stay with Tandy when you are upgrading your computer.

OS-9 Level 2

Paris Radio Electronics won the race to be first to release OS-9 Level 2 in Australia.

Their first shipment hit just before the weekend of the 10th and was an immediate sellout!

I've no doubt that Jackie has restocked since then! I'm told Blaxland Computer Services have also received stocks. So now is the time to go look at this amazing operating system!

It is certainly not for the faint hearted - the manuals are quite large, but the study is worth it!

Tandy's stocks are now said to be due this month....

Blaxland's New Software Deals

Bruce from Blaxland Computer Services has been creating some growth of his own of late!

With a new store and new staff, he has almost single handedly replaced some of the verve that has been missing from the CoCo world over the last 12 months.

And now he has a few answers for OS-9 users in the form of hard drives, 512 K upgrade kits and public domain software for the CoCo 3 using OS-9 Level 2.

All in all a long haul - a lot of work has gone into all this, but we as CoCo 3 users will be the very lucky beneficiaries.

Wilfred Eggert

I almost forgot.

How could I forget Wilfred!

Wilfred attended the last CoCoConf and many of you were able to meet this very capable executive of the InterTan organisation there.

Currently, with Ken out of the scene, Wilfred is the stand in buyer for Computer Products.

He does a lot of other things at InterTan, so I'm sure he is very busy, but InterTan have not changed any policies. If you have a problem with your computer which you can't resolve with your local Tandy store, call Wilfred on Sydney 02 675 1222.

He will help.

CoCoMax Changes

We published a fix for CoCo 3 owners who own CoCoMax last month.

It appears that the fix does not work on some computers.

Anyone with further info - HELP!!

CoCo 3 Disk #3

This disk is now available and continues our commitment to provide you with software for your CoCo 3.

The programs are from both Softgold and CoCo magazines over the last few months.

So if you don't buy Softgold on disk or CoCoOz, at least purchase this one, and catch up!

There is a tape version available also, and either disk or tape is available for $16.50.

Conf '87

Following on from last month's release of information regarding Conf '87, we'd like to hear from those who would be interested in riding with us on the bus we will be hiring to go from Brisbane to Bundeena.

We are also prepared to organise a similar facility for the Melbournians.

Keep in mind that you will probably need at least the preceeding Friday free for travel on this bus.

Costs will be kept as low as possible. They should be cheaper than alternative public means.

We need to know if you want to go to Bundeena this way THIS MONTH, because we especially need to know what we have to do for the Victorians.

We can probably arrange for pickups along the logical route of the buses travels.

Competitions

Again I would remind you that the programming competitions are in full swing.

Martha's Grafix competition is attracting some nice entries, as is the utilities competition.

In the Games area, I have to admit that it is very much anyone's competition yet. So come on, let's see those entries flowing in!

■

the GOLDSOFT WISHBOOK

The following products are available on order from us.

To order, contact us by phone, Viatel or letter, giving us your name, address, phone number and credit card number, as well as the Item # shown beside the product as listed below.

All items include post and packing.

CoCo Hardware

Item #	Description	Cost
B 001	512K upgrade kit for CoCo 3 owners	$199.00
B 002	10 Mb Hard drive inc software	$1299.00
B 003	Coming! CoCoNet! Networking extrodinaire!	TBA
G 001	The CoCoConnection - Use your CoCo to control models, alarms - anything electrical	$206.00
G 002	Video Amplifier with sound - attach your CoCo 1 or 2 to a Video monitor	$35.00
G 003	The Probe - A temperature sensing unit you plugin to the joy stick port.	$49.95
G 004	64K Upgrade Kit - upgrade your CoCo2's memory to 64K!	$55.00

CoCo Software

B 1001	The Wiz! The ultimate OS-9 L2 Coms package! Multi Windowing, VT52 Emulator, 300-19200 baud, RS 232 protocol	$159.00
B 1002	IMS - Relational Data Base written in 4GL & VERY fast! OS-9 L2	$299.00
G 1001	Say the Wordz - two Curriculum based speller programs for your Tandy Speech / Sound Pack (32K ECB)	$29.95
	The CoCo 3 Tape/Disk	
G 1002	# 1	$16.00
G 1003	# 2	$16.00
G 1004	# 3	$16.00
	The Best of CoCoOz	
G 1005	# 1 Education	$16.00
G 1006	# 2 Part 1 16K Games	$16.00
G 1007	# 2 Part 2 32K Games	$16.00
G 1008	# 3 Utilities	$16.00
G 1009	# 4 Business	$16.00
G 1010	# 5 Adventure Games	$16.00
G 1011	# 6 Preschool Edn	$16.00
G 1012	# 7 Graphics	$16.00
G 1013	# 8 16K Games	$16.00
G 1014	# 9 32K Games	$16.00
G 1015	#10 Education	$16.00
G 1016	#11 Education (Disk only)	$16.00

Tandy and IBM PC Hardware

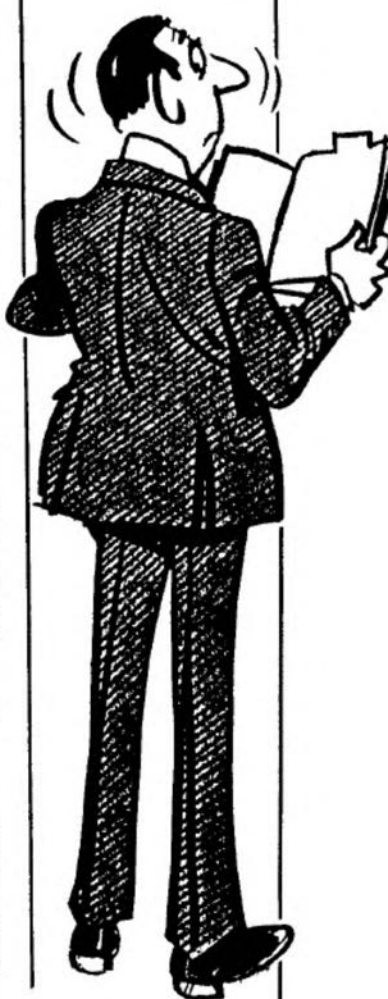
Item #	Description	Cost
Q 001	Colour Monitor (DTX 2001)	$680.00
Q 002	Mono Monitor	$190.00
Q 003	Mouse	$90.00
Q 004	80286 Speed Card	$550.00

Tandy and IBM PC Software

Business		
Q 1001	dBase II	$1043.00
Q 1002	dBase III	$1470.00
Q 1003	Sidekick	$215.00
Q 1004	Turbo Pascal 8087	$261.00
Q 1005	Turbo Pascal BCD & 8087	$244.00
Q 1006	Crosstalk	$306.00
Q 1007	Lotus 123	$1054.00
Q 1008	Wordstar 2000+	$927.00
Fun		
Q 1020	Ancient Art of War	$96.00
Q 1021	Print Shop	$119.00
Q 1022	Gato	$68.00
Q 1023	Sargon III	$96.00
Q 1024	Zork I	$79.00
Q 1025	Zork II	$79.00
Q 1026	Zork III	$79.00
Q 1027	Trinity	$79.00
Q 1028	Ballyhoo	$79.00
Q 1029	Hitch Hicker's Guide to the Galaxy	$79.00
Q 1030	Crossword Magic	$68.00
Q 1031	The American Challenge	$68.00
Q 1032	Balance of Power	$89.00
Q 1033	Racter	$79.00
Q 1034	Jet	$114.00
Q 1035	Moonmist	$79.00
Q 1036	Shanghai	$68.00
Q 1037	Championship Golf	$89.00
Q 1038	Borrowed Time	$68.00
Specials!		
Q 1050	Side Print	$72.00

Miscellaneous Items

Item #	Description	Cost
G 2001	Box of 10 DSDD Disks	$20.00
	10 Boxes plus (per box)	$16.10
G 2002	Blank C30 Cassettes	$2.00
	12 Cassettes	$18.00
G 2003	Tape cases, 12 for	$5.00
G 2004	Help - Manual for CoCo	$9.95
G 2005	MiCo Help - for MC 10	$9.95



REVIEWS

POWER DOS

Technical
Classification: Utility
Cost: $???
RAM: 64K DECB / EPROM
Supplier: Arty Tron Marketing,
P.O. Box 68,
East Brunswick,
3057
Extras: None.

What is it?
Power Dos, like SDos & ADos, enhances your Disk Operating System. But unlike the other DOS's, you are given a DOS chip to replace your current DOS chip, thereby replacing all previous commands with new ones.
After replacing your old DOS chip with the new DOS chip, you are given an extra 25 commands above your normal commands including 2 modified commands. Here is a list of these commands and what they do:
AUTO: Automatic line numbering.
BAUD: Set your own baud rate.
COLD: Cold starts the computer.
DIR: Will pause after 13 directory entries. Also shows free granules on disk.
DOS: Enable OS-9.
DPEEK: Returns a 16-bit value.
DPOKE: Stores a 16-bit value.
FAST: Like a POKE65497,0.
HIDE: Hides your directory.
LOCK: Protects a file from destruction on disk.
OLD: Restores a NEW
PDIR: Prints your directory on a printer.
RAM: Access 64K.
RATE: Alters the stepping rate of your drives.
RUNM: LOADM & EXEC a binary file.
SCAN$: Replaces
10 A$=INKEY$:IF A$=""THEN10
SCOPY: Copy multiples files on a disk.
SLIST: Better than <SHIFT><@>.
SLOW: Like POKE65494,0.
SYS: Defines number of tracks per drive.
SYSTEM: Sets number of drives & tracks for each drive.
TRK: Prints number of tracks that will be formatted for each drive.
UNHIDE: Restores your directory.
UNLOCK: Restores the LOCK command.
VERIFY: Verifies the save when saving a file.

Problems?
* None, really. It comes back to what I said in the other DOS packages (December '86 Australian Rainbow Magazine); if you find some things in there attractive, then by all means compare them all and then buy. The good advantage with this one is that if you really like it, you don't have to go to the trouble of having to burn it onto an EPROM.

General Comments!
All in all fairly good! Like I said in the other DOS packages; great if you find them interesting or like having a better DOS than you already have.

Rating...
Workability: 6/10
Usefulness: 6.5/10
Instructions: 7/10
Overall: 6.5/10

PROGRAMMERS' UTILITY

by Alex.

Technical Dept.
Classification: Utility
Cost: $29.95 Tape or Disk
System: 16 - 64K CoCo 2 or 3
Supplier: Economy Computing,
P.O. Box 595,
Taree, 2430
(065) 50 7275
Extras: Nil.

What It Is Dept.
This is an extract from the instruction booklet:

"Programmers' Utility" is designed to make writing BASIC programs quick and easy and gives you a number of time saving features at the press of a button. It is written entirely in Machine Language.
Programmer's Utility features:

* Repeat last entry
* Edit on error
* Screen editor
* LIST up or down
* Auto line numbering
* Binary converter
* Forced CLOAD
* Dump screen to printer
* Merge programs
* Memory display
* Orange screen on/off
* Program Summary
* Recover/unNEW
* Search program
* PCLEAR 0
* Single key command entry"

End of Quote.

General Comments Dept.
Programmers Utility is a great little package for those efficient people out there. Upon receiving the cassette you also get a small 11 page booklet. The booklet in itself is easy to understand and all new functions and commands are well explained.
Some commands, I find, are quite good and useful. For example,
* you can LIST as fast or as slow as you want, upwards or downwards.
* Repeat your last command.
* Force CLOAD a program (this option keeps trying to load a program - ie instead of an I/O error being generated, it keeps searching for the start of a program.
* PCLEAR0: getting the most amount of memory without losing the current program in memory.
* Keysaver: By pressing 'CLEAR' and, say, the letter "A", you get the command "RUN" on the screen. This can be customized to suit your tastes by adjusting the BASIC program following "Programmers' Utility"

I find it a good program and recommend it to anyone who wants to write their programs fast.

Rates 9/10

PORSCHE

GRAPHICS

by Robert Davies

PORSCHE WAS ORIGINALLY programmed by Robert Davis (April '85 US. Rainbow Magazine) and has been converted for the CoCo 3 by Victor Koss.

After the Porsche has been drawn, pressing the SPACEBAR will 'paint' the Porsche in all of the 64 colours available.

The Listing:

```
0 'COCO 3 VERSION OF "PORSCHE"
1 'IN APR 1985  US RAINBOW
2 'CONVERTED TO COCO 3 BY
3 'VICTOR KOSS
4 '
5 GOTO10
6 SAVE"191:3":END
10 ON BRK GOTO 1080
20 PALETTE CMP
30 '*****************************'
40 '*    PORSCHE 930 TURBO      *'
50 '*     BY ROBERT DAVIS       *'
60 '*         MAY 1984          *'
70 '*****************************'
80 HSCREEN 2:PALETTE 0,0
90 HCLS0:HCOLOR 4,0
100 HLINE(0,10)-(320,180),PSET,B
110 HLINE(0,134)-(320,134),PSET
120 '** DECK LID **'
130 HLINE(23,90)-(45,81),PSET
140 GOTO220
150 '** LETTERING "PORSCHE" **'
160 HDRAW"BM60,164;U8R6D4L6;BM68
,164;U8R6D8L6;BM76,164;U8R6D4L6R
5D2R1D2;BM84,164;R6U5L6U3R6;BM98
,164;L6U8R6;BM100,164;U8D4R6U4D8
;BM114,164;L6U5R5L5U3R6"
170 '** LETTERING "930" **'
180 HDRAW"BM140,164;U8L6D5R6;BM1
42,164;R6U5L4R4U3L6;BM156,164;L6
U8R6D8"
190 '** LETTERING "TURBO" **'
200 HDRAW"BM177,164;U8L3R6;BM182
,156;D8R6U8;BM190,164;U8R6D4L6R5
D2R1D2;BM198,164;U8R5D3L5R6D5L6;
BM206,164;U8R6D8L6"
210 GOTO850
220 '** TIRES AND WHEELWELLS **'
230 HCIRCLE(67,115),19
240 HCIRCLE(204,115),19
250 HCIRCLE(67,115),12
260 HCIRCLE(204,115),12
270 HCIRCLE(67,115),10
280 HCIRCLE(204,115),10
290 HCIRCLE(67,115),2
300 HCIRCLE(204,115),2
310 HCIRCLE(67,115),23,,1,.5,0
320 HCIRCLE(204,115),23,,1,.5,0
330 '** STRIPE AT BOTTOM **'
340 HDRAW"BM85,116R5BM180,116R5"
:HLINE(90,114)-(180,116),PSET,BF
350 '** FRONT LIGHT **'
360 HDRAW"BM240,98;M236,85;L3;M2
36,98;R3"
370 '** HOOD AND WINDSHIELD **'
380 HDRAW"BM173,78;M139,59;G3M16
1,81;R3E1R3E1R2E1;BM233,85;H1L2H
1L2H1M165,82;BM173,78;M210,82"
390 '** ARCS UNDER STRIPE **'
400 HCIRCLE(96,116),6,,1,.25,.5
410 HCIRCLE(176,116),6,,1,0,.25
420 '** AIR INTAKE **'
430 HCIRCLE(67,115),27,,1,.87,0
440 HDRAW"BM84,95;R6F1R2D1F3D1F2
D1F3D1F4R1F3"
450 '** REAR WINDOW **'
460 HDRAW"BM46,80;M85,60;D3;M56,
77H1L1"
470 '** BOTTOM OF BODY **'
480 HLINE(96,122)-(176,122),PSET
490 '** WHALE TAIL **'
500 HDRAW"BM24,90;U4H3L5H2U3E2R2
0;M45,80;G2L21"
510 '** REAR BUMPER & LIGHT **'
520 HDRAW"BM48,104;L32U4R12F4R4G
8L14L2H1U1R7U4L7D4U5E1R10D8L6D1F
2R1F2R1F2R20NR5U3"
530 '*ARC BETWEEN LIGHT & TAIL*'
540 HDRAW"BM16,100;U2E1U1E1U1E3R
1"
550 '** FRONT BUMPER **'
560 HDRAW"BM224,102;ND3R24L4H3F3
R4F2D1L9D4R9U4D4G2L10U9D9L11R19G
3D2F1D1G1L12U1L2U1L2U1L1NL4U5R3F
6R7"
570 '** ROOF **'
580 HDRAW"BM85,60;R2E1R3E1R43F1R
1F1"
590 '** SIDE WINDOW **'
600 HDRAW"BM155,81;U4L5D5U5L2D5R
5D3R4U2H2;BM150,77;M136,66;H1L1H
1L39G1L2G1L1M69,76;BM72,82;R74NR
5L8M130,64;BM100,82;M95,64R3M103
,82"610 HCIRCLE(72,78),4,,1,.25,
.70
620 '** DOOR **'
630 HDRAW"BM103,82;G3D2M109,107;
R1D1R1D1R56E1R1E1U3E1U4E1U2H1U4H
1U3H1U1H1L1H1L9;BM108,88;R6L6U1R
2D2L2"
640 '** SPOKES (BACK WHEEL) **'
650 HDRAW"BM67,115;M66,106;R1M67
,115;M68,106;M67,115;M58,115;U1M
67,115;M59,113;U1M67,115;M77,115
;U1M67,115;M77,113;U1M67,115;M61
,121;F1M67,115;M63,123;M67,115;M
73,123;E1M67,115;M75,121;M67,115
"
660 '** SPOKES (RIGHT WHEEL) **'
670 HDRAW"BM204,115;M203,106;R1M
204,115;M205,106;M204,115;M196,1
15;U1M204,115;M196,113;M204,115;
M214,115;U1M204,115;M213,113;M20
4,115;M198,121;F1M204,115;M200,1
23;M204,115;M210,123;E1M204,115;
M211,120;M204,115"
680 HPAINT(24,80),4,4
690 HPAINT(28,108),4,4
700 HPAINT(16,108),4,4
710 HPAINT(248,108),4,4
720 HPAINT(232,108),4,4
730 HPAINT(228,114),4,4
740 HPAINT(95,104),4,4
750 HPAINT(101,80),4,4
760 HPAINT(237,92),4,4
770 PALETTE 2,9
780 HPAINT(12,12),2,4
790 HPAINT(130,130),2,4
800 PALETTE 1,3
810 HPAINT(10,150),1,4
820 PALETTE 9,36
830 HPAINT(25,102),9,4
840 GOTO150
850 PALETTE 10,44
860 HPAINT(130,100),10,4
870 HPAINT(200,85),10,4
880 HPAINT(200,95),10,4
890 HPAINT(67,95),10,4
900 HPAINT(180,80),10,4
910 HPAINT(158,82),10,4
920 HPAINT(155,82),10,4
930 HPAINT(30,85),10,4
940 HPAINT(30,117),10,4
950 HPAINT(240,114),10,4
960 HPAINT(150,118),10,4
970 HCOLOR4,0:HPRINT(0,24),"By R
. Davis.  Converted by Victor Ko
ss."
980 HCOLOR0,0:HLINE(1,1)-(318,9)
,PSET,BF
990 HCOLOR10,0
1000 HPRINT(1,0),"Press <SPACE>
to change to next color "
1010 X=0
1020 A$=INKEY$:IFA$=CHR$(32)THEN
1030ELSE1020
1030 PALETTE 10,X
1040 X=X+1
1050 IF X>63 THEN X=0
1060 GOTO1020
1070 GOTO1070
1080 PALETTE CMP:END
```

■

MOTORBIKE

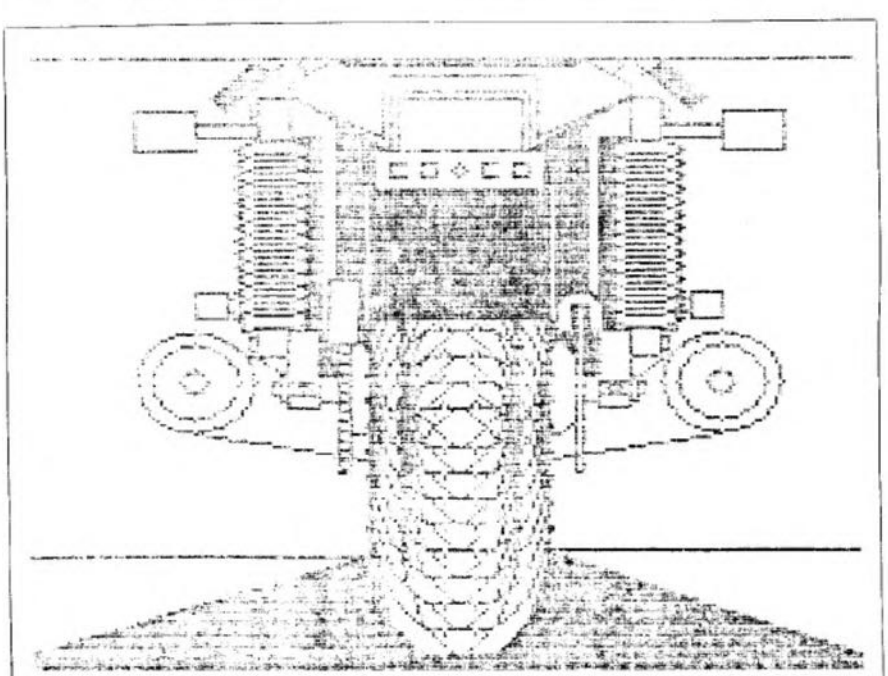
16K ECB
GRAPHICS

by Erin Kersten

MOTORBIKE IS AN entry in the graphics competition. In actual fact it is two graphics programs; one program depicting the motorbike from the front, and the second graphics program viewed from the back.

Listing One:

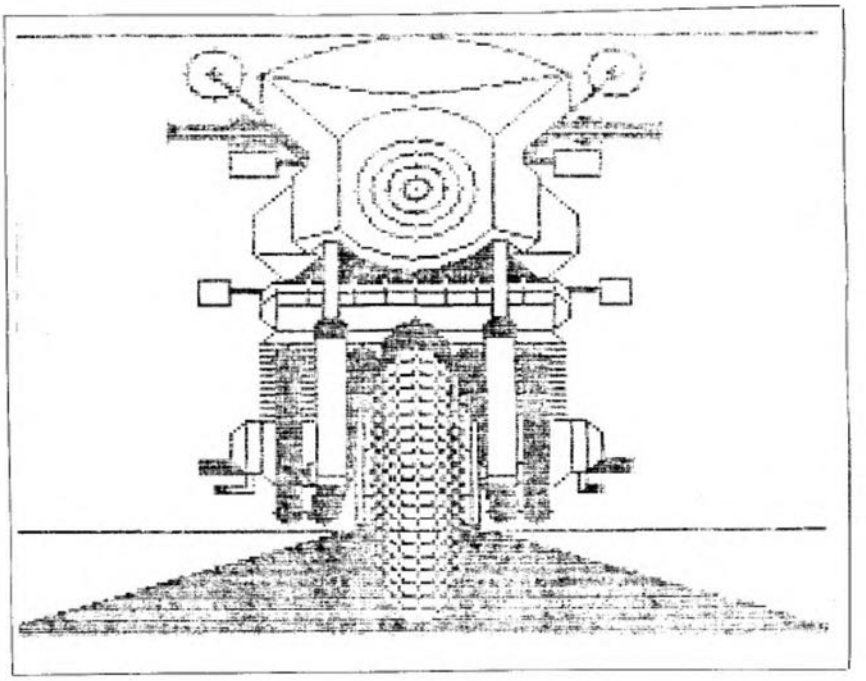
```
  0 GOTO10
1 '** MOTORBIKE - REAR  VIEW ***
   ****** ERIN KERSTEN *********
3 SAVE"209:3":END'6
10 PMODE4
20 PCLS1
30 SCREEN1,1
40 COLOR0
50 LINE(0,0)-(255,192),PSET,B
60 FORI=128TO140 STEP 8
70 FOR K=82 TO 152 STEP8
80 CIRCLE(I,K),20,,2,.0,.0
90 NEXT K,I
100 DRAW"BM108,88L4D28R4BM154,88
R4F4D20G4L4
110 DRAW"BM162,76D52R2U52L2BM164
,98R4D8L4BM104,88D40L3U40R3
120 FOR A=102 TO 104
130 FOR S=88 TO 128 STEP 3
140 LINE(A,S)-(A,S),PSET
150 NEXT S,A
160 DRAW"BM102,98L6D8R8BM96,104L
8U4R8BM88,100U16R8D24L8U4
170 DRAW"BM92,92BL4U8R4U4D4L4D8L
8U8L4U4
180 FOR D=28 TO 80 STEP 4
190 CIRCLE(84,D),10,,.4
200 NEXT D
210 CIRCLE(92,102),2
220 DRAW"BM88,98L2D8R2L2U2L2U4R2
BM80,86L2D4R2BM106,88L8U20R8D20
230 DRAW"BM72,80L8U8R8D8U2R4L4U4
R2
240 CIRCLE(64,100),16
250 CIRCLE(64,100),12
260 CIRCLE(64,100),4
270 DRAW"BM157,84R4U8R4D8R4U8H4L
4G4D8
280 DRAW"BM168,104D4R8U24L8D16R8
D4L8R8D2R2U2R2U4L2D4U6L2
290 DRAW"BM176,92R8U8D2R2D4L2U6R
4U4D4L4BL8L4U4
300 FOR D=28 TO 80 STEP 4
310 CIRCLE(180,D),10,,.4
320 NEXT D
330 DRAW"BM192,80R8U8L8D2L4BD4R4
D2U8
340 CIRCLE(172,102),2
350 CIRCLE(200,100),16
360 CIRCLE(200,100),12
370 CIRCLE(200,100),4
380 DRAW"BM116,12R32D16L32U16
390 DRAW"BM110,40R44U12L44D12
400 LINE(109,40)-(155,80),PSET,B
F
410 DRAW"BM80,24U12D2L2D4R2U6R8D
12
420 DRAW"BM176,24U12R8D2R2D4L2D6
U12
430 DRAW"BM114,28U17E1R34F1D17
440 LINE(114,28)-(88,11),PSET
450 LINE-(112,24),PSET
460 DRAW"BM112,24U18E1R38F1D18
470 LINE(152,24)-(176,12),PSET
480 LINE-(149,28),PSET
490 DRAW"BM88,12U4E12F2BM176,12U
4H12G2L60H2R64
500 DRAW"BM88,24R8D60R2U16R2U48
510 PAINT(94,57),0
520 DRAW"BM108,38R2L3D45
530 PAINT(104,40),0
540 DRAW"BM176,24L8D50H3U54
550 DRAW"BM155,38R2D45
560 PAINT(169,44),0
570 PAINT(158,56),0
580 LINE(68,85)-(88,98),PSET
590 LINE(64,116)-(108,124),PSET
600 LINE(192,87)-(176,100),PSET
610 LINE(200,116)-(156,124),PSET
620 DRAW"BM80,24L16D4U12L16D12R1
6U8R16
630 DRAW"BM184,24R16D4U12R16D12L
16U8L16
640 LINE(L,K)-(L,K),PSET
650 DRAW"BM80,0G12F4E16G4F1H7
660 DRAW"BM184,0F12G4H16F4G1E7
670 DRAW"BM92,4H2E2BM172,4E2H2
680 DRAW"BM118,32L4D4R4BR4U4R4D4
L4R4BR4BU2E2F2G2H2BD2BR8U4R4L4D4
R4BR4U4R4D4L4
690 PAINT(72,12),0:PAINT(192,12)
,0:PAINT(98,92),0:PAINT(107,86),
0:PAINT(158,86),0:PAINT(166,90),
0:PAINT(170,82),0:PAINT(170,78),
0:PAINT(124,1),0
700 LINE(0,192)-(108,154),PSET
710 LINE(255,192)-(154,152),PSET
720 LINE(0,154)-(110,154),PSET
730 LINE(154,152)-(255,152),PSET
740 PAINT(128,189),0
750 GOTO 750
```



■

Listing Two:

```
0 GOTO10
1 '** MOTORBIKE - FRONT VIEW ***
   ***** ERIN KERSTEN **********
3 SAVE"209A:3":END'6
10 PMODE4
20 PCLS1
30 SCREEN1,1
40 COLOR0
50 LINE(0,0)-(255,192),PSET,B
60 FORI=128TO128
70 FOR K=112 TO 160 STEP4
80 CIRCLE(I,K),12,,2,.0,.0
90 NEXT K,I
100 DRAW"BM116,128G2D12F2H2D10L2
U32R2D12BM140,128F2D12G2E2D10R2U
32L2D12
110 DRAW"BM144,134R2D4L2R2D2R2D1
U1R4D1U1R2U50H2L4G2D4R8L8D46R8U1
R2U2R1U2L1U2L2R2D4BL2L8
120 DRAW"BM112,134L2D4R2L2D2L2D1
U1L4D1U1L2U50E2R4F2D4L8R8D46L8U1
L2U2L1U2R1U2R2L2D4BR2R8
130 DRAW"BM108,88U24L4D24R4BR40U
24R4D24
140 CIRCLE(128,46),28,,1,.13,.39
150 CIRCLE(128,48),16
160 DRAW"BM104,64G4H4U20E4U4H12U
12F20D28BM152,64F4E4U20H4U4E12U1
2G20D28
170 CIRCLE(128,42),28,,1,.13,.39
180 CIRCLE(128,50),28,,1,.64,.89
190 LINE(108,60)-(96,64),PSET:LI
NE(148,60)-(160,64),PSET
200 CIRCLE(128,74),74,,1,.67,.85
210 LINE(88,12)-(108,16),PSET:LI
NE(168,12)-(148,16),PSET
220 CIRCLE(128,13),24,,.25,.12,.
40
230 PAINT(124,96),0:PAINT(124,93
),0:PAINT(122,96),0:PAINT(121,10
1),0:PAINT(135,95),0:PAINT(135,1
01),0:PAINT(138,104),0:PAINT(118
,104),0
240 DRAW"BM100,40L8D4L12U8R12D4R
8L8U1R8BM156,40  R8D4R12U8L12D3L
8
250 DRAW"BM100,40H8L28D1U6D1R16U
2D8U6R12BM80,28E2R1U1R2D1L2R3F2D
4G2L4H2
260 PAINT(84,29),0:PAINT(84,27),
0:PAINT(84,33),0
270 DRAW"BM88,29L20D2R20":PAINT(
66,29),0:DRAW"BM88,28U4D10R4U6":
PAINT(89,29),0:PAINT(89,26),0:PA
INT(89,33),0
280 COLOR1:LINE(80,30)-(90,30),P
SET,B
290 COLOR0:DRAW"BM84,34D2":PAINT
(96,34),0
300 DRAW"BM156,40E8R28D1U6D1L16U
2D8U6L12":PAINT(172,30),0:DRAW"B
M176,28H2L4G2BD4F2R4E2":PAINT(17
2,27),0:PAINT(172,33),0:PAINT(18
4,30),0
310 COLOR1:LINE(165,30)-(187,30)
,PSET,B
320 COLOR0
330 DRAW"BM92,92H4U4E4U1D13R10L1
0U8R12U4L4U1D5U4L8R12BR4R4U1D1R8
U1D1R8U1D1R8U1D1R8U1D1R4BR4R4U1D
1R8U1D1R4D4G4L10R10U12D4L8U4D4L4
BL4L4U4D4L8U4D4L8U4D4L8U4D4L8U4D
4L4R4BD8L2R12BR12R12
340 DRAW"BM92,92G4R14BR8R9BR16R8
BR10R12H4
350 FOR A=88 TO 92
360 FOR B=98 TO 119 STEP 2
370 LINE(A,B)-(A,B),PSET
380 NEXT B,A
390 CIRCLE(95,101),4:CIRCLE(95,1
48),4:DRAW"BM92,102D48R6U48":PAI
NT(95,101),0:PAINT(93,108),0:PAI
NT(95,148),0
400 DRAW"BM92,120L8G4D8F4U16D16R
4U16D16F4
410 FOR A=96 TO 102
420 FOR B=98 TO 120 STEP 2
430 LINE(A,B)-(A,B),PSET
440 NEXT B,A
450 DRAW"BM96,140R6
460 CIRCLE(104,148),4:LINE(107,1
50)-(110,140),PSET
470 LINE(100,150)-(102,140),PSET
480 CIRCLE(114,101),4
490 LINE(118,101)-(114,120),PSET
500 PAINT(114,101),0:PAINT(114,1
06),0:PAINT(104,148),0:PAINT(105
,144),0
510 FOR A=163 TO 167
520 FOR B=98 TO 119 STEP 2
530 LINE(A,B)-(A,B),PSET
540 NEXT B,A
550 CIRCLE(159,101),4:DRAW"BM156
,101D48R6U48":CIRCLE(159,148),4:
PAINT(159,147),0:PAINT(159,138),
0:PAINT(159,101),0:PAINT(159,150
),0
560 FOR A=154 TO 162
570 FOR B=98 TO 119 STEP 2
580 LINE(A,B)-(A,B),PSET
590 NEXT B,A
600 CIRCLE(142,101),4:CIRCLE(150
,148),4
610 LINE(147,146)-(145,140),PSET
:LINE(154,146)-(152,140),PSET:LI
NE(138,100)-(142,120),PSET
620 PAINT(142,101),0:PAINT(150,1
46),0:PAINT(142,106),0:PAINT(150
,142),0
630 DRAW"BM164,120R8F4D8G4U16D16
L4U16D16G4;BM104,68G8L2H8U12E9
640 DRAW"BM152,68F8R2E8U12H9
650 DRAW"BM88,68R14BM168,68L14
660 DRAW"BM88,80D2U2E3R1U1D1R8U1
D1R4BR4R4U1D1R8U1D1R8U1D1R8U1D1R
8U1D1R4BR4R4U1D1R8U1D1R1F3D2
670 PAINT(112,72),0:PAINT(102,73
),0:PAINT(153,72),0:PAINT(108,91
),0:PAINT(150,91),0
680 COLOR1:DRAW"BM100,77U1BM112,
77U1BM120,77U1BM128,77U1BM136,77
U1BM144,77U1BM156,77U1
690 COLOR0
700 DRAW"BM80,132L8D4R12BM176,13
2R8D4L12
710 PAINT(180,134),0:PAINT(76,13
4),0
720 DRAW"BM76,140D2R4U2D2R6U6L2D
4L8BM172,136D4R4D2L4U2D2L2U6
730 PAINT(78,141),0:PAINT(174,14
1),0
740 DRAW"BM88,80L8D4L8U8R8D3R8BM
168,80R8D4R8U8L8D3L8BM92,28H16D1
F16BM164,28E16D1G16
750 CIRCLE(76,12),8:CIRCLE(180,1
2),8:CIRCLE(76,12),2:CIRCLE(180,
12),2
760 FOR A=0 TO 12 STEP 4
770 CIRCLE(128,48),A
780 NEXT A
790 LINE(0,192)-(116,152),PSET
800 LINE(256,192)-(140,152),PSET
810 LINE(0,154)-(256,192),PSET,B
820 PAINT(128,190),0
830 GOTO 830
```



■

WORD-PRO

32K ECB
BUSINESS

by Neville McDonald

FOR THE PAST year or so, I have been toying around with my own version of a Word Processing program.

I had a look at a few programs going around, but found them much too complicated to use, or you have to learn too many new symbols to get the thing to work properly.

At last, I have come up with a program that suits my needs as an occasional letter writer. It offers a few essential editing options and also takes the hassle out of worrying about print width (This program was written for the DMP-110 printer).

I really do not know how suitable others may find it because of it's length and the time that would be required to type it in. Anyhow, if anyone is interested in using it or looking at it for ideas of their own, I would be happy to supply copies of the program to people who send a blank tape to record it onto.

I am not sure of the system requirements, other than I know it doesn't fit in an 16KECB CoCo (I have a 64KECB CoCo).

Anyhow, here is a list of what it can do.

The following are the parameters of the program:-

1) 200 lines of input, with each line holding a maximum of 255 characters.
2) Choice of colour screen, either green or orange.
3) Load previously saved listing from tape.
4) Save listing to tape.
5) Scroll the listing, either automatically or manually with the option of commencing scrolling from a certain line.
6) Alter words or characters.
7) Inserting words or characters.
8) Deleting words or characters.
9) Inserting blank lines so as to seperate text.
10) Delete whole lines of text.
11) Move a line or a group of lines to a different position in the text.
12) Printout, with a choice of three (3) printing styles.
13) Extend an already existing file. ie continue on as though you had not left the input coutine.
14) Automatic prevention of wrap-around during printing with the DMP-110 printer.
15) Proofing copies and multiple copies of the final text are available.
16) Automatic extension of a sentence by designating the LAST character to be a " ", causing the computer to join the next line of text to the current one being printed.
17) Selectable printing baud rate, either 600 or 1200.
18) Automatic warning when reaching the limit of input when line 190 is reached.

As I said, it is a bit long, but it does what I want it to do and that is the main thing!

The Listing:

```
0 GOTO5
1 '********** WORD-PRO *********
   ***** NEVILLE MCDONALD ******
3 SAVE"118:3":END'3
5 CLS(0):PRINT@32*7+9,"WORD PROC
ESSOR.";
10 PRINT@32*9+7,"BY neville mcdo
nald.";
15 FOR Y=1 TO 3000:NEXT Y
20 BAUD=87:STYL=19
25 CLS:PRINT@32*5,"DO YOU WANT T
O USE THE  GREEN   SCREEN OR WOU
LD YOU LIKE TO     USE AN ORANGE
 SCREEN ?"
30 PRINT:PRINT" 1 - GREEN":PRINT
" 2 - ORANGE"
35 T$=INKEY$:IF T$="1" THEN 50
40 IF T$="2" THEN 45 ELSE 35
45 SCREEN 0,1:POKE 359,57:GOTO 5
5
50 SCREEN 0,1:POKE 359,126
55 CLS(0):PRINT@32*5+5,"I=instru
ctions  M=menu";
60 PRINT@32*7+5,"S=start input";
65 T$=INKEY$:IF T$="I" THEN 150
70 IF T$="M" THEN 80
75 IF T$="S" THEN 235 ELSE 65
80 '
85 '
90 '    **MENU**
95 '
100 '
105 MOTOR OFF:AUDIO OFF
110 CLS:PRINT:PRINT"
,**menu**":PRINT:PRINT"1=LOAD FR
OM TAPE  2=SAVE TO TAPE
                        3=SCROLL
        4=ALTER WORD"
115 PRINT:PRINT"5=DELETE WORD
  6=INSERT WORD
                7=INSERT LINE
  8=DELETE LINE"
120 PRINT:PRINT"9=MOVE LINE
  P=PRINTOUT":PRINT:PRINT"X=EXTE
ND          Q=QUIT"
125 T$=INKEY$:IF T$="P" THEN 140
5
130 IF T$="Q" THEN 2505
135 IF T$="X" THEN 2460
140 IF T$<CHR$(48) OR T$>CHR$(57
) THEN 125
145 Y=VAL (T$):ON Y GOTO 2145,23
25,350,765,845,920,1005,1210,190
0,1405
150 CLS
155 PRINT:PRINT"THERE ARE 200 LI
NES AVAILABLE   FOR USE. IF YOU
NEED MORE,      SAVE FILE TO TAP
E."
160 PRINT:PRINT"INSTRUCTIONS FOR
 SAVING TO TAPE AND LOADING FROM
 TAPE ARE       INCLUDED WITH TH
OSE SECTIONS."
165 PRINT:PRINT:PRINT"PRESS ANY
KEY TO CONTINUE."
170 T$=INKEY$:IF T$="" THEN 170
175 CLS
180 PRINT"THERE ARE 3 TYPES OF S
CROLLING.":PRINT:PRINT"autoscrol
l WHERE THE COMPUTER   WILL AUTO
MATICALLY SCROLL"
185 PRINT"NEW LINES EVERY 3 SECO
NDS APP.":PRINT:PRINT"manualscro
ll WHERE YOU SCROLL   EACH LINE
WHEN YOU PRESS        THE <SPACE
BAR>."
190 PRINT:PRINT"from a line to e
nd WHICH STARTS SCROLLING FROM A
 SPECIFIED LINE TO THE END.
                  PRESS ANY KEY
 TO CONTINUE."
195 T$=INKEY$:IF T$="" THEN 195
200 CLS:PRINT"TO LEAVE ANY MODE,
 PRESS THE ";CHR$(34);"^";CHR$(3
4);"IN RESPONSE TO A Y/N QUESTIO
N"
205 PRINT"         or
        AT ANY TIME DURING THE
 SCROLLINGROUTINE."
210 PRINT:PRINT:PRINT"TO LEAVE T
HE INPUT ROUTINE,     PRESS THE"
```

```
;CHR$(34);"^";CHR$(34)
215 PRINT"AS A SEPERATE LINE, TH
EN        PRESS THE <ENTER> KEY"
220 PRINT:PRINT"PRESS ANY KEY TO
 CONTINUE."
225 T$=INKEY$:IF T$="" THEN 225
230 GOTO 55
235 '
240 '
245 '     **INPUT ROUTINE**
250 '
255 '
260 CLS:CLEAR5000:BAUD=87:STYL=1
9
265 DIM S$(200)
270 Y=1
275 H=1
280 PRINT"          start input no
w"
285 FOR X=Y TO 200
290 LINE INPUT S$(X)
295 IF S$(X)="" THEN S$(X)=" "
300 IF S$(X)="^" THEN 330
305 IF X>190 AND H=1 THEN 310 EL
SE 320
310 CLS:SOUND 200,2:FOR A=1 TO 5
0:NEXT A:SOUND 200,2
315 PRINT CHR$(182);"TAKE CARE,
ONLY 10 LINES LEFT.";CHR$(185):F
OR A=1 TO 3000:NEXT A:H=0
320 NEXT X
325 PRINT:PRINT"            memory
 full"
330 PRINT:PRINT"           M=menu
Q=quit"
335 T$=INKEY$:IF T$="M" THEN 80
340 IF T$="Q" THEN 2505 ELSE 335
345 CLS(RND(8)):PRINT@32*7+9,"M=
menu Q=quit";:GOTO 335
350 '
355 '
360 '    **SCROLLING ROUTINE**
365 '
370 '
375 CLS:PRINT@32*4,"WHICH TYPE O
F SCROLL ?"
380 PRINT:PRINT"A = autoscroll M
 = manual scroll"
385 PRINT"F = from a line to the
 end"
390 T$=INKEY$:IF T$="A" THEN 410
395 IF T$="M" THEN 450
400 IF T$="F" THEN 480
405 IF T$="^" THEN 345 ELSE 390
410 N=1
415 CLS:FOR M=N TO (X-1)
420 PRINT M;":";S$(M)
425 FOR Y=1 TO 550
430 T$=INKEY$:IF T$="^" THEN 345
435 NEXT Y
440 NEXT M
445 GOTO 330
450 N=1
455 CLS:FOR M=N TO (X-1)
460 PRINT M;":";S$(M)
465 T$=INKEY$:IF T$="^" THEN 345
470 IF T$=CHR$(32) THEN NEXT M E
LSE 465
475 GOTO 330
480 CLS:PRINT@32*4,"WHICH LINE D
O YOU WANT TO SCROLLFROM ";:INPU
T N
485 PRINT"A = autoscroll  M = ma
nualscroll"
490 T$=INKEY$:IF T$="A" THEN 415
495 IF T$="M" THEN 455
500 IF T$="^" THEN 345 ELSE 490
505 IF H=1 THEN 540:CLS
510 '
515 '
520 '   **MID$ ROUTINE**
525 '
530 PRINT"WHICH LINE ?";K$
535 INPUT L
540 CLS:PRINT S$(L)
545 PRINT:PRINT"IS THIS THE LINE
 ? Y/N";K$
550 T$=INKEY$:IF T$="E" THEN 445
555 IF T$="N" THEN 735
560 IF T$="Y" THEN 570
565 IF T$="^" THEN 345 ELSE 550
570 PRINT:PRINT R$;"WHICH WORD(S
) ?"
575 LINE INPUT W$
580 IF W$="" THEN W$=" "
585 A=LEN(S$(L))
590 B=LEN(W$)
595 N=1
600 WORD=INSTR(N,S$(L),W$)
605 IF WORD=0 THEN 725
610 '
615 '
620 '
625 C$=LEFT$(S$(L),(WORD-1))
630 D$=RIGHT$(S$(L),(A-WORD-B+1)
)
635 F$=""
640 FOR E=1 TO B
645 Q$=MID$(W$,E,E):IF Q$>CHR$(6
4) AND Q$<CHR$(91) THEN F$=F$+CH
R$(ASC(Q$)+32) ELSE 655
650 GOTO 670
655 IF Q$>CHR$(96) AND Q$<CHR$(1
23) THEN F$=F$+CHR$(ASC(Q$)-32)
ELSE 665
660 GOTO 670
665 F$=F$+CHR$(128)
670 NEXT E
675 CLS
680 PRINT"IS THIS THE WORD ? Y/N
";K$
685 PRINT@96,C$+F$+D$
690 FOR Y=1 TO 25:NEXT Y
695 PRINT@96+LEN(C$),W$;
700 T$=INKEY$:IF T$="Y" THEN RET
URN
705 IF T$="N" THEN 715
710 IF T$="^" THEN 345 ELSE 685
715 IF WORD>(A-B+1) THEN 725
720 N=WORD+1:GOTO 600
725 CLS:PRINT"NO CAN FIND ";CHR$
(34);W$;CHR$(34):FOR Y=1 TO 2000
:NEXT Y
730 GOTO 540
735 CLS:PRINT"YOU ARE IN ";K$;"
MODE."
740 PRINT:PRINT"DO YOU WANT TO S
TAY IN THIS     MODE ?  Y/N"
745 T$=INKEY$:IF T$="Y" THEN 760
750 IF T$="N" THEN 345
755 IF T$="^" THEN 345 ELSE 745
760 H=0:GOTO 505
765 H=0:CLS
770 '
775 '   **ALTER WORD ROUTINE**
780 '
785 '
790 R$="":K$="*ALTER*":GOSUB 505
795 CLS:PRINT"TYPE NEW WORD(S) N
OW.";K$
800 LINE INPUT W$
805 S$(L)=C$+W$+D$:CLS
810 PRINT@96,"LINE "L" NOW READS
 :-"
815 PRINT@160,S$(L)
820 PRINT:PRINT"ANY MORE SAME LI
NE ? Y/N";K$
825 T$=INKEY$:IF T$="Y" THEN 840
830 IF T$="N" THEN 345
835 IF T$="^" THEN 345 ELSE 825
840 H=1:GOTO 790
845 H=0:CLS
850 '
855 '   **DELETE WORD ROUTINE**
860 '
865 '
870 R$="":K$="*DELETE*":GOSUB 50
5
875 CLS:S$(L)=C$+D$
880 IF S$(L)="" THEN S$(L)=" "
885 PRINT@96,"LINE "L" NOW READS
 :-"
890 PRINT@160,S$(L)
895 PRINT:PRINT"ANY MORE SAME LI
NE ? Y/N";K$
900 T$=INKEY$:IF T$="Y" THEN 915
905 IF T$="N" THEN 345
910 IF T$="^" THEN 345 ELSE 900
915 H=1:GOTO 870
920 H=0:CLS
925 '
930 '   **INSERT WORD ROUTINE**
935 '
940 '

945 R$="AFTER ":K$="*INSERT*":GO
SUB 505
950 CLS:C$=C$+W$
955 PRINT"TYPE NEW WORD NOW.";K$
960 LINE INPUT W$
965 CLS:S$(L)=C$+W$+D$
970 PRINT@96,"LINE "L" NOW READS
 :-"
975 PRINT@160,S$(L)
980 PRINT:PRINT"ANY MORE SAME LI
NE ? Y/N";K$
985 T$=INKEY$:IF T$="Y" THEN 100
0
990 IF T$="N" THEN 345
995 IF T$="^" THEN 345 ELSE 985
1000 H=1:GOTO 945
1005 '
1010 '
1015 '   **INSERT LINE ROUTINE**
1020 '
1025 '
1030 CLS:PRINT"INSERT LINE ROUTI
NE."
1035 PRINT:PRINT"ONLY INSERTS BL
ANK LINES."
1040 PRINT"USE ALTER ROUTINE TO
INSERT NEW WORDS OR SENTENCES."
1045 PRINT:PRINT"TO INSERT NEW S
ENTENCES, ALTER  THE FIRST CHARA
CTER <SPACEBAR>"
1050 PRINT:PRINT"THEN TYPE NEW W
ORD OR SENTENCE."
1055 PRINT:PRINT"PRESS ANY KEY T
O CONTINUE."
1060 T$=INKEY$:IF T$="" THEN 106
0
1065 CLS:PRINT"WHICH LINE DO YOU
 WANT TO INSERTBLANK LINES AFTER
 ?"
1070 PRINT:INPUT M
1075 PRINT:PRINT"HOW MANY LINES
TO BE INSERTED ?"
1080 PRINT:INPUT G
1085 CLS:PRINT"IS THIS THE LINE
TO INSERT FROM?"
1090 PRINT:PRINT S$(M)
```

```
1095 PRINT:PRINT"Y/N"
1100 T$=INKEY$:IF T$="N" THEN 11
85
1105 IF T$="Y" THEN 1115
1110 IF T$="^" THEN 345 ELSE 110
0
1115 CLS(0):PRINT@32*5+4,"RE-NUM
BERING NOW";
1120 PRINT@32*8+4,"PLEASE WAIT."
;
1125 FOR Y=1 TO (X-M-1)
1130 S$(X+G-Y)=S$(X-Y)
1135 NEXT Y
1140 FOR Y=(M+1) TO M+G
1145 S$(Y)=" "
1150 NEXT Y
1155 LET X=X+G
1160 FOR Y=1 TO 1000:NEXT Y
1165 CLS(0):PRINT@32*5+4,"RENUMB
ERING COMPLETE.";
1170 PRINT@32*8+4,"PRESS ANY KEY
 TO CONTINUE.";
1175 T$=INKEY$:IF T$="" THEN 117
5
1180 GOTO 345
1185 CLS:PRINT"YOU ARE IN THE IN
SERT LINE MODE."
1190 PRINT:PRINT"DO YOU WANT TO
STAY IN THIS      MODE ? Y/N"
1195 T$=INKEY$:IF T$="N" THEN 11
80
1200 IF T$="Y" THEN 1065
1205 IF T$="^" THEN 345 ELSE 119
5
1210 '
1215 '
1220 '  ** DELETE LINE(S) ROUTIN
E **
1225 '
1230 '
1235 CLS:PRINT"DELETE LINE(S) RO
UTINE."
1240 PRINT"  ** TAKE NOTE **"
1245 PRINT:PRINT"TAKE EXTREME CA
RE SELECTING      LINE(S) TO BE D
ELETED."
1250 PRINT:PRINT"ONCE LINE(S) HA
VE BEEN DELETED, THEY CANNOT BE
RESTORED EXCEPT"
1255 PRINT"BY USING THE INSERT L
INE(S) ROUTINE."
1260 PRINT:PRINT"PRESS ANY KEY T
O CONTINUE."
1265 T$=INKEY$:IF T$="" THEN 126
5
1270 CLS:PRINT"WHICH IS THE firs
t LINE TO BE   DELETED ?"
1275 INPUT M
1280 PRINT"WHICH IS THE last LIN
E TO BE    DELETED ?"
1285 INPUT G
1290 CLS:PRINT S$(M):PRINT S$(G)
1295 PRINT:PRINT"ARE THESE THE F
IRST AND LAST     LINES TO BE DEL
ETED ?"
1300 PRINT:PRINT"Y/N"
1305 T$=INKEY$:IF T$="Y" THEN 13
20
1310 IF T$="N" THEN 1380
1315 IF T$="^" THEN 345 ELSE 130
5
1320 CLS(0):PRINT@32*5+4,"RE-NUM
BERING NOW.";
1325 PRINT@32*8+4,"PLEASE WAIT."
;
1330 FOR Y=(G+1) TO X
1335 S$(M)=S$(Y)
1340 M=M+1
1345 NEXT Y
1350 X=M-1
1355 FOR Y=1 TO 1000:NEXT Y
1360 CLS(0):PRINT@32*5+4,"RENUMB
ERING COMPLETE.";
1365 PRINT@32*8+4,"PRESS ANY KEY
 TO CONTINUE.";
1370 T$=INKEY$:IF T$="" THEN 137
0
1375 GOTO 345
1380 CLS:PRINT"YOU ARE IN DELETE
 LINE MODE"
1385 PRINT:PRINT"DO YOU WANT TO
STAY IN THIS      MODE ?  Y/N"
1390 T$=INKEY$:IF T$="Y" THEN 12
70
1395 IF T$="N" THEN 345
1400 IF T$="^" THEN 345 ELSE 139
0
1405 '
1410 '
1415 '  ** PRINTING ROUTINE **
1420 '
1425 '
1430 CLS(0):PRINT@72,"PRINTING R
OUTINE.";
1435 IF PEEK(65314)/2<>INT(PEEK(
65314)/2) THEN 1625
1440 PRINT@128,"THE COMPUTER IS
SET FOR :-";
1445 IF STYL=19 THEN STY$="STAND
ARD DATA PROCESSING."
1450 IF STYL=18 THEN STY$="STAND
ARD WORD PROCESSING."
1455 IF STYL=29 THEN STY$="ELITE
 WORD PROCESSING."
1460 PRINT@192,STY$;
1465 IF BAUD=87 THEN BAU$="600"
1470 IF BAUD=41 THEN BAU$="1200"
1475 PRINT@256,BAU$;" BAUD";
1480 PRINT@320,"DO YOU WANT TO C
HANGE ANY Y/N ?";
1485 T$=INKEY$:IF T$="Y" THEN 15
00
1490 IF T$="^" THEN 345
1495 IF T$="N" THEN 1650 ELSE 14
85
1500 CLS:PRINT@64,"DO YOU WANT T
O CHANGE THE       PRINTER BAUD
RATE ? Y/N"
1505 T$=INKEY$:IF T$="Y" THEN 15
20
1510 IF T$="^" THEN 345
1515 IF T$="N" THEN 1560 ELSE 15
05
1520 CLS:PRINT@192,"WHICH BAUD R
ATE IS THE PRINTER  SET FOR ?"
1525 PRINT@256,"press 1 - 600 BA
UD
                        2 - 1200 B
AUD."
1530 T$=INKEY$:IF T$="" THEN 153
0
1535 IF T$="1" THEN BAUD=87 ELSE
 GOTO 1545
1540 POKE 149,0:POKE 150,87:GOTO
 1560
1545 IF T$="2" THEN BAUD=41 ELSE
 GOTO 1530
1550 POKE 149,0:POKE 150,41
1555 IF T$="^" THEN 345
1560 CLS:PRINT@64,"DO YOU WANT T
O  CHANGE THE      PRINTING STYL
E ? Y/N"
1565 T$=INKEY$:IF T$="" THEN 156
5
1570 IF T$="Y" THEN 1580
1575 IF T$="N" THEN 1650 ELSE 15
65
1580 CLS:PRINT@64,"WHICH STYLE O
F PRINTING ?"
1585 PRINT@128," 1 - STANDARD DA
TA PRINTING."
1590 PRINT" 2 - STANDARD word PR
INTING."
1595 PRINT" 3 - ELITE WORD PRINT
ING."
1600 T$=INKEY$:IF T$="" THEN 160
0
1605 IF T$="1" THEN STYL=19
1610 IF T$="2" THEN STYL=18
1615 IF T$="3" THEN STYL=29
1620 CLS(0):GOTO 1440
1625 PRINT@229,"PRINTER NOT TURN
ED ON.";
1630 PRINT@294,"TURN ON NOW PLEA
SE.";
1635 PRINT@355,"PRESS ANY KEY TO
 CONTINUE.";
1640 FOR Y=1 TO 5:SOUND RND(150)
,2:NEXT Y
1645 T$=INKEY$:IF T$="" THEN 164
5 ELSE 1430
1650 CLS:PRINT@32*7+6,"PROOFING
COPY ? Y/N";
1655 T$=INKEY$:IF T$="" THEN 165
5
1660 IF T$="Y" THEN 1675
1665 IF T$="N" THEN 1740
1670 IF T$="^" THEN 345 ELSE 165
0
1675 CLS(7):PRINT@32*5+2,"PRINTI
NG PROOFING COPY ONLY.";
1680 PRINT #-2,CHR$(27) CHR$(19)
1685 PRINT #-2,"  *** PRINTING
PROOFING COPY ONLY  ***"
1690 PRINT#-2,CHR$(27)CHR$(52)CH
R$(10):PRINT#-2,CHR$(12):PRINT#-
2,CHR$(27)CHR$(52)CHR$(66)
1695 FOR Y=1 TO (X-1)
1700 PRINT#-2,Y;": ";S$(Y)
1705 NEXT Y
1710 PRINT #-2,CHR$(27)CHR$(52)C
HR$(43):PRINT#-2,CHR$(12):PRINT#
-2,CHR$(27)CHR$(52)CHR$(66)
1715 CLS:PRINT@32*7+4,"M=menu Q=
quit P=printout"
1720 T$=INKEY$:IF T$="Q" THEN 25
05
1725 IF T$="P" THEN 1740
1730 IF T$="M" THEN 80
1735 IF T$="^" THEN 80 ELSE 1720
1740 CLS:PRINT@32*7+2,"NUMBER OF
 COPIES REQUIRED";
1745 INPUT J
1750 PRINT #-2,CHR$(27)CHR$(52)C
HR$(66):PRINT#-2,CHR$(27) CHR$(S
TYL)
1755 CLS(0):PRINT@32*7+6,"PRINTI
NG, PLEASE WAIT.";
1760 IF STYL=29 THEN LNTH=96 ELS
E LNTH=80
1765 FOR M=1 TO J
1770 FOR Y=1 TO (X-1)
1775 Z=0:W$=S$(Y)
1780 LGT=LEN(W$):IF LGT<(LNTH+1)
 THEN 1805
1785 T$=MID$(W$,LNTH-Z,1)
1790 IF T$=" " THEN 1815
1795 IF T$="-" THEN 1815
```

```
1800 Z=Z+1:GOTO 1785
1805 T$=MID$(W$,LGT,1):IF T$="_"
 THEN 1845
1810 PRINT #-2,W$:GOTO 1825
1815 PRINT #-2,LEFT$(W$,LNTH-Z)
1820 W$=RIGHT$(W$,LGT-LNTH+Z):Z=
0:GOTO 1780
1825 NEXT Y
1830 PRINT #-2,CHR$(12)
1835 NEXT M
1840 PRINT #-2,CHR$(12):GOTO 187
5
1845 Q$=W$:W$=S$(Y+1)
1850 IF (LEN(Q$)+LEN(W$)))>254 TH
EN 1865
1855 W$=LEFT$(Q$,(LEN(Q$)-1))+W$
1860 Y=Y+1:GOTO 1780
1865 T$=LEFT$(Q$,(LEN(Q$)-1))+LE
FT$(W$,(LNTH-(LEN(Q$)-1))):PRINT
#-2,T$
1870 W$=RIGHT$(W$,(LEN(W$)-LEN(Q
$))):Z=0:GOTO 1780
1875 CLS(0):PRINT@32*8+12,"THE E
ND.";
1880 FOR Y=1 TO 10:SOUND RND(150
),2:NEXT Y
1885 PRINT@32*8+6,"ANY MORE COPI
ES Y/N";
1890 T$=INKEY$:IF T$="Y" THEN 17
40
1895 IF T$="N" THEN 345 ELSE 189
0
1900 '
1905 '
1910 '  **PARA. MOVING ROUTINE**
1915 '
1920 '
1925 CLS:PRINT"MOVE LINE(S)   RO
UTINE."
1930 PRINT:PRINT"WHICH IS THE fi
rst LINE TO BE   MOVED ";
1935 INPUT M:PRINT S$(M)
1940 PRINT:PRINT"WHICH IS THE la
st LINE TO BE    MOVED ";
1945 INPUT G:PRINT S$(G)
1950 PRINT:PRINT"AFTER WHICH LIN
E DO YOU WANT TO INSERT THE LINE
(S)";
1955 INPUT I:PRINT S$(I)
1960 PRINT"ARE THESE THE CORRECT
 LINES ?    Y/N"
1965 T$=INKEY$:IF T$="N" THEN 20
75
1970 IF T$="Y" THEN 1980
1975 IF T$="^" THEN 345 ELSE 196
5
1980 CLS(0):PRINT@32*5+4,"RENUMB
ERING NOW.";
1985 PRINT@32*8+4,"PLEASE WAIT."
;
1990 FOR Y=1 TO (X-I)
1995 S$(X+G-M-Y+1)=S$(X-Y)
2000 NEXT Y
2005 IF M>I THEN 2100
2010 Q=M
2015 FOR Y=(I+1) TO (I+G-M+1)
2020 S$(Y)=S$(Q):Q=Q+1
2025 NEXT Y
2030 X=X+G-M+1:Q=M
2035 FOR Y=(G+1) TO X
2040 S$(Q)=S$(Y):Q=Q+1
2045 NEXT Y
2050 X=Q-1
2055 CLS(0):PRINT@32*5+4,"RE-NUM
BERING COMPLETE.";
2060 PRINT@32*8+4,"PRESS ANY KEY
 TO CONTINUE.";
2065 T$=INKEY$:IF T$="" THEN 206
5
2070 GOTO 345
2075 CLS:PRINT"YOU ARE IN THE LI
NE(S)          MOVING ROUTINE."
2080 PRINT:PRINT"DO YOU WANT TO
STAY IN THIS     MODE ? Y/N"
2085 T$=INKEY$:IF T$="Y" THEN 19
00
2090 IF T$="N" THEN 345
2095 IF T$="^" THEN 345 ELSE 208
5
2100 Q=G+1
2105 FOR Y=(I+1) TO (I+G-M+1)
2110 S$(Y)=S$(Q):Q=Q+1
2115 NEXT Y
2120 X=X+G-M+1:Q=G+1
2125 FOR Y=Q TO X
2130 S$(Y)=S$(Q+G-M+1):Q=Q+1
2135 NEXT Y
2140 X=X+M-G-1:GOTO 2055
2145 '
2150 '
2155 ' **LOAD FROM TAPE ROUTINE*
2160 '
2165 '
2170 CLS:PRINT@96,"LOADING FILE
FROM TAPE ROUTINE."
2175 GOSUB 2475
2180 CLS:PRINT:PRINT"LOADING FIL
E FROM TAPE ROUTINE."
2185 CLEAR 5000
2190 PRINT:PRINT"WHAT IS THE NAM
E OF THE FILE ?"
2195 PRINT:INPUT W$
2200 CLS:PRINT@32*4,"PLEASE CHEC
K BEFORE I SEARCH    FOR THE FIL
E."
2205 PRINT:PRINT"THE FILE NAME I
S ";CHR$(34);W$;CHR$(34)
2210 PRINT:PRINT"IS THIS CORRECT
 Y/N"
2215 T$=INKEY$:IF T$="Y" THEN 22
30
2220 IF T$="N" THEN 2180
2225 IF T$="^" THEN 80 ELSE 2215
2230 CLS:PRINT:PRINT"PRESS THE p
lay BUTTON NOW":PRINT:PRINT"PRES
S ANY KEY TO START SEARCH."
2235 T$=INKEY$:IF T$="" THEN 223
5
2240 DIM S$(200):X=0
2245 CLS(0):PRINT@32*6,"SEARCHIN
G FOR FILE ";CHR$(34);W$;CHR$(34
);
2250 OPEN "I",#-1,W$
2255 INPUT #-1,S$
2260 IF EOF(-1) THEN 2265
2265 CLOSE #-1
2270 IF S$<>W$ THEN 2310
2275 OPEN "I",#-1,W$
2280 FOR X=1 TO 200
2285 INPUT #-1,S$(X)
2290 IF EOF(-1) THEN 2300
2295 NEXT X
2300 CLOSE #-1
2305 GOTO 80
2310 IF X>0 THEN 80
2315 SKIPF S$
2320 GOTO 2230
2325 '
2330 '
2335 '  **SAVING TO TAPE**
2340 '
2345 '
2350 CLS:PRINT@96,"SAVING TO TAP
E ROUTINE."
2355 GOSUB 2475
2360 CLS:PRINT@96,"SAVING TO TAP
E ROUTINE."
2365 PRINT:PRINT"WHAT IS THE FIL
E TO BE CALLED ?"
2370 PRINT:PRINT"note:- MAXIMUM
8 CHARACTERS.      "
2375 PRINT:INPUT W$
2380 CLS:PRINT@32*3,"PRESS THE p
lay & record BUTTONS NOW."
2385 PRINT@32*9,"PRESS ";CHR$(34
);"Y";CHR$(34);" WHEN READY TO":
PRINT"SAVE FILE TO TAPE."
2390 T$=INKEY$:IF T$="Y" THEN 24
00
2395 IF T$="^" THEN 80 ELSE 2390
2400 CLS(3):PRINT@32*9+6,"SAVING
 TO TAPE NOW.";
2405 OPEN "O",#-1,W$
2410 PRINT#-1,W$
2415 CLOSE #-1
2420 OPEN "O",#-1,W$
2425 FOR Y=1 TO X
2430 PRINT#-1,S$(Y)
2435 NEXT Y
2440 CLOSE #-1
2445 CLS(3):PRINT@32*9+7,"SAVING
 COMPLETED.";
2450 FOR Y=1 TO 2000:NEXT Y
2455 GOTO 80
2460 CLS:IF X=0 THEN 235
2465 IF X>199 THEN 325
2470 Y=X:GOTO 275
2475 PRINT:PRINT"do not disconne
ct tape recorder"
2480 PRINT:PRINT"BY LISTENING TO
 THE T.V.,       POSITION THE TA
PE BY USING THE  fastf & rewind
AND play BUTTONS."
2485 PRINT"WHEN POSITIONED, PRES
S THE stop BUTTON, THEN PRESS AN
Y KEY TO   GO TO THE NEXT STAGE.
"
2490 AUDIO ON:MOTOR ON
2495 T$=INKEY$:IF T$="" THEN 249
5
2500 MOTOR OFF:AUDIO OFF:RETURN
2505 CLS(6):PRINT@32*6,"THIS PRO
GRAM IS FINISHED.";
2510 PRINT@32*8,"TO USE AGAIN, T
YPE RUN AND PRESSENTER.";
2515 FOR Y=1 TO 5000:NEXT Y
2520 PRINT#-2,CHR$(27)CHR$(52)CH
R$(66):PRINT#-2,CHR$(27)CHR$(19)
:CLS
```

■

Cassette tape cleaning

ALWAYS totally erase a tape before doing any CSAVE's on it. This will eliminate any bits of garbage that will cause the dreaded IO Errors.

This is best done with a BULK ERASER, but may also be done by placing the tape in a cassette player and pressing play and record simultaneously and by typing 'MOTORON'.

It takes a while, but it is worth it.

Kevin Gowan

GUNFIGHT

32K ECB
GAME

by Craig Stewart

GUNFIGHT IS A REFLEX game with some hi-rez pictures. Up to four players can play. Just press the <Spacebar> when the gunfighter prompts you and press the <Spacebar> again when he draws his gun.

You are the smiling chap with the fancy waistcoat (or the one on the right hand side). If you ever make it past round six, you're doing well!

The Listing:

```
1 GOTO10
2 '******** GUNFIGHT ************
   ***** CRAIG STEWART *********
3 SAVE"1461:3":END'1
10 REM CLEAR MEMORY IF NECESSARY
20 FOR I= 18688TO 23701
30 READ Q$:POKEI,VAL("&H"+Q$):NE
XTI
40 EXEC 18688
50 DATA 7E,4A,DF,8E,EA,60,30,1F,
8C,0,0,27,2,20,F7,39,AD,9F,A0,0,
81,0,27,F8,39,8E,E,0,86,FF,A7,80
,8C,26,0,25,F9,39,10,8E,4E,20,8E
,10,18,C6,6,A6,A0,A7,80,5A,27,2,
20,F7,30,88,1A,10,8C,51,B3,22,2,
20,EA,39,10,8E,52,8,8E,10,2
60 DATA C6,6,A6,A0,A7,80,5A,27,2
,20,F7,30,88,1A,10,8C,55,9B,22,2
,20,EA,39,10,8E,55,F0,8E,17,43,C
6,9,A6,A0,A7,80,5A,27,2,20,F7,30
,88,17,10,8C,57,56,22,2,20,EA,39
,10,8E,57,6C,8E,17,54,C6,9,A6,A0
,A7,80,5A,27,2,20,F7,30,88,17,10
70 DATA 8C,58,CE,22,2,20,EA,39,1
0,8E,5B,CC,8E,17,2B,C6,6,A6,A0,A
7,80,5A,27,2,20,F7,30,88,1A,10,8
C,5C,3F,22,2,20,EA,39,10,8E,59,D
8,8E,17,2F,C6,6,A6,A0,A7,80,5A,2
7,2,20,F7,30,88,1A,10,8C,5A,4B,2
2,2,20,EA,39,34,12,B6,FF,23,8A,8
80 DATA B7,FF,23,7F,1,47,8E,B7,9
8,A6,80,8D,16,A6,80,8D,12,A6,80,
8D,E,7A,1,47,26,EF,B6,FF,23,35,1
2,17,FF,0,39,B4,1,47,B7,FF,20,86
,5A,4A,26,FD,39,10,BE,1,13,31,A8
,EC,10,8C,0,15,22,F7,B6,1,13,84,
7F,8B,A,C6,46,3D,1F,1,39,17,FE
90 DATA EC,17,FF,14,17,FE,F3,17,
FF,DA,30,1F,27,12,BF,4B,25,AD,9F
,A0,0,81,0,10,26,0,22,BE,4B,25,2
0,EA,17,FF,13,BE,4B,21,30,1F,8C,
0,0,27,10,BF,4B,25,AD,9F,A0,0,81
,0,26,14,BE,4B,25,20,E9,17,FE,F6
,17,FF,2F,17,FF,68,86,1,B7,4B,23
100 DATA 39,17,FF,5,17,FF,3E,17,
FF,59,7F,4B,23,39,7F,4B,24,B6,4B
,23,81,1,26,11,F6,4B,1A,8E,4B,1B
,3A,A7,84,8E,4B,B7,17,1,7F,20,6,
8E,4B,D0,17,1,77,F6,4B,1A,5C,F7,
4B,1A,C1,5,26,14,C6,1,F7,4B,1A,B
E,4B,21,30,88,C9,BF,4B,21,7C,4B,
27
110 DATA 17,1,C3,B6,4B,24,81,5,2
5,3,16,1,D3,7C,4B,24,F6,4B,1A,8E
,4B,1B,3A,A6,84,31,1,27,C7,F6,4B
,1A,C1,1,26,8,8E,4B,2A,17,1,31,2
0,20,C1,2,26,8,8E,4B,46,17,1,25,
20,14,C1,3,26,8,8E,4B,60,17,1,19
,20,8,8E,4B,7E,17,1,11,20
120 DATA 0,17,FD,FF,17,1,22,17,F
F,13,16,FF,6B,1,0,0,0,0,0,0,1,F4
,0,0,0,0,0,0,0,50,4C,41,59,20,31
,20,2D,20,52,45,41,44,59,20,54,4
F,20,42,45,20,53,48,4F,54,20,3F,
5E,50,4C,41,59,20,32,20,2D,20,50
,52,45,50,41,52,45,20,54
130 DATA 4F,20,44,49,45,20,21,5E
,50,4C,41,59,33,20,2D,20,41,52,4
5,20,59,41,20,52,45,41,44,59,20,
43,4F,57,41,52,44,20,3F,5E,50,4C
,41,59,45,52,20,34,20,2D,20,47,4
5,54,20,52,45,41,44,59,20,53,4C,
4F,57,50,4F,4B,45,20,21,5E,49,20
,47,4F,54
140 DATA 20,59,41,20,41,4C,4C,20
,2D,20,46,49,4E,41,4C,4C,59,21,2
1,5E,48,41,21,20,48,41,21,20,48,
41,21,20,20,2D,20,20,49,20,47,4F
,54,20,59,41,5E,55,47,47,47,47,2
1,21,21,21,21,20,20,20,59,41,20,
47,4F,54,20,4D,45,20,5E,57,4F,55
,4C,44,20
150 DATA 59,4F,55,20,4C,49,4B,45
,20,54,4F,20,50,4C,41,59,20,41,4
7,41,49,4E,3F,5E,48,4F,57,20,4D,
41,4E,59,20,41,52,45,20,50,4C,41
,59,49,4E,47,20,3F,5E,17,0,3F,34
,10,BD,A9,28,35,10,A6,80,81,5E,2
7,5,BD,A3,A,20,F5,17,FC,CE,39,7F
,FF,C0
160 DATA 7F,FF,C3,7F,FF,C5,B6,FF
,22,84,7,8A,F8,B7,FF,22,7F,FF,C7
,7F,FF,C9,7F,FF,CB,7F,FF,CC,7F,F
F,CE,7F,FF,D0,7F,FF,D2,39,7F,FF,
C0,7F,FF,C2,7F,FF,C4,B6,FF,22,84
,7,8A,8,B7,FF,22,7F,FF,C6,7F,FF,
C9,7F,FF,CA,7F,FF,CC,7F,FF,CE,7F
,FF,D0
170 DATA 7F,FF,D2,39,B6,4B,27,8B
,31,8E,4,0,C6,AF,E7,80,8C,6,0,25
,F9,8E,4,F0,A7,84,17,FC,62,39,8E
,4B,9E,17,FF,75,17,FC,58,17,FC,5
5,8E,4B,E8,17,FF,69,AD,9F,A0,0,2
7,FA,81,59,27,1,39,CC,0,0,FD,4B,
1B,FD,4B,1D,FD,4B,1F,FD,4B,27,FD
180 DATA 4B,25,FD,4B,23,CC,1,F4,
FD,4B,21,B7,4B,1A,7E,49,0,3F,35,
1C,38,FD,4B,40,FD,4B,3E,FD,4B,3C
,CC,1,F4,FD,4E,3A,B7,4B,33,7E,49
,0,3F,35,8E,84,AE,84,20,F0,DC,2B
,9E,19,EE,84,27,9,10,A3,2,23,6,A
E,84,20,F3,1A,1,9F,47,39,26,FB,9
E
190 DATA 19,6F,80,6F,80,9F,1B,9E
,19,BD,AE,BB,9E,27,9F,23,BD,AD,E
4,9E,1B,9F,1D,9F,1F,8E,1,A9,9F,B
,AE,E4,10,DE,21,6F,E2,F,2D,F,2E,
F,8,6E,84,86,80,97,8,BD,AF,89,BD
,AB,F9,32,62,26,4,9E,F,32,85,C6,
9,BD,AC,33,BD,AE,E8,DC,68,34,16
200 DATA C6,A5,BD,B2,6F,BD,B1,43
,BD,B1,41,D6,54,CA,7F,D4,50,D7,5
0,10,8E,AD,7F,7E,B1,EA,8E,BA,C5,
BD,BC,14,9D,A5,81,A9,26,5,9D,9F,
BD,B1,41,BD,BC,6D,BD,B1,E6,DC,3B
,34,6,86,80,34,2,BD,1,9A,1C,AF,8
D,46,9E,A6,9F,2F,A6,80,27,7,81,3
A,27
210 DATA F,7E,B2,77,A6,81,97,0,2
7,5B,EC,80,DD,68,9F,A6,9D,9F,8D,
2,20,D8,27,78,4D,10,2A,1,BC,81,A
3,22,B,BE,1,23,48,1F,89,3A,9D,9F
,6E,94,81,B4,23,D1,6E,9F,1,2D,9E
,19,30,1F,9F,33,39,BD,A1,C1,27,A
,81,3,27,15,81,13,27,3,97,87,39
220 DATA BD,A1,C1,27,FB,20,EE,BD
,A4,26,9D,A5,20,2,1A,1,26,33,9E,
A6,9F,2F,6,0,32,62,9E,68,8C,FF,F
F,27,6,9F,29,9E,2F,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FE
,FF,FF,FF,FF,FF,FE,7F,FF,FF,FF,F
F,FE
230 DATA 3F,FF,FF,FF,FF,FE,1F,FF
,FF,FF,FF,FE,F,FF,FF,FF,FF,FE,7,
E3,FF,FF,FF,FE,3,1,FF,FF,FF,FE,1
,80,FF,FF,FF,FF,0,C0,7F,FF,FF,FF
,80,60,3F,FF,FF,FF,E0,30,1F,FF,F
F,FD,B8,18,F,FF,FF,E0,C,E,7,FF,F
F,C0,6,2,87,FF,FF,80,3,3,7
240 DATA FF,FF,80,1,81,87,FF,FF,
80,0,D0,C7,FF,FF,F0,0,68,47,FF,F
F,E8,40,24,47,FF,FF,EF,0,12,67,F
F,FF,DF,F2,19,37,FF,FF,D9,F8,8C,
8F,FF,FF,BF,FC,6,47,FF,FF,7F,FB,
43,23,FF,FF,1F,F7,41,91,FF,FF,DF
,EF,80,C0,FF,FF,DD,DF,C0,C0,7F,F
F,D3
250 DATA BF,C1,E0,7F,FF,DF,7F,C1
,FF,FF,FF,DE,FF,C1,FF,FF,FF,ED,1
F,C1,FF,FF,FF,F0,CF,C3,FF,FF,FF,
FF,EF,C3,FF,FF,FF,FF,E7,F7,FF,FF
```

```
,FF,FF,F0,7,FF,FF,FF,FF,F5,57,FF
,FF,FF,FF,EA,AB,FF,FF,FF,FF,D5,5
7,FF,FF,FF,FF,6D,D5,FF,FF,FF,FE,
53,DD,FF
260 DATA FF,FF,FC,EE,E,FF,FF,FF,
FB,5C,7,7F,FF,FF,F7,E0,0,7F,FF,F
F,FA,C0,0,3F,FF,FF,F4,C0,E0,3F,F
F,FF,EA,80,E0,3F,FF,FF,F5,0,F0,3
F,FF,FF,E2,80,F0,3F,FF,FF,EB,80,
F0,3F,FF,FF,F5,80,F0,3F,FF,FF,ED
,80,F0,3F,FF,FF,EA,80,F0,3F,FF,F
F,E5
270 DATA 80,F0,3F,FF,FF,E2,80,F0
,3F,FF,FF,E5,80,F0,3F,FF,FF,E2,C
0,F0,3F,FF,FF,E5,60,F0,3F,FF,FF,
EA,A0,F0,3F,FF,FF,E5,60,F0,3F,FF
,FF,EA,C0,F0,1F,FF,FF,F5,C0,F0,1
F,FF,FF,EA,A0,F0,5F,FF,FF,F5,40,
F0,5F,FF,FF,EA,A0,F0,9F,FF,FF,EA
,B0,F0,5F
280 DATA FF,FF,F5,40,F0,9F,FF,FF
,E0,10,F1,5F,FF,FF,EA,A8,F2,9F,F
F,FF,F5,A0,F1,5F,FF,FF,E2,A8,F2,
9F,FF,FF,E8,0,F1,1F,FF,FF,EA,98,
F2,BF,FF,FF,F5,50,F1,3F,FF,FF,F5
,50,F5,3F,FF,FF,F2,A8,F2,BF,FF,F
F,F9,50,E5,3F,FF,FF,FA,A2,E0,3F,
FF,FF,F8
290 DATA 51,E5,7F,FF,FF,FB,1,E0,
7F,FF,FF,FB,71,C7,7F,FF,FF,FB,23
,C5,7F,FF,FF,FB,A3,CD,7F,FF,FF,F
8,33,CD,7F,FF,FF,F8,73,DF,7F,FF,
FF,F0,3,C0,7F,FF,FF,F0,C3,C0,FF,
FF,FF,F0,D3,C0,7F,FF,FF,F0,D1,8F
,7F,FF,FF,E0,EF,EF,7F,FF,FF,F0,E
F,EF,7F
300 DATA FF,FF,F8,EF,EF,7F,FF,FF
,F8,EF,EF,7F,FF,FF,F8,E6,8F,7F,F
F,FF,F8,E6,8F,7F,FF,FF,F8,D6,8F,
7F,FF,FF,F8,E0,87,7F,FF,FF,FC,7F
,80,FF,FF,FF,FC,7F,81,FF,FF,FF,F
C,3F,C1,FF,FF,FF,FE,1F,C1,FF,FF,
FF,FE,F,C1,FF,FF,FF,FE,0,3,FF,FF
,FF,FE
310 DATA 3,3,FF,FF,FF,FE,F,3,FF,
FF,FF,FF,F,3,FF,FF,FF,FF,F,3,FF,
FF,FF,FF,F,7,FF,FF,FF,FF,F,7,FF,
FF,FF,FF,8F,F,FF,FF,FF,FF,F,F,FF
,FF,FF,FF,8F,F,FF,FF,FF,FF,8F,F,
FF,FF,FF,FF,8F,1F,FF,FF,FF,FF,8F
,1F,FF,FF,FF,FF,8F,1F,FF
320 DATA FF,FF,FF,8F,F,FF,FF,FF,
FF,8F,F,FF,FF,FF,FF,8F,7,FF,FF,F
F,FF,8F,7,FF,FF,FF,FF,8F,7,FF,FF
,FF,FF,8F,7,FF,FF,FF,FF,8F,7,FF,
FF,FF,FF,8F,3,FF,FF,FF,FF,8F,3,F
F,FF,FF,FF,8F,3,FF,FF,FF,FF,8F,3
,FF,FF,FF,FF,8F,3,FF,FF,FF,FF
330 DATA 8F,3,FF,FF,FF,FF,8F,3,F
F,FF,FF,FF,AF,7,FF,FF,FF,FF,AF,7
,FF,FF,FF,FF,8F,7,FF,FF,FF,FF,8F
,F,FF,FF,FF,FF,8F,F,FF,FF,FF,FF,
8F,F,FF,FF,FF,FF,BF,EF,FF,FF,FF,
FF,80,F,FF,FF,FF,FF,BF,CF,FF,FF,
FF,FF,BF,DF,FF,FF,FF,FF,BF,DF,FF
340 DATA FF,FF,FF,BF,DF,FF,FF,FF
,FE,42,1F,FF,FF,FF,FE,EF,CF,7F,F
F,FF,E4,0,B,5F,FF,FF,A0,0,D,3F,F
F,FC,48,0,0,F,FF,FD,0,0,D,3F,FF,
FE,0,7F,B,5F,FF,FF,FF,FF,F,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FE,B6,DC,D
F,FF,FF,FF,FF,FF,FF,FF,FE,B4
350 DATA DD,DF,FF,FF,FF,FF,FF,FF
,FF,FE,B6,DC,DF,FF,FF,FF,FF,FF,F
F,FF,FE,FF,FE,5F,FF,FF,FF,FF,FF,
FF,FF,FE,0,0,F,FF,FF,FF,FF,FF,FF
,FF,FF,0,3,F,FF,FF,FF,FF,FF,FF,F
F,FE,0,B,F,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,E3,FF,FF
360 DATA FF,FF,FF,C3,FF,FF,FF,FF
,FF,83,FF,FF,FF,FF,FF,3,FF,FF,FF
,FF,FE,3,FF,FF,FF,FF,FC,3,FF,FF,
FF,FF,F8,7,FF,FF,FF,FF,F0,F,FF,F
F,FF,C7,E0,1F,FF,FF,FF,80,C0,3F,
FF,FF,FF,1,80,7F,FF,FF,FE,3,0,FF
,FF,FF,FC,6,1,FF,FF,FF,F8,C,7
370 DATA FF,FF,FF,F0,18,1D,FF,FF
,FF,E0,70,30,7F,FF,FF,E1,40,60,3
F,FF,FF,E0,C0,C0,1F,FF,FF,E1,81,
80,1F,FF,FF,E3,B,0,F,FF,FF,E2,16
,0,F,FF,FF,E2,24,2,17,FF,FF,E6,4
8,0,F7,FF,FF,EC,98,4F,FB,FF,FF,F
1,31,1F,BB,FF,FF,E2,60,3F,FD,FF,
FF
380 DATA C4,C2,DF,FE,FF,FF,89,82
,EF,F8,FF,FF,3,1,F7,FB,FF,FE,3,3
,FB,FB,FF,FC,7,83,FD,CB,FF,F8,2F
,EF,FE,BB,FF,F0,3F,E7,FF,7B,FF,E
0,7F,EF,F8,B7,FF,F0,FF,EF,F3,F,F
F,FF,FF,EF,F7,FF,FF,FF,FF,EF,E7,
FF,FF,FF,FF,E0,F,FF,FF,FF,FF,C0,
7
390 DATA FF,FF,FF,FF,C0,6,FF,FF,
FF,FF,C0,2,7F,FF,FF,FF,80,3,7F,F
F,FF,FF,0,3,3F,FF,FF,FF,0,3,3F,F
F,FF,FC,0,2,9F,FF,FF,FC,0,3,5F,F
F,FF,FC,0,3,5F,FF,FF,FC,0,3,1F,F
F,FF,FC,0,1,4F,FF,FF,FC,0,0,AF,F
F,FF,FC,0,1,47,FF,FF
400 DATA FC,0,1,97,FF,FF,FC,0,1,
A7,FF,FF,FC,0,1,87,FF,FF,FC,0,1,
57,FF,FF,FC,0,1,A3,FF,FF,FC,0,1,
4B,FF,FF,FC,0,1,AB,FF,FF,FC,0,3,
4B,FF,FF,FC,0,6,AB,FF,FF,FC,0,5,
53,FF,FF,FC,0,6,A7,FF,FF,F8,0,3,
57,FF,FF,F8,0,15
410 DATA A7,FF,FF,FA,0,6,AF,FF,F
F,FA,0,5,57,FF,FF,F9,0,A,AF,FF,F
F,FA,0,5,57,FF,FF,F9,0,A,AF,FF,F
F,FA,80,5,57,FF,FF,F9,40,A,AF,FF
,FF,FA,A0,D,57,FF,FF,F9,40,A,AF,
FF,FF,F8,A0,5,57,FF,FF,FD,40,9,5
7,FF,FF,FC,A0,A,AF,FF,FF
420 DATA FC,A0,A,AF,FF,FF,FD,50,
5,4F,FF,FF,FC,A0,A,5F,FF,FF,FC,0
,4,5F,FF,FF,FE,A0,A,5F,FF,FF,FE,
0,0,DF,FF,FF,FE,F0,E,5F,FF,FF,FE
,B2,C,DF,FF,FF,FE,B0,8D,DF,FF,FF
,FE,B2,F8,DF,FF,FF,FE,FF,F8,5F,F
F,FF,FE,7,FC,F,FF,FF,FF,7,FF
430 DATA F,FF,FF,FE,7,FF,F,FF,FF
,FE,F3,B7,F,FF,FF,FE,D3,FF,7,FF,
FF,FE,53,DB,F,FF,FF,FE,53,DB,1F,
FF,FF,FE,71,DB,1F,FF,FF,FE,71,E3
,1F,FF,FF,FE,71,85,1F,FF,FF,FE,7
1,83,1F,FF,FF,FF,E1,83,1F,FF,FF,
FF,1,86,3F,FF,FF,FF,81,86,3F,FF,
FF
440 DATA FF,83,8C,3F,FF,FF,FF,83
,98,7F,FF,FF,FF,83,F0,7F,FF,FF,F
F,C0,0,7F,FF,FF,FF,C0,0,7F,FF,FF
,FF,C0,0,7F,FF,FF,FF,C0,0,FF,FF,
FF,FF,C0,0,FF,FF,FF,FF,E0,0,FF,F
F,FF,FF,E0,0,FF,FF,FF,FF,F0,61,F
F,FF,FF,FF,F0,60,FF,FF,FF,FF,F0,
61
450 DATA FF,FF,FF,FF,F0,61,FF,FF
,FF,FF,F8,1,FF,FF,FF,FF,F8,1,FF,
FF,FF,FF,F8,1,FF,FF,FF,FF,F0,1,F
F,FF,FF,FF,F0,1,FF,FF,FF,FF,E0,1
,FF,FF,FF,FF,E0,61,FF,FF,FF,FF,E
0,61,FF,FF,FF,FF,E0,61,FF,FF,FF,
FF,E0,61,FF,FF,FF,FF,C0,1,FF,FF,
FF
460 DATA FF,C0,1,FF,FF,FF,FF,C0,
1,FF,FF,FF,FF,C0,61,FF,FF,FF,FF,
C0,61,FF,FF,FF,FF,C0,61,FF,FF,FF
,FF,C0,61,FF,FF,FF,FF,E0,5,FF,FF
,FF,FF,E0,5,FF,FF,FF,FF,E0,3,FF,
FF,FF,FF,F0,63,FF,FF,FF,FF,F0,63
,FF,FF,FF,FF,F0,63,FF,FF,FF,FF,F
0,63
470 DATA FF,FF,FF,FF,F0,63,FF,FF
,FF,FF,F8,3,FF,FF,FF,FF,F8,3,FF,
FF,FF,FF,FC,3,FF,FF,FF,FF,F8,3,F
F,FF,FF,FF,F8,2,7F,FF,FF,FE,F3,F
7,7F,FF,FF,FA,D0,0,27,FF,FF,FC,B
0,0,5,FF,FF,F0,0,0,12,3F,FF,FC,B
0,0,0,BF,FF,FA,D0,FE,0,7F,FF
480 DATA FF,F0,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF
490 DATA FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FC,0,1,AB,FF,FF,FF,
FF,FF,FC,0,3,4B,FF,FF,FF,FF,FF,F
C,0,6,AB,FF,FF,FF,FF,FF,FC,0,5,5
3,FF,FF,FF,FF,FF,FC,0,6,A7,FF,FE
,FF,FF,FF,F8,0,3,57,FF,FF,7F,FF,
FF,F8,0,1,A7,FF,F8,BF,FF,FF
500 DATA FA,0,0,0,3,F0,D1,FF,BF,
FA,0,0,0,0,6B,C0,0,3F,F9,0,0,0,3
,FF,E0,0,BF,FA,0,0,0,3,FF,8,0,3F
,F9,0,0,0,3,FF,D8,7F,FF,FA,80,0,
0,3,FE,BB,FF,FF,F9,40,0,0,3,FF,2
7,FF,FF,FA,A0,0,0,2,1A,1F,FF,FF,
F9,40,0
510 DATA 0,1F,E0,FF,FF,FF,F8,A0,
0,17,FF,E1,FF,FF,FF,FD,40,19,57,
FF,C7,FF,FF,FF,FC,AA,AA,AF,FF,FF
,FF,FF,FF,FC,AA,AA,AF,FF,FF,FF,F
F,FF,FD,55,55,4F,FF,FF,FF,FF,FF,
FC,AA,AA,5F,FF,FF,FF,FF,FF,FC,0,
44,5F,FF,FF,FF,FF,FF,FE,AA,AA,5F
,FF,FF
520 DATA FF,FF,FF,FE,0,0,DF,FF,F
F,FF,FF,FF,FE,FF,FE,5F,FF,FF,FF,
FF,FF,FE,B6,DC,DF,FF,FF,FF,FF,FF
,FE,B4,DD,DF,FF,FF,FF,FF,FF,FE,B
6,DC,DF,FF,FF,FF,FF,FF,FE,FF,FE,
5F,FF,FF,FF,FF,FF,FE,0,0,F,FF,FF
,FF,FF,FF,FF,0,3,F,FF,FF,FF,FF,F
F
530 DATA FE,0,B,F,FF,FF,FF,FF,FF
,FE,F1,9B,F,FF,FF,FF,FF,FF,FE,D1
,FF,7,FF,FF,FF,FF,FF,FE,51,DF,F,
FF,FF,FF,FF,FF,FE,51,83,1F,FF,FF
,FF,FF,FF,FE,71,83,1F,FF,FF,FF,F
F,FF,FE,71,83,1F,FF,FF,FF,FF,FF,
FE,71,85,1F,FF,FF,FF,FF,FF,FE,71
,83
540 DATA 1F,FF,FF,FF,FF,FF,FF,E1
,83,1F,FF,FF,FF,FF,FF,FF,FC,FF,F
F,FF,FF,FF,E5,81,F0,3F,FF,FF,FF,
FF,FF,E2,C1,F0,3F,FF,FF,FF,FF,FF
,E5,60,F0,3F,FF,FF,FF,FF,FF,EA,A
0,F0,3F,FF,FF,FF,7F,FF,B5,60,F0,
3F,FF,FF,FE,FF,FF,EA,C0,F0,1F,FF
,FF,FD,1F
550 DATA FF,E5,81,B0,1F,FC,FF,8B
,F,C0,0,1,E0,5F,FC,0,3,D6,0,0,1,
E0,5F,FC,0,7,FF,C0,0,1,E0,9F,FC,
0,10,FF,CF,FF,FF,C0,5F,FF,FE,1B,
```

continued on page 19

A screen dump with a few added features

PICTURES

16K Disk
UTILITY

by Charles Syms & Damien Clarke

THE IDEA BEHIND writing this program was that after using screen dump programs for veiwing pictures two problems arose, firstly having lots of paper around and secondly wearing out printer ribbons very quickly.

To use the program you save pictures to disk by using the following command:

```
SAVEM"name",3584,9727,0
```

...and put the pictures program on the disk last. With this method you can have up to 22 pictures and by running "PICTURES" you can view the pictures on the screen by selecting the picture numbers and while viewing the picture you can reverse the colours by hitting "X".

You can load pictures from a cassette system by typing in:

```
CLOADM"picture",2048
```

... for the required offset and then saving to disk as above.

The reason behind saving the pictures first (on a newly formatted disk) is that the program only looks at the first 22 programs. You can load "PICTURES" and substitute another disk with pictures (in the first 22 programs) and then run the program, it will look at pictures or M/L programs but will ignore basic programs.

You can experiment with the program by changing the PMODE in line 230 and by changing the SCREEN COLOURSET in lines 310 and 350 to obtain different effects.

If you wish you can add a screen dump (there is space on the disk for 22 pictures, the PICTURES program and a 1 granule screen dump), as an exercise I modified the program to call the "VERSADUMP" program from Hot Coco. To do this you add to line 20

```
... :DEFUSR1=$H7A00:POKE150
     ,18:LOADM"DUMP"
- delete line 250
- add to line 260
  ..."Y# SCREEN DUMP"
- add line 295
  295 IFD$="Y"THEN360
- after GOSUB350 in line 320
  add
    ... ELSEIFZ$="Y"THEN390
- add line 385
  385 END
- add to line 360
  ... OR 'Y' ...
- after 'X' and add line 390
  390 Z=USR1(0).
```

You now need to save the program "DUMP" screen dump to the disk.

In the program I have used DSKI$ to read the disk directories, in line 70 there is a test for basic programs to delete them from the menu for the pictures.

In lines 205 to 220 the reverse subroutine (in machine language) is poked into memory. This subroutine was written by Damien Clarke of Canberra and I use it with his permission, I was using a basic routine which took one minute to run whereas this routine works **extremely fast.**

In line 260 I use the ASCII values of the letters of the alphabet to write the alphabet on the menu as this saves a lot of programming space. An alternative to this being defining strings, A$(1)="A":A$(2)="B": etc.

Lines 300-310 use a similar system to select which picture is loaded from disk.

The Listing:

```
0 GOTO10
3 SAVE"214:3":END'8
10 'WRITTEN BY C.SYMS. 9 JAN 86.
    REV SUBROUTINE BY D.CLARKE.
20 CLS:CLEAR2000,&H79FF:DEF USR0
=&H7F01:DIM A$(22)
25 'READ DIR FIRST 8 PROGRAMS
30 DSKI$ 0,17,3,B$,C$
40 E$=B$+LEFT$(C$,127)
50 P=1:FORC=1TO8
60 A$(C)=MID$(E$,P,8)
70 H$=MID$(E$,P+8,3):IFH$="BAS"T
HENA$(C)=""
80 P=P+32:NEXTC
85 'READ DIR NEXT 8 PROGRAMS
90 DSKI$ 0,17,4,B$,C$
100 E$=B$+LEFT$(C$,127)
110 P=1:FORC=9TO16
120 A$(C)=MID$(E$,P,8)
130 H$=MID$(E$,P+8,3):IFH$="BAS"
THENA$(C)=""
140 P=P+32:NEXTC
145 'READ DIR LAST 6 PROGRAMS
150 DSKI$ 0,17,5,B$,C$
160 E$=B$+LEFT$(C$,127)
170 P=1:FORC=17TO22
180 A$(C)=MID$(E$,P,8)
190 H$=MID$(E$,P+8,3):IFH$="BAS"
THENA$(C)=""
200 P=P+32:NEXTC
205 'REVerse SUBROUTINE
210 FORA=&H7F01 TO &H7F0B:READB:
POKEA,B:NEXTA
220 DATA 142,14,0,99,128,140,38,
0,37,249,57
230 PMODE4,1
235 'DISPLAY MENU
240 CLS:PRINT"SELECT PICTURE BY
LETTER"
250 PRINT"----------------------
--"
260 L=65:FORC=1TO22:PRINTCHR$(L)
;CHR$(58);" ";A$(C),:L=L+1:NEXTC
:PRINT "W: END",,"X: REVERSE THE
 PICTURE"
265 'SELECT PICTURE NUMBER
270 D$=INKEY$:IFD$=""THEN270
280 IFD$="W"THEN380
290 IFD$="X"THEN360
300 D=ASC(D$):D=D-64
310 PCLS1:SCREEN1,0:LOADMA$(D)
320 Z$=INKEY$:IFZ$="X"THENGOSUB3
50ELSEIFZ$=""THEN320ELSE240
330 GOTO320
340 END
345 'JUMP TO REVerse SUBROUTINE
350 SCREEN1,0:Z=USR0(0):RETURN
355 'ERROR TRAPPING
360 CLS:PRINT"USE THE 'X' FUNCTI
ON ONLY WHILE VEIWING A PICTURE

              HIT A KEY"
370 IF INKEY$=""THEN370ELSE240
380 CLS:PRINT"THANK YOU!!"
```

THE COLOURFUL COCO 3

ARTICLE

by Brian Bere–Streeter

THERE IS NO NEED to repeat the new features of the CoCo 3, as these are, by now, well known. However some of the new commands and functions have a great deal of power and demand exploration.

Amongst the new or revised commands are PALETTE, ATTR, CLS and WIDTH.

PALETTE

Unlike the CoCo 1 or 2 where the colour range was restricted to 8 plus black, the CoCo 3 has a total of 64 colours available.

At any one time a maximum of 16 colours are immediately available, and on power-up and on using the new command PALETTE RGB, the standard set of 16 colours (8 foreground and 8 background) is loaded into 16 'slots' ready for use.

The 8 background colours are the standard green, yellow, blue, red, buff, cyan, magenta, orange, but the foreground only uses black, green, buff, orange, this way maintaining compatability with the old PMODES and 32 x 16 text screens.

Now the good news. You are not restricted to these 8 colours!

Using the new command PALETTE x,y (PALETTE slot no., colour no.) you can load any of the 64 colours into any of the 16 slots in any combination.

Not only can you run a program using your custom set of 16 colours, but during the program run you can, at any time, by calling PALETTE substitute a colour in a particular slot for another colour, and in fact use all 64 colours in the one program, but of course only 16 at any one time.

Also calling PALETTE in a new program line at the start of old programs will enable a colour change to most existing CoCo 1 & 2 programs in both text and graphic modes.

The tables in the back of the manual will show which slots for which colours in the old text and PMODES, and the new coloured text looks nice as cyan or yellow on black, and PMODE 4 as blue on cyan or brown on yellow.

ATTR

This is a new command which lets you put a range of coloured texts and coloured backgrounds on the screen, and uses the standard 8 foreground and 8 background colours, to create in essence, 64 different text colour combinations.

For obvious reasons some combinations are either blank (e.g. blue on blue) or virtually unreadable (e.g. orange on red).

ATTR also provides underlining of text and blinking of text.

Again you are not restricted to the 8 background and 8 (4) foreground colours. Using PALETTE you can set up any of the 64 colours for foreground or background, and the program 'COLRTEXT' demonstrates all combinations of the 64 foreground colours on the 8 standard background colours, all combinations of the 8(4) foreground colours on 64 background colours and a combination of 8 custom foregrounds on the 64 backgrounds.

The 8 custom foregrounds require pre-loading of 8 new colours into the relevant foreground slots (examining the program will show you how) before invoking the ATTR command.

CLS

This is a modified version of the old CLS command, which on the CoCo 1 or 2 clears the screen to the selected colour, but retains the black border around the screen. The modified form of this command in the CoCo 3 clears the entire screen to the selected colour, in the 40 & 80 column modes.

You can give your text intensive programs a colourful lift by calling a CLS colour then an ATTR combination to have, say, blue text on a cyan screen with a red border.

The program 'COLRTEXT' also demonstrates all combinations of the standard 8 colours used in CLS with the standard 8 background colours of the default ATTR 0,x (black text).

Similarly you can modify your programs to display any of the 64 colours in foreground, background or border by using PALETTE to load a new colour into the relevant slot.

WIDTH

This is also a new command and will set either 40 column or 80 column text with full upper and lower case. Note that the 80 column text screen may not be usable on some TV screens.

Samples of both screen formats are shown in the program 'COLRTEXT'.

General.

The program 'COLRTEXT' is more than just a demonstration program, but can be used as a reference when you want to find a new colour combination to put into a program, as the relevant CLS, ATTR and PALETTE numbers are shown on screen with the colours selected.

Take note that when using a colour TV set some colours 'bleed' into each other and therefore some combinations are unusable.

When using the RGB monitor, all colours are crisp & sharp and the improvement over a standard tv is certainly worthwhile. In fact, before I bought my CoCo 3, I even went through the motions of evaluating other alternative computer systems for my CoCo 2 replacement (perish the thought!!), and in doing so evaluated other RGB monitors.

Whilst the price of $700 for the Tandy CM-8 seems high on the surface, a check of other magazine ads shows that RGB monitors range from about $600

to $1200 for digital style and about $750 to $1400 for analogue style (the CM-8 is analogue), so Tandy's price seems competitive.

I certainly feel the price is justified to get the best out of the CoCo 3. If using a monochrome TV, try using:

```
PALETTE0,0:PALETTE 8,63:CLS1
```

... for your 40 or 80 text screen, and adjust the contrast control for clarity.

Alternatively, using standard power-up colours, try using CLS5 to clear to a uniform shade, before using text.

Finally when you use any of the new commands, and want to run a new program, without powering down, remember to reset the default colours with PALETTE RGB before running the new program, or your previous colour selection may do unexpected things to the new program.

Enjoy your CoCo 3, and if you discover other new things about it that are undocumented, please send them in to Australian Coco, for all to share.

The Listing:

```
0 GOTO10
1 '******** COLOUR TEXT ********
   **** BRIAN BERE-STREETER ****
3 SAVE"207C:3":END'10
10 WIDTH30:ON BRK GOTO 5000
20 PALETTE0,0:PALETTE8,18:CLS1
30 LOCATE0,0:ATTR7,0:PRINT STRIN
G$(80,127)
40 FOR X=1TO21:LOCATE0,X:PRINT C
HR$(124):LOCATE 79,X:PRINT CHR$(
124):NEXT
50 LOCATE0,22:PRINT CHR$(124)+ST
RING$(78,127)+CHR$(124);
60 LOCATE20,2:ATTR1,0,U:PRINT"Mu
lti-colour Text Screen Demonstra
tions";:ATTR3,0
70 LOCATE10,4:PRINT"1. Shows def
ault black text on 8 default bac
kground colours";
75 LOCATE10,5:PRINT"   with 8 st
andard colours for borders.";
80 LOCATE10,8:PRINT"2. Shows tex
t in 64 foreground colours on 8
default";
85 LOCATE10,9:PRINT"   backgroun
d colours.";
90 LOCATE10,12:PRINT"3. Shows te
xt in 8 default foreground colou
rs on 64";
95 LOCATE10,13:PRINT"   backgrou
nd colours.";
100 LOCATE10,16:PRINT"4. Shows t
ext in 8 custom foreground colou
rs on 64";
105 LOCATE10,17:PRINT"   backgro
und colours";
150 LOCATE28,20:ATTR1,0,B:PRINT"
Make a Selection > 1 to 4 ";:ATT
R1,0
160 A$=INKEY$:IF A$=""THEN160
170 IF A$="1"OR A$="2"OR A$="3"O
R A$="4" THEN180ELSE160
180 A=VAL(A$)
190 ON A GOTO 1000,2000,3000,400
0
1000 ON BRK GOTO1190
1010 WIDTH40:PALETTE RGB
1020 FOR X=1TO8:CLS X
1030 FOR Y=0TO7:ATTR0,Y
1040 PRINT:PRINT:PRINT:PRINT:PRI
NT:PRINT:PRINT:PRINT:PRINT
1050 PRINT:PRINT:PRINT:PRINT:PRI
NT:PRINT:PRINT:PRINT:PRINT:PRINT
:PRINT:PRINT:PRINT
1060 PRINT"
                    ";
1070 LOCATE0,1:PRINT"    ";:ATTR
0,Y,U:PRINT"COLOUR TEXT SCREENS
USING CLS/ATTR";:ATTR0,Y
1080 LOCATE0,7:PRINT"    Now is
the time for all good men to  c
ome to the aid of the party."
1090 LOCATE0,10:PRINT"    The qu
ick brown fox jumps over the
lazy dog."'COPYRIGHT 1986 BERE-S
TREETER.
1100 LOCATE5,15:PRINT"USE CLS";X
;"THEN ATTR 0,";Y
1110 LOCATE5,18:ATTR0,Y,B:PRINT"
PRESS SPACE FOR NEXT TEXT SCREEN
":ATTR0,Y
1120 LOCATE5,20:PRINT"PRESS BREA
K TO EXIT
1130 A$=INKEY$:IF A$=""THEN1130
1140 IF ASC(A$)=3THEN1150
1150 CLSX
1160 NEXT Y
1170 NEXT X
1180 GOTO1020
1190 PALETTE RGB:WIDTH 40:ATTR2,
0:CLS1:GOTO10
2000 ON BRK GOTO2230
2010 WIDTH40:PALETTE RGB:CLS1
2020 PRINT"    ";:ATTR2,0,U:PRI
NT"TEXT COLOURS USING PALETTE/AT
TR";:ATTR2,0:PRINT
2030 FOR X=0TO63:PALETTE 8,X
2040 LOCATE0,3
2050 ATTR0,0:PRINT"          ";:
ATTR0,1:PRINT"          ";:ATTR0
,2:PRINT"          ";:ATTR0,3:PR
INT"          ";
2060 ATTR0,0:PRINT"   ATTR   ";:
ATTR0,1:PRINT"   ATTR   ";:ATTR0
0,2:PRINT"   ATTR   ";:ATTR0,3:P
RINT"   ATTR   ";
2070 ATTR0,0:PRINT"          ";:
ATTR0,1:PRINT"          ";:ATTR0
,2:PRINT"          ";:ATTR0,3:PR
INT"          ";:ATTR2,0:PRINT
2080 PRINT"   0,0       0,1
  0,2       0,3   ":PRINT
2090 ATTR0,4:PRINT"          ";:
ATTR0,5:PRINT"          ";:ATTR0
,6:PRINT"          ";:ATTR0,7:PR
INT"          ";'COPYRIGHT 1986
BERE-STREETER.
2100 ATTR0,4:PRINT"   ATTR   ";:
ATTR0,5:PRINT"   ATTR   ";:ATTR0
,6:PRINT"   ATTR   ";:ATTR0,7:PR
INT"   ATTR   ";
2110 ATTR0,4:PRINT"          ";:
ATTR0,5:PRINT"          ";:ATTR0
,6:PRINT"          ";:ATTR0,7:PR
INT"          ";:ATTR2,0
2120 PRINT
2130 PRINT"   0,4       0,5
  0,6       0,7":PRINT:PRINT
2140 PRINT"    ";:PRINT"USE PAL
ETTE 8,";X;"THEN ATTR 0, X"
2150 PRINT:PRINT:PRINT"    ";:A
TTR2,0,B:PRINT"PRESS SPACE FOR N
EXT TEXT COLOUR":ATTR2,0
2160 PRINT:PRINT"    PRESS BREA
K TO EXIT"
2170 IF X=0OR X=9OR X=18OR X=27O
R X=36OR X=38OR X=45OR X=54OR X=
63 THEN GOTO2180ELSE2190
2180 LOCATE5,17:PRINT"(STANDARD
ATTRIBUTE COLOUR SET)
2190 A$=INKEY$:IF A$=""THEN2190
2200 IF ASC(A$)=3THEN2210
2210 NEXT
2220 GOTO2010
2230 PALETTE CMP:ATTR2,0:WIDTH40
:GOTO10
3000 ON BRK GOTO3230
3010 WIDTH40:PALETTE RGB:CLS1
3020 PRINT"    ";:ATTR2,0,U:PRI
NT"TEXT COLOURS USING PALETTE/AT
TR";:ATTR2,0:PRINT
3030 FOR X=0TO63:PALETTE 1,X
3040 LOCATE0,3
3050 ATTR0,1:PRINT"          ";:
ATTR1,1:PRINT"          ";:ATTR2
,1:PRINT"          ";:ATTR3,1:PR
INT"          ";
3060 ATTR0,1:PRINT"   ATTR   ";:
ATTR1,1:PRINT"   ATTR   ";:ATTR2
,1:PRINT"   ATTR   ";:ATTR3,1:PR
INT"   ATTR   ";
3070 ATTR0,1:PRINT"          ";:
ATTR1,1:PRINT"          ";:ATTR2
,1:PRINT"          ";:ATTR3,1:PR
INT"          ";:ATTR2,0:PRINT
3080 PRINT"   0,1       1,1
  2,1       3,1   ":PRINT
3090 ATTR4,1:PRINT"          ";:
ATTR5,1:PRINT"          ";:ATTR6
,1:PRINT"          ";:ATTR7,1:PR
INT"          ";'COPYRIGHT 1986
BERE-STREETER.
3100 ATTR4,1:PRINT"   ATTR   ";:
ATTR5,1:PRINT"   ATTR   ";:ATTR6
,1:PRINT"   ATTR   ";:ATTR7,1:PR
INT"   ATTR   ";
3110 ATTR4,1:PRINT"          ";:
ATTR5,1:PRINT"          ";:ATTR6
,1:PRINT"          ";:ATTR7,1:PR
INT"          ";:ATTR2,0
3120 PRINT
3130 PRINT"   4,1       5,1
  6,1       7,1":PRINT:PRINT
3140 PRINT"    ";:PRINT"USE PAL
ETTE 1,";X;"THEN ATTR X, 1"
3150 PRINT:PRINT:PRINT"    ";:A
TTR2,0,B:PRINT"PRESS SPACE FOR N
EXT TEXT COLOUR":ATTR2,0
3160 PRINT:PRINT"    PRESS BREA
K TO EXIT"
3170 IF X=0OR X=9OR X=18OR X=27O
```

continued next page

continued from previous page

```
R X=36OR X=38OR X=45OR X=54OR X=
63 THEN GOTO3180ELSE3190
3180 LOCATE5,17:PRINT"(STANDARD
ATTRIBUTE COLOUR SET)
3190 A$=INKEY$:IF A$=""THEN3190
3200 IF ASC(A$)=3THEN3210
3210 NEXT
3220 GOTO3010
3230 PALETTE CMP:ATTR2,0:WIDTH40
:GOTO10
4000 ON BRK GOTO4240
4010 WIDTH40:PALETTE RGB:CLS1
4020 PALETTE8,45:PALETTE9,38:PAL
ETTE10,0:PALETTE11,27:PALETTE12,
36:PALETTE13,54:PALETTE14,18:PAL
ETTE15,9

4030 PRINT"      ";:ATTR2,0,U:PRI
NT"TEXT COLOURS USING PALETTE/AT
TR";:ATTR2,0:PRINT
4040 FOR X=0TO63:PALETTE 1,X
4050 LOCATE0,3
4060 ATTR0,1:PRINT"            ";:
ATTR1,1:PRINT"          ";:ATTR2
,1:PRINT"          ";:ATTR3,1:PR
INT"          ";
4070 ATTR0,1:PRINT"   ATTR   ";:
ATTR1,1:PRINT"   ATTR   ";:ATTR2
,1:PRINT"   ATTR   ";:ATTR3,1:PR
INT"   ATTR   ";
4080 ATTR0,1:PRINT"            ";:
ATTR1,1:PRINT"          ";:ATTR2
,1:PRINT"          ";:ATTR3,1:PR
INT"          ";:ATTR2,0:PRINT
4090 PRINT"   0,1       1,1
  2,1       3,1   ":PRINT
4100 ATTR4,1:PRINT"            ";:
ATTR5,1:PRINT"          ";:ATTR6
,1:PRINT"          ";:ATTR7,1:PR
INT"          ";'COPYRIGHT 1986
BERE-STREETER.
4110 ATTR4,1:PRINT"   ATTR   ";:
ATTR5,1:PRINT"   ATTR   ";:ATTR6
,1:PRINT"   ATTR   ";:ATTR7,1:PR
INT"   ATTR   ";
4120 ATTR4,1:PRINT"            ";:
ATTR5,1:PRINT"          ";:ATTR6
,1:PRINT"          ";:ATTR7,1:PR
INT"          ";:ATTR2,0
4130 PRINT
4140 PRINT"   4,1       5,1
  6,1       7,1":PRINT:PRINT
4150 PRINT"      ";:PRINT"USE PAL
ETTE 1,";X;"THEN ATTR X, 1"
4160 PRINT:PRINT:PRINT"      ";:A
TTR2,0,B:PRINT"PRESS SPACE FOR N
EXT TEXT COLOUR":ATTR2,0
4170 PRINT:PRINT"      PRESS BREA
K TO EXIT"
4180 IF X=0OR X=9OR X=18OR X=27O
R X=36OR X=38OR X=45OR X=54OR X=
63 THEN GOTO4190ELSE4200
4190 LOCATE5,17:PRINT"(STANDARD
ATTRIBUTE COLOUR SET)
4200 A$=INKEY$:IF A$=""THEN4200
4210 IF ASC(A$)=3THEN4220
4220 NEXT
4230 GOTO4010
4240 PALETTE RGB:PALETTE15,38:AT
TR2,0:WIDTH40:GOTO10
5000 END
```

continued from page 15

```
FF,CF,FF,FF,80,9F,FF,FF,DD,7F,C0
,0,0,1,5F,FF,FF,E4,FF,C0,0,0,2,9
F,FF,FF,F8,58,40,0,0
560 DATA 5,5F,FF,FF,FF,7,F8,0,0,
2,9F,FF,FF,FF,87,FF,E8,0,5,1F,FF
,FF,FF,E3,FF,EA,98,2,BF,FF,FF,FF
,FF,FF,F5,55,55,3F,FF,FF,FF,FF,F
F,F5,55,55,3F,FF,FF,FF,FF,FF,F2,
AA,AA,BF,FF,FF,FF,FF,FF,F8,55,55
,3F,FF,FF,FF,FF,FF,F8,22,0,3F,FF
570 DATA FF,FF,FF,FF,F8,55,55,7F
,FF,FF,FF,FF,FF,FB,0,0,7F,FF,FF,
FF,FF,FF,FB,7F,FF,7F,FF,FF,FF,FF
,FF,FB,3B,6D,7F,FF,FF,FF,FF,FF,F
B,BB,2D,7F,FF,FF,FF,FF,FF,F8,3B,
6D,7F,FF,FF,FF,FF,FF,F8,7F,FF,7F
,FF,FF,FF,FF,FF,FF0,0,0,7F,FF,FF,
FF,FF
580 DATA FF,F0,C0,0,FF,FF,FF,FF,
FF,FF,F0,D0,0,7F,FF,FF,FF,FF,FF,
F0,D9,8F,7F,FF,FF,FF,FF,FF,E0,FF
,8F,7F,FF,FF,FF,FF,FF,F0,FF,8F,7
F,FF,FF,FF,FF,FF,F8,FF,8F,7F,FF,
FF,FF,FF,FF,F8,FF,8F,7F,FF,FF,FF
,FF,FF,F8,FF,8F,7F,FF,FF,FF,FF,F
F,F8,FF
590 DATA 8F,7F,FF,FF,FF,FF,FF,F8
,FF,8F,7F,FF,FF,FF,FF,FF,F8,FF,8
7,7F,8F,7F,FF,FF,FF,FF,FF,F8,FF,
8F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0
600 DATA 0,0,0,D5,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
610 DATA 0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF
620 DATA FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FB,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,BF,FF,FF,
FF,FF,FF,FD,DF,FF,FF,FF,FF,FF,FF
,FF,FF,FF
630 DATA FF,FF,FD,FF,FE,3E,F6,C3
,DF,FF,FE,DD,B2,DF,FD,BC,FE,BD,B
4,DF,DF,74,FE,3C,76,D2,F3,EC,FE,
DD,B6,DB,FF,BC,FE,3D,B6,C3,BB,FF
,FF,FF,FF,FB,FD,FF,FF,FF,FD,FF,E
F,FF,FF,FF,FF,F7,7F,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,BF,FF,FF,FF,FF
,FF,FD,FF
640 DATA FF,FF,FF,FF,FF,FF,FF,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF
650 DATA FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF
660 DATA FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,6C,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
670 DATA 0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
680 DATA 0,0,0,0,0,0,0,0,0,0,0,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF
690 DATA FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,FF,FF,FF,7F,
FF,FF,FF,77,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,DE,FB,FF,FF,FF,FF,FF,D
F,FF,FF,FF,FE,F7,FF,FF,FF,FF,BF,
FF,E3,CD,A3,FF,3D,BF,6D,B4,AF,FF
,BF,7D,E3
700 DATA B5,29,FF,3B,EF,ED,85,AD
,FF,FF,7F,63,B5,A1,FF,FF,F7,FF,F
F,FF,FF,FF,FF,FB,FF,FF,FF,FF,FF,
7F,FF,FF,FF,FF,FF,F7,7F,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,F
F,FF,FF,FF,FF,FF,FF,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
710 DATA 0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,FF,
FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF
```

■

HINT....

Printer Baud Rates

Want to change the printer baud rate of your computer? Then follow the table below and choose your baud rate.

If you want ...	Type in ...
300 baud	POKE150,180
600 baud	POKE150,87
1200 baud	POKE150,41
1800 baud	POKE150,25
2000 baud	POKE150,23
2400 baud	POKE150,18
3600 baud	POKE150,10
4800 baud	POKE150,7
7200 baud	POKE150,3
9600 baud	POKE150,1

MISSION INFILTRATE

32K DECB
ADVENTURE

by Scott Harvey

YOU ARE A member of a highly trained elite of crack commando agents. You have just been told by a Government agent that you have been picked by the government to enter into the KARAN BASE - a former top security military installation and stop KGB agents from developing a super weapon that is being worked upon by their scientists.

You will be further debriefed during the game so prepare to face MISSION INFILTRATE.

About the Program

Mission Infiltrate is a 32K ECB disk Graphic Adventure game. The actual game is made up from two programs.

The first program, "CREATE", when run will write onto the disk a machine language routine which creates the character generator and also creates 15 graphic screens used in the program.

The second program, "INFLTRT" is the actual game and is played on the Hi-res graphic screen.

So if you press break you will need to type SCREEN 0,0 to get back to the text screen.

The game uses to word commands such as GET MACHETE and directions such as N,S,E,W and Up and Down. The rest of the game is fairly self explanatory so enjoy playing it.

The Listing:

```
0 GOTO10
1 '******** CREATE **************
  ****** SCOTT HARVEY *********
3 SAVE"196":END
10 '
15 CLS:PRINT"THIS PROGRAM WILL C
REATE A      MACHINE LANGUAGE PR
OGRAM AND    SAVE IT TO DISK AND
 THEN CREATE 14 PICTURE FILES FO
R THE        ADVENTURE GAME. THI
S WILL TAKE  SOME TIME."
20 CLEAR200,32210
30 FOR X=32210 TO 32714:READY:PO
KEX,Y:NEXTX
40 DATA255,48,141,0,25,191,1,104
,48,141,0,118,191,1,107,134,126,
183,1,103,183,1,106,134,57,167,1
40,229,57,0,52,55,214,111,38,92,
31,2,220,136,196,224,231,140,240
,134,12,61,219,137,224,140,232,2
11,188,195,0,96,158,136,140,5,0,
37,3
50 DATA195,12,0,30,2,129,255,39,
22,129,13,39,26,129,8,39,22,129,
32,39,18,129,47,47,10,129,91,44,
6,32,10,134,46,32,6,134,91,32,2,
134,47,142,126,82,128,45,48,8,74
,38,251,198,8,166,128,167,164,49
,168,32,90,38,246,53,183,52,55,1
34,255
60 DATA32,156,234,234,234,234,23
4,234,234,234,255,255,255,255,25
5,255,255,255,199,187,187,187,18
7,187,199,255,239,207,239,239,23
9,239,131,255,199,187,251,231,22
3,191,131,255,199,187,251,231,25
1,187,199,255,187,187,187,129,25
1,251
70 DATA251,255,131,191,199,251,2
51,187,199,255,199,191,191,167,1
55,187,199,255,131,251,247,239,2
23,191,191,255,199,187,187,199,1
87,187,199,255,199,187,187,195,2
51,251,199,255,255,191,191,255,2
55,191,191,255,255,255,255,255,2
39,239
80 DATA223,191,247,239,223,191,2
23,239,247,255,255,255,255,131,2
55,255,255,255,191,223,239,247,2
39,223,191,255,199,187,251,231,2
39,255,239,255,255,255,255,255,2
55,255,255,255,239,215,187,131,1
87,187,187,255,135,187,187,135,1
87,187
90 DATA135,255,199,187,191,191,1
91,187,199,255,135,187,187,187,1
87,187,135,255,131,191,191,135,1
91,191,131,255,131,191,191,135,1
91,191,191,255,195,191,191,179,1
87,187,195,255,187,187,187,131,1
87,187,187,255,131,239,239,239,2
39,239
100 DATA131,255,251,251,251,251,
251,187,199,255,187,183,175,159,
175,183,187,255,191,191,191,191,
191,191,131,255,187,147,171,187,
187,187,187,255,187,155,171,179,
187,187,187,255,199,187,187,187,
187,187,199,255,135,187,187,135,
191,191

110 DATA191,255,199,187,187,187,
171,183,203,255,135,187,187,135,
175,183,187,255,199,187,191,199,
251,187,199,255,131,239,239,239,
239,239,239,255,187,187,187,187,
187,187,199,255,187,187,187,187,
187,215,239,255,187,187,187,187,
171,147
120 DATA187,255,187,187,215,239,
215,187,187,255,187,187,215,239,
239,239,239,255,131,251,247,239,
223,191,131,255,255,255,255,255,
255,255,255,255,234
190  SAVEM"CHRGEN",32210,32714,3
211
195 CLS:PRINT"CREATING PICTURE F
ILES"
200 PMODE4,1:PCLS5:PMODE3,1:COLO
R5,0
210 LINE(0,0)-(255,84),PSET,B
220 LINE(45,0)-(45,30),PSET:LINE
-(0,84),PSET:LINE(50,20)-(120,40
),PSET,B:LINE(140,10)-(160,40),P
SET,B:LINE(180,15)-(200,40),PSET
,B:LINE(183,23)-(198,37),PSET,BF
230 CIRCLE(190,10),2,5:CIRCLE(20
0,10),2,5
240 DRAW"BM50,20;E10R25U2R2D2R3R
7U2R2D2R1R30G10D20E10U20;BM80,12
;G5R20E5L20"
250 DRAW"BM90,10;U5R5D1"
260 DRAW"BM140,10E10R20D30G10U30
E10;BM180,15;E10U5R20D5L20R20G10
D25E10U25G10"
270 DRAW"BM90,68;D15U15R50E20L50
G20D5R50D10U10E20D10U15"
280 DRAW"BM45,30R5;BM130,30R10;B
M170,30R10;BM210,30R45"
290 PAINT(2,10),5,5:PAINT(60,2),
2,5
300 PAINT(180,2),2,5:PAINT(220,2
),2,5
310 LINE(155,17)-(157,25),PSET,B
F
320 POKE178,30:PAINT(60,60),,5:P
AINT(120,82),,5
330 POKE178,254:PAINT(90,35),,5:
PAINT(125,25),,5
340 POKE178,0
350  SAVEM"SCRN1",3584,6303,0
360 PCLS:LINE(0,0)-(255,84),PSET
,B
370 LINE(40,0)-(40,50),PSET:LINE
-(0,84),PSET
380 DRAW"BM130,5;G5R20E5L20R20D1
0G10E5U10D10L5D5R10E5L10R10G5D20
G10L30E10R30L30U20E5G5R10U5L5U10
;BM155,20;D25G15E10D20R5U25G15U5
D5D20L5U20R5L5L15R5D10L5U10L10U5
D5D20R5U20"
390 PAINT(146,12),5,5:PAINT(142,
21),5,5:PAINT(152,35),5,5:PAINT(
120,59),5,5
400 DRAW"BM50,70;U10R30D10U10E20
D10U10L30G20"
410 DRAW"BM70,50;R15U2L6U13R1E5D
3R1D4L1U4D4D3H5L1L4D13L6D2"
420 DRAW"BM40,50;R20;BM90,50R25;
BM155,50R100"
430 POKE178,190:PAINT(100,10),,5
```

```
:PAINT(20,40),,5
440 POKE178,0
450 SAVEM "SCRN2",3584,6303,0
460 PCLS:LINE(0,0)-(255,84),PSET
,B
470 LINE(0,70)-(255,70),PSET:LIN
E(50,10)-(80,40),PSET,B:LINE(55,
15)-(75,35),PSET,BF:LINE(120,10)
-(170,70),PSET,B:LINE(210,10)-(2
40,40),PSET,B:LINE(215,15)-(235,
35),PSET,BF
480 DRAW"BM120,70;U10R50L50E5R45
L40U10R40L40E5R35L30U10R30L30E5R
25L20U10R20L20E5"
490 PAINT(130,20),5,5
500 PAINT(160,65),2,5:PAINT(160,
58),3,5:PAINT(160,50),2,5:PAINT(
160,44),3,5:PAINT(160,35),2,5:PA
INT(160,28),3,5:PAINT(160,20),2,
5:PAINT(160,14),3,5
510 POKE178,150:PAINT(10,10),,5:
POKE178,0
520 SAVEM "SCRN3",3584,6303,0
530 PCLS:LINE(0,0)-(255,84),PSET
,B
540 LINE(40,0)-(40,60),PSET:LINE
-(255,60),PSET:LINE(40,60)-(0,84
),PSET
550 LINE(60,10)-(90,40),PSET,B:C
OLOR2:LINE(65,15)-(85,35),PSET,B
F:COLOR5:LINE(129,10)-(145,40),P
SET,B:LINE(145,10)-(160,40),PSET
,B:LINE(190,10)-(220,40),PSET,B:
COLOR3:LINE(195,15)-(215,35),PSE
T,BF:COLOR5
560 CIRCLE(145,10),17.5,,.50,.50
,,0
570 LINE(129,10)-(160,40),PSET:L
INE(160,10)-(129,40),PSET:PAINT(
132,25),1,5:PAINT(158,25),1,5:PA
INT(140,15),2,5:PAINT(150,35),2,
5:PAINT(140,35),3,5:PAINT(150,15
),3,5
580 DRAW"BM55,84;E15R150G15;BM60
,84E10R140G10"
590 PAINT(120,73),3,5
600 POKE178, 60:PAINT(10,10),,5:
PAINT(110,10),,5
610 POKE178,0
620 SAVEM "SCRN4",3584,6303,0
630 PCLS:LINE(0,0)-(255,84),PSET
,B
640 LINE(50,0)-(50,50),PSET:LINE
-(255,50),PSET:LINE(50,50)-(0,84
),PSET
650 DRAW"BM85,20;U10L15D2L3R3D2R
3D2R2U2R5D6R5;BM80,25D15R15U15L5
D5L5U5L5D2R5L5D2R5L5D2R15D2L15D2
R15D2L15D2R15U14D2L5R5D2L5R5D2L5
"
660 DRAW"BM95,15;E5R10L7U2R2U5D5
R2D2R3F5L20R8U5D5R2U5D5R2U5D5R8L
8D8L5U8;BM120,35;R2U5E2H2D4U4L3D
4U4L4D4U4L5D1L2G1L1R1F1R2D1R10L2
D2R2U2D2D3"
670 DRAW"BM134,25U15D15L2R6L2D5D
1R2L5R1U5R2U16":PSET(135,10,5)
680 DRAW"BM149,10;E5D5C0D3C0D7C5
H5U5E5D5R4U5F6D5G5U7D17L2U17L1;B
M195,35;L50D2R50U2"
690 DRAW"BM180,10;D15R2U15L2R1U1
R1R8D1L1R2D15L2U15;BM200,10;D5R2
0L3D2R3U2D2D3R3U5R7U5L30R30R1U1D
6"
700 DRAW"BM210,25;R5D5U5L5D15R5U
5R5D5R5U15L5D5L5R5U5R5D5R2D1R2U2
R2D2U2R2U9R2U6D6R1D12R2D3L2U3D3R
7D3L10U15"
710 POKE178,255:PAINT(60,10),,5:
PAINT(10,10),,5:POKE178,2:PAINT(
130,65),,5:POKE178,0
720 SAVEM "SCRN5",3584,6303,0
730 PCLS:LINE(0,0)-(255,84),PSET
,B
740 LINE(60,0)-(60,50),PSET:LINE
-(50,55),PSET:LINE(60,50)-(90,50
),PSET:LINE(210,50)-(220,50),PSE
T:LINE(90,15)-(200,30),PSET,B:LI
NE-(90,45),PSET,B:LINE-(200,60),
PSET,B:LINE(220,50)-(255,60),PSE
T,B
750 DRAW"BM90,15;E10R110G10E10D4
5G10E10R10E10R25"
760 LINE(40,45)-(0,61),PSET:LINE
-(10,70),PSET:LINE-(50,51),PSET:
LINE-(40,45),PSET:LINE(10,70)-(1
0,84),PSET:LINE-(50,60),PSET:LIN
E-(50,50),PSET
770 FORX=90 TO 195 STEP 4:DRAW"B
M"+STR$(X)+",30;U10R4D10U10L4E2"
:NEXT X
780 FOR X=90 TO 195 STEP4:DRAW"B
M"+STR$(X)+",45;U10R4D10U10L4E2"
:NEXTX:FOR X=90 TO 195 STEP4:DRA
W"BM"+STR$(X)+",60;U10R4D10U10L4
E2":NEXT X
790 PAINT(10,10),3,5:PAINT(70,10
),3,5:POKE179,200:PAINT(70,70),1
,5:POKE179,0
800 LINE(210,10)-(212,45),PSET,B
F
810 PAINT(140,25),2,5
820 POKE178,246:PAINT(20,60),,5:
PAINT(20,70),,5:PAINT(5,75),,5
830 POKE178,0
840 SAVEM "SCRN6",3584,6303,0
850 PCLS0:LINE(0,0)-(255,84),PSE
T,B
860 LINE(120,10)-(170,70),PSET,B
870 LINE(0,70)-(255,70),PSET
880 DRAW"BM125,70;U10R45L40U9R40
L35U8R35L30U7R30L25U6R25L20U5R20
L15U4R15L10U3R10L5U2R5L4U2R4L3U2
R3L2U1R2L1U1R1L0"
890 PAINT(130,20),5,5:POKE178,11
:PAINT(10,10),,5
900 POKE178,1:FORY=69 TO10 STEP-
1:X=169:PAINT(X,Y),,5:NEXTY
910 POKE178,2:PAINT(140,75),,5:P
OKE178,0
920  SAVEM"SCRN7",3584,6303,0
930 PCLS0:LINE(0,0)-(255,84),PSE
T,B:LINE(50,0)-(50,60),PSET:LINE
-(0,84),PSET:LINE(120,10)-(210,4
0),PSET,B:LINE(125,15)-(205,35),
PSET,B
940 DRAW"BM90,0;D10G10R20H10G1D2
L1D2R4U2L1U2;BM89,15;D1L1D2R4U2L
1U1"
950 DRAW"BM125,35E5U5R5U5R5D10R1
0D5U5L10U5R5U10D5R15U5D5L5D5R10D
5L5D5U5R5U5R5D5R10U10R10U5D5L5D5
R5D10U10R10D5R5"
960 DRAW"BM100,80;U10E20R90D10U1
0G20D10U10L90;BM50,60;R60;BM200,
60R55"
970 POKE178,2:PAINT(130,20),,5:P
OKE178,10:PAINT(135,30),,5:POKE1
78,100:PAINT(150,28),,5:POKE178,
70:PAINT(155,30),,5:POKE178,234:
PAINT(175,20),,5:POKE178,245:PAI
NT(185,30),,5:POKE78,30:PAINT(19
5,30),,5:POKE178,0
980 POKE178,11:PAINT(200,20),,5:
POKE178,210:PAINT(10,10),5,5:PAI
NT(70,10),,5:PAINT(206,57),,5:PO
KE178,0
990 SAVEM"SCRN8",3584,6303,0
1000 PCLS0:LINE(0,0)-(255,84),PS
ET,B
1010 FORX=0TO250STEP20:DRAW"BM"+
STR$(X)+",40;U10R10D10R10":NEXTX
1020 S=S+1:IFS=10 THEN GOTO1030E
LSEX=RND(255):Y=RND(20):FORN=0 T
O10:CIRCLE(X,Y),N,3,.25:NEXTN:GO
TO1020
1030 POKE178,167:PAINT(128,60),,
5:POKE178,0
1040 SAVEM "SCRN9",3584,6303,0
1050 PCLS0:LINE(0,0)-(255,84),PS
ET,B
1060 LINE(50,0)-(50,60),PSET:LIN
E-(0,84),PSET:LINE(70,40)-(140,7
0),PSET,B:LINE(70,30)-(140,40),P
SET,B:LINE(105,35)-(110,50),PSET
,B
1070 DRAW"BM70,30;E10R70G10D10E1
0U10G10L70;BM106,40C0;R3C5;BM150
,30;D30G10"
1080 LINE(180,10)-(200,40),PSET,
B:DRAW"BM185,12;D8R2D5R6U5R2U8L1
0;BM185,25;U2L2D2R2F10D2R2U2L2;B
M195,25;U2R2D2L2G10D2L2U2R2;BM18
7,14;D2R2U2L2;BM192,14;D2R2U2L2;
BM189,17;D2R2U2L2"
1090 LINE(50,60)-(70,60),PSET:LI
NE(150,60)-(255,60),PSET
1100 POKE178,34:PAINT(20,10),,5:
PAINT(90,10),,5:POKE178,4:PAINT(
100,60),,5:PAINT(145,50),,5:POKE
178,2:PAINT(90,35),,5:PAINT(100,
25),,5:PAINT(145,30),,5:POKE178,
0
1110  SAVEM"SCRN10",3584,6303,0
1120 PCLS0:LINE(0,0)-(255,84),PS
ET,B
1130 LINE(40,0)-(40,40),PSET:LIN
E-(255,40),PSET:LINE(40,40)-(0,8
4),PSET:LINE(130,0)-(170,40),PSE
T,B
1140 DRAW"BM130,30;R40L40E5R35L3
0U10R30L30E5R25L20U10;BM70,80E30
R120G30L120"
1150 PAINT(160,5),3,5:PAINT(160,
13),2,5:PAINT(160,20),3,5:PAINT(
160,28),2,5:PAINT(160,35),3,5
1160 POKE178,1:PAINT(150,60),,5:
POKE178,89:PAINT(20,10),,5:PAINT
(60,10),,5:PAINT(200,10),,5:POKE
178,0
1170  SAVEM"SCRN11",3584,6303,0
1180 PCLS0:LINE(0,0)-(255,84),PS
ET,B:LINE(50,0)-(50,70),PSET:LIN
E-(255,70),PSET:LINE(50,70)-(0,8
4),PSET
1190 DRAW"BM130,45;U5D5R10D5R5U1
0R5D5R5U5R5U5R10U5L5U10L5D5L10U5
L5D5L5U5L5D10R5D5L5D5L5"
1200 PAINT(150,35),5,5:POKE178,2
52:PAINT(10,40),,5:PAINT(90,40),
,5:POKE178,0
1210  SAVEM "SCRN12",3584,6303,0
1220 PCLS0:LINE(0,0)-(255,84),PS
ET,B:LINE(50,0)-(50,70),PSET:LIN
E-(255,70),PSET:LINE(50,70)-(0,8
4),PSET:LINE(140,10)-(180,50),PS
ET,B:LINE(145,15)-(175,45),PSET,
B:DRAW"BM152,20;D2R4U2L4;BM164,2
0;D2R4U2L4"
1230 PAINT(20,10),5,5:POKE178,25
2:PAINT(90,10),,5:POKE178,0
```

```
1240 SAVEM "SCRN13",3584,6303,0
1250 PCLS0:LINE(0,0)-(255,84),PS
ET,B:LINE(50,0)-(50,50),PSET:LIN
E-(0,84),PSET:LINE(70,40)-(190,6
0),PSET,B:LINE(75,45)-(185,55),P
SET,B:LINE(230,10)-(255,60),PSET
,B:LINE(235,15)-(245,25),PSET,B:
LINE(238,18)-(242,23),PSET,BF
1260 DRAW"BM70,40;E20R40;BM155,2
0;R55G20D20E20U20G20;BM195,40;E1
0D10G10U10;BM110,35;E10R10L10G10
R45E10L10"
1270 DRAW"BM130,30;U10R20D10L20U
10E5R5L5G5R20E5D10G5U10E5L10R10E
5L5D5E5D5E5D5E5D5E5D5E5D5E5D5E5D
5E5D5E5D5E5D5E5D5E5R5F10"
1280 DRAW"BM140,18;R5U8R3E2R5U1L
5H2L4U2R3D2R3L3L6D2L15L3L3R9L9D1
L3R3L6R6R15D2R10L5D8"
1290 DRAW"BM250,15;D5R5L5D5;BM25
5,25G5D5F5;BM235,30;R10D5L10D5R1
0D5L10D5R10D5L10"
1300 DRAW"BM230,10;E10"
1310 PAINT(130,50),5,5:PAINT(200
,40),5,5:PAINT(120,43),3,5:PAINT
(207,30),3,5:PAINT(100,30),3,5
1320 PAINT(196,35),3,5:PAINT(195
,37),3,5:PAINT(197,34),3,5:PAINT
(198,33),3,5:PAINT(200,31),3,5
1330 LINE(50,50)-(70,50),PSET:LI
NE(200,50)-(230,50),PSET
1340 POKE178,6:PAINT(20,10),5,5:
PAINT(70,10),,5:PAINT(220,20),,5
:POKE178,0
1350  SAVEM"SCRN14",3584,6303,0
1360 PCLS0:LINE(0,0)-(255,84),PS
ET,B:LINE(70,70)-(170,80),PSET,B
:LINE(0,80)-(255,80),PSET:LINE(1
20,70)-(135,80),PSET,B
1370 DRAW"BM110,30;R70G10L70E10G
10D20L10D10U10R10U20R70D20R10L10
E10U20D20R10G10;BM170,70;E25L15R
15D10G25U10;BM70,70;E25R5;BM130,
65;D5U5R10D5"
1380 FOR X=110 TO 165 STEP10:DRA
W"BM"+STR$(X)+",50;U5R5D5L5":NEX
TX
1390 FOR X=105 TO 160 STEP10:DRA
W"BM"+STR$(X)+",60;U5R5D5L5":NEX
TX
1400 FOR X=5TO65 STEP10:CIRCLE(X
,75),5:PAINT(X,75),5,5:NEXTX
1410 FOR X=250 TO 180 STEP-10:CI
RCLE(X,75),5:PAINT(X,75),5,5:NEX
TX
1420 PAINT(90,55),5,5:PAINT(185,
46),5,5
1430 PAINT(90,75),3,5:PAINT(160,
75),3,5:PAINT(185,60),3,5:PAINT(
140,35),2,5:PAINT(155,65),2,5:PA
INT(175,40),2,5:PAINT(180,55),2,
5
1440 LINE(0,15)-(255,15),PSET
1450 CIRCLE(138,15),10,,.55,.50,
.0
1460 POKE178,2:PAINT(10,5),,5:PO
KE178,0
1470  SAVEM"SCRN15",3584,6303,0
1480 CLS:PRINT"COMPLETED."
```

■

The Listing:

```
0 GOTO10
1 '********* INFILTRATER *******
   ****** SCOTT HARVEY *********
3 SAVE"196-:3":END
10 IF PEEK(32211)=57 THEN GOTO 2
0ELSECLEAR200,32210:LOADM"CHRGEN
":POKE492,PEEK(360):POKE493,PEEK
(361):POKE494,PEEK(363):POKE495,
PEEK(364):POKE32211,48:EXEC32211
20 CLEAR400:GOSUB2880:PMODE4,1:P
CLS5:EXEC32211:SCREEN1,1:PMODE3,
1
30 T1=0:PH=0
40 COLOR5,0:GOSUB3270
50 POKE 65495,0
60 DIMR$(30),RM(30),D(30,6),V$(2
2),O$(20,3),LO(18),SC(18),C$(6)
70 REM DATA
80 DATA IN A FOREST,0,0,2,0,0,0
90 DATA OUTSIDE WEST WALL,0,0,3,
1,0,0
100 DATA OUTSIDE FORTRESS WALL,0
,0,4,2,0,0
110 DATA IN FRONT OF A HUGE GATE
,0,0,5,3,0,0
120 DATA NEAR SOME THICK BUSH,0,
0,0,4,0,0
130 DATA IN A CAVE,0,0,0,5,0,0
140 DATA IN CELL NO.1,0,0,8,0,0,
0
150 DATA IN THE ELECTROSHOCK ROO
M,0,14,0,7,0,0
160 DATA IN A LONG CORRIDOR,0,15
,10,0,0,0
170 DATA IN A LARGE FOYER,1,16,0
,9,0,0
180 DATA IN A DINING ROOM,0,17,1
2,0,0,0
190 DATA IN THE KITCHEN,0,0,0,11
,0,0
200 DATA IN CELL NO. 2,7,0,0,0,0
,0
210 DATA IN THE INTERROGATION RO
OM,8,20,15,0,0,0
220 DATA IN A CORRIDOR,9,21,0,14
,0,0
230 DATA IN AN EMPTY ROOM,10,22,
17,0,0,0
240 DATA AT A STAIRWAY,11,0,0,16
,18,0
250 DATA IN THE ATTIC,0,0,0,0,0,
17
260 DATA IN THE WEAPONS ROOM,0,2
5,20,0,0,0
270 DATA AT A STAIRWAY,14,0,21,1
9,0,26
280 DATA CORRIDOR,15,27,0,20,0,0
290 DATA IN THE LIBRARY,16,0,23,
0,0,0
300 DATA IN THE H.Q. ROOM,0,0,24
,22,0,0
310 DATA OUT ON A TURRET,0,0,0,2
3,0,0
320 DATA IN THE GUARD ROOM,19,0,
0,0,0,0
330 DATA IN A DARK BASEMENT,0,0,
0,0,20,0
340 DATA AT THE CORRIDOR'S END,2
1,0,0,0,0,0
350 DATA IN THE SECRET CHAMBER,2
2,0,29,0,0,0
360 DATA SECRET PASSAGE,0,0,30,2
8,0,0
370 DATA IN THE LABORATORY,0,0,0
,29,0,0
380 R= 30:FORI=1TOR:READR$(I):FO
RA=1TO6:READD(I,A):NEXT:NEXT
390 REM FORMAT FOR OBJECT DATA=D
ESCRIPTION,KEYWORD,ROOM #LOCATIO
N,POINT #VALUE,RESPONSE WHEN EXA
MINED
400 DATA MACHETE,MACHETE,1,1,STU
RDY AND SHARP.
410 DATA LOCK,LOCK,4,-2,IT USES
PASS KEYS TO OPEN.
420 DATA DEAD BODY,BODY,6,-1,APP
EARS TO BE A DEAD GUARD.
430 DATA PASS KEY,KEY,0,1,LOOKS
LIKE IT BELONGS IN A LOCK
440 DATA LONG HANDLED SHOVEL,SHO
VEL,8,1,IT IS WELL MADE AND STRO
NG
450 DATA THICK BUSH,BUSH,5,-1,LO
OKS PRETTY THICK
460 DATA BOMB TIMING DEVICE,TIME
R,13,25,IT IS A MK3 BOMB TIMING
DEVICE
470 DATA LASER GUN,GUN,19,0,IT I
S CHARGED
480 DATA LARGE BOOKCASE,CASE,22,
0,LOOKS INTERESTING
490 DATA EXPLOSIVES BOOK,BOOK,0,
0,LOOKS INTERESTING
500 DATA LARGE OAK CHEST,CHEST,2
5,-1,IT IS MADE OF OAK
510 DATA PLASTIC EXPLOSIVES,EXPL
OSIVE,0,0,LOOK PRETTY DANGEROUS
520 DATA RED RUG,RUG,26,-1,IT IS
 A RED PERSIAN RUG.
530 DATA BOMB CASING,CASING,0,0,
IT IS HOLLOW
540 DATA BIG PADLOCK,PADL,25,-1,
LOOKS TOUGH TO CRACK.
550 DATA A BENCH,BENCH,8,-1,
560 DATA REFRIGERATOR,FRIDGE,12,
-1,MAKES NOISE.. COCKROACHES??
570 DATA A TIME BOMB,BOMB,0,-1,I
T ISN'T SET.
580 O= 18:FORI=1TOO:READO$(I,1),
O$(I,2),LO(I),SC(I),O$(I,3):NEXT
:T=RND(-TIMER):FORI=1TOO:IFLO(I)
=-2 THENLO(I)=RND(O):NEXT:ELSENE
XT
590 V$( 1)="EXAMINE"
600 V$( 2)="INVENTORY"
610 V$( 3)="QUIT"
620 V$( 4)="SCORE"
630 V$( 5)="HELP"
640 V$( 6)="LOAD"
650 V$( 7)="SAVE"
660 V$( 8)="GET"
670 V$( 9)="DROP"
680 V$( 10)="CUT"
690 V$( 11)="OPEN"
700 V$( 12)="SEARCH"
710 V$( 13)="DIG"
720 V$( 14)="SET"
730 V$( 15)="FIRE"
740 V$( 16)="LIFT"
750 V$( 17)="BUILD"
760 V$( 18)="RUN"
770 V$( 19)="READ"
780 V$( 20)="PUSH"
790 V$( 21)=""
800 V$( 22)=""
810 NV= 22:FORI=1TONV
820 IFLEN(V$(I))<4 THEN V$(I)=V$
(I)+CHR$(32):GOTO 820
830 V1$=V1$+LEFT$(V$(I),4):NEXT.
840 FORI=1TO O
850 IFLEN(O$(I,2))<4 THENO$(I,2)
=O$(I,2)+CHR$(32):GOTO 850
860 N1$=N1$+LEFT$(O$(I,2),4):NEX
T
```

```
870 DATANORTH,SOUTH,EAST,WEST,UP
,DOWN:FORDD=1TO6:READC$(DD):NEXT
DD
880 L= 1:L5= 1:T=0:SG$=STRING$(3
2,217):EL$=STRING$(32,32):CLS:LN
=0
890 FOR I=1TO600:NEXTI:PCLS:LINE
(0,0)-(255,84),PSET,B:PRINT@224,
"YOU ARE "R$(L)"."
900 ON L GOSUB 1990,2040,2070,21
00,2150,2190,2300,2350,2420,2460
,2530,2610,2300,2650,2420,2620,2
660,2670,2680,2690,2420,2700,271
0,2720,2730,2740,2620,2750,2760,
2770
910 PRINT"YOU SEE:";
920 Z=0:FORA=1TOO
930 IFLO(A)=L AND POS(0)+LEN(O$(
A,1))>32 THENPRINT
940 IFLO(A)=L THENPRINTO$(A,1)+C
HR$(44);:Z=1
950 NEXT:PRINTCHR$(8);".";
960 IFZ=0THENPRINTCHR$(8)+":NOTH
ING OF INTEREST."
970 PRINT:PRINT"OBVIOUS EXITS LE
AD: "
980 FORG=1TO6:IFD(L,G)<>0THENPRI
NTC$(G)+CHR$(32);
990 NEXT
1000 IFLN>0 THENFORJ=P TO LN-102
4 STEP32:PRINT@J,EL$;:NEXTJ:LN=0
1010 FORI=1TO700:NEXTI:FORI=416+
9 TO 480+31:PRINT@I,CHR$(32);:NE
XTI:PRINT@416,;:TURNS=TURNS+1:I$
="":LINEINPUT"WHAT NOW? ";I$
1020 COLOR5,0
1030 IFI$=""THENPRINT@448,"WHAT?
":GOTO 1010
1040 IF L=10 AND I$="N"ANDTI=0 T
HEN GOTO 2780
1050 IF L=20 AND I$="D" AND CA=>
2 THEN GOTO 2860
1060 IF L=10 AND I$="N"ANDPH=1TH
ENL=4:GOTO890
1070 IF PH=1 THEN PRINT@480,"TIM
E LEFT:";TI:TI=TI-1
1080 IF PH=1 AND TI=0 AND L<>1TH
EN GOTO 3100
1090 IFI$="LOOK"THEN 890
1100 IFLEN(I$)>1THEN 1140
1110 L5=L
1120 G=INSTR("NSEWUD",I$):IFG=0T
HENPRINT@448,"I DON'T UNDERSTAND
.":GOTO 1010
1130 IFD(L,G)>0THEN L5=D(L,G):L=
L5:GOTO 890:ELSEPRINT@416,"YOU C
AN'T GO THAT WAY.":GOTO 1010
1140 I$=I$+" ":SP=INSTR(I$,CHR$(
32))
1150 V2$=LEFT$(I$,SP-1):N2$=MID$
(I$,SP+1):V$=LEFT$(V2$,4):N$=LEF
T$(N2$,4):V=INSTR(V1$,V$):N=INST
R(N1$,N$)
1160 IFV=0THENPRINT"I DON'T UNDE
RSTAND.":GOTO 1010:ELSEV=(V-1)/4
+1
1170 IFN=0THENPRINT"I DON'T UNDE
RSTAND.";GOTO 1010:ELSEN=(N-1)/4
+1
1180 ON V GOTO1190,1250,1320,134
0,1360,1380,1400,1420,1500,1530,
1570,1640,1670,1700,1740,1780,18
10,1850,1870,1910,1940,1960
1190 REM VERB # 1 EXAMINE
1200 IF L=8 AND N=16 THEN GOTO28
20
1210 IF LO(N)<>-1 AND LO(N)<>L T
HENPRINT"YOU CAN'T EXAMINE SOMET
HING YOU DO NOT HAVE OR CANNOT S
EE.":GOTO 1010
1220 IFO$(N,3)=""THENPRINT"NOTHI
NG SPECIAL.":GOTO 1010
1230 PRINTO$(N,3):GOTO 1010
1240 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1250 REM VERB # 2 INVENTORY
1260 PCLS:CLS
1270 PRINT"YOUR INVENTORY:":NH=0
1280 FORI=1TOO:IFLO(I)=-1THENH=1
:PRINTO$(I,1)
1290 NEXT:IFNH=0THENPRINT"NOTHIN
G."
1300 GOTO 1010
1310 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1320 REM VERB # 3 QUIT
1330 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1340 REM VERB # 4 SCORE
1350 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1360 REM VERB # 5 HELP
1370 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1380 REM VERB # 6 LOAD
1390 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1400 REM VERB # 7 SAVE
1410 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1420 REM VERB # 8
1430 IF L=25 AND N=12 AND LO(14)
<>-1 THEN GOTO2800
1440 IF LO(N)=-1 THEN PRINT"YOU
ALREADY HAVE IT.":GOTO3230
1450 IF LO(N)<>L THEN PRINT"I DO
N'T SEE IT.":GOTO3230
1460 IF CA=5 THEN PRINT"YOUR ARM
S ARE FULL.":GOTO3230
1470 IF SC(N)=-1 THENPRINT"YOU C
AN'T GET THAT.":GOTO 3230
1480 LO(N)=-1:CA=CA+1:PRINT"OKAY
. YOU HAVE IT.":GOTO3230
1490 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1500 REM VERB # 9
1510 IF LO(N)=1 THEN PRINT "YOU
ARE NOT CARRYING IT.":GOTO3230
1520 LO(N)=L:CA=CA-1:PRINT"OKAY
DROPPED.":GOTO3230
1530 REM VERB # 10
1540 IF L=5 AND LO(1)=-1 AND MID
$(N1$,21,4)="BUSH"THEN :PCLS:CLS
:PRINT "YOU HAVE HACKED THROUGH
THE     BUSH WHICH NOW REVEALS A
 CAVE   ENTRANCE WHICH YOU ENTER
.":L=6:GOTO 3230
1550 IF L=5 AND N<>1 THEN PRINT
"YOU CAN'T CUT WITH THAT.":GOTO3
230
1560 PRINT "YOU CAN'T CUT THAT."
:GOTO 3230
1570 REM VERB # 11
1580 IF L=12 AND N=17 THEN GOTO
2840
1590 IFL=4 AND N=2ANDLO(4)=-1THE
N PCLS:CLS:PRINT"THE GATE OPENS
AND YOU ENTER THE FOYER OF THE L
ARGE FORTRESS":L=10:GOTO3230
1600 IF L=4 AND N=2AND LO(4)=0TH
EN PRINT "YOU NEED A KEY..":GOTO
3230
1610 IF L=25 AND N=11ANDLO(15)=0
 THEN PRINT "THE CHEST OPENS REV
EALING. SOME EXPLOSIVES.":LO(12)
=25:GOTO3230
1620 IF L=25 AND N=11ANDLO(15)=2
5 THENPRINT"IT'S LOCKED TIGHT":G
OTO3230
1630 PRINT"YOU CAN'T OPEN THAT."
:GOTO1010
1640 IF L=6  AND N=3THEN PRINT "
YOU FIND A PASS KEY IN HIS LEFT
POCKET.":LO(4)=6:GOTO3230
1650 IF L=22 AND N=9 THEN PRINT
"YOU HAVE FOUND A BOOK ON
 EXPLOSIVES":LO(10)=22:GOTO 3230
1660 PRINT"YOU FIND NOTHING.":GO
TO 1010
1670 REM VERB # 13
1680 IF L=7 AND LO(5)=-1 THEN PC
LS:CLS:PRINT"YOU HAVE DUG A TUNN
EL AND ARE   NOW IN CELL NO .2":
L=13:GOTO3230
1690 PRINT"YOU FIND NOTHING.":GO
TO1010
1700 REM VERB # 14
1710 IF L=30  AND LO(18)=30AND N
=18 THEN PCLS:CLS:INPUT "HOW MAN
Y SECONDS 0-60";TI:PRINT "OKAY S
ET TO ";TI;" SECONDS.YOU HAD BET
TER GET BACK TO THE FOREST  QUIC
KLY.":PH=1:GOTO 3230
1720 IF N<>7 THEN PRINT "YOU CAN
'T SET THAT.":GOTO3230
1730 PRINT"YOU CAN'T DO THAT YET
.":GOTO1010
1740 REM VERB # 15
1750 IF L=25 AND LO(8)=-1 AND N=
8THEN PCLS:CLS:PRINT "THE LASER
GUN EMITS A LOW BUZZ  AND A THIN
 BEAM OF RADIANT      LIGHT SHOO
TS FROM THE GUN TO    THE PADLOC
K MELTING IT INTO A   POOL OF BU
BBLING MOLTEN METAL.":LO(15)=0:G
OTO 3230
1760 IF N<>8 THEN PRINT "YOU CAN
'T FIRE THAT.":GOTO 3230
1770 PRINT"NOTHING HAPPENS.":GOT
O1010
1780 REM VERB # 16
1790 IF L=26 AND N=13THENPRINT"Y
OU FOUND A BOMB CASING":LO(14)=2
6:GOTO 3230
1800 PRINT"YOU CAN'T LIFT THAT."
:GOTO1010
1810 REM VERB # 17
1820 IF L=30 AND N=7 AND LO(7)=-
1ANDLO(12)=-1ANDLO(14)=-1THENPRI
NT"YOU HAVE MADE THE TIME BOMB."
:LO(12)=0:LO(14)=0:LO(7)=0:LO(18
)=30:GOTO 3230
1830 IF L<>30ANDN=7ANDLO(7)=-1AN
DLO(12)=-1ANDLO(14)=-1THENPRINT"
YOU CAN'T YET.":GOTO3230
1840 PRINT"YOU CAN'T BUILD THAT.
":GOTO1010
1850 REM VERB # 18
1860 PRINT"WHY?":GOTO1010
1870 REM VERB # 19
1880 IF LO(10)=-1 AND N=10 THENP
CLS:CLS:PRINT"THE BOOK SAYS: THE
 MK3 TIMING   DEVICE WHEN USED I
N CONJUNCTION WITH PLASTIC EXPLO
SIVES MUST    ALWAYS BE SET TO 3
4 SECONDS NO  MATTER WHAT.ALSO T
HE CASING     MUST FIRST BE FOUN
D FOR THE     ";
1890 IF LO(10)=-1 AND N=10 THEN
PRINT"EXPLOSIVES BEFORE HANDLING
 OR   ELSE THEY WILL TURN VOLATI
LE    AND DETONATE.":GOTO3230
1900 PRINT "YOU CAN'T READ THAT.
":GOTO1010
```

```
1910 REM VERB # 20
1920 IF L=22 AND N=9 THEN  PCLS:
CLS:PRINT "IT REVEALS A SECRET C
HAMBER      WHICH YOU ENTER":L=28
:GOTO3230
1930 PRINT"NOTHING HAPPENS.":GOT
O1010
1940 REM VERB # 21
1950 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1960 REM VERB # 22
1970 PRINT"I DON'T UNDERSTAND.":
GOTO 1010
1980 GOTO 3230
1990 IF TI=26 THEN GOTO3100ELSE
IF PH=1 AND TI<>26 THEN GOTO 315
0ELSEFOR X=10 TO 250 STEP 30:LIN
E(X,10)-(X+10,84),PSET,B:NEXTX
2000 FOR X=2 TO 250 STEP 46:CIRC
LE(X,15),40,3,.60,.15,.75:NEXTX
2010 FOR X=2 TO 250 STEP 45:PAIN
T(X,15),5,3:NEXTX
2020 IF LO(1)=1 THEN PUT(160,68)
-(197,83),Z2,PSET
2030 RETURN
2040 LINE(1,15)-(254,15),PSET:FO
R X=0 TO 255 STEP 5:DRAW"BM"+STR
$(X)+",15;D4R5U4":NEXT X
2050 PAINT(128,54),3,5
2060 RETURN
2070 LINE(1,15)-(254,15),PSET:FO
RX=0TO250 STEP 10:DRAW"BM"+STR$(
X)+",15;D4R5U4L4":NEXT X
2080 PAINT(128,54),3,5
2090 RETURN
2100 LINE(1,15)-(98,15),PSET:LIN
E(158,15)-(254,15),PSET:LINE(99,
10)-(99,84),PSET:LINE(157,10)-(1
57,84),PSET:CIRCLE(128,10),30,,.
25,.50,.0
2110 LINE(160,54)-(168,50),PSET,
BF
2120 LINE(98,24)-(104,26),PSET:L
INE-(98,28),PSET:LINE(98,64)-(10
4,66),PSET:LINE-(98,68),PSET
2130 PAINT(50,54),3,5:PAINT(200,
54),3,5:PAINT(128,54),2,5
2140 RETURN
2150 FORX=2 TO 250 STEP 35:CIRCL
E(X,54),30,3,.55,.15,.85:NEXTX
2160 FORX=2 TO 250 STEP 35:PAINT
(X,54),5,3:NEXTX
2170 PAINT(128,82),3,5
2180 RETURN
2190 LINE(255,81)-(205,71),PSET,
B:LINE(255,71)-(210,65),PSET,B
2200 LINE(205,72)-(202,72),PSET:
LINE(205,81)-(203,81),PSET
2210 LINE(202,72)-(200,66),PSET:
LINE-(200,62),PSET:LINE-(197,62)
,PSET:LINE-(197,81),PSET:LINE-(2
05,81),PSET
2220 LINE(210,66)-(208,66),PSET:
LINE-(206,60),PSET:LINE-(206,56)
,PSET:LINE-(203,56),PSET:LINE-(2
03,72),PSET
2230 PAINT(245,66),3,5:PAINT(245
,78),3,5
2240 CIRCLE(128,55),30,5,3,.50,.
0
2250 LINE(0,54)-(255,54),PSET
2260 PAINT(128,53),5,5
2270 PAINT(2,2),2,5:PAINT(234,10
),2,5
2280 IF LO(4)=6 THEN PUT(10,65)-
(27,81),Z3,PSET
2290 RETURN
2300 LINE(30,0)-(30,54),PSET:LIN
E-(255,54),PSET:LINE(30,54)-(0,8
4),PSET
2310 DRAW"BM42,37;G10D5R100U5L10
0R100E10L100R100D5G10U5"
2320 LINE(40,45)-(60,25),PSET:LI
NE(125,45)-(145,27),PSET
2330 IF L=13ANDLO(7)=13 THEN PUT
(180,75)-(158,60),Z4,PSET
2340 RETURN
2350 LINE(50,0)-(50,34),PSET:LIN
E-(255,34),PSET:LINE(50,34)-(0,8
4),PSET
2360 DRAW"C2BM108,44;E30R30D10G3
0U10L30D20U10R30D10U20E30G30D10L
30R30E30D10"
2370 PAINT(134,24),3,2:PAINT(163
,26),3,2
2380 DRAW"C5BM124,33;E5R5F5L15E5
L3U3H3D3H3D3H3D3H3D3H3D3H3D3H3D3
H3D3H3D3D15"
2390 LINE(88,36)-(98,50),PSET,BF
2400 IF LO(5)=8 THEN PUT(160,65)
-(215,78),Z1,PSET
2410 RETURN
2420 LINE(0,0)-(113,30),PSET:LIN
E(0,84)-(113,54),PSET:LINE(255,0
)-(143,30),PSET:LINE(255,84)-(14
3,54),PSET
2430 LINE(113,30)-(143,54),PSET,
B
2440 PAINT(2,20),2,5:PAINT(253,2
0),2,5
2450 RETURN
2460 LINE(20,0)-(20,30),PSET:LIN
E-(125,30),PSET:LINE(150,30)-(25
5,30),PSET:LINE(20,30)-(0,84),PS
ET:LINE(55,0)-(75,15),PSET,B:LIN
E(50,0)-(80,20),PSET,B:LINE(190,
0)-(220,20),PSET,B:LINE(195,0)-(
215,15),PSET,B
2470 DRAW"BM65,60;D10U10E20R20D1
0U10G20D10U10L20":DRAW"BM120,35;
D10U5R20D5U10L20E10R20D10U5G10U5
E10U15L20D15R20"
2480 LINE(120,84)-(130,60),PSET:
LINE-(160,60),PSET:LINE-(150,84)
,PSET
2490 PAINT(2,4),2,5:PAINT(128,3)
,2,5
2500 PAINT(65,5),5,5:PAINT(205,5
),5,5
2510 PAINT(140,75),3,5
2520 RETURN
2530 LINE(40,0)-(40,30),PSET:LIN
E-(255,30),PSET:LINE(40,30)-(0,8
4),PSET
2540 LINE(60,65)-(190,65),PSET:L
INE-(210,35),PSET:LINE-(80,35),P
SET:LINE-(60,65),PSET:LINE(90,40
)-(190,40),PSET:LINE-(185,50),PS
ET:LINE-(85,50),PSET:LINE-(90,40
),PSET
2550 FOR S=90 TO 205 STEP 25:DRA
W"C2BM"+STR$(S)+",35;U15R15D15L1
5C5":NEXT S
2560 FOR S=70 TO 185 STEP 25:DRA
W"C2BM"+STR$(S)+",84;U10R15D10U1
0U15L15D15C5":NEXT S
2570 FOR S=75 TO 175 STEP 25:PAI
NT(S,60),3,2:NEXT S
2580 FOR S=95 TO 210 STEP 25:PAI
NT(S,25),3,2:NEXT S
2590 LINE(60,65)-(60,84),PSET:LI
NE(210,35)-(210,50),PSET:LINE(19
0,65)-(190,84),PSET
2600 RETURN
2610 GOSUB3190:POKE491,10:LOADM"
SCRN1":EXEC32211:POKE65495,0:RET
URN
2620 LINE(0,0)-(255,84),PSET,B:L
INE(60,0)-(60,50),PSET:LINE-(255
,50),PSET:LINE(60,50)-(0,84),PSE
T
2630 POKE178,132:PAINT(10,10),,5
:PAINT(70,10),,5:POKE178,1:PAINT
(60,75),,5:POKE178,0
2640 RETURN
2650 GOSUB3190:POKE491,10:LOADM"
SCRN2":EXEC32211:POKE65495,0:RET
URN
2660 GOSUB3190:POKE491,10:LOADM"
SCRN3":EXEC32211:POKE65495,0:RET
URN
2670 GOSUB3190:POKE491,10:LOADM"
SCRN4":EXEC32211:POKE65495,0:RET
URN
2680 GOSUB3190:POKE491,10:LOADM"
SCRN5/BIN":EXEC32211:POKE65495,0
:IF L=19ANDLO(8)=19THENPUT(190,8
3)-(170,68),Z9,PSET:PAINT(178,70
),3,5:PAINT(178,80),3,5:PAINT(18
9,69),3,5:RETURN:ELSERETURN
2690 GOSUB3190:POKE491,10:LOADM"
SCRN7":EXEC32211:POKE65495,0:RET
URN
2700 GOSUB3190:POKE491,10:LOADM"
SCRN6":EXEC32211:POKE65495,0:IF
LO(10)=22 THEN PUT(85,82)-(60,57
),Z5,PSET:RETURNELSERETURN
2710 GOSUB3190:POKE491,10:LOADM"
SCRN8":EXEC32211:POKE65495,0:RET
URN
2720 GOSUB3190:POKE491,10:LOADM"
SCRN9":EXEC32211:POKE65495,0:RET
URN
2730 GOSUB3190:POKE491,10:LOADM"
SCRN10":EXEC32211:POKE65495,0:IF
 LO(15)=25THEN PUT(100,41)-(111,
54),Z8,PSET:RETURNELSEIFLO(12)=2
5THENPUT,(219,60)-(235,80),Z6,PSE
T:RETURNELSERETURN
2740 GOSUB3190:POKE491,10:LOADM"
SCRN11":EXEC32211:POKE65495,0:IF
LO(14)=26THENPUT(50,50)-(65,70),
Z7,PSET:RETURNELSERETURN
2750 GOSUB3190:POKE491,10:LOADM"
SCRN12":EXEC32211:POKE65495,0:RE
TURN
2760 GOSUB3190:POKE491,10:LOADM"
SCRN13":EXEC32211:POKE65495,0:RE
TURN
2770 GOSUB3190:POKE491,10:LOADM"
SCRN14":EXEC32211:IFLO(18)=30 TH
EN PUT(198,30)-(254,76),X1,PSET:
POKE65495,0:RETURNELSEPOKE65495,
0: RETURN
2780 PCLS:CLS:PRINT"YOU HAVE LEF
T THE FORTRESS BUT  THERE WERE G
UARDS WALKING       OUTSIDE AND
UNFORTUNATELY THEY  HAVE SEEN YO
U. IN DESPERATION   YOU QUICKLY
CONTACT H.Q. ON     YOUR WATCH C
.B. SET TO THE      DISTRESS FRE
QUENCY. ALL YOU GET IS "
2790 PRINT"A VOICE  SAYING YOU H
AVE FAILED AND MUST BE TERMINATE
D RATHER    THAN CAUGHT. A CYANID
E FILLED    TOOTH IMPLANTED IN Y
OUR MOUTH   IS ELECTRONICALLY ACT
IVATED AND YOU DIE PAINFULLY.
            YOUR MISSION IS AT AN
 END.....":GOTO3200
2800 PCLS:CLS:PRINT"THE EXPLOSIV
ES WERE HIGHLY      UNSTABLE AND
 EXPLODED BLOWING   MOST OF YOU
```

```
RIGHT BACK TO THE   FOREST. YOU
SHOULD HAVE FIGURED OUT HOW TO C
ONTAIN THEM         CORRECTLY. I
 WOULD SAY THIS     MEANS YOU CA
NNOT FULLFILL YOUR  ";
2810 PRINT"MISSION SO THIS IS
          THE END.....":GOTO3
200
2820 PCLS:CLS:PRINT"AS YOU BEND
DOWN TO EXAMINE THE BENCH YOU AC
CIDENTLY SET OFF A  LOW ALARM BE
AM THAT WAS         CROSSING THE
 FLOOR NEAR THE     BENCH. IMMED
IATELY TWO GUARDS   RUSH IN AND
BECAUSE YOU WERE SO CURIOUS ARE
KIND ENOUGH TO GIVE ";
2830 PRINT"YOU A DEMONSTRATION O
F HOW IT    WORKS. WITH YOU AS TH
E           DEMONSTRATION MODEL.
THEY SET    IT SO HIGH THAT IT KI
LLS YOU.    ACCIDENTLY OF COURSE.
 IT SEEMS   YOU HAVE FAILED AND Y
OUR         MISSION IS AT AN END.
...":GOTO3200
2840 PCLS:CLS:PRINT"ONCE OPEN YO
U FIND OUT WHAT WAS MAKING THE N
OISE. IT WAS A KGB  AGENT HOLDIN
G A DEVICE THAT     MAKES COCKRO
ACH NOISES. HE THEN GRABS ANOTHE
R DEVICE WHICH      MAKES A RAT-
TAT-TAT NOISE AND   FIRES IT AT
YOU. INCREDIBLY YOU ";
2850 PRINT"NO LONGER CARE ABOUT
YOUR        MISSION AS YOU ARE FL
OATING      ABOVE YOUR BODY. I DO
 BELIEVE    YOU ARE DEAD.
            YOU HAVE FAILED YOUR
MISSION.    THE END....":GOTO3200
2860 PCLS:CLS:PRINT"YOU WERE CAR
RYING TOO MUCH AND  HAVE TRIGGER
ED OFF A WEIGHT     SENSITIVE ST
EP CONNECTED TO A   SERIES OF GU
NS IN THE WALLS     WHICH HAVE N
OW COMMENCED        SHOOTING. UN
ABLE TO AVOID THE   FIRE IN THE
CONFINED SPACE THEY ";
2870 PRINT"HAVE TAKEN YOUR LIFE.
YOU HAVE    FAILED YOUR MISSION.
            THE END.....":GOTO320
0

2880 REM
2890 DIM Z1(20),Z2(15),Z3(8),Z4(
10),Z5(20),Z6(15),Z7(10),Z8(5),Z
9(8)
2900 PMODE4,1:PCLS5:PMODE3,1
2910 DRAW"C5BM200,9;F6R6U4R40U4L
40U4L6G6":PAINT(210,11),2,5
2920 GET(200,3)-(255,16),Z1,G
2930 DRAW"C5BM55,20;E2R26U2D2R7D
2L7D2U4;BM55,20;F5":LINE(60,25)-
(83,20),PSET:PAINT(63,23),3,5
2940 GET(55,15)-(92,30),Z2,G
2950 PCLS0:DRAW"BM100,15;R2D2L2U
2D4R2D2L2U2D4D2R4D6L4U6"
2960 GET(100,15)-(117,31),Z3,G
2970 DRAW"BM220,15;D4H4D4H4D4H4D
4H4D4H4D4":CIRCLE(210,25),6:DRAW
"BM210,25;D2"
2980 GET(222,15)-(200,30),Z4,G
2990 PCLS0:DRAW"BM170,10;E3R20G3
L20D20R20U20D20E3U20G3":LINE(175
,15)-(185,15),PSET:LINE(175,17)-
(185,17),PSET:LINE(175,19)-(185,
19),PSET
3000 GET(170,30)-(195,5),Z5,G
3010 PCLS0:FORX=170 TO 185 STEP4
:DRAW"BM"+STR$(X)+",10;D20R4U20L
4":NEXTX:LINE(170,15)-(185,18),P
SET,BF:LINE(170,25)-(185,22),PSE
T,BF3020 GET(170,10)-(186,30),Z6
,G
3030 PCLS0:LINE(170,10)-(185,30)
,PSET,B:LINE(175,15)-(180,25),PS
ET,BF
3040'GET(170,10)-(185,30),Z7,G
3050 PCLS0:LINE(230,15)-(235,20)
,PSET,B:LINE(227,20)-(238,28),PS
ET,BF
3060 GET(227,15)-(238,28),Z8,G
3070 PCLS0:DRAW"BM60,10;D10U10L3
U5L1D5R1L1L10D2L5D1R5D2R10D5R5L5
U3L2U2"
3080 GET(60,5)-(40,20),Z9,G
3090 RETURN
3100 GOSUB3190:POKE491,10:LOADM"
SCRN15":EXEC32211:POKE65495,0:FO
RX=1TO1000:NEXTX
3110 LINE(130,40)-(150,80),PSET,
BF
3120 K=140:A=140:FORX=1 TO30:K=K
+1:A=A-1:Y=40:CIRCLE(K,Y),20,1:C
IRCLE(A,Y),20,5:NEXT X:EXEC44539
:PCLS:CLS
3130 IFL<>1THEN3170ELSEPRINT"CON
GATULATIONS YOU HAVE        COM
PLETED THE TASK SET DOWN FOR YOU
. YOU ARE NOW A NATIONAL    HER
O AND HAVE BEEN PRMOTED. YOU HAV
E FINISHED MISSION          ";
3140 PRINT"INFILTRATE. WELL DONE
 BUT WHATS YOUR NEXT MISSION?":G
OTO 3200
3150 PCLS:CLS:PRINT"YOU SET THE
BOMB TO THE         INCORRECT TI
ME AND THEY HAVE    DEFUSED IT.
YOU SHOULD HAVE     FOUND OUT TH
E CORRECT TIME      SETTING. YOU
 HAVE FAILED YOUR   MISSION. THE
 ONLY THING YOU CAN DO IS RUN AW
AY TO STH AMERICA   ";
3160 PRINT"AND HOPE YOU ARE NEVE
R TRACKED   DOWN OR YOU WILL BE K
ILLED.      THE END.....":GOTO320
0
3170 PCLS:CLS:PRINT"YOU DIDN'T G
ET BACK TO THE      FOREST IN TI
ME I'M SORRY AND    HAVE BEEN CA
UGHT IN THE         EXPLOSION AN
D HAVE DIED. YOU    COMPLETED YO
UR MISSION BUT DID  NOT SURVIVE.
NEXT TIME DISCOVER  WHAT TIME TO
 SET THE BOMB TOO.  ";
3180 PRINT"THE END.....":GOTO320
0

3190 POKE 360,PEEK(492):POKE361,
PEEK(493):POKE363,PEEK(494):POKE
364,PEEK(495):POKE32211,48:POKE6
5494,0:RETURN
3200 EXEC 44539:PCLS:CLS:PRINT"A
NOTHER GAME Y/N?"
3210 FG$=" ":FG$=INKEY$:IF FG$="
Y" THEN SCREEN0,0:GOTO 10 ELSE I
F FG$="N" THEN PRINT"GOODBYE.":S
CREEN0,0:END
3220 GOTO 3210
3230 IFV$="GET"THEN 890
3240 IFV$="DROP"THEN 890
3250 IFV$="SEAR"THEN890
3260 GOTO1010
3270 PCLS:CLS:LINE(0,0)-(255,191
),PSET,B:PRINT@96+12,"MISSION";:
PRINT@128+10,"INFILTRATE";:PRINT
@160+14,"BY";:PRINT@192+10,"SCOT
T HARVEY"
3280 DRAW"BM100,156;U30R5D30L5R1
0U30R5D30L5R10U30R5D30R5U30R5D30
R5U30"
3290 LINE(100,156)-(146,126),PSE
T,B
3300 LINE(100,131)-(146,136),PSE
T,BF:LINE(100,151)-(146,146),PSE
T,BF
3310 CIRCLE(136,138),9,3
3320 LINE(100,156)-(109,126),PRE
SET,BF
3330 PAINT(136,138),0,3
3340 DRAW"BM145,131;E5;U3E3U3E3H
3L3G3H3G3H3G3H3G3H3G3H3G3D3F3D3G
3"
3350 DIMX1(70):GET(100,110)-(156
,156),X1,G
3360 PRINT@449,"DO YOU NEED INST
RUCTIONS Y/N"
3370 FOR S=0 TO 3
3380 DRAW"A"+STR$(S)+"BM136,138;
U4":FORI=1TO100:NEXTI:PAINT(136,
138),0,3
3390 A$=INKEY$:IFA$="Y" THENDRAW
"A0": GOTO3430
3400 IF A$="N" THEN DRAW"A0":GOT
O50
3410 NEXT S
3420 GOTO 3370
3430 PCLS:CLS:A$="CODE: ONE TRAN
SMISSION":PRINT@0,"";:GOSUB3570
3440 A$="YOU MUST INFILTRATE THE
 KARAN ":PRINT@32,"";:GOSUB3570
3450 A$="BASE WHICH IS NOW UNDER
 THE":PRINT@64,"";:GOSUB3570
3460 A$="CONTROL OF THE KGB. ONC
E INSIDE":PRINT@96,"";:GOSUB3570
3470 A$="YOU MUST FIND THE THREE
 PARTS":PRINT@128,"";:GOSUB3570
3480 A$="OF A TIME BOMB HIDDEN W
ITHIN ":PRINT@160,"";:GOSUB3570
3490 A$="THE FORTRESS BY PREVIOU
S AGENTS.":PRINT@192,"";:GOSUB35
70
3500 A$="YOU MUST THEN PLANT THE
 BOMB IN":PRINT@224,"";:GOSUB357
0
3510 A$="THE LABORATORY WHERE TH
E KGB ":PRINT@256,"";:GOSUB3570:
A$="ARE WORKING ON A NEW SUPER":
PRINT@288,"";:GOSUB3570:A$="WEAP
ON THAT COULD CAUSE THE":PRINT@3
20,"";:GOSUB3570
3520 A$="DESTRUCTION OF MILLIONS
 OF ":PRINT@352,"";:GOSUB3570:A$
="LIVES. DETONATE THE BOMB AND":
PRINT@384,"";:GOSUB3570:A$="GET
OUT.":PRINT@416,"";:GOSUB3570
3530 A$="OPERATION: INFILTRATE":
PRINT@448,"";:GOSUB3570:A$="SCRE
EN WILL NOW SELF DESTRUCT.":PRIN
T@480,"";:GOSUB3570
3540 FORI=1TO200:NEXTI:FOR X=0 T
O 100:A=RND(255):B=RND(191):LINE
(128,96)-(A,B),PSET:NEXT X
3550 GOTO50
3560 GOTO3560
3570 FORX=1TOLEN(A$)
3580 POKE140,150
3590 B$=MID$(A$,X,1):PRINTB$;:EX
EC43359:FORZ=1TO RND(50):NEXTZ:N
EXTX
3600 RETURN
```

■

COME TO at Bundee

Conf '87 this year is to be held at the Uniting Church's campsite in Bundeena NSW.

This is a particularly pretty area of Sydney, situated on the northern tip of the Royal National Park, in Port Hacking.

The water views are fabulous, and the bushwalks are amoungst the best in Australia.

Not that you'll have anytime during conference for these things, because as usual, the conference will be jam packed with all sorts of things to see and do!

The big news this year will obviously be the growing use of OS-9 Level 2 on the CoCo 3's; and Conf '87 will be the definitive place to see this excellent system.

By that time initial users will have had time to sort the system out and create some really interesting stuff.

But it is not just OS-9 that is of interest this year.

With the release of the new T1000 EX and SX, interest in these machines has never been higher. We'll have a number of these computers at the conference, as well as their big brothers, the T3000 series, which we'll be putting through their paces.

We've had continuing interest in some of the more diverse subjects covered in the magazine at past conferences, so again this year we'll have tutorials on hardware mods and on Forth.

There'll be Basic Basic and Advanced Basic courses, and an Assembly Language tutorial as well.

Other computers will be discussed, principally the 68000 series of computers, and of course, we'll be showing Goldlink 642 on Viatel - and Videotex in general.

Conference is a place to meet old friends, to meet the people behind the names in the magazine, to learn a lot of new information, to see the latest Tandy equipment.

We hope you'll come. We're sure you'll be glad you did. But please hurry your booking, because accommodation (which is not obligatory) and places at the conference, are both limited by the size of the centre.

The cost is increased over previous years due entirely to the fact that we are doing it in Sydney which is a good deal more expensive than the Gold Coast!

On the other hand, many of you will save by not having the additional travelling expenses associated with getting to the Gold Coast.

We aim to make the conference a family affair, and the location is a good one for people with families who are less interested in computers, but who would still like to be with dad or mum for the weekend.

The family can take a ferry trip, go for bush walks, or just laze on the beach, whilst you do your thing at the conference.

CONF '87

na N.S.W.

CONF '87

Rates

Accommodated (1) $87.00
Family of 2, + $68.00 = $155.00
Additional family members $52.00 ea

Includes supper Friday evening, breakfast lunch and dinner on Saturday and breakfast and lunch on Sunday plus all accommodation.

Non Accommodated Rates

	One day	Two Days
One person	$40.00	$58.00
Sat Evening Meal	$12.00	$12.00
	$52.00	$70.00

Additional family $31.50 $45.50 /person
Includes morning / afternoon tea and lunch.

$20.00 deposit required with booking; final payment to be made by 15th July 1987.

LOCATION:-
Uniting Church's campsite
Bundeena NSW

DATE:- 8th & 9th August, 1987

REGISTER NOW!!
We can only accept a limited number of people this year. DON'T MISS OUT! on a top weekend of FUN, FRIENDSHIP and LEARNING.

Name: .....
Address: .....
Phone: .....
No. People attending: .....

SPEAK UP!:- Now is your chance to suggest your ideas for any tutorials we may not have mentioned. (participants only).

Tutorials likely to attend: .....
.....

Please find enclosed:
chq/money order/bankcard/visa/mastercard
Card No. .....
Signature: .....

Getting it all together.

COCO MERGE

32K ECB
TAPE UTILITY

by John L. Nicolettos

COCO MERGE IS A small utility to help you merge two or more basic programs together, whether they be on tape or to be typed in.

The program is fully documented and when you've finished reading the instructions, CoCo Merge is ready and rearing to go!

The Listing:

```
10 '  ***************************
20 '  *       CO CO MERGE       *
30 '  *           BY            *
40 '  *   JOHN L. NICOLETTOS    *
50 '  *      APRIL 1, 1983      *
60 '  *  ALL  RIGHTS RESERVED   *
70 '  ***************************
80 CLS
90 PRINT"           CO CO MERGE":
PRINT@32,STRING$(32,131);
100 PRINT"   THIS PROGRAM WILL L
OAD,":PRINT"EXECUTE, AND PROTECT
 A MACHINE"
110 PRINT"LANGUAGE UTILITY WHICH
 WILL":PRINT"ALLOW YOU TO MERGE
TWO OR MORE"
120 PRINT"CASSETTE PROGRAMS.  TO
 USE":PRINT"SIMPLY TYPE OR CLOAD
 YOUR FIRST"
130 PRINT"PROGRAM.  BEFORE ENTER
ING THE':PRINT"NEXT PROGRAM PRES
S THE [SHIFT]"
140 PRINT"AND CLEAR KEYS.  TYPE
OR CLOAD":PRINT"THE NEXT PROGRAM
.  WHEN DONE"
150 PRINT"PRESS THE [SHIFT] AND
CLEAR KEYS";:PRINT"ONCE AGAIN TO
 COMPLETE MERGE."
160 PRINT@484,"PRESS ENTER TO CO
NTINUE";:LINEINPUT ZZ$
170 ED=PEEK(39)*256+PEEK(40)
180 ST=ED- 118
190 FOR X=ST TO ED
200 READ D:POKE X,D:SUM=SUM+D
210 NEXT X
220 IF SUM <> 12939THEN CLS:PRIN
T@263,"!!!DATA ERROR!!!":END
230 EXEC ST
240 CLEAR 200,ST
250 DATA 48, 141, 0, 43, 191, 1,
 104, 111
260 DATA 141, 0, 3, 57, 0, 255,
16, 77
270 DATA 69, 82, 71, 69, 32, 73,
 78, 73
280 DATA 84, 73, 65, 84, 69, 68,
 13, 77
290 DATA 69, 82, 71, 69, 32, 67,
 79, 77
300 DATA 80, 76, 69, 84, 69, 68,
 13, 129
310 DATA 92, 38, 54, 109, 141, 2
55, 215, 38
320 DATA 27, 158, 25, 175, 141,
255, 205, 158
330 DATA 27, 48, 30, 159, 25, 49
, 140, 199
340 DATA 99, 141, 255, 194, 141,
 30, 142, 171
350 DATA 239, 126, 172, 121, 174
, 141, 255, 180
360 DATA 159, 25, 49, 140, 194,
111, 141, 255
370 DATA 173, 141, 9, 142, 171,
239, 126, 172
380 DATA 121, 126, 130, 115, 198
, 16, 166, 160
390 DATA 189, 162, 130, 90, 38,
248, 57
400 CLS
410 PRINT@128,"       CO"
420 PRINT:PRINT"            CO"
430 PRINT:PRINT"               MERG
E"
440 PRINT:PRINT"                  L
OADED"
450 END
```

■

Timer - Australian Context

The use of the TIMER function can give a result in seconds, making it suitable to time operator response. Australia uses 50Hz Mains frequency, whereas in the U.S.A., 60Hz is used.

BASIC programs written for the American market use the (TIMER/60) function to approximate seconds. If this formula is used in Australian computers, slower times than normal will result.

You must convert the formula to (TIMER/50) to suit.

Kevin Gowan

COCO CON

32K ECB
UTILITY

COCO LIST CONTROL is a small utility to help list your program on the screen using your right joystick.

All you do is RUN the program, and when you're finished reading the instructions, CoCo List Control is engaged and ready.

To list your progam in the normal speed, push the joystick *towards* the fire button. To slow the listing down, pull the joystick button *away* from the fire button.

When your listing has finished, return the joystick *towards* the fire button.

The Listing:

```
10 '  ***************************
20 '  *                         *
30 '  *   CO CO LIST CONTROL    *
40 '  *                         *
50 '  *           BY            *
60 '  *                         *
70 '  *   JOHN L. NICOLETTOS    *
80 '  *                         *
90 '  *  ALL  RIGHTS RESERVED   *
100 ' *                         *
110 ' *     MARCH 1,  1983      *
120 ' *                         *
130 ' ***************************
140 '
150 '
160 '
170 CLS
180 PRINT"        CO CO LIST CONT
ROL":PRINT@32,STRING$(32,131);
190 PRINT"THIS PROGRAM WILL LOAD
 A MACHINE";:PRINT"LANGUAGE PROG
RAM INIO UPPER RAM"
200 PRINT"MEMORY.  IT WILL AUTOM
ATICALLY":PRINT"EXECUTE AND PROT
ECT THE MACHINE"
210 PRINT"LANGUAGE PROGRAM.":PRI
NT
220 PRINT"YOU MUST HAVE THE RIGH
T JOYSTICK";:PRINT"CONNECTED TO
THE COMPUTER.  ALSO";
230 PRINT"YOU MUST HAVE THE JOYS
TICK IN":PRINT"THE TOP VERTICAL
POSITION"
240 PRINT"(NEAREST TO THE BUTTON
)."
250 PRINT@484,"PRESS ENTER TO CO
NTINUE";:LINEINPUT ZZ$
260 ED=PEEK(39)*256+PEEK(40)
```

LIST TROL

by John L. Nicolettos

```
270 ST=ED- 45
280 FOR X=ST TO ED
290 READ D:POKE X,D:SUM=SUM+D
300 NEXT X
310 IF SUM <> 4342THEN CLS:PRINT
@263,"!!!DATA ERROR!!!":END
320 EXEC ST
330 CLEAR 200,ST
340 DATA 48, 141, 0, 4, 191, 1,
104, 57
350 DATA 129, 13, 38, 31, 52, 11
8, 189, 169
360 DATA 222, 125, 1, 91, 39, 19
, 182, 1
370 DATA 91, 129, 63, 39, 241, 1
42, 10, 255
380 DATA 48, 31, 38, 252, 122, 1
, 91, 38
390 DATA 244, 53, 118, 126, 130,
 115
400 CLS
410 PRINT"        CO CO LIST CONT
ROL":PRINT@32,STRING$(32,131);
420 PRINT"PROGRAM IS LOADED.  TO
 USE THE"
430 PRINT"LIST CONTROL SIMPLY MO
VE YOUR":PRINT"RIGHT JOYSTICK AW
AY FROM THE"
440 PRINT"BUTTON.  THE LISTING S
PEED WILL":PRINT"DECREASE AS YOU
 MOVE THE"
450 PRINT"JOYSTICK.  WHEN YOU RE
ACH THE":PRINT"JOYSTICK'S LIMIT
THE SCROLLING"
460 PRINT"WILL STOP.  TO CONTINU
E SIMPLY":PRINT"MOVE YOUR JOYSTI
CK TOWARDS THE"
470 PRINT"BUTTON.  REMEMBER TO K
EEP THE":PRINT"JOYSTICK AT MAX S
PEED WHEN NOT"
480 PRINT"USING THE LIST CONTROL
 "
```

For those whose screen dump programs only prints halfwidth just put your DMP-110 into elongation mode!

Send to the printer:

```
PRINT#-2,CHR$(27);CHR$(14)
```

To turn it off send:

```
PRINT#-2,CHR$(27);CHR$(15)
```

PI-COS

any MiCo, any CoCo
UTILITY

by John Day

JOHN DAY RECENTLY supplied us with the Pi-C.O.S. (Pi - Cassette Operating System). We have run it through on our machines and must say we are thoroughly pleased with it's perfomance, handling and easy-to-understand commands as well as user - friendlyness.

We recommend this to anyone who is serious about cassette-only operations as this utility is a great help and the next best thing to a disk drive.

The Listing:

```
0 '**********PI-COS*************
1 '
2 '     CASSETTE OPERATING
3 '          SYSTEM
4 '
5 '    (C)1972 N. CHILADAS
6 '
7 '
8 '*****************************
9 CLS:DIMQW(12)
10 GOSUB290
20 CLS:PRINT@32," WHAT TYPE OF S
YSTEM:-"
30 PRINT" 1 - MC-10",," 2 - 16K
COLOR BASIC"," 3 - 16K ECB",," 4
 - 64K ECB",," 5 - DOS (ANY)"
40 A$=INKEY$:IFA$=""THEN40
50 S=VAL(A$):IF S<1 ORS>5 THEN S
OUND 1,1:GOTO20
60 CLS
70 ON S GOTO 140,170,200,230,260
80 FOR L=0TO511:POKEL+X,32:NEXT
90 FOR L=1TO16:READ D:POKEL+X+96
,D:NEXT
95 FOR Q=200TO245:SOUNDQ,1:NEXT
100 FOR L=1TO26:READ D:POKE228+L
+X,D:NEXT
110 GOTO110
120 DATA23,8,1,20,39,19,32,1,32,
16,9,45,3,15,19,63
130 DATA1,2,15,21,20,32,1,32,4,1
5,12,12,1,18,32,9,14,32,19,21,18
,6,5,18,19,33
140 X=16384:XX=X*(256/X)*256:Y=I
NT(Y/X)
150 XY=INT(X-QW(5))*X/256
160 GOTO80
170 X=1024:XX=X*(256/X)*256:Y=IN
T(Y/X)
180 XY=INT(X-QW(2)/3.14159)*256
190 GOTO80
200 X=1024:XX=X*(256/X)*256:Y=IN
T(Y/X)
210 XY=INT(X-QW(2)/3.14159)*X/25
6*256
220 GOTO80
230 X=1024:XX=X*(256/X)*256:Y=IN
T(Y/X)
240 XY=INT(X-QW(12)/3.14159)*X/2
56
250 GOTO80
260 X=1024:XX=X*(256/X)*256:Y=IN
T(Y/X)
270 XY=INT(X-(RS-B)*256)/256
280 GOTO80
290 CLS2
300 PRINT@32,"    ***** PI-C.O.S
 *****
310 PRINT:PRINT" THE "CHR$(34)"P
I-COS"CHR$(34)" CASSETTE OPERATI
NG SYSTEM IS NOW READY TO ENGAGE
."
320 PRINT:PRINT" PLEASE ENSURE T
HAT YOUR TAPE    RECORDER IS CON
NECTED TO THE    COMPUTER; THAT
POWER TO  THE    RECORDER IS OFF
; AND THAT NO     CASSETTE IS IN
THE RECORDER."
330 PRINT" IF THESE INSTRUCTIONS
 AREN'T    CARRIED OUT, YOU MAY
NOT  BE    PLEASED WITH THE RESU
LTS."
340 PRINT@448,"    PRESS A KEY T
O CONTINUE"
350 IF INKEY$=""THEN350
360 RETURN
370 END
380 SAVE"180C:3":END'8
```

■

CoCo 2 or CoCo 3?

If you're creating software to run on either the CoCo 2 or the CoCo 3 and want to take advantage of both computers using the one piece of software, then use this PEEK.

It checks **whether** the program is running in a CoCo 1, 2 or 3. 'PEEK' is any number between 65456 and 65471; if you get anything but a '126' then your program is running under a CoCo 3.

Steve Youngberry

Here's a tricky one.

PYRAMINX

32K ECB
GAME

by Bob Delbourgo

REMEMBER RUBIKS CUBE and how everyone in the beginning couldn't get it right? Well you're probably bored silly trying to get the colours right on a cube. Ever thought about trying to get the colours right with a pyramid?

That's the aim of this game!
You will get a plan view of this tetrahedral puzzle as well as a view of it's hidden bottom face.

You may rotate it about all four verticies including upper, right, left and back, as will be marked on the diagram. You can rotate a single corner or a mini-tetrahedron.

The instructions are in the program, so type it in and try out your thinking abilities!

The Listing:

```
0 GOTO10
1 '********** PYRAMINX *********
   ****** BOB DELBOURGO ********
3 SAVE"205C:3":END'1
10 PMODE4,1:PCLS1:SCREEN1,1
20 LINE(112,36)-(128,12),PRESET:
LINE-(144,36),PRESET:LINE-(112,3
6),PRESET:LINE-(128,28),PRESET:L
INE-(128,12),PRESET:LINE(128,28)
-(144,36),PRESET
30 DRAW"C0BM48,68;U12E4F4D12U12E
4F4D12BR20L8H8R16L16E8R8BR20L8G8
R16L16D8ER20U8R16L16E8R8BR20L8G8
R16L16F8R8BR4U16R12F4G4L12R8F8BR
12U16L4G4BR16H4L4BR12BU4U4BD8BR1
6L8G8R16G8L8"
40 DRAW"BM48,100;U16R12F4G4L12D8
BR28U8H8F8E8BR4BD16U16R12F4G4L12
R8F8BR4U8E8F8L16R16D8BR4U12E4F4D
12U12E4F4D12BR8R8L4U16G4E4F4BD12
BR8U16F16U16BR4F16BU16G16"
50 DRAW"BM68,132;U16R12F4G4L12D8
BR4BE16D12F4R8E4U12BF4E4R12G16R1
2E4BU8BR4E4R12G16R12E4BF4R16L16U
16BR36L8G8R16L16F8R8"
60 DRAW"BM12,164;U8L2G2BR8H2L2BR
8D8BR4U8F8U8BR8G4F4E4H4BR20BD4G4
L4E8L8F8BR12U8R4F4G4L4BR12U4E4F4
L8R8D4BR4U8F8U8BR4D8BR12L4H4R8L8
E4R4BR4D8R8"
70 DRAW"BM144,164;U8R4F4G4L4BR20
L4H4R8L8E4R4BR4D8R8BR4U8R4F4L8R8
G4L4BR16H4E4F4G4BE8D6F2R4E2U6BR4
BD8U8R6F2G2L6R4F4BR8BU4R4D4L4H4E
4R4BR8G4F4E4H4"
80 FORI=1TO2:PLAY"V30O2L12AA#AGL
6F#F#L12AGF#D#L3D":NEXTI
100 CLS3:GOSUB1600:PRINT@64,"YOU
 WILL GET A PLAN VIEW OF THISTET
RAHEDRAL PUZZLE AS WELL AS A VIE
W OF ITS HIDDEN BOTTOM FACE."
110 PRINT@192,"YOU MAY ROTATE IT
 ABOUT ALL FOURVERTICES, UPPER,
RIGHT, LEFT ANDBACK, AS WILL BE
MARKED ON THE  DIAGRAM. YOU CAN
ROTATE A SINGLECORNER OR A MINI-
TETRAHEDRON."
120 PRINT@384,"THE CODES FOR THE
 VARIOUS TURNS ARE AS FOLLOWS:":
GOSUB1610
140 CLS4:PRINT"FOR SINGLE CORNER
 TURNS PRESS :":PRINT@64,"<u> TH
EN <1> - CLOCKWISE UPPER, <u> TH

EN <2> - ANTICLOCK.UPPER, <r> TH
EN <1> - CLOCKWISE RIGHT, <r> TH
EN <2> - ANTICLOCK.RIGHT,"
150 PRINT"<l> THEN <1> - CLOCKWI
SE LEFT,  <l> THEN <2> - ANTICLO
CK.LEFT,  <b> THEN <1> - CLOCKWI
SE BACK,  <b> THEN <2> - ANTICLO
CK.BACK."
160 PRINT@352,"SIMILARLY FOR MIN
ITETRAHEDRA :  <U> THEN <1> - CL
OCKWISE UPPER  <U> THEN <2> - AN
TICLOCK.UPPER, AND SO ON..":GOSU
B1610
180 PMODE1,1:PCLS:SCREEN1,0
190 LINE(11,141)-(125,141),PSET:
LINE-(68,41),PSET:LINE-(11,141),
PSET:LINE-(68,106),PSET:LINE-(68
,41),PSET:LINE(68,106)-(125,141)
,PSET
191 LINE(137,141)-(251,141),PSET
:LINE-(194,42),PSET:LINE-(137,14
1),PSET
192 LINE(175,75)-(213,75),PSET:L
INE-(175,141),PSET:LINE-(156,108
),PSET:LINE-(232,108),PSET:LINE-
(213,141),PSET:LINE-(175,75),PSE
T
193 LINE(68,86)-(48,119),PSET:LI
NE-(48,75),PSET:LINE-(68,64),PSE
T:LINE-(29,130),PSET:LINE-(29,10
8),PSET:LINE-(68,86),PSET
194 LINE(68,86)-(87,119),PSET:LI
NE-(87,75),PSET:LINE-(68,64),PSE
T:LINE-(106,130),PSET:LINE-(106,
108),PSET:LINE-(68,86),PSET
195 LINE(48,119)-(87,119),PSET:L
INE-(48,141),PSET:LINE-(29,130),
PSET:LINE-(106,130),PSET:LINE-(8
7,141),PSET:LINE-(48,119),PSET
196 DRAW"C3BM64,36;U12R4F4D2L8R8
D2G4L4":DRAW"BM8,144;D12R8":DRAW
"BM120,156;U12R4F4G4L4R4F4":DRAW
"BM120,100;D12R8U12"
197 DRAW"BM44,164;U12R4F4G4L4D4B
E12D12R8BR4U8E4F4D2L8R8D6BR4U12F
4D4F4U12":DRAW"BM44,172;D8F4E4U8
BR8G2E2F2H2D12L2R4BR14L4H4U2R8L8
U2E4R4BR4D8F4U8D8E4U8"
198 DRAW"BM164,164;U12R4F4D2L8R8
D2G4L4BR16H4U4E4F4D4G4BR12U12G4E
4F4BR4E4F4H4D12BR12H4U4E4F4D4G4B
R8U12F4D4U4E4D12":DRAW"BM184,172
;L4G4D2R8L8D6BR12U8E4F4D2L8R8D6B
R12L4H4U4E4R4BR12L4G4D2R8L8D2F4R
4"
199 T$="U12L2G4BR12H4L2BR12D8F4R
4E4U8BR4D12U12R8F4G4L8R8F4BR4U12
F12U12BF12U2BU2E4H4G4":T1$="E4D1
2":T2$="E4F4G8R8"
200 DIMX(36),Y(36),C(36),P(4),Q(
4)
201 DATA63,100,73,100,68,113,53,
100,83,100,68,126,43,113,73,81,8
7,124,63,81,92,113,48,124
202 DATA33,113,83,81,87,135,53,8
1,102,113,48,133,23,124,72,57,10
6,135,62,57,112,124,29,136
203 DATA42,95,95,95,68,135,194,6
4,232,130,156,130,176,96,214,96,
194,130,194,92,214,126,176,126
204 FORI=1TO36:READX(I),Y(I):NEX
TI
205 FORI=1TO25STEP3:C(I)=2:C(I+1
)=3:C(I+2)=4:NEXTI:FORI=28TO36:C
(I)=1:NEXTI
206 PCOPY1TO3:PCOPY2TO4:PMODE1,1
:GOSUB1000:PLAY"P1"
207 P(1)=132:Q(1)=104:P(2)=132:Q
(2)=148:P(3)=20:Q(3)=148:P(4)=76
:Q(4)=28
210 CLS2:GOSUB1600
211 PRINT@64,"";:INPUT"ENTER THE
 NUMBER OF RANDOM TURNSYOU WANT
THE COMPUTER TO MAKE  (MAXIMUM
OF 36)";N:N=INT(N):IFN<0THEN211
212 IFN>36THEN211
213 IFN=0THEN245
```

```
214 PRINT@256,"NOW UNSCRAMBLE TH
E PYRAMINX IN":PRINT255-N;" TURN
S, slowly please...":PRINT"PRESS
 <C> IF YOU WISH TO GIVE UPAND W
ANT THE COMPUTER TO SOLVE  IT FO
R YOU!"
215 PRINT@416,"PRESS <Q> TO RESI
GN OR RESTART."
216 E=RND(4):F=RND(4):IFE=F THEN
216
217 G=RND(4):IFG=E THEN217
218 IFG=F THEN217
219 H=10-G-F-E:FORI=1TO25STEP3:C
(I)=E:C(I+1)=F:C(I+2)=G:NEXTI:FO
RI=28TO36:C(I)=H:NEXTI:GOSUB1000
220 R=RND(-TIMER):U=0:R$="":FORJ
=1TON:R=RND(16)-1:R$=R$+HEX$(R):
SOUNDRND(255),1:PRINT@192,"move
";J
221 ON R+1 GOTO500,520,540,560,5
80,600,620,640,660,680,700,720,7
40,760,780,800
222 NEXTJ
223 FORT=1TOLEN(R$):V1=0:V=VAL("
&H"+MID$(R$,T,1)):IFV>=0ANDV<4TH
ENV1=V+4
224 IFV>3ANDV<8THENV1=V-4
225 IFV>7ANDV<12THENV1=V+4
226 IFV>11THENV1=V-4
227 V$=HEX$(V1)
240 MID$(R$,T,1)=RIGHT$(V$,1):NE
XTT:GOSUB1610
245 SCREEN1,0:R=0:U=1
250 POKE282,0:I$=INKEY$:DRAW"C3B
M96,20;XT$;"
251 IFI$="c"THENPOKE282,255:GOTO
1500
252 IFI$="q"THENPOKE282,255:RUN1
80

253 IFI$="U"THENR=0:GOTO270
254 IFI$="R"THENR=1:GOTO270
255 IFI$="L"THENR=2:GOTO270
256 IFI$="B"THENR=3:GOTO270
257 IFI$="u"THENR=8:GOTO270
258 IFI$="r"THENR=9:GOTO270
259 IFI$="l"THENR=10:GOTO270
260 IFI$="b"THENR=11:GOTO270
261 DRAW"C2BM96,20;XT$;":GOTO250
270 SOUND100,1:POKE282,255
271 I$=INKEY$:DRAW"C3BM96,20;XT$
;":IFI$="1"THEN280
272 IFI$="2"THENR=R+4:GOTO280
273 DRAW"C2BM96,20;XT$;":GOTO271
280 SOUND200,1
281 IFR>=0ANDR<4THENV=R+4
282 IFR>3ANDR<8THENV=R-4
283 IFR>7ANDR<12THENV=R+4
284 IFR>11THENV=R-4
285 R$=R$+HEX$(V)
286 IFLEN(R$)=255THENSOUND250,5:
GOTO1500 ELSEON R+1 GOTO500,520,
540,560,580,600,620,640,660,680,
700,720,740,760,780,800
500 S=PPOINT(X(1),Y(1)):GOSUB101
0
501 GOSUB1620:GOTO810
520 S=PPOINT(X(21),Y(21)):GOSUB1
010
521 GOSUB1630:GOTO810
540 S=PPOINT(X(19),Y(19)):GOSUB1
010
541 GOSUB1640:GOTO810
560 S=PPOINT(X(20),Y(20)):GOSUB1
010
561 GOSUB1650:GOTO810
580 S=PPOINT(X(1),Y(1)):GOSUB101
0
581 GOSUB1660:GOTO810
600 S=PPOINT(X(21),Y(21)):GOSUB1
010
601 GOSUB1670:GOTO810
620 S=PPOINT(X(19),Y(19)):GOSUB1
010
621 GOSUB1680:GOTO810
640 S=PPOINT(X(20),Y(20)):GOSUB1
010
641 GOSUB1690:GOTO810
660 S=PPOINT(X(1),Y(1)):S1=PPOIN
T(X(7),Y(7)):S2=PPOINT(X(4),Y(4)
):S3=PPOINT(X(10),Y(10)):GOSUB10
10
661 GOSUB1620:C(7)=C(9):C(9)=C(8
):C(8)=S1:C(4)=C(6):C(6)=C(5):C(
5)=S2:C(10)=C(12):C(12)=C(11):C(
11)=S3:GOTO810
680 S=PPOINT(X(21),Y(21)):S1=PPO
INT(X(27),Y(27)):S2=PPOINT(X(15)
,Y(15)):S3=PPOINT(X(9),Y(9)):GOS
UB1010
681 GOSUB1630:C(27)=C(32):C(32)=
C(11):C(11)=S1:C(15)=C(35):C(35)
=C(17):C(17)=S2:C(9)=C(33):C(33)
=C(26):C(26)=S3:GOTO810
700 S=PPOINT(X(19),Y(19)):S1=PPO
INT(X(25),Y(25)):S2=PPOINT(X(13)
,Y(13)):S3=PPOINT(X(7),Y(7)):GOS
UB1010
702 GOSUB1640:C(25)=C(33):C(33)=
C(12):C(12)=S1:C(13)=C(36):C(36)
=C(18):C(18)=S2:C(7)=C(31):C(31)
=C(27):C(27)=S3:GOTO810
720 S=PPOINT(X(20),Y(20)):S1=PPO
INT(X(26),Y(26)):S2=PPOINT(X(14)
,Y(14)):S3=PPOINT(X(8),Y(8)):GOS
UB1010
721 GOSUB1650:C(26)=C(31):C(31)=
C(10):C(10)=S1:C(14)=C(34):C(34)
=C(16):C(16)=S2:C(8)=C(32):C(32)
=C(25):C(25)=S3:GOTO810
740 S=PPOINT(X(1),Y(1)):S1=PPOIN
T(X(7),Y(7)):S2=PPOINT(X(4),Y(4)
):S3=PPOINT(X(10),Y(10)):GOSUB10
10

741 GOSUB1660:C(7)=C(8):C(8)=C(9
):C(9)=S1:C(4)=C(5):C(5)=C(6):C(
6)=S2:C(10)=C(11):C(11)=C(12):C(
12)=S3:GOTO810
760 S=PPOINT(X(21),Y(21)):S1=PPO
INT(X(27),Y(27)):S2=PPOINT(X(15)
,Y(15)):S3=PPOINT(X(9),Y(9)):GOS
UB1010
761 GOSUB1670:C(27)=C(11):C(11)=
C(32):C(32)=S1:C(15)=C(17):C(17)
=C(35):C(35)=S2:C(9)=C(26):C(26)
=C(33):C(33)=S3:GOTO810
780 S=PPOINT(X(19),Y(19)):S1=PPO
INT(X(25),Y(25)):S2=PPOINT(X(13)
,Y(13)):S3=PPOINT(X(7),Y(7)):GOS
UB1010
781 GOSUB1680:C(25)=C(12):C(12)=
C(33):C(33)=S1:C(13)=C(18):C(18)
=C(36):C(36)=S2:C(7)=C(27):C(27)
=C(31):C(31)=S3:GOTO810
800 S=PPOINT(X(20),Y(20)):S1=PPO
INT(X(26),Y(26)):S2=PPOINT(X(14)
,Y(14)):S3=PPOINT(X(8),Y(8)):GOS
UB1010
801 GOSUB1690:C(26)=C(10):C(10)=
C(31):C(31)=S1:C(14)=C(16):C(16)
=C(34):C(34)=S2:C(8)=C(25):C(25)
=C(32):C(32)=S3:GOTO810
810 GOSUB1000:IFW=1THEN1502ELSEI
FU=0THEN222ELSE245
1000 FORI=1TO36:PAINT(X(I),Y(I))
,C(I),4:NEXTI:RETURN
1010 PMODE1,1:PCOPY3TO1:PCOPY4TO
2:RETURN
1500 W=1:FORD=LEN(R$)TO1STEP-1:R
=VAL("&H"+MID$(R$,D,1))
1501 ON R+1 GOTO500,520,540,560,
580,600,620,640,660,680,700,720,
740,760,780,800
1502 SOUNDD,1
1503 C=2*INT(R/8):K=1+R-4*INT(R/
4):L=1+INT(R/4)-C
1504 DRAW"BM"+STR$(P(K))+","+STR
$(Q(K))+";C"+STR$(C)+";XT"+STR$(
L)+"$;":PLAY"P1":DRAW"C1;XT"+STR
$(L)+"$;"
1535 NEXTD
1536 PLAY"V30O2L8D#L12D#DL8D#L12
EGL8G#P88L12A#L8G#GP88L8EL12ED#L
8EL14GG#A#G#GEL6D#"
1540 IFINKEY$<>"Q"THEN1540ELSERU
N180
1600 PRINT@8,"pyraminx puzzle";:
RETURN
1610 PRINT@487,"inkey to continu
e";:IFINKEY$=""THEN1610ELSERETUR
N
1620 C(1)=C(3):C(3)=C(2):C(2)=S:
RETURN
1630 C(21)=C(29):C(29)=C(23):C(2
3)=S:RETURN
1640 C(19)=C(30):C(30)=C(24):C(2
4)=S:RETURN
1650 C(20)=C(28):C(28)=C(22):C(2
2)=S:RETURN
1660 C(1)=C(2):C(2)=C(3):C(3)=S:
RETURN
1670 C(21)=C(23):C(23)=C(29):C(2
9)=S:RETURN
1680 C(19)=C(24):C(24)=C(30):C(3
0)=S:RETURN
1690 C(20)=C(22):C(22)=C(28):C(2
8)=S:RETURN
```

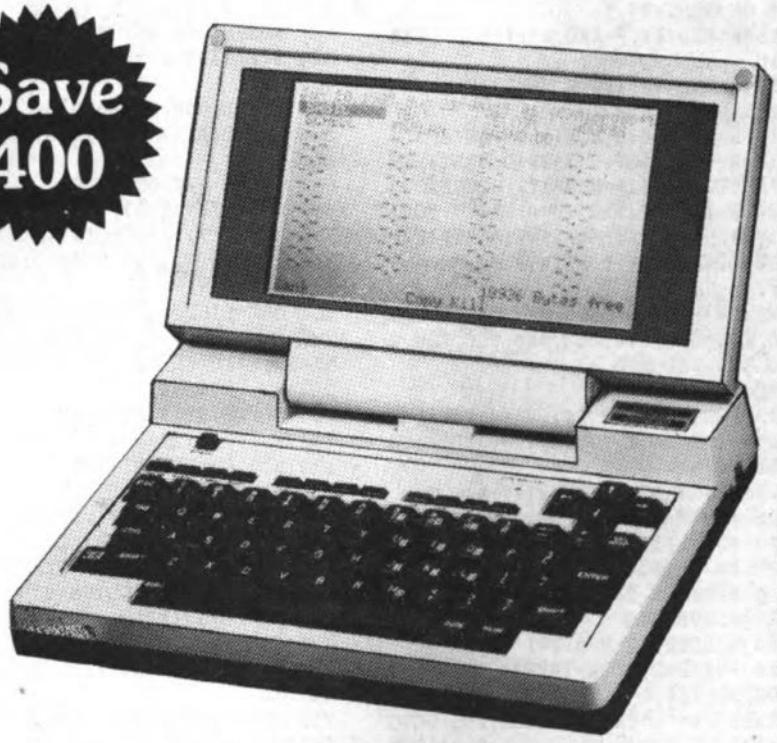
Tandy ELECTRONICS

The Ultimate In Lap-Top Computing!

Reg **1399.00**

$999

Save $400

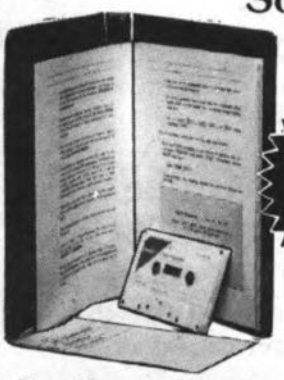


- **Features Six Built-In Programs Including MultiPlan**
- **24,000 Character Memory**

Tandy 200. Whether you're a student, journalist or professional person, the only way to compute on the go and get results is with the Tandy 200 portable computer. For complex spreadsheet and calculations, we put MultiPlan software into the 200's permanent memory enabling it to give sales forecasts, profit and loss, budgeting, pricing and a lot more. You can also write your own programs in BASIC language. The Tandy 200 has a high-contrast flip-top LCD incorporating 240 x 128 resolution on a 16-line 40 character text. Includes parallel printer RS 232C interface. A unique offer! 26-3860

SPECIFICATIONS. Microprocessor: 8-bit 80C85 CMOS. **Clock Speed:** 2.4 MHz. **Memory:** 72K ROM, expandable to 104K; 24K RAM, expandable to 72K in banks of 24K. **Keyboard:** Full-sized 56 key typewriter-style with embedded data pad, plus 8 programmable function keys. **Display: 40 x 16 liquid crystal display, upper and lower case ASCII, characters, 240 x 128 dot-matrix graphics. Input/output:** Parallel printer interface. RS-232C serial communications interface. Cassette tape interface. Standard bar-code reader interface. **Dimensions:** 5.5 x 29.8 x 21.6cm. **Weight:** 2kg. **Power Supply:** Operations — Up to 16 hours on 4 "AA" alkaline batteries, and can use nickel-cadmium rechargeable batteries with minor additional-cost modification. Optional AC/DC charger/adapter available. Memory—internal rechargeable nickel-cadmium batteries will retain memory for up to 15 days with power off.

Versatile Tandy 200 Software

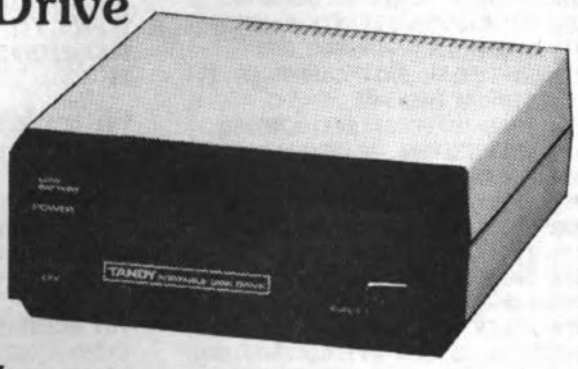


Save Up To $20

Data Manager. For all your data. 26-3886
Reg 79.95 . . . . . **Sale! 59.95**
Project Scheduler. Segment your work. 26-3893 Reg 79.95 . . . . . **Sale! 59.95**
Text Power 200. Features advanced formatting capabilities in word processing. 26-3881 Reg 79.95 . . . . . **Sale! 59.95**
Time Minder. Appointment reminder. 26-3883 Reg 79.95 . . . . . **Sale! 59.95**
Romulus Chess. With 7 echelons of play. 26-3890 Reg 45.95 . . . . . **Sale! 39.95**

Portable 8.89cm Floppy Disk Drive

- **PortableDisk Storage For Your 100**



Save $200

Reg **399.95**

199.95

Now you can be in a position to access an extra 100K of data whenever your away from the office. Includes initialization software. Weighs only 793g. Requires 4 'AAA' batteries. 26-3808

Student Computer Desk

- Suits Most Hardware

Save $50

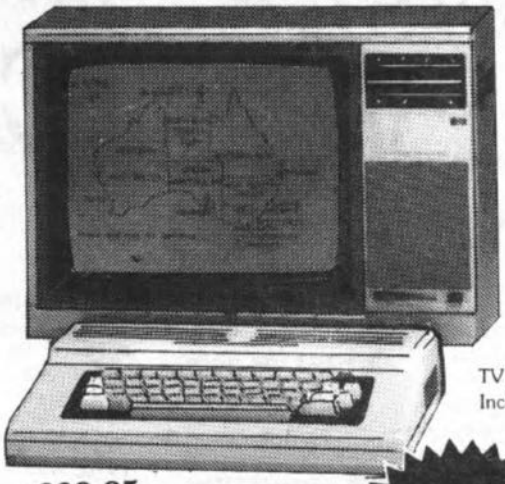
Computer and Accessories Not Included

Reg 199.95

149.95

This robust and decorative piece is a must for the student! Gives easy passage of paper from storage shelf to printer in a three tier configuration. 26-9408

Color Computer 2



TV Not Included

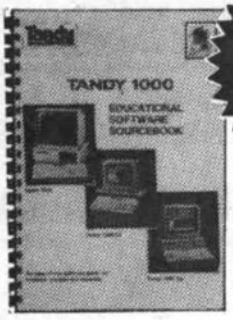
Reg 299.95

199.95

Save $100

Now you can both educate and entertain your family hour after hour by simply attaching Color Computer 2, to your T.V. and plugging in a Program Pak. The 16K Extended will teach your children all about maths, science, reading and computer fundamentals. 26-3136

Education Source Book



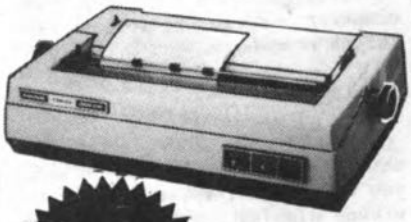
Save $10

Reg 14.95

4.95

A thorough guide to locating and buying software. An offer that's exclusive to Goldsoft readers! Present this ad at a Tandy Store for discount 26-7991

Daisy Wheel Printer



Save $1500

Reg 2499

$999

Our model 510 professional word processing printer averages 43 cps and prints 10/12 cpi or relative spacing (switch select). Includes a courier 10 wheel. 26-1270

Single Diskettes

Save $4

Reg 6.95

- Fully Certified Ensuring Quality

2.95

All diskettes are unformatted, double density and soft sector. Exclusive price to Goldsoft readers! Just take ad along to your local Tandy store. 26-0305

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

INVERSE SWITCH

HARDWARE MODIFICATION

THIS IS A fairly old hardware modification, but seeing that there are still a lot of people with CoCo 1's and 2's out there, I've decided to bring it out of the bag again.

This inverse switch will enable your CoCo to have a green on black text screen (as opposed to a black on green screen). The results are less eye strain and sometimes even a better display. Mind you, it's great for late night work, when your eyes are very sensitive to any amount of brightness for any length of time.

The project in itself is fairly easy to put together, the only tough thing about it would be deciding where you want your switch to reside (you have to drill a hole somewhere in your CoCo to fit your plug on.)

Requirements:

- o 1 DPSS switch
- o 3 wires; all about 25cm long
- o Solder and soldering iron
- o Drill and drill bit the size of your switch.

I'm referring to a long grey case computer, so if you are trying this on any other computer, it might work, it might not. I think it should.

Method:

o Get one wire and solder onto the left side of the switch and solder it onto pin 1 of the SN741S02N (this is a small 14 pin chip)

o Get another wire and solder it to the middle of the switch. Take the other end and solder it to pin 9 of the MC6847P (a 40 pin chip). Remember that pin 9 has been bent out of it's socket and is in no way attached to the socket it was once attached to.

o Solder the last remaining wire to the right of the switch. This particular wire attaches itself to pins 2 AND 3 to the SN741S02N chip (the same 14 pin chip described in step one).

From pin three of the SN741S02N the wire then goes on to pin 2 of the MC6847P.

o Drill a small hole into your **computer so as to attach the switch somewhere.**

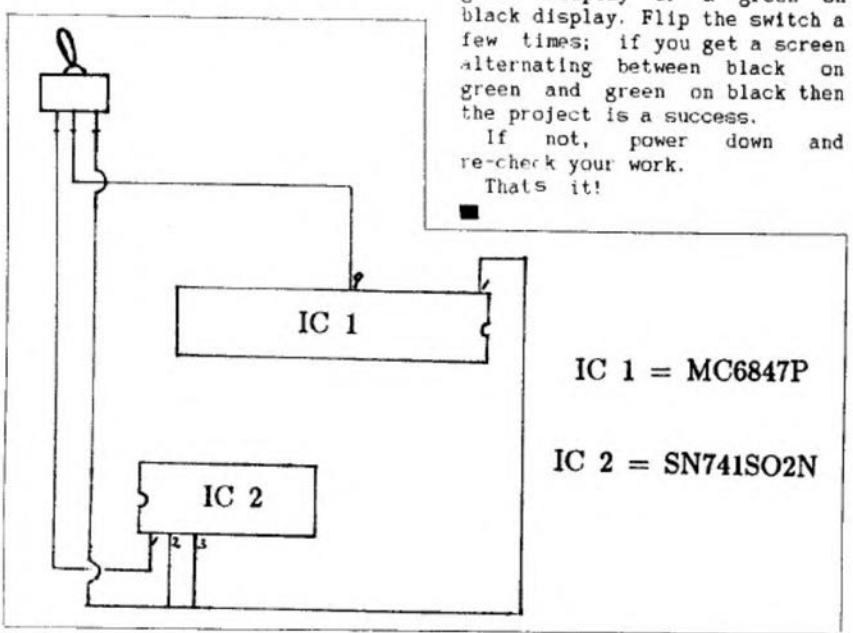
Testing:

o Power up the computer; depending how the switch is set, you should get either a black on green display or a green on black display. Flip the switch a few times; if you get a screen alternating between black on green and green on black then the project is a success.

If not, power down and re-check your work.

Thats it!

■



SINLINE

by Justin Lipton

YOU MAY REMEMBER an article I wrote some time ago about sine waves. Here is the original program.

```
10 PMODE 4,1:SCREEN1,1:PCLS
 :REM SET UP HIGH-RES SCREEN
20 FOR A=0 TO 256
 :REM HORIZONTAL LOCATION ON
  SCREEN
30 B=INT(SIN(A*6)*6)+100
 :REM FIND SINE VALUE OF A
  MULTIPLY BY 6 AND MOVE IT
  DOWN 100 PIXELS
40 PSET(A,B,5)
 :REM PLOT THE POINTS
100 NEXT A
110 GOTO 110
 :REM STAY ON GRAPHICS SCREEN
```

Since first writing this program I have discovered some other effects you may want to try. First try ...

```
60 LINE(A,B)-(128,0),PSET
```

Interesting Huh? Now try adding line 70 below for a picture that looks something like a clam ...

```
70 LINE(A,B+20)-(128,196),PSET
```

Let's try something else. Delete line 60 and 70 and we'll try using the 'B' (box) function of the LINE command ...

```
60 LINE(A,B)-(A+9,B+9),PSET,B
```

We now get large exaggerated thick curves. I wonder what would happen if we left the 'B' out.

```
60 LINE(A,B)-(A+9,B+9),PSET
```

We have now produced 3-dimensional looking waves. Remember one of the best ways to learn to program is to sit down and experiment with Coco's various commands.

■

CoCo 3 Program

CONVERSION of BASIC PROGRAMS for the COCO 3

by George McLintock

UTILITY

THE NEW GRAPHIC screen commands for the CoCo 3 high resolution graphic screen follow the same general structure as used for the old CoCo graphic screen. They are similar enough to make it worth while using another Basic program to do most of the source code conversion for you.

Convert is a BASIC program to do this. It will automatically change most of a Basic program for the old graphic screen to run using the new high res screen of the CoCo 3. The program won't do it all for you, some additional changes have to be made manually.

But in general these are restricted to the initial set up of parameters for the program, and are straight forward.

Which leads to the question of why bother. If you have an existing program for the old CoCo, it will still run on the CoCo 3 without any modification.

The main advantage of converting existing programs is that it provides a basis for using and experimenting with the extra graphic capabilities of the CoCo 3.

There are a lot of good BASIC graphic programs around for the old CoCo. At the time they were written, they were limited to the graphic capabilities of that machine. 128 x 192 in 4 colours, or 256 x 192 in 2 colours.

Pictures drawn on these screens will convert directly to the new 320 x 192 structure, with similar proportions and appearance, and a 64 pixil right margin. The big difference of course is that you now have 16 colours available at this resolution.

So that if you do a direct conversion of an existing program that you like, you can then experiment with the new commands in a more meaningful way.

Using CONVERT

The BASIC program to be converted must first be saved in ASCII format, ie use (C)SAVE "Name",A.

Convert will read these source statements as an input file, change the code as required, and write the modified statements out to a new output file in a similar format. This new file can then be (C)LOAD'ed back into Basic as a normal BASIC program.

The program submitted is written for a disk system with both the input file and output file open at the same time. Each program line is input in sequence, modified if required, and written out again.

This is not possible with a cassette based system, so I have indicated alternative code required to read the whole program into memory at one time, change it, and write it out again. This will work with a cassette system, and there should be sufficient memory available for most programs.

OTHER CHANGES REQUIRED

Convert will alter all program statements which are different between the screens, to the correct syntax for the CoCo 3 screen. Most other changes required to get the program to work are made at the logical start of the program which initialises parameters, eg the DIM, SCREEN, PMODE etc, which are set before the program actually starts to do anything.

All PMODE, PCLEAR, and PCOPY statements are removed.

All SCREEN statements are altered to HSCREEN ZQ, where ZQ is a variable to be set to 1-4 (depending on the screen required) before the first HSCREEN is executed.

If the program contains more than one SCREEN statement, use POKE &HE6C6,18: POKE &HE6C7,18 to prevent later HSCREEN commands clearing the screen.

The original program will have PCLS (converted to HCLS) commands as required to clear the screen.

If the program alternates between the graphic screen and text screen (as with some games and screen draw programs), then you will have to insert a HSCREEN 0 command before the program attempts to write to the text screen after displaying the graphic screen.

PSET and PRESET are altered only when required. The test used is that if they are preceeded by a comma (eg as part of a LINE or PUT) then they are left unaltered. If not then the P is replaced wi an H.

PCLS and PPOINT commands have the first P replaced with an H, and all other graphic commands have an H inserted in front of them.

GET and PUT are changed to HGET and HPUT. The array name used in the old program is left as a simple variable which should be set equal to the buffer number as defined by the equivalent HBUFF command.

The normal situation for the old program would be along the lines of ...

```
DIM A1(140)
GET (x,y)-(x1,y1),A1
PUT (x2,y2)-(x3,y3),A1
```

Convert will alter the GET and PUT statements. To make it work, you have to replace the DIM statement with A1=1: HBUFF A1,140 (It won't be 140 though)

Under some circumstances GET requires a ",G" after the buffer name. This is not allowed with HGET, and *Convert* will remove it if it's there.

I have also included an outline of an alternative way of calculating the buffer size required. It is most unlikely to be the same as the DIM value.

If the program happens to use A1 as a simple variable as well as an array, then you will have to alter each HGET and HPUT command to insert a new variable as the buffer number.

The program performs other checks along the way to prevent changes being made where they should not be. eg inside strings, in DATA lines, and to exclude the PUT in INPUT.

Comments and remarks are not protected, and will be changed in the same way as the rest of the program.

After making these changes, the program should work on the high res screen, and you can start changing some colours with HCOLOR and PALETTE.

REFERENCE TABLE

When you start altering the program to vary the graphics, or simply to find some particular use of a command, it can be very helpful to have a reference to line numbers containing any particular command.

Convert produces a reference table of all line numbers containing a graphic command which is altered by it. It also produces a reference to DIM as this must be altered when GET's are used. As a further check, it also provides a reference for possible ML routines (DEFUSR and EXEC) and PEEK's and POKE's.

When the program has been converted, you get an option to print this table or not. The strings SE$ and EE$ (line 1160) are used to elongate part of the printout. These can be altered or set to null if they don't suit your printer.

The program also checks for very long lines. If the original source line is greater then 253 characters, then these are noted in the table. You may have lost something off the end of the original line.

The program will increase the length of some lines. If the new line would be longer than 255 bytes, the change is not made, and these line numbers are noted in the table.

CASSETTE SYSTEM

The basic change required for a cassette system is to read the whole program into an array before you start. The control logic is altered to modify each line from this array, and when finished, write the whole program out again.

You also have to delete the addition of "/BAS" in lines 20 and 30.

To do this:

* Add an extra array, say P$(500) to the DIM's in line 1100.

* Replace lines 40 to 60 with something like ...

```
40 OPEN "I",#-1,N$:P=0
45 IF EOF(-1) THEN CLOSE
 :GOTO60
50 P=P+1:LINEINPUT#-1,P$(P)
55 GOTO 45
60 FOR W = 1 TO P:A$=P$(W)
```

... and replace lines 620-630 with something like

```
620 P$(W)=A$:NEXT W
622 INPUT "prompt for
 cassette ready";A$
624 OPEN "O",#-1,N1$
626 FOR X=1 TO P
628 PRINT #-1,P$(X)
630 NEXT X:CLOSE
632 GOTO 940
```

* For programs too large to fit in memory as provided, you can CLEAR more space,

* Increase the size of P$ array

* **Eliminate the** reference table, or make it smaller,

* Do the program in two parts and merge them together again.

SIZES for HBUFF

The manual provides a description of how to calculate the size of HBUFF required for a HGET command. I find it difficult to follow and use an alternative procedure derived from basic principles.

The HGET command moves whole bytes from the graphic screen to the buffer. It does not take individual bits as occurs with GET.

The buffer size required is specified as the number of bytes required to be held there, with the count starting from zero. eg to hold 160 bytes requires a HBUFF of 159 which is bytes 0 to 159 or a total of 160 bytes.

Following from this, some minimum sizes can be derived.

To hold a single pixil width line from the top of the screen to the bottom (eg for LINE (1,0)-(1,191)), requires a buffer of 192 bytes.

The graphic screen is held in memory addresses in much the same way as the normal text screen. With adjoining memory locations holding pixels running across the screen. So that to take a single **pixel** from the top line of the screen, you have to take one byte, rather then just the number of bits in the **pixel**. With 192 lines required, you have to take 192 bytes.

Similarly, to hold a single **pixel** width line running right across the screen eg LINE (0,1)-(319,1) or LINE (0,1)-(639,1), requires either 80 or 160 bytes depending on the HSCREEN used.

The number of bytes required is the number of pixels across the screen divided by the number of pixels per byte. The 16 colour screen has 2 pixels per byte, the 4 colour ones have 4 and the 2 colour one has 8 pixels per byte. For this line you require the number of bytes used to display that many pixels.

Following from this, the number of bytes required to hold an area of the screen depends mainly on the number of bytes required to hold each line running across the screen. Once this is known, then the total number of bytes required is the number of bytes per line across the screen, multiplied by the number of lines deep for the area to be saved.

The number of bytes per line across the screen depends on the number of pixels per byte and how it aligns with byte boundaries.

If the area to be got is from a fixed position of the screen, you can often use one less byte per line then if you have to allow for getting the same area (number of pixels) from anywhere on the screen. This occurs when each line starts or ends on a byte boundary.

For example, with a 4 colour screen, a buffer size of 192 (HBUFF 1,191) will hold an area of 4 pixels wide for the full depth of the screen if the lines across the screen start on a byte boundary. But the same area will require a buffer of 384 bytes if it starts anywhere else. ie you can HGET (0,0)-(3,191) into a buffer of 192 bytes, but you require 384 bytes to HGET (1,0)-(4,191). The difference occurs because you require 2 bytes to hold the pixels from 1 to 4 across the screen. The first byte for pixels 1 to 3, and the second byte for pixil no 4.

Hence the number of bytes per line is the number of pixels required, divided by the number of pixels per byte. If the answer is not a whole number, round up to the next highest number. If the line being got does not always start and end within this number of bytes, add one more to the number of bytes per line.

With the 16 colour screen, you only ever have to add one once. If the number of pixels per line is odd, then it must always start **or end on a byte boundary**. So after dividing by 2 and rounding up, you can always get that number of pixels per line from anywhere on the screen, with that many bytes.

A similar situation can occur with the other screens. eg if you want 5 pixels per line with a 4 colour screen, then you can always get them in 2 bytes. But if you want a line with 6 pixels from anywhere on the screen you have to allow three bytes per

line, ie add the extra byte per line after rounding up.

If in doubt, go back to the basic principles and work it through for the particular requirement.

The Listing:

```
1 '**   CONVERT
        BY GEORGE MCLINTOCK
        30/11/86
2 GOTO 10
3 SAVE"120A:3":END
4 STOP
5 '
6 ' PROGRAM TO CONVERT SOURCE ST
ATEMENTS FOR OLD COCO GRAPHIC SC
REEN TO HIGH RES SCREEN FOR COCO
 3
7 '
10 CLS:PRINT"PROGRAM TO CONVERT
A LOW RES    SCREEN BASIC PROGRA
M TO HIGH RESFOR THE COCO 3":PRI
NT:PRINT"THE PROGRAM MUST BE SAV
ED TO    DISK IN ASCII FIRST":PR
INT
20 PCLEAR 1:CLEAR 15000:GOSUB 10
90:PRINT "ENTER NAME OF EXISTING
 PROGRAM":INPUT N$:K=INSTR(N$,"/
"):IF K=0 THEN N$=N$+"/BAS"
30 PRINT:PRINT"ENTER NAME FOR NE
W PROGRAM":INPUT N1$:K=INSTR(N1$
,"/"):IF K=0 THEN N1$=N1$+"/BAS"
40 OPEN "I",#1,N$:OPEN "O",#2,N1
$
50 IF EOF(1) THEN 940
60 LINE INPUT #1,A$
70 IF LEN(A$)=0 THEN 620 ELSE IF
 LEN(A$) > 253 THEN J=16:GOSUB 8
10
80 K=INSTR(A$," "):LN$=LEFT$(A$,
K):PRINT LN$:PRINTA$
90 IF MID$(A$,K,1)=" " THEN K=K+
1:GOTO 90
100 IF MID$(A$,K,4)="DATA" THEN
620
110 GOSUB 870
120 J=1:K=INSTR(A$,C$(J)):IF K<>
0 THEN GOSUB 810 'DIM
130 J=2:FOR X=1 TO 3 'PMODE PCLE
AR PCOPY
140 K=INSTR(A$,H$(X)):IF K=0 THE
N 200
150 GOSUB 650:IF SW=1 THEN 140
160 K1=INSTR(K,A$,":"):A1$=LEFT$
(A$,K-1):GOSUB 810
170 IF K1=0 THEN A2$="" ELSE A2$
=MID$(A$,K1)
180 IF INSTR(A1$,":")=0 AND INST
R(A2$,":")=0 THEN A$=A1$+"REM"+A
2$:GOTO 200
190 A$=A1$+A2$:GOTO 140
200 NEXT X
210 '
220 J=3:L=1 'SCREEN
230 K=INSTR(L,A$,C$(J)):IF K=0 T
HEN 290
240 GOSUB 650:IF SW=1 THEN 230
250 K1=INSTR(K,A$,":"):A1$=LEFT$
(A$,K-1):L=K+LEN(C$(J)):GOSUB 81
0
260 IF K1=0 THEN A2$="" ELSE A2$
=MID$(A$,K1)
270 A$=A1$+"HSCREEN ZQ"+A2$:GOTO
 230
280 '
290 FOR J=4 TO 5:L=1 'PSET PRESE
T
300 K=INSTR(L,A$,C$(J)):IF K=0 T
HEN 360
310 GOSUB 650:IF SW=1 THEN 300
320 X=K-1:L=K+LEN(C$(J))
330 IF MID$(A$,X,1)=" " THEN X=X
-1:GOTO 330
340 IF MID$(A$,X,1)="," THEN 300
350 GOSUB 810:MID$(A$,K)="H":GOT
O 300
360 NEXT J
370 '
380 FOR J=6 TO 7:L=1 'PCLS PPOIN
T
390 K=INSTR(L,A$,C$(J)):IF K=0 T
HEN 420
400 GOSUB 650:IF SW=1 THEN 390
410 MID$(A$,K)="H":L=K+LEN(C$(J)
):GOSUB 810:GOTO 390
420 NEXT J
430 '
440 FOR J=8 TO 14:L=1 'ADD H TO
THESE
450 IF LEN(A$) > 254 THEN J1=J:J
=15:GOSUB 810:J=J1:GOTO 520
460 K=INSTR(L,A$,C$(J)):IF K=0 T
HEN 520
470 GOSUB 650:IF SW=1 THEN 460
480 IF C$(J)="GET" THEN GOSUB 70
0
490 IF C$(J) = "PUT" THEN IF MID
$(A$,K-2,2) = "IN" THEN L=K+3:GO
TO 460
500 A1$=LEFT$(A$,K-1):A2$=MID$(A
$,K):L=K+LEN(C$(J))+1
510 A$=A1$+"H"+A2$:GOSUB 810:GOT
O 450
520 NEXT J
530 '
540 J=17:FOR X=4 TO 5
550 K=INSTR(A$,H$(X)):IF K <> 0
THEN GOSUB 810
560 NEXT X
570 '
580 J=18:FOR X=6 TO 7
590 K=INSTR(A$,H$(X)):IF K <> 0
THEN GOSUB 810
600 NEXT X
610 '
620 PRINT #2,A$:PRINTA$
630 GOTO 50
640 'TEST FOR INSIDE QOUTES
650 SW=0:IF Q=0 THEN RETURN
660 FOR T=1 TO Q
670 IF K>Q(T,0) AND K<Q(T,1) THE
N SW=1: L=Q(T,1): T=Q+2
680 NEXT T:RETURN
690 '
700 K1=INSTR(K,A$,":"):IF K1=0 T
HEN GOSUB 770 ELSE K1=K1-1
710 IF MID$(A$,K1,1) = " " THEN
K1=K1-1:GOTO 710
720 Y=K1-1
730 IF MID$(A$,Y,1)=" " THEN Y=Y
-1:GOTO 730
740 IF MID$(A$,Y,1)="," AND MID$
(A$,K1,1)="G" THEN 750 ELSE RETU
RN
750 A1$=LEFT$(A$,Y-1):A2$=MID$(A
$,K1+1):A$=A1$+A2$
760 RETURN
770 K1=INSTR(K,A$,"'"):IF K1=0 T
HEN K1=LEN(A$) ELSE K1=K1-1
780 RETURN
790 '
800 'RECORD LINE NUMBERS
810 IF LEN(L$(M(J,1))) = 0 THEN
L$(M(J,1))=LN$:RETURN
820 IF INSTR(L$(M(J,1)),LN$) <>
0 THEN RETURN
830 IF LEN(L$(M(J,1)))+LEN(LN$)
> 255 THEN M(J,1)=M(J,1)+1:IF M(
J,1) >= M(J+1,0) THEN PRINT:PRIN
T "OUT OF SPACE FOR ";C$(J):PRIN
T"ADJUST M ARRAY":CLOSE:STOP
840 L$(M(J,1)) = L$(M(J,1)) + LN
$
850 RETURN
860 'SET UP QUOTE ARRAY
870 Q=0:L=1
880 K=INSTR(L,A$,CHR$(34)):IF K=
0 THEN RETURN
890 Q=Q+1:Q(Q,0)=K:L=K+1
900 K=INSTR(L,A$,CHR$(34)):IF K=
0 THEN K=LEN(A$)-1
910 Q(Q,1)=K:L=K+1:GOTO 880
920 '
930 ' PRINT REFERENCE TABLE
940 CLS:CLOSE:PRINT "ENTER  P  T
O PRINT TABLE":INPUT "ELSE PRESS
 ENTER";A$:IF A$="" THEN STOP
950 PRINT#-2,SE$;N$;EE$:PRINT#-2
,"LINE NUMBERS CONTAINING SELECT
ED COMMANDS"
960 FOR J=1 TO 18
970 PRINT#-2,SE$;C$(J);EE$;" ";
980 IF LEN(L$(M(J,0))) = 0 THEN
PRINT#-2,"NONE":GOTO 1060
990 IF LEN(L$(M(J,0))) <= 80 - L
EN(C$(J))*2+2 THEN PRINT#-2,L$(M
(J,0)):GOTO 1060 ELSE PRINT#-2
1000 FOR X=M(J,0) TO M(J,1)
1010 IF LEN(L$(X)) <= 80 THEN PR
INT#-2,L$(X):GOTO 1050
1020 Y=80
1030 IF MID$(L$(X),Y,1) = " " TH
EN PRINT#-2,LEFT$(L$(X),Y-1):L$(
X) = MID$(L$(X),Y+1):GOTO 1010
1040 Y=Y-1:GOTO 1030
1050 NEXT X
1060 NEXT J
1070 '
1080 STOP
1090 'SET PARAMS
1100 DIM C$(19),M(19,1),L$(190),
H$(10),Q(50,1)
1110 FOR J=1 TO 19:M(J,0)=(J-1)*
10+1:M(J,1)=M(J,0):NEXT J
1120 C$(1)="DIM":C$(2)="REMOVED"
:C$(3)="SCREEN":C$(4)="PSET"
1130 C$(5)="PRESET":C$(6)="PCLS"
:C$(7)="PPOINT":C$(8)="PAINT"
1140 C$(9)="PUT":C$(10)="COLOR":
C$(11)="CIRCLE":C$(12)="LINE"
1150 C$(13)="GET":C$(14)="DRAW":
C$(15)="TOO LONG":C$(16)="> 253"
1160 SE$=CHR$(14):EE$=CHR$(15)
1170 C$(17)="POSS ML":C$(18)="PE
EK/POKE"
1180 H$(1)="PMODE":H$(2)="PCOPY"
:H$(3)="PCLEAR":H$(4)="DEFUSR"
1190 H$(5)="EXEC":H$(6)="PEEK":H
$(7)="POKE"
1200 RETURN
```

PETER GUNN

CoCo + Orchestra 90cc
MUSIC

by Michael Monck

THE *PETER GUNN* theme has been used in many ways, ranging from a nice instrumental to being used in movies ("The Blues Brothers") to video arcade machines ("Spy Hunter").

Even a pop group has re-written this piece ("The Art of Noise")!

Well, it had to come sometime!

Here is a listing for the Peter Gunn theme for the Orchestra-90CC, done by the ever talented Michael Monck, who gave us "Axel F", "Popcorn" and other great pieces!

Enjoy it!

The Listing:

```
/PETER GUNN (HENRY MANCINI)
/ENTERED & ARRANGED BY
/MICHAEL MONCK AH059.75-4790

NQ=80
V1YA V2YD V3YA V4YA V5YD
K1&
P01
M
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
P02
R01
R01
P03
M
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"SC"C"C"C"C"C"C"C"
M
_H.-5&I-5-7
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
_X-7-8-9-A-B-C-D-E
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$S8"C"C"C"
M
_W-5&
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
*I0-4#X-5-6-7-8-9-A-B
V2_I-B%,-B%,-A,-B%,S-A#-9I-B%-8
V2_-9
V5_Q8"IC"8"8"$C"
M
_I$-7-5&-4Q:-3&;-3&;-3&;
V2_I-B%,-B%,-A,-B%,S-A#-9I-B%-8
V2_-9
V5_Q8"IC"8"8"$C"
M
_Q:-3&;-4-5&-7-8-7
V2_I-B%,-B%,-A,-B%,S-A#-9I-B%-8
V2_-9
V5_Q8"IC"8"8"$C"C"
M
_I-A#-9X-A-B-C-D-E-F
V2_I-B,-B,-A%,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"
M
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"SC"C"C"C"A"A"A"A"
P04
M
*H.2%I2&0
V3*H.0I0-2
V4_H.-3%I-3%;-5#
V2_I-B,-B,-A,-B,S-A-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
*X0_-1-2-3-4-5-6-7
V3_X-2-3-4-5-6-7-8-9
V4_X-5#-6-7-8-9-A-B-C
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"
M
*W2&
V3*W0
V4*W-2
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
*I71#X0-1-2-3-4-5-6
V3*I0_-6#X-7-8-9-A-B-C-D
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"
M
*I$02&3I:3%4&,3%4&,3%4&,
V3_I$7-5&-4Q:-3&,-3&,
V3_-3&
V2_I-B%,-B%,-A%,-B%,S-A#-9I-B%
V2_-8-9
V5_Q8"IC"8"8"$C"C"
M
*Q:4&32&0-10
V3_Q:-3&-4%-5&-7-8-7
V2_I-B%,-B%,-A%,-B%,S-A#-9I-B%
V2_-8-9
V5_QC"I8"C"C"$C"8"
P05
M
_I-3#-2X-3-4-5-6-7-8-9
V3_I-A#-9X-A-B-C-D-E-F
V2_I-B%,-B%,-A%,-B%,S-A#,-9
V2_I-B%-8-9
V5_Q8"IC"8"SC"C"C"C"8"8"8"C"
M
*H$I$7,7,7,
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"C"C"C"C"
M
*I7AS:9&A9I7153Q$
V2_I-4$Q$$-B,
V5_Q8"$$8"
M
*I7AS:9&A9I67$$$
V2_H$I$-E,-E,$
V3_H$I$-7,-7,$
V5_QH$I$8"8"
M
*I7AS9&A97I5&356
M
*I7,A,A,A,A,A,A,A,
V3*I3,3,4,4,4#,4,5,5,
V2_I$-B,-B,-B,-B,-B,-B,-B,
V5_I8"8"8"8"8"8"8"8"
P06
M
*HAIA,A9&7
V3*H7I7,765
V4*H5I5,543
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"
M
*IA,A9&77,7,7,7
V3*7,7655,5,5,5
V4*5,5,433,3,3,3
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
*QAI9&7QAI97
V3*Q7I65Q7I65
V4*Q5I43Q5I43
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"
M
*IA,A,9&7,H7
V3*I7,7,65,H5
V4*I5,5,43,H3
V2_I-B,-B,-A,-B,S-A#-9I-B-8-9
V5_Q8"IC"8"8"$C"C"
M
*HCIC,C,B9&
V3*HA%IA%,A%,9&6
V4*H7I77,6,4
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
V2_-9%
V5_Q8"IC"8"8"$C"C"
M
*IC,CB9&9&,9&,9&,9&,
V3*IA,A9&66,6,6,6,
V4*I7,7644,4,4,4,
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
V2_-9%
V5_Q8"IC"8"8"$C"C"
M
*QCIB9&QCIB9&
V3*QAI9&6QAI9&6
V4*Q7I64Q7I64
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
```

```
V2_-9%
V5_Q8"IC"8"8"$C"8"
M
*IC,CB9&9&,9&,9&,9&,
V3*IA,A9&66,6,6,6,
V4*I7,7644,4,4,4,
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
V2_-9%
V5_Q8"IC"8"8"8"8"C"
P07
R04
P08
M
_I-3#-2X-3-4-5-6-7-8-9
V3_I-A#-9X-A-B-C-D-E-F
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
V2_-9%
V5_Q8"IC"8"SC"C"8"8"C"C"8"C"
M
*H$I$9&,9&,9&,
V3*H$I$7,7,7,
V4*H$I$4%,4%,4%,
V2_I-B,-B,-A,-B,S-A#,-9%I-B-8
V2_-9%
V5_Q8"IC"8"SC"8"C"8"C"8"C"8"
P09
R04
P0A
M
I
*I4#5H$I$3%
V3*I3#4%
V4*I12&
V2_I-B%,-B%,-A,-B%,S-A#,-9%
V2_I-B%-8-9%
V5_Q8"IC"8"8"$C"8"
M
*I4#5H$I$A%
V3*I3#4%
V4*I12&
V2_I-B%,-B%,-A,-B%S-A#,-9%
V2_I-B%-8-9%
V5_Q8"IC"8"8"$C"C"
M
*IB#CH$I$A%
V3*IA#B%
V4*I89&
V2_I-B%,-B%,-A,-B%,S-A#,-9%
V2_I-B%-8-9%
V5_Q8"IC"8"8"$C"8"
M
*IB#CQ$I$S2%3%,3%517
V3*IA#B%
V4*I89&
V2_I-B%,-B%,-A,-B%,S-A#,-9%
V2_I-B%-8-9%
V5_Q8"IC"8"SC"8"C"8"C"8"C"8"
M
*Q9&,9,9,9,
V3*Q7,7,7,7,
V4*Q5&,5,5,5,
V2_Q-A&,-A,-A,-A,
M
*Q9&,9,9,I9B
V3*Q7,7,7,I78
V4*Q5&,5,5,I56%
V2_Q-A&,-A,-A,I-A-B
M
*H$Q$BXA98765432
V3*H$Q$8X76543210
V4*H$Q$6%X543210-1-2
V2_H$Q$-B
V5_IC"8"C"8"C"8"
```

■

RECOVER

16K ECB TAPE UTILITY

by Grahame Pollock

RECOVER IS A VERY handy utility. It continues to CLOAD or CLOADM after an I/O error so that you get the entire program with a bit of garbage in the middle.

To use Recover, you must first run the main program and follow the instructions.

Overall, it's a very handy utility!

The Listing:

```
2 GOTO10
3 SAVE"189B:3":END
10 PRINT"IO TAPE RECOVERY":PRINT
"FOR THE COCO"
20 PRINT"BY GRAHAME POLLOCK"
30 PRINT"24 KENT ST ,
40 PRINT"MINTO, 2566."
50 PRINT"AUST.":PRINT:PRINT
60 PRINT"TO RECOVER BASIC OR ML
PROGRAMS FROM TAPE"
70 INPUT"PRESS ENTER";X
80 CLS:PRINT"DO YOU WANT RECOVER
Y TO SIT"
90 PRINT"1. BELOW MEMORY TOP"
100 PRINT"2. ELSEWHERE"
110 INPUTM
120 IFM=2THEN200
140 P=256*PEEK(&H74)+PEEK(&H75)
150 P1=P-265
160 CLEAR100,P1
170 P=256*PEEK(&H74)+PEEK(&H75)
180 P1=P-265
190 GOTO270
200 INPUT"WHERE";P1
210 P2=INT(P1/256)
220 P3=P1-P2*256
240 POKE&H400,P2:POKE&H401,P3
250 CLEAR100,P1
260 P1=256*PEEK(&H400)+PEEK(&H40
1)
270 PRINT"RECOVERY IS NOW LOADIN
G IN      MEMORY FROM"
280 PRINTP1"TO"P1+265
290 PRINT"&H"HEX$(P1)" TO &H"HEX
$(P1+265)
300 FORX=0TO265
310 READ A
320 POKE P1+X,A
325 A1=A1+A
330 NEXT X
335 IFA1<>32605 THEN CLS:PRINT"D
ATA ERROR":END
340 PRINT:PRINT"EXEC &H"HEX$(P1)
" OR EXEC"P1"TO    RECOVER A BA
SIC PROGRAM"
350 PRINT"EXEC &H"HEX$(P1+102)"
OR EXEC"P1+102"TO    RECOVER A M
.L.    PROGRAM"
360 PRINT:PRINT"BASIC WILL NEED
EDITING AFTER   RECOVERY"
370 PRINT"M.L. WILL NEED BLOCK M
EMORY     MOVES AFTER RECOVERY"
380 DATA 15,120,129,77,1,1,50,98
,189,165,197,189,166,72,125,1,22
8,39,29,182,1,227,39,29,189,173,
25,134,255,151,111,12,120,189,16
6,53,126,172,124,189,1,133,189,1
64,45,126,172,115,182,1,226,39,3
,126
390 DATA 166,22,189,173,25,189,1
67,124,158,25,159,126,220,126,76
,189,172,55,189,167,11,38,127,15
0,124,39,123,42,237,159,27,141,7
6,142,171,236,189,185,156,126,17
2,233,189,173,25,126,166,25,157,
159
400 DATA 141,118,189,166,72,158,
138,157,165,39,6,189,178,109,189
,183,61,182,1,226,129,2,38,181,2
52,1,229,51,139,223,157,125,1,22
8,38,169,252,1,231,48,139,159,12
6,189,167
410 DATA 124,189,167,11,38,58,15
9,126,13,124,39,52,42,243,126,16
7,233,39,5,189,183,61,159,157,11
0,159,0,157,189,1,127,150,111,76
,39,80,126,173,235,189,179,228,1
31,1
420 DATA 255,16,34,14,236,195,5,
255,221,136,57,134,69,173,159,16
0,2,22,255,107,134,69,173,159,16
0,2,32,185,16,16,16,142,1,209,11
1,128,134,32,167,128,140,1,218,3
8,249,157,165,39,23,189,177,86
430 DATA 189,182,84,206,1,209,23
1,192,39,10,140,198,8,166,128,16
7,192,90,38,249,57
```

■

Wow! I love adventures, but I hate typing them in. This one must be the shortest I've ever written. If you can find your way out without modifying the program, you are better at it than me!

```
10 PRINT"YOU ARE LOST IN T
HE DESERT."
20 PRINT"YOU ARE OUT OF WA
TER."
30 INPUT"NORTH, SOUTH, EAS
T OR WEST";D$
40 GOTO 10
```

PRETTY PRINTER PATTERNS

32K ECB
PRINTER APPLICATION

by Bob Delbourgo

THIS PROGRAM is disk based and is a utility for designing your own printer pattern borders. Your creations may be stored on disk and recalled when needed for printing.

The method uses a form of fat bits that are found on CoCoMax. Before running ...

```
OPEN"O",#1,"PATTERNS.DAT"
 :CLOSE#1
```

The Listing:

```
0 GOTO5
1 '******** PATTERN ********** ******* BOB DELBOURGO ********
3 SAVE"205J:3":END'10
5 PCLEAR1:CLEAR1000:H=0:W=0:PMODE0,1:PCLS1
9 'main menu
10 CLS:FORI=0TO15:POKE1024+I,134:POKE1535-I,134:POKE1055-I,134:POKE1505+I,134:POKE1024+32*I,134:POKE1535-32*I,134:NEXTI
20 PRINT@66,"printer pattern productions";:PRINT@104,"BOB DELBOURGO";
30 PRINT@162,"<C>REATE <M>ODIFY/VIEW";:PRINT@194,"<P>RINT HARDCOPY <R>ERUN";:PRINT@226,"<S>AVE OR <L>OAD DISK FILE";
40 Q$=INKEY$:IFQ$=""THEN40
50 Q=INSTR("CMPSLR",Q$):IFQ=0THENSOUND10,5:GOTO40
60 ONQ GOTO100,200,300,400,500,550
99 'create new pattern
100 IFH>0THENPRINT@386,"PATTERN ALREADY DRAWN!!!";:GOSUB630:PRINT@386,STRING$(29,143);:GOTO40
105 PRINT@386,"ENTER WIDTH (3-42)";:INPUTW:POKE1439,134:IFW<3ORW>42THENSOUND10,5:GOTO105
106 PRINT@418,"<S>INGLE OR <D>OUBLE STRIP";:INPUTH$:POKE1471,134:H=INSTR("SD",H$):IFH=0THENSOUND10,5:GOTO106
107 PRINT@354,"ARROW KEYS TO MOVE CURSOR";:PRINT@386,"SPACEBAR TO SET/RESET FATBIT";:PRINT@418,"<ENTER> WHEN YOU'VE FINISHED ";
108 H=7*H:DIMP$(W):FORI=1TOW:P$(I)=STRING$(H,"0"):NEXTI:PP$=STRING$(W,128)
109 FORK=1TO3:GOSUB630:NEXTK
110 GOSUB600
115 I=1:J=1
120 IFMID$(P$(I),J,1)="1"THENLINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PSET,B ELSELINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PRESET,B
121 FORT=1TO20:NEXTT:IFMID$(P$(I),J,1)="1"THENLINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PRESET,B ELSELINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PSET,B
122 I$=INKEY$:IFI$=""THEN120
123 IFI$=CHR$(9)THENI=I+1:IFI>WTHENSOUND1,5:I=W
124 IFI$=CHR$(8)THENI=I-1:IFI<1THENSOUND1,5:I=1
125 IFI$=CHR$(94)THENJ=J-1:IFJ<1THENSOUND1,5:J=1
126 IFI$=CHR$(10)THENJ=J+1:IFJ>H THENSOUND1,5:J=H
127 IFI$=CHR$(13)THENGOSUB620:IFFL=1THENGOSUB1000:GOTO10
128 IFI$=CHR$(32)THENIFMID$(P$(I),J,1)="0"THENMID$(P$(I),J,1)="1":LINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PSET,BF ELSEMID$(P$(I),J,1)="0":LINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PRESET,BF
129 IFI$="R"THENRUN
130 GOTO120
199 'modify existing pattern
200 IFPP$=""THENPRINT@418,"there is no pattern to modify";:GOSUB630:PRINT@418,STRING$(28," ");:GOTO10
205 GOSUB600:GOSUB610
210 GOTO115
299 'hard copy of pattern
300 PRINT@386,"<L>EFT,<C>ENTRE OR <R>IGHT";:PRINT@418,"JUSTIFICATION ON PRINTER?";:Q$=INKEY$:LE=0:IFQ$=""THEN300
301 Q=INSTR("LCR",Q$):IFQ=0THENSOUND1,1:GOTO300
302 RE=INT(480/W):PRINT@322,"# REPETITIONS (1 -";RE;")";:INPUTR:POKE351,134:IFR>RE ORR<1THENSOUND1,1:GOTO302
303 L=480-R*W:IFQ=2THENLE=INT(L/2)
304 IFQ=3THENLE=L
310 PRINT#-2:PRINT#-2,CHR$(18);
311 FORLL=1TOLE:PRINT#-2,CHR$(128);:NEXTLL
312 FORLL=1TOR:PRINT#-2,LEFT$(PP$,W);:NEXTLL:IFQ<3THENPRINT#-2,CHR$(13);
313 IFH=14THENFORLL=1TOLE:PRINT#-2,CHR$(128);:NEXTLL:FORLL=1TOR:PRINT#-2,MID$(PP$,INSTR(PP$," ")+1);:NEXTLL:PRINT#-2,CHR$(13)
314 PRINT#-2,CHR$(30);
315 GOTO10
399 'save pattern
400 IFH=0THENPRINT@318,"pattern not yet created!!";:SOUND1,1:GOSUB630:GOTO10 ELSEGOSUB1100
401 DIMPN$(NN+1),PA$(NN+1):KK=1
402 OPEN"I",#1,"PATTERNS.DAT"
403 IFEOF(1)THEN406
404 INPUT#1,PN$(KK):INPUT#1,PA$(KK):KK=KK+1
405 GOTO403
406 CLOSE#1:INPUT"NEW SHAPE (<9 LETTERS)";PN$:IFLEN(PN$)>8THENPN$=LEFT$(PN$,8)
407 PN$(NN+1)=PN$:PA$(NN+1)=PP$
408 OPEN"O",#1,"PATTERNS.DAT"
409 FORKK=1TONN+1:PRINT#1,PN$(KK):PRINT#1,PA$(KK):NEXTKK
410 PRINT:PRINT"pattern shape has been saved.":GOSUB630:GOTO10
499 'load pattern
500 RUN501
501 GOSUB1100:PRINT:INPUT"WHICH SHAPE TO LOAD";TN$
502 OPEN"I",#1,"PATTERNS.DAT"
503 IFEOF(1)THEN510
504 INPUT#1,PN$,P$
505 IFTN$=PN$THENCLOSE#1:PP$=P$:GOSUB1200:GOTO10
506 GOTO503
510 CLOSE#1:PRINT"there is no such shape!":SOUND1,5:GOSUB630:GOTO10
550 RUN
599 'draw pattern grid, width W, height H
600 PCLS1:SCREEN1,0:FORI=1TOW:FORJ=1TOH:LINE(6*I-6,6*J-6)-(6*I,6*J),PRESET,B:NEXTJ,I:RETURN
609 'enter pattern on grid
610 FORI=1TOW:FORJ=1TOH:PLAY"L255T255C":IFMID$(P$(I),J,1)="1"THENLINE(6*I-4,6*J-4)-(6*I-2,6*J-2),PRESET,B
611 NEXTJ,I:LINE(0,100)-(100,110),PSET,BF:RETURN
620 DRAW"C0BM0,110;U8R6D8L6BR10R2BR4U8BR4G4F4BR4R2BR4U2E4H2L4G2;BM50,110H2U4E2BR8F4E4G4D4BR12U8D8BR8U8F8U8BR8F2D4G2C1"
621 I$=INKEY$:IFI$="Y"THENFL=1:LINE(0,100)-(100,110),PSET,BF:RETURN
622 IFI$="N"THENFL=0:LINE(0,110)-(100,100),PSET,BF:DRAW"BM0,100;D4F4E4F4E4U4BF8;U4E4F4LRR8D4BR8;U8BR8;R8L4D8C1":RETURN
623 GOTO621
```

continued next page

OZZIE OS9

by Fred Bisseling

FIRSTLY, I WOULD LIKE to quote a small part of an article by Dale L. Puckett in Aust. Rainbow Mar.84.

"First, we must understand that OS9 itself is only an operating system. It is not an application package or a language used to write programs. It was written TO PROVIDE the SUPPORT necessary FOR BASIC09, a state of the art language designed by Microware and Motorola."

So you see, you have just discovered the main reason for the existence of OS9.....BASIC09.

As I've mentioned in previous articles, I am not an expert on OS9. It is with this in mind, that I purchased two books called- "The Official Basic09 Tour Guide", and the "Complete Rainbow Guide to OS9", both by Dale L.Puckett.

Not only do these books simplify the learning of Basic09, it has helped me to understand a lot more about OS9. I cannot expound the virtues of these books strongly enough. Why the hierarchy of Radio Shack allowed the publication of the OS9 Manuals in their present form (the BASIC09 manual is better) is a mystery. Surely, the supply of readable manuals such as the one by Puckett could only increase the sales of OS9 and associated software.

Basic09, is a well structured language, that strips BASIC of a lot of cumbersome points and adds the readability of PASCAL. There are many commands in Basic09 that anyone familiar with BASIC would recognise. But there are a lot more that would be new to the average programmer. With a lot more variations on the IF THEN ELSE statements and several ON ERROR functions, which RS BASIC sadly lacks.

The Editor of Basic09 includes automatic error checking while entering your program. As you enter each line (line numbers are not required) Basic09 immediatly compiles it into I Code (Intermediate Code) and indicates any errors and their location. Once you have written and debugged your program, use the PACK command to do a final compilation which cannot be decompiled (not easily anyway), listed or even edited. This PACKed version, not only provides security, I believe it to be just as fast as any assembly language version. By using Basic09, you have the advantage of writing your programs in a HIGHLY Structured Language that is almost plain english.

Just for interest, try LISTing or DUMPing a program that has been PACKed: How did you go? Try using OS9's Editor and DEBUG modes? That didn't work either. So unless you provide someone with the Source, they won't be able to modify the Procedure.

Don't forget: if you have any items of interest, drop me a line. My address is:

Fred Bisseling
P.O. Box 770,
COOMA 2630

RECOMENDED READING

Rainbow
Oct.83 Page6. NEW 64K CoCo (OS-9).
Rainbow Nov.83 Page15. OS-9 POWER
Rainbow Jan.84 Page41. Coming To Life With OS-9.
Rainbow Mar.84 Page43. POTPOURRI of OS-9.
Rainbow Jun.84 Page26. ONE DISK IS BETTER. (this article is on how to install the BOOT Files onto the SYSTEM disk. Very useful to those who don't own a CoCo2.) See also Aust. R'bow Apr. 1985 Page57. GET A 'BOOT' OUT OF OS9.
HOT COCO Aug.84 Page19. BASIC09. (describes Basic09 and how to place it on the SYSTEM disk.)
Rainbow Aug.84 Page84. OPERATING SYSTEMS.
CoCo Aug, Sept, Nov, Dec. 1984 and Feb.85 articles on OS9 by Bob T.
Rainbow Feb.85 Page85. 128K THE EASY WAY.
Rainbow Apr.85 Pages57&58. GET A BOOT OUT OF OS9 and LISTFILE.

■

continued from previous page

```
629 'wait
630 FORT=1TO1000:NEXT:RETURN
1000 PP$="":FORII=1TOW:A=1:P=0:F
ORJJ=1TO7:P=P+A*VAL(MID$(P$(II),
JJ,1)):A=A*2:NEXTJJ:PP$=PP$+CHR$
(128+P):NEXTII
1001 IFH=14THENPP$=PP$+" ":FORII
=1TOW:A=1:P=0:FORJJ=8TO14:P=P+A*
VAL(MID$(P$(II),JJ,1)):A=A*2:NEX
TJJ:PP$=PP$+CHR$(128+P):NEXTII
1002 RETURN
1100 CLS:PRINT"HERE ARE YOUR SHA
PE NAMES:-":PRINT
1101 NN=0:OPEN"I",#1,"PATTERNS.D
AT"
1102 IFEOF(1)THEN1110 ELSE NN=NN
+1
1103 INPUT#1,PN$:INPUT#1,P$:PRIN
TPN$+STRING$(16-LEN(PN$)," ");
1104 GOTO1102
1109 'patterns file details
1110 CLOSE#1:PRINT:RETURN
1200 IFINSTR(PP$," ")THENH=14ELS
EH=7
1201 IFH=14THENW=(LEN(PP$)-1)/2
ELSEW=LEN(PP$)
1202 DIMP$(W):FORII=1TOW:C=ASC(M
ID$(PP$,II,1))-128:C$="0000000":
A=1
1203 FORJJ=1TO7:FL=C AND A:IFFL<
>0THENMID$(C$,JJ,1)="1"
1204 A=2*A:NEXTJJ:P$(II)=C$:IFH=
7THEN1210
1205 C=ASC(MID$(PP$,W+1+II,1))-1
28:C$="0000000":A=1:FORJJ=1TO7:F
L=C AND A:IFFL<>0THENMID$(C$,JJ,
1)="1"
1206 A=2*A:NEXTJJ:P$(II)=P$(II)+
C$
1210 NEXTII:RETURN
```

■

This is a program that will give you a 64 character screen with 80 size writing. DO NOT RESET... it will crash.

```
1 '***WIDTH 64 SCREEN ***
10 POKE 57414,17
20 POKE 63052,64
30 POKE 63105,64
40 POKE 63112,44
50 POKE 63113,0
60 POKE 63601,128
70 POKE 63605,43
80 POKE 63606,128
90 WIDTH 64
```

■

A WEEKEND with the COCO 3

ARTRICLE

by Barry Sidebottom

WITH ALL THE raves & 'expert' opinions about the new COCO 3 I thought you might be interested in a "layman's" views. I am not a computer programmer by trade & my only experience & knowledge is that which I've gained since buying my grey case CoCo.

I will admit however, that I am totally hooked on this hobby & have sunk quite a deal of money into it. I currently have a 16K ECB upgraded to 64K, twin disk drives (with TRS DOS & RDOS), a shuttle 300 modem, and an AMUST 80DT printer.

So, MANY hours were spent that weekend having a look at the new model & checking compatibility with programs I already possess. I realise there must be a lot of others out there who, like me are thinking about upgrading to the new model.

Firstly, let me say that the new high resolution screens are absolutely fantastic to program on. Eight standard color backgrounds with three standard foreground colors gives a lot of room for personal choice - I found white on dark blue very easy to understand.

And the new high resolution graphics screens are totally addictive. To see screens & colors appearing that I had only previously been able to see when forced to watch my workmates gloat about their Commodores was very satisfying!

I also found that when on the standard 32 column screen that the characters have been very discreetly improved. But boy what a great improvement! I was constantly thinking how much better they looked, and how much clearer the screen appeared to be.

It also appeared that in many little ways the extra speed of the new model showed through. DIR, MERGE, RENUM, cold start POKE all seemed much faster.

The LONG hours spent, both day & night (& morning), working with the manual & computer to write the demo program were good & very reminiscent of when I first got my grey case. I finally finished late Sunday leaving me Sunday night (& Monday morning)!

Then came the crunch! What a disappointment to find that although I had a 76% compatibility with applications software, the 24% that didn't fit the bill represented the greatest outlay in cash & my most used programs. These included the VIP range of software.

In the games department we had a better result with 81.5% compatability. The main disappointment here was that the best flight simulators-Worlds of Flight & P51 Mustang-are in the non-compatibility group. (I've included a full list of all the software at the bottom of the list.)

This left me with an overall result of 78.5% compatibility and 21.25% incompatibility. Not bad as figures but not so good when you look at the actual programs that are incompatible.

I would also like to say here that back in 1984 when I upgraded my CoCo 1 to 64K I felt totally cheated & conned to find that in fact I could only access 32K unless I was a programmer (as these are the majority of the people adept at ML).

I am also becoming increasingly concerned about the downfall in the manufacturing quality of the Color Computers. When I first became involved (1983) we were able to brag about the extremely low number of breakdowns & faults in the CoCo.

So, after a very mixed (but still enjoyable) weekend I am now in a dilemma. The new model is great & undoubtedly the best value around for a new buyer. But to upgrade when so much has already been invested, well I'm not so sure!

I hope this can be of some benefit to anyone who, like me, has been reading about the new model and was wondering whether the time is right to upgrade, but isn't able to do a test run first. As for me I'm lucky that I could!

Test results: Compatible Applications: Color Quaver, Lyra, Filmaster, Clone, Disk Manager, Superdup (supplied with Graphicom), Worksaver 2, Disk Edtasm (32 width mode only), Super Back Up, Musica (disk), Macro 80C, Telewriter 64, Calligrapher, Cookbook, Heart Lung Circulation,

Incompatible Applications: Zap (Factory Programming), Disk Utility, VIP Database, VIP Terminal, VIP Writer, COCO MAX 2 (& I would assume 1)

Compatible Games & Simulations: Backgammon, Baseball, Berserk, Invader, Lancer, Lemans, Megabug, Poltergeist, Wildcat, Pinball, Zaksund, MsGobble, Double Back, Invasion, Galax Attax, Whirly Bird, Dunkey Monkey, Tennis, Defense, 3D Bricks, Colorzap, Seadragon, Gobbler, Junior's Revenge

Incompatible Games & Simulations: Popcorn, Rail Runner, Quasarcommand, 8 Ball, Storm Fighters, Zaxxon, Decathlon, P51 Mustang, Worlds of Flight, Hall of the King.

Rom Packs, Compatible: Chess, Color Cubes, Edtasm +

Rom Packs, Incompatible: Audio Spectrum Analyser.

Special Thanks to Jack Smit & Mark from Smits Appliances Sunbury. (Sunbury Tandy agent).

Footnote- The results & comments above were achieved in 1 weekend of hectic programming with the CoCo 3 by, as stated, a relative layman. They are presented to all the others out there who aren't professional programmers and might be wondering.

Barry Sidebottom. SUNBURY VIC. ph. 744-6281

■

This one only takes one to play.

SOLO SCRABBLE

32K ECB
GAME

by Bob Delbourgo

SOLO SCRAMBLE ('SCRABOLO') is a game for one player. The aim is to score the maximum number of points with your hand randomly dealt (up to seven letters) by the CoCo. The normal rules apply:

* Each letter has a point value and your entry must consist of joined-up letters along vertical or horizontal lines.

* There are a total of 100 letters including two blanks.

Key in your letter in your hand from the upper right to insert it on the board and press ENTER when satisfied with your entry, otherwise press CLEAR to erase.

You may change all your letters (or else none) by pressing '?', incurring a 10 point penalty.

Scoring is automatic.

You may save your game in any round by keying '>'.

Alternatively, you may load a previous game by pressing '<'.

The Listing:

```
0 GOTO9
1 '********** SCRABOLO **********
   ***** BOB DELBOURGO **********
3 SAVE"205H:3":END'1
9 GOTO20000
10 CLS0:CLEAR1900:DIM A$(100):R=R
ND(-TIMER):POKE359,57:SCREEN0,1
20 A$="scrabolo":FORI=1TO8:PRINT
@32*I+47,MID$(A$,I,1);:SOUND20*I
,1:NEXTI
21 PRINT@208,"OB";:PRINT@295,"DE
LBOURG";
30 PRINT@354,"SAVE/LOAD:  <T>APE
 OR <D>ISK?"
31 I$=INKEY$:IFI$="T"THENDV=-1EL
SEIFI$="D"THENDV=1 ELSE31
32 PRINT@354,"DIRECTIONS/INSTRUC
TIONS <Y/N>?";
33 I$=INKEY$:IFI$="Y"THEN50ELSEI
FI$="N"THEN100ELSE33
49 'directions
50 CLS:PRINT@1,A$" IS JUST SOLO
SCRABBLE":PRINT:PRINT"THE AIM IS
 TO SCORE THE MAXIMUM NUMBER OF
POINTS WITH YOUR HAND RANDOMLY D
EALT (UP TO 7 LETTERS)BY COCO. T
HE NORMAL RULES APPLY:"
51 PRINT"EACH LETTER HAS A POINT
 VALUE  AND YOUR ENTRY MUST CON
SIST OF  JOINED-UP LETTERS ALONG
 VERTICALor HORIZONTAL LINES."
52 PRINT:PRINT"THERE IS A TOTAL
OF 100 LETTERS AND THIS INCLUDES
 TWO BLANKS  DESIGNATED BY @ I
N THE GAME."
54 GOSUB1050
55 CLS:PRINTCHR$(223)" DOUBLES T
HE LETTER VALUE":PRINTCHR$(175)"
 TRIPLES THE LETTER VALUE":PRINT
CHR$(239)" DOUBLES THE WORD VALU
E  AND":PRINTCHR$(191)" TRIPLES
THE WORD VALUE."
56 PRINT"THIS APPLIES TO ALL NEW
 WORDS  THAT ARE REALISED BY YO
UR ENTRY.":PRINT"KEY IN THE LETT
ER IN YOUR HAND  AT THE UPPER RI
GHT TO INSERT IT ON THE BOARD."
57 PRINT"PRESS ENTER WHEN SATISF
IED WITH YOUR ENTRY, OR CLEAR TO
 ERASE.  YOU MAY CHANGE ALL YOUR
 LETTERS (OR ELSE NONE) BY PRESS
ING <?>, INCURRING A 10-POINT PE
NALTY!"
58 GOSUB1050
60 CLS:PRINT"SCORING IN SCRABOLO
 IS AUTOMATIC":PRINT"YOU MAY SAV
E YOUR GAME IN ANY  ROUND BY KE
YING >.":PRINT"ALTERNATIVELY, YO
U MAY LOAD A  PREVIOUS GAME BY
PRESSING <."
61 PRINT:PRINT"REMEMBER TO HAVE
YOUR DISK OR  CASSETTE AT THE R
EADY AS SOON ASYOU ENTER THE FIL
ENAME."
62 PRINT:PRINT"THE LETTERS, THEI
R NUMBER AND  THEIR POINT VALUE
S APPEAR IN 2  COLUMNS ON THE BO
ARD RIGHT IN  HEXADECIMAL NOTAT
ION."
63 GOSUB1050
99 'initialize
100 A$="A1A1A1A1A1A1A1A1A1B3B3C3
C3D2D2D2D2E1E1E1E1E1E1E1E1E1E1E1
E1F4F4G2G2G2H4H4I1I1I1I1I1I1I1I1
I1J8K5L1L1L1L1M3M3N1N1N1N1N1O1
O1O1O1O1O1O1P3P3QAR1R1R1R1R1R1
S1S1S1S1T1T1T1T1T1T1U1U1U1U1V4V4
W4W4X8Y4Y4ZA@0@0"
101 B$=STRING$(225,32):P$=STRING
$(225,"0"):SP$=" "+CHR$(175)+CHR
$(191)+CHR$(239)+CHR$(223)
110 CLS2:FORI=32TO480STEP32:PRIN
T@I,STRING$(15,32);:NEXTI
111 FORI=1TO8:READD:V=INT(D/32):
H=D-32*V:POKE1024+D,191:MID$(B$,
H+15*V-14,1)=CHR$(191):NEXTI
112 FORI=1TO12:READD:V=INT(D/32)
:H=D-32*V:POKE1024+D,175:MID$(B$
,H+15*V-14,1)=CHR$(175):NEXTI
113 FORI=1TO24:READD:V=INT(D/32)
:H=D-32*V:POKE1024+D,223:MID$(B$
,H+15*V-14,1)=CHR$(223):NEXTI
114 FORI=1TO17:READD:V=INT(D/32)
:H=D-32*V:POKE1024+D,239:MID$(B$
,H+15*V-14,1)=CHR$(239):NEXTI
115 FORI=0TO12:READD$:PRINT@80+3
2*I,D$;:NEXTI:FORI=0TO12:READD$:
PRINT@84+32*I,D$;:NEXTI
116 PRINT@88,"KEY IN ";:PRINT@12
0,"LETTERS";:PRINT@152,"IN HAND"
;:PRINT@184,"& enter";
117 PRINT@248,"'clear'";:PRINT@2
80,"DELETES";:PRINT@312,"ENTRY.
";:PRINT@344,"'?' TO ";:PRINT@37
6,"CHANGE ";:PRINT@408,"LETTERS"
;
118 PRINT@440,"'>'SAVE";:PRINT@4
72,"'<'LOAD";:PRINT@1,"scrabolo"
;:PRINT@10,"hand=";
119 'shuffle letters
120 PRINT@16,"SHUFFLING..";:FORK
=1TO200
121 I=RND(100):J=RND(100):IFI=J
THEN121
122 I$=MID$(A$,2*I-1,2):J$=MID$(
A$,2*J-1,2)
123 MID$(A$,2*I-1,2)=J$:MID$(A$,
2*J-1,2)=I$:NEXTK
124 'first deal
125 H$=LEFT$(A$,14):A$=MID$(A$,1
5):GOSUB1000:CI=8:CJ=8:N=0
129 'round N, start of entries
130 N=N+1:PRINT@1,"round";N;:A$(
N)=""
131 PRINT@23,CHR$(159);"#=";CHR$
(159);:PRINT@27,MID$(STR$(PT),2-
INSTR(STR$(PT),"-"));:IFA$=""AND
H$=""THEN500
135 C$=INKEY$:PO=CI+15*CJ-15:PE=
1023+CI+32*CJ:CH=PEEK(PE):POKE P
E,207:FORT=1TO30:NEXT:POKE PE,CH
:FORT=1TO10:NEXTT:IFC$=""THEN135
136 IFC$=CHR$(8)THENCI=CI-1:IFCI
=0THENSOUND1,1:CI=1
137 IFC$=CHR$(9)THENCI=CI+1:IFCI
=16THENSOUND1,1:CI=15
138 IFC$=CHR$(10)THENCJ=CJ+1:IFC
J=16THENSOUND1,1:CJ=15
139 IFC$=CHR$(94)THENCJ=CJ-1:IFC
J=0THENSOUND1,1:CJ=1
140 IFC$=CHR$(13)THENGOSUB200:GO
TO130
141 IFC$=CHR$(12)THENGOSUB250:GO
TO135
142 IFC$=">"THENGOSUB300:GOTO135
143 IFC$="<"THENGOSUB350:GOSUB10
00:GOTO130
144 IFC$="?"THENA$(N)="....":PT=
PT-10:PRINT@27,MID$(STR$(PT),2-I
```

```
NSTR(STR$(PT),"-"));:GOSUB1020:G
OTO130
145 IFINSTR(H$,C$)=0ORCH<27OR(CH
>63ANDCH<91)THEN135
146 IFC$<"@"ORC$>"Z"THEN135
147 D$="":FORI=1TOLEN(H$)/2:D$=D
$+MID$(H$,2*I-1,1):NEXTI:I=INSTR
(1,D$,C$):I=2*I-1:A$(N)=A$(N)+CH
R$(CH)+CHR$(PO+30)+MID$(H$,I,2):
H$=LEFT$(H$,I-1)+MID$(H$,I+2):GO
SUB1000
148 C=ASC(C$):C=C-64:POKEPE,C
150 GOTO135
199 'entry
200 IFLEFT$(A$(N),1)="."THENSOUN
D1,10:RETURN
201 GOSUB1030:GOSUB2000:RETURN
249 'clear and rentry
250 FORI=1TOLEN(A$(N))/4:CH$=MID
$(A$(N),4*I-3,1):P=ASC(MID$(A$(N
),4*I-2,1))-30:IFCH$=CHR$(96)THE
NCH$=" "
251 V=1+INT((P-1)/15):H=P+15-15*
V:PRINT@H+32*V-1,CH$;:SOUND30*I,
1
252 H$=H$+MID$(A$(N),4*I-1,2):NE
XTI
253 A$(N)="":GOSUB1000:RETURN
299 'save
300 IFN=1THENSOUND200,10:RETURN
ELSEPOKE359,126:SCREEN0,0:PRINT@
48,"save:";:INPUTF$:IFDV=1THENF$
=F$+"/DAT"
301 HH$="":IFA$(N)=""THENHH$=H$:
GOTO302 ELSEFORJ=1TOLEN(A$(N))/4
:HH$=HH$+MID$(A$(N),4*J-1,2):NEX
TJ:HH$=HH$+H$
302 OPEN"O",#DV,F$:PRINT#DV,N-1,
PT:IFHH$=""THENHH$="@@"
303 FORJ=1TON-1:GOSUB1060:PRINT#
DV,K$:NEXTJ:IFA$=""THENA$="@@@@"
304 PRINT#DV,A$:PRINT#DV,B$:PRIN
T#DV,P$:PRINT#DV,HH$
305 CLOSE#DV:GOSUB1040:POKE359,5
7:SCREEN0,1:RETURN
349 'load
350 POKE359,126:SCREEN0,0:PRINT@
48,"load:";:INPUTF$:IFDV=1THENF$
=F$+"/DAT"
351 OPEN"I",#DV,F$
352 INPUT#DV,N,PT
353 FORJ=1TON:INPUT#DV,K$:GOSUB1
065:NEXTJ:INPUT#DV,A$:INPUT#DV,B
$:INPUT#DV,P$:INPUT#DV,H$
354 CLOSE#DV:GOSUB1040:IFA$="@@@
@"THENA$=""
355 IFHH$="@@"ORH$="@@"THENH$=""
356 FORI=1TO15:PRINT@32*I,MID$(B
$,15*I-14,15);:NEXTI:POKE359,57:
SCREEN0,1:RETURN
499 'gameend
500 PRINT@48,"SAVE GAME <Y/N>?";
501 I$=INKEY$:IFI$=""THEN501
502 IFI$="N"THEN510
503 IFI$="Y"THEN505
504 GOTO501
505 GOSUB1040
506 GOSUB300:GOTO500
510 END
999 'display hand
1000 IFLEN(H$)=0THEND$=STRING$(7
,32) ELSED$="":FORI=1TOLEN(H$)/2
:M=ASC(MID$(H$,2*I-1,1)):M=M+32:
D$=D$+CHR$(M):NEXTI:IFLEN(D$)<7T
HEND$=D$+STRING$(7-LEN(D$),32)
1001 PRINT@16,D$;:RETURN
1019 'change all letters
1020 IF LEN(H$)<14THENSOUND1,10:
RETURN
1021 FORI=1TO7:R=RND(LEN(A$)/2)-
1
1022 A$=LEFT$(A$,2*R)+MID$(H$,2*
I-1,2)+MID$(A$,2*R+1):SOUNDR+1,1
:NEXTI
1023 H$=LEFT$(A$,14):A$=MID$(A$,
15):GOSUB1000:RETURN
1029 'update hand and board
1030 IFLEN(H$)=14THENSOUND10,10:
RETURN
1031 FORI=1TOLEN(A$(N))/4:P=ASC(
MID$(A$(N),4*I-2,1))-30:V=1+INT(
(P-1)/15):H=P+15-15*V:PRINT@H+32
*V-1,MID$(A$(N),4*I-1,1);:SOUNDH
+154*V,1:MID$(B$,P,1)=MID$(A$(N),
4*I-1,1):MID$(P$,P,1)=MID$(A$(N)
,4*I,1):NEXTI
1032 L=14-LEN(H$):H$=H$+LEFT$(A$
,L):A$=MID$(A$,L+1):GOSUB1000:RE
TURN
1034 'bonus
1035 PRINT@48,"bonus of 50!!!";:
FORL=10TO250STEP10:SOUNDL,1:NEXT
L:PT=PT+50
1039 'cleanup
1040 FORJ=1072TO1087:POKEJ,159:N
EXTJ:RETURN
1049 'freeze
1050 PRINT@485,"any key to conti
nue";
1051 IFINKEY$=""THEN1051
1052 RETURN
1059 'output hex code
1060 K$="":FORK=1TOLEN(A$(J)):KK
=ASC(MID$(A$(J),K,1)):K1=INT(KK/
16):K2=KK-16*K1:K$=K$+HEX$(K1)+H
EX$(K2):NEXTK:RETURN
1064 'input hex decode
1065 A$(J)="":FORK=1TOLEN(K$)/2:
KK=VAL("&H"+MID$(K$,2*K-1,2)):A$
(J)=A$(J)+CHR$(KK):NEXTK:RETURN
1999 'vert or horiz  entry?
2000 SC=0:PRINT@48,"scoring..";:
P=ASC(MID$(A$(N),2,1))-30
2001 IFLEN(A$(N))=4THEN2010
2002 IFABS(P+30-ASC(MID$(A$(N),6
,1)))<15THEN2020
2009 'find position p1 of first
letter in vert. word, p2 of last
 letter
2010 P=P-15:IFP<1THENP=P+15:P1=P
:GOTO2015
2011 IFINSTR(SP$,MID$(B$,P,1))TH
EN2012ELSE2010
2012 P1=P+15:P=P1
2015 P=P+15:IFP>225THENP2=P-15:G
OTO2018
2016 IFINSTR(SP$,MID$(B$,P,1))TH
EN2017ELSE2015
2017 P2=P-15
2018 GOSUB2100:GOSUB1040:RETURN
2019 'find p1 and p2 for horiz.
word
2020 P=P-1:IFINT((P-1)/15)<>INT(
P/15)THENP=P+1:P1=P:GOTO2025
2021 IFINSTR(SP$,MID$(B$,P,1))TH
EN2022ELSE2020
2022 P1=P+1:P=P1
2025 P=P+1:IF15*INT((P-1)/15)=P-
1 THENP2=P-1:GOTO2028
2026 IFINSTR(SP$,MID$(B$,P,1))TH
EN2027ELSE2025
2027 P2=P-1
2028 GOSUB2200:GOSUB1040:RETURN
2099 'vert. word score
2100 IFP1=P2 THEN2110ELSEFORP=P1
 TOP2 STEP15:SC=SC+VAL("&H"+MID$
(P$,P,1)):NEXTP
2101 FORK=1TOLEN(A$(N))/4:IFMID$
(A$(N),4*K-3,1)=CHR$(223)THENSC=
SC+VAL("&H"+MID$(A$(N),4*K,1))EL
SEIFMID$(A$(N),4*K-3,1)=CHR$(175
)THENSC=SC+2*VAL("&H"+MID$(A$(N)
,4*K,1))
2102 NEXTK
2103 FORK=1TOLEN(A$(N))/4:IFMID$
(A$(N),4*K-3,1)=CHR$(239)THENSC=
2*SC ELSEIFMID$(A$(N),4*K-3,1)=C
HR$(191)THENSC=3*SC
2104 NEXTK:PT=PT+SC:IFLEN(A$(N))
=28THENGOSUB1035
2110 FORP=1TOLEN(A$(N))/4:SC=0:I
=ASC(MID$(A$(N),4*P-2,1))-30
2111 I=I-1:IFINT((I-1)/15)<>INT(
I/15)THENI=I+1:P1=I:GOTO2115
2112 IFINSTR(SP$,MID$(B$,I,1))TH
EN2113ELSE2111
2113 P1=I+1:I=P1
2115 I=I+1:IF15*INT((I-1)/15)=I-
1 THENP2=I-1:IFP1=P2 THEN2124ELS
E2120
2116 IFINSTR(SP$,MID$(B$,I,1))TH
EN2117ELSE2115
2117 P2=I-1:IFP1=P2 THEN2124
2119 'horiz. addup
2120 FORK=P1 TO P2:SC=SC+VAL("&H
"+MID$(P$,K,1)):NEXTK
2121 IFMID$(A$(N),4*P-3,1)=CHR$(
223)THENSC=SC+VAL("&H"+MID$(A$(N
),4*P,1))ELSEIFMID$(A$(N),4*P-3,
1)=CHR$(175)THENSC=SC+2*VAL("&H"
+MID$(A$(N),4*P,1))
2122 IFMID$(A$(N),4*P-3,1)=CHR$(
239)THENSC=2*SC ELSEIFMID$(A$(N)
,4*P-3,1)=CHR$(191)THENSC=3*SC
2123 PT=PT+SC
2124 NEXTP:RETURN
2199 'horiz. word score
2200 FORP=P1 TOP2:SC=SC+VAL("&H"
+MID$(P$,P,1)):NEXTP
2201 FORK=1TOLEN(A$(N))/4:IFMID$
(A$(N),4*K-3,1)=CHR$(223)THENSC=
SC+VAL("&H"+MID$(A$(N),4*K,1))EL
SEIFMID$(A$(N),4*K-3,1)=CHR$(175
)THENSC=SC+2*VAL("&H"+MID$(A$(N)
,4*K,1))
2202 NEXTK
2203 FORK=1TOLEN(A$(N))/4:IFMID$
(A$(N),4*K-3,1)=CHR$(239)THENSC=
2*SC ELSEIFMID$(A$(N),4*K-3,1)=C
HR$(191)THENSC=3*SC
2204 NEXTK:PT=PT+SC:IFLEN(A$(N))
=28THENGOSUB1035
2210 FORP=1TOLEN(A$(N))/4:SC=0:I
=ASC(MID$(A$(N),4*P-2,1))-30
2211 I=I-15:IFI<1THENI=I+15:P1=I
:GOTO2215
2212 IFINSTR(SP$,MID$(B$,I,1))TH
EN2213ELSE2211
2213 P1=I+15:I=P1
2215 I=I+15:IFI>225THENI=I-15:P2
=I:IFP1=P2 THEN2224ELSE2220
2216 IFINSTR(SP$,MID$(B$,I,1))TH
EN2217ELSE2215
2217 P2=I-15:IFP1=P2 THEN2224
2219 'vert. addup
2220 FORK=P1 TOP2 STEP15:SC=SC+V
```

continued on page 47

This one does everything but pay the bills.

BILLS, DATA & DATES

32K ECB
BUSINESS

by Glenn Blomfield

BILLS, DATA & DATES IS for use with a savings A/C to pay all regular bills, ie insurance, car registration, power, telephone, etc.

When all the data has been entered, the computer will then show all savings per week/month and also shows the balance of the bills A/C.

The program is all subroutined, as shown below:

```
  5- 100 Numerical variables
105- 200 Main program
205- 285 Date check
305- 375 How you are payed
405- 465 Paydays
475- 495 Continue/Menu
505- 535 Next pay
605- 660 Bills sort by date
705- 780 Find day of week
805- 850 Print day of week
905- 970 Setting up bills file
1000-1100 Bills data
1105-1200 Bills total
1205-1390 Bills savings
1405-1470 CSAVE bills file
1505-1575 CLOAD bills file
1605-1700 Add/Delete from bills
1705-1740 Instructions
```

To go through each subroutine more clearly ...

'Date Check'- Checks the date. This is only 6 characters long and has to be a valid date, which is then converted to two strings: [YYMMDD] and [DDMMYY].

'How you are paid'- Will let you select your pay period and then set three variables:

1. Pay period in days,
2. Pays per year, and
3. Pay period

'Paydays'- Will take your next date of pay, ask you your pay period (time between pays) and then display day & date of all pays for the year.

'Next Pay'- Will determine if 'this year' is a leap year, add the pay period and adjust 'DDMMYY' variables.

'Entries Correct'- Will continue the listing if answered 'Y' or else will list the incorrect bill for correction.

'Bill sort by Date'- !! NOTE ERROR !! - PRINT@400 should be PRINT@480 in line 605; this is a bubble sort and can take some time if there is a large change in the date of the data.

'Find day of week'- is based on a reference date of 2/1/84 and compares this with the date of the day you wish to know and will increment the reference date until it equals the date in question. There IS a shorter way of doing this, but this way was the quickest to create.

'Print day of week'- Works with the above.

'Setting up bills file'- Shows format for bills data and lists all bills.

'Bills data'- This checks the data as it is entered and formats it into one string for each bill.

'Bills total'- will total all the bills, then divide them by the pay period to give you an amount to be saved each payday.

'Bills savings'- This is the guts of the program. Anyone can add up their annual & regular bills and divide them by the number of pays they get. This program goes one step further and tells you how much you need in the bank at any date to meet the bills when they fall due.

'CSAVE bills file'- Will save the bills data to tape only.

'CLOAD bills data'- will load the bills data from tape only.

'Add/Delete from bills file'- as the name implies.

'Instructions'- Very rough outline of what the program does.

The program makes heavy use of strings and string functions, PRINTUSING functions and error checking of input data.

I've used this program over the last three years and have removed all the bugs I could find. It was my first major program I wrote and it taught me a lot about BASIC and the CoCo.

The program was originally screen orientated and then changed to the printer. The screen display is not the best as it is set for 80 columns and should be written as a separate routine.

I'll gladly answer any questions anyone may have. Please send a Self Addressed envelope to:

Glenn Blomfield,
21 Flower St,
Bulls,
NZ

The Listing:

```
0 GOTO5
1 '**** BILLS, DATA & DATES ****
    ****** GLENN BLOMFIELD *****
3 SAVE"183:3":END
5 REM NUMERICAL VARIBLES
10 D=0:M=0:Y=84:X=0:C=0:H=0:I=0
15 SB=0:SD=0:B=0:PP=0:PA=0:E=0
20 F=0:G=0:J=0:K=0:N=0:O=0:S=0
25 RD=02:RM=01:RY=84:T=0
55 REM STRING VARIBLES
60 H$="DATE OF NEXT BILL"
65 DS$="DATE AS STRING<YYMMDD>"
70 SD$="DATE AS STRING<DDMMYY>"
75 SH$="DATE OF NEXT BILL"
80 DA$="DATE INPUT"
85 N$="INPUT":S$="PAY PERIOD"
90 A$="INKEY":RF$="DAY OF WEEK"
95 T$="TEMP":TD$="TODAYS DATE"
100 NP$="DATE OF NEXT PAY":G$="B
ILL DATE CHANGE"
105 CLS:REM MAIN PROGRAM
110 CLEAR 1500
115 DIM L(12),B$(50)
120 DATA 31,28,31,30,31,30,31,31
,30,31,30,31
125 FOR X=1 TO 12:READ L(X):NEXT
 X
130 PRINT"   BILLS,DATA & DATES
PROGRAM"
135 PRINT:PRINT"     SELECT OPTI
ONS 1-9":PRINT
140 PRINT" 1. INSTRUCTIONS"
145 PRINT" 2. SET-UP A NEW BILLS
 FILE"
150 PRINT" 3. LIST/CHANGE BILLS
FILE"
155 PRINT" 4. CLOAD BILLS FILE"
160 PRINT" 5. CSAVE BILLS FILE"
165 PRINT" 6. PRINT PAYDAYS & DA
Y OF WEEK"
170 PRINT" 7. CALCULATE AMOUNT F
OR BILLS"
175 PRINT" 8. FIND DAY OF WEEK "
180 PRINT" 9. ADD TO OR DELETE F
ROM BILLS     FILE"
185 INPUT"            SELECT 1-9
";N
190 IF N<1 OR N>9 THEN 200
```

```
195 ON N GOSUB 1705,905,945,1505
,1405,405,1105,805,1605
200 CLS:GOTO 120
205 REM DATE CHECK
210 IF LEN(DA$)<>6 THEN 280
215 D=VAL(MID$(DA$,1,2)):F=0
220 M=VAL(MID$(DA$,3,2))
225 Y=VAL(MID$(DA$,5,2))
230 IF D>31 OR M>12 OR Y>99 THEN
 280
235 IF D<1 OR M<1 OR Y<84 THEN 2
80
240 IF M=2 AND D>29 AND Y/4=INT(
Y/4) THEN 280
245 IF M=2 AND D>28 AND Y/4<>INT
(Y/4) THEN 280
250 Y$=MID$(STR$(Y),2,2)
255 M$=RIGHT$(STR$(M+100),2)
260 D$=RIGHT$(STR$(D+100),2)
265 DS$=Y$+M$+D$
270 SD$=D$+M$+Y$
275 RETURN
280 PRINT"ERROR IN DATE"
285 F=1:RETURN
305 CLS:PRINT"  HOW ARE YOU PAYE
D   1-5":PRINT
310 PRINT TAB(8)"1. WEEKLY"
315 PRINT TAB(8)"2. FORTNIGHTLY"
320 PRINT TAB(8)"3. EVERY 28 DAY
S"
325 PRINT TAB(8)"4. EACH CALANDE
R MONTH"
330 PRINT TAB(8)"5. OTHER"
335 INPUT S
340 IF S<1 OR S>5 THEN PRINT"SEL
ECTION ERROR":GOTO 335
345 IF S=5 THEN INPUT"INPUT THE
NUMBER OF DAYS BETWEEN PAYS";PP:
PA=INT(365/PP)
350 S$="EVERY"+STR$(PP)+" DAYS"
355 IF S=1 THEN PP=7:PA=52:S$="W
EEKLY"
360 IF S=2 THEN PP=14:PA=26:S$="
FORTNIGHILY"
365 IF S=3 THEN PP=28:PA=13:S$="
EVERY 28 DAYS"
370 IF S=4 THEN PA=12:S$="EACH C
ALANDER MONTH"
375 RETURN
405 CLS:REM PAYDAYS
410 INPUT " INPUT NEXT PAYDAY DA
TE                    -USE<DDMMYY
>";DA$
415 GOSUB 210
420 IF F=1 THEN 410 ELSE PRINT
425 GOSUB 305:CLS
430 FOR X=1 TO PA
435 GOSUB 505:GOSUB 705
440 PRINT USING"%       % ### ##
# ###";RF$,D,M,Y
445 IF INT(X/10)=X/10 THEN GOSUB
 475:CLS
450 IF F=1 THEN RETURN
455 NEXT X
460 PRINT:PRINT:GOSUB 480
465 RETURN
475 PRINT:PRINT:PRINT"PRESS <C>
TO CONTINUE"
480 PRINT"PRESS <M> TO RETURN TO
 MENU"
485 A$=INKEY$:IF A$="" THEN 485
490 IF A$="C" THEN F=0: RETURN
495 IF A$="M" THEN F=1 ELSE 485:
RETURN
505 E=L(2):REM NEXT PAY
510 IF INT(Y/4)=Y/4 THEN L(2)=29
515 IF PA=12 THEN PP=L(M)
520 D=D+PP
525 IF D>L(M) THEN D=D-L(M):M=M+
1
530 IF M>12 THEN M=M-12:Y=Y+1
535 L(2)=E:RETURN
550 INPUT"ARE ALL ENTRIES CORREC
T -<Y/N>";N$
555 IF N$="Y" THEN RETURN
560 IF N$="N" THEN 565 ELSE 550
565 INPUT"WHICH ENTRY NUMBER IS
INCORRECT";N
570 IF N>A OR N<1 THEN 565
575 CLS:PRINT@256,STRING$(32,45)
580 PRINT@288,"   BILL DATA TO B
E CHANGED
585 PRINT@352," BILL NAME -  ";M
ID$(B$(N),1,12)
590 PRINT@384," DATE DUE  -  ";M
ID$(B$(N),14,6)
595 PRINT@416,"  AMOUNT   - ";MI
D$(B$(N),22,7)
600 X=N:GOSUB 1010:PRINT@256,STR
ING$(160,32):PRINT@224,;:RETURN
605 PRINT@400,"BILLS BEING SORTE
D BY DATE"
610 F=0:FOR X=1 TO A-1
615 H$=MID$(B$(X),18,2)+MID$(B$(
X),16,2)+MID$(B$(X),14,2)
620 I$=MID$(B$(X+1),18,2)+MID$(B
$(X+1),16,2)+MID$(B$(X+1),14,2)
625 IF I$<H$ THEN 630 ELSE 650
630 T$=B$(X)
635 B$(X)=B$(X+1)
640 B$(X+1)=T$
645 F=1
650 NEXT X
655 IF F=1 THEN 610
660 RETURN
705 RD=2:RM=1:RY=84:REM FIND DAY
 OF WEEK
710 IF Y=RY THEN 735
715 IF INT(RY/4)=RY/4 THEN RD=RD
-2 ELSE RD=RD-1
720 IF RD=0 THEN RD=7
725 RY=RY+1
730 IF RY=Y THEN 735 ELSE 715
735 E=L(2)
740 IF INT(RY/4)=RY/4 THEN L(2)=
29
745 IF RM < M THEN RD=RD+7 ELSE
760
750 IF RD >L(RM) THEN RD=RD-L(RM
):RM=RM+1
755 GOTO 745
760 IF RD<D THEN RD=RD+7:GOTO 76
0
765 IF RD=D THEN RF$="MONDAY":L(
2)=E:RETURN
770 IF RD>D THEN G=RD-D
775 RF$=MID$("SUNDAY   SATURDAY
FRIDAY   THURSDAY WEDNESDAYTUESD
AY  ",(G-1)*9+1,9)
780 L(2)=E:RETURN
805 CLS:REM PRINTS DAY OF WEEK
810 PRINT"WORKS ON DATES AFTER 0
20184 ONLY"
815 GOSUB 475:IF F=1 THEN RETURN
820 CLS:INPUT"TO FIND DAY OF WEE
K-INPUT DATE  USING <DDMMYY>";DA
$:F=0
825 GOSUB 210
830 IF F=1 THEN 810 ELSE PRINT:P
RINT
835 GOSUB 705
840 PRINT D;M;Y;" IS A ";RF$
845 GOSUB 475
850 IF F=0 THEN 820 ELSE RETURN
905 CLS:A=0:REM SETTING UP BILLS
 FILE
910 PRINT" PLEASE INPUT BILLS US
ING THE    FOLLOWING FORMAT":PRI
NT
915 PRINT"    NAME    DATE DUE
AMOUNT"
920 PRINT" 12 LETTERS  DDMMYY  #
###.##"
925 GOSUB 475:IF F=1 THEN RETURN
:ELSE CLS
930 FOR X=1 TO 50
935 GOSUB 1005
940 NEXT X
945 CLS:IF A=0 THEN PRINT@225,"
    THERE'S NO BILLS FILE":FOR X
=1TO 1000:NEXT X:RETURN
950 FOR X=1 TO A
955 PRINT X;B$(X)
960 IF X/10=INT(X/10) OR X=A THE
N PRINT:GOSUB 550:CLS
965 NEXT X
970 GOSUB 605:RETURN
1005 A=A+1:CLS:REM BILLS DATA
1010 PRINT@0,"TO STOP INPUTTING
BILLS-USE <ZZ>FOR NAME. FOR DATE
 USE <DDMMYY>"
1015 PRINT:PRINT
1020 INPUT" BILL NAME -";N$
1025 IF LEN(N$)>12 THEN PRINT"NA
ME TOO LONG":GOTO 1015
1030 IF LEN(N$)<1 THEN PRINT"ERR
OR IN NAME":GOTO 1015
1035 IF VAL(N$)>10 THEN PRINT"ER
ROR-NUMBER>10 IN NAME":GOTO 1015
1040 IF N$="ZZ" THEN X=50:A=A-1:
RETURN
1045 B$(X)=N$+STRING$(13-LEN(N$)
,32)
1050 INPUT" DATE DUE  -";DA$
1055 GOSUB 210:IF F=1 THEN 1050
1060 B$(X)=B$(X)+DA$+STRING$(1,3
2)
1065 INPUT"  AMOUNT   -";N$
1070 IF LEN(N$)>7 OR LEN(N$)<1 T
HEN PRINT"ERROR IN AMOUNT":GOTO
1065
1075 IF INSTR(N$,".")=0 THEN N$=
N$+".00":GOTO 1095
1080 K=LEN(N$)-INSTR(N$,".")
1085 IF K=1 THEN N$=N$+"0":GOTO
1095
1090 IF K=0 THEN N$=N$+"00":GOTO
 1095
1095 B$(X)=B$(X)+STRING$((7-LEN(
N$)),32)+N$
1100 RETURN
1105 CLS:T=0:REM BILLS TOTAL
1110 IF A=0 THEN PRINT"YOU NEED
TO SET-UP A BILLS FILE OR CLOAD
AN EXISTING BILLS FILE"ELSE 1125
1115 PRINT:PRINT:GOSUB 480
1120 RETURN
1125 INPUT"DO YOU WANT TO CHECK
THE BILLS  FILE FOR ERRORS -Y/N"
;N$
1130 IF N$="N" THEN 1140
1135 IF N$="Y" THEN GOSUB 945 EL
SE 1125
1140 FOR X=1 TO A
1145 T=T+VAL(MID$(B$(X),22,7))
1150 NEXT X
1155 CLS:PRINT USING"TOTAL OF AL
L BILLS =$$#####.##";T
1160 GOSUB 475:CLS
1165 IF S$="" THEN GOSUB 305
1170 B=INT(T/PA*100+.5)/100:CLS
```

```
1175 PRINT USING"YOU NEED TO SET
 ASIDE $$###.##  EACH PAYDAY";B
1180 PRINT:PRINT
1185 PRINT USING"IF YOU ARE NOT
ALEADY SETTING   ASIDE AN AMOUNT
 EACH PAYDAY OR  THAT AMOUNT IS
SMALL COMPARED   WITH $$###.## T
HEN YOU MAY NEED TO PUT ASIDE A
LUMPSUM WHICH    WITH  $$###.##
EACH PAYDAY WILL COVER YOUR  BIL
LS";B;B:PRINT
1190 INPUT"INPUT YOUR SAVINGS FO
R BILLS,IF NIL ENTER <0>";O
1195 IF B<O THEN B=O:GOSUB 475
1200 CLS:IF F=1 THEN RETURN
1205 INPUT"   INPUT TODAYS DATE
                    -USE<DDMMYY>
";DA$
1210 GOSUB 210
1215 IF F=1 THEN 1205
1220 TD$=DS$:PRINT:PRINT
1225 INPUT"   NEXT PAYDAYS DATE
                    -USE<DDMMYY>
";DA$
1230 GOSUB 210
1235 IF F=1 THEN 1225
1240 FOR X=1 TO A
1245 H$=MID$(B$(X),18,2)+MID$(B$
(X),16,2)+MID$(B$(X),14,2)
1250 IF H$ < TD$ THEN 1255 ELSE
1270
1255 H=VAL(H$):H=H+10000:G$=STR$
(H)
1260 MID$(B$(X),18,2)=MID$(G$,2,
2)
1265 NEXT X:IF X=A+1 THEN 1240
1270 X=A:NEXT X:CLS:GOSUB 605:CL
S
1275 INPUT"  DO YOU REQUIRE A PR
INTED COPY   Y/N";N$:CLS
1280 IF N$="Y"THEN J=2 ELSE J=0
1285 INPUT"INPUT BALANCE OF A/C
OR IF YOU  DON'T HAVE AN A/C FOR
 YOUR BILLSTHEN INPUT ZERO";SB
1290 SD=0:CLS:NP$=DS$
1295 I=0:SD=SD+SB:SB=SD:PRINT#-J
,USING"                STARTING BA
LANCE                  $$#####.##";S
B
1300 DA$=MID$(NP$,5,2)+MID$(NP$,
3,2)+MID$(NP$,1,2):GOSUB 210
1305 FOR X=1 TO A
1310 H$=MID$(B$(X),18,2)+MID$(B$
(X),16,2)+MID$(B$(X),14,2)
1315 SH$=MID$(B$(X),14,6)
1320 IF X>A THEN 1330
1325 IF I>=PA OR DS$>H$ THEN 134
5
1330 SB=SB+B:PRINT#-J,USING"
%          %  PAY                +$$
####.##  $$#####.##";SD$;B;SB
1335 GOSUB 505:GOSUB 250
1340 I=I+1:IF I>=PA AND X>A THEN
 1380 ELSE 1320
1345 C$=MID$(B$(X),22,7):C=VAL(C
$):B$=MID$(B$(X),1,12)
1350 SB=SB-C:PRINT#-J,USING"
%          %  %              %  -$$
####.##  $$#####.##";SH$;B$;C;SB
1355 IF SB>=0 THEN 1370
1360 SB=ABS(SB)
1365 X=A:NEXT X:GOTO 1295
1370 NEXT X
1375 IF I<PA+1 THEN 1320
1380 CLS:PRINT:PRINT USING"YOU W
ILL REQUIRE$$#####.## SET   ASIDE
 NOW AND WITH$$###.## ADDEDEACH
PAYDAY THIS WILL MEET YOUR ANNUA
L/REGULAR BILLS  AS  THEY  FALL
DUE.";SD+1;B:PRINT:PRINT
1385 GOSUB 480
1390 RETURN
1405 CLS:REM CSAVE BILLS FILE
1410 PRINT"POSITION TAPE FOR REC
ORDING"
1415 PRINT:PRINT"      PRESS <PLA
Y AND RECORD>"
1420 GOSUB 475:GOSUB 605
1425 IF F=1 THEN RETURN ELSE 143
0
1430 PRINT:PRINT:PRINT"      CSAV
ING NOW!"
1435 OPEN"O",#-1,"BILLS"
1440 FOR X=1 TO A
1445 PRINT#-1,B$(X)
1450 NEXT X
1455 CLOSE#-1
1460 PRINT:PRINT"      BILLS FILE
 CSAVED":PRINT:PRINT
1465 GOSUB 480
1470 RETURN
1505 CLS:REM CLOAD BILLS FILE
1510 PRINT"POSITION TAPE FOR PLA
YING"
1515 PRINT:PRINT"      PRESS <PLA
Y>"
1520 GOSUB 475
1525 IF F=1 THEN RETURN ELSE 153
0
1530 PRINT:PRINT:PRINT"      CLOA
DING NOW!"
1535 OPEN"I",#-1,"BILLS"
1540 FOR X=1 TO 50
1545 IF EOF (-1) THEN A=X-1:X=50
:GOTO 1555
1550 INPUT #-1,B$(X)
1555 NEXT X
1560 CLOSE #-1
1565 PRINT:PRINT"      BILLS FILE
 CLOADED":PRINT:PRINT
1570 GOSUB 480:CLS
1575 RETURN
1605 CLS:PRINT" TO ADD TO OR DEL
ETE FROM THE             BILLS FI
LE":PRINT
1610 INPUT"INPUT<A>FOR ADD OR<D>
FOR DELETE";N$
1615 IF N$="D" THEN 1670 ELSE IF
N$="A" THEN 1625 ELSE 1610
1620 RETURN
1625 CLS:PRINT" PLEASE INPUT BIL
LS USING THE           FOLLOWING
FORMAT":PRINT
1630 PRINT"    NAME      DATE DUE
  AMOUNT    12 LETTERS   DDMMYY
  ####.##"
1635 GOSUB 475:IF F=1 THEN RETUR
N
1640 CLS:IF A=0 THEN PRINT"THERE
'S NO BILLS FILE!":PRINT:GOSUB 4
80
1645 IF F=1 THEN RETURN
1650 FOR X=A+1 TO 50
1655 GOSUB 1005
1660 NEXT X
1665 GOSUB 945:RETURN
1670 CLS:PRINT"  TO DELETE A REC
ORD FROM THE             BILLS F
ILE                IT'S NUMBER MU
ST BE KNOWN"
1675 GOSUB 475:CLS:IF F=1 THEN R
ETURN
1680 INPUT"INPUT RECORD NUMBER F
OR DELETING";N$:Z=VAL(N$)
1685 FOR X=Z TO A-1
1690 B$(X)=B$(X+1)
1695 NEXT X
1700 A=A-1:RETURN
1705 CLS:PRINT"WRITTEN BY   GLEN
N    BLOMFIELD              21 F
LOWER ST BULLS THE PROGRAM IS FO
R USE WITH A   SAVINGS A/C TO PA
Y ALL REGULAR  BILLS:-INSURANCE,
CAR REG,POWER  PHONE,ETC.WHICH W
HEN ENTERED    WILL SHOW THE SAV
INGS PER WEEK/ MONTH ETC ";
1720 PRINT"AND  SHOW  WHAT  THE
 BALANCE OF THE BILLS A/C SHOULD
 BE ,WHEN FIRST STARTING OR FOR
 ANY YEARLY PERIOD"
1725 GOSUB 475
1730 CLS:PRINT"THE FOLLOWING ARE
 NOT REQUIRED  IF MEMORY IS LIMI
TED                          5-10
0                           405-46
5                           705-78
0                           805-85
0"
1735 GOSUB 475
1740 RETURN
```

■

SOLO SCRABBLE

continued from page 44

```
AL("&H"+MID$(P$,K,1)):NEXTK
2221 IFMID$(A$(N),4*P-3,1)=CHR$(
223)THENSC=SC+VAL("&H"+MID$(A$(N
),4*P,1))ELSEIFMID$(A$(N),4*P-3,
1)=CHR$(175)THENSC=SC+2*VAL("&H"
+MID$(A$(N),4*P,1))
2222 IFMID$(A$(N),4*P-3,1)=CHR$(
239)THENSC=2*SC ELSEIFMID$(A$(N)
,4*P-3,1)=CHR$(191)THENSC=3*SC
2223 PT=PT+SC
2224 NEXTP:RETURN
9999 'board data
10000 DATA 32,39,46,256,270,480,
487,494
10010 DATA 69,73,193,197,201,205
,321,325,329,333,453,457
10020 DATA 35,43,102,104,128,135
,142,226,230,232,236,259,267,290
,294,296,300,384,391,398,422,424
,483,491
10030 DATA 65,77,98,108,131,139,
164,170,263,356,362,387,395,418,
428,449,461
10049 'letter data
10050 DATA 1a9,3b2,3c2,2d4,1e0,4
f2,2g3,4h2,1i9,8j1,5k1,1l4,3m2
10060 DATA 1n6,1o8,3p2,Aq1,1r6,1
s4,1t6,1u4,4v2,4w2,8x1,4y2,Az1
20000 PCLEAR1:GOTO10
```

Melbourne Colour Computer Club's

COCOLOGO

16K ECB
GRAPHIX ENTRY

by Val Stephenson

COCOLOGO IS AN entry for the Melbourne Colour Computer Club competition for a logo. The mountains on page one are purloined from "Astronut" with many additions.

The Listing:

```
0 GOTO10
3 SAVE"210:3":END'6
10 CLS:PRINT@68,"COCOLOGO BY VAL
 STEPHEN":PRINT@330,"COPYRIGHT":
PLAY"P1P1"
20 '*******COCOLOGO**************
30 '******************************
40 '**********BY****************
50 '********VAL T.STEPHEN********
60 '******************************
70 '**********COPYRIGHT********
80 PMODE4,1:PCLS :SCREEN1,1:POKE
 179,0
90 QQ=0
100 DRAW"BM0,168E4R4E6R3E6R2F12R
6F8E8R6E8R4E2R2F18R4E4R4U6E6R3F1
0R3F5R4E12R2F6R3F6E3R4E10R2F10D1
E4R4E6R3E6R2F12R7F8E9"
110 POKE179,4:PAINT(2,170)
120 FORB=0TO255 STEP2:LINE(B,190
-RND(18))-(B,190),PRESET:NEXTB
130 FOR Y=1TO8:CIRCLE(RND(255),R
ND(160)),2,,1:NEXT Y
150 FOR X=1TO300:PSET(RND(255),R
ND(191)):NEXT X
160 LINE(50,10)-(45,40),PSET
170 LINE(50,10)-(55,30),PSET
180 LINE (55,30)-(60,10),PSET
190 LINE(60,10)-(65,40),PSET
200 LINE (65,10)-(70,10),PSET
210 LINE(75,25)-(70,25),PSET
220 LINE(70,40)-(85,40),PSET
230 LINE (70,10)-(70,40),PSET
240 LINE (90,10)-(90,40),PSET
250 LINE(90,40)-(100,40),PSET
260 LINE (105,10)-(105,40),PSET
270 LINE(106,10)-(106,40),PSET
280 LINE (105,10)-(115,10),PSET
290 LINE(105,25)-(115,25),PSET
300 LINE (115,40)-(105,40),PSET
310 LINE (115,10)-(115,40),PSET
320 LINE(120,10)-(130,40),PSET,B
330 LINE(135,10)-(135,40),PSET
340 LINE (145,10)-(145,40),PSET
350 LINE (135,40)-(145,40),PSET
360 LINE (150,10)-(150,40),PSET
370 LINE (150,10)-(160,10),PSET
380 LINE (160,10)-(160,25),PSET
390 LINE(150,25)-(160,25),PSET
400 LINE(158,25)-(160,40),PSET
410 LINE(165,10)-(175,40),PSET
420 LINE(175,10)-(175,40),PSET
430 LINE (190,10)-(180,10),PSET
440 LINE (180,10)-(180,40),PSET
450 LINE (180,40)-(190,40),PSET
460 LINE (180,25)-(185,25),PSET
470 LINE(20,50)-(10,50),PSET
480 LINE(10,50)-(10,80),PSET
490 LINE (10,80)-(20,80),PSET
500 LINE(25,50)-(35,80),PSET,B
510 LINE(40,50)-(40,80),PSET
520 LINE(40,80)-(50,80),PSET
530 LINE (55,50)-(65,80),PSET,B
540 LINE (70,50)-(85,65),PSET,B
550 LINE(70,65)-(70,80),PSET
560 LINE(80,65)-(85,80),PSET
570 LINE(100,50)-(110,50),PSET
580 LINE(100,80)-(110,80),PSET
590 LINE(100,50)-(100,80),PSET
600 LINE(115,50)-(125,80),PSET,B
610 LINE(130,80)-(135,50),PSET
620 LINE(135,50)-(140,70),PSET
630 LINE(140,70)-(145,50),PSET
640 LINE (145,50)-(150,80),PSET
650 LINE(155,50)-(165,65),PSET,B
660 LINE(155,65)-(155,80),PSET
670 LINE (170,50)-(170,80),PSET
680 LINE( 180,50)-(180,80),PSET
690 LINE(170,80)-(180,80),PSET
700 LINE(195,50)-(195,80),PSET
710 LINE (185,50)-(205,50),PSET
720 LINE(220,50)-(210,50),PSET
730 LINE( 210,50)-(210,80),PSET
740 LINE(210,80)-(220,80),PSET
750 LINE(210,65)-(215,65),PSET
760 LINE (225,50)-(235,65),PSET,
B
770 LINE(225,65)-(225,80),PSET
780 LINE(230,65)-(235,80),PSET
790 LINE(90,90)-(70,90),PSET
800 LINE(70,90)-(70,130),PSET
810 LINE(70,130)-(90,130),PSET
820 LINE (100,90)-(100,130),PSET
830 LINE(100,130)-(120,130),PSET
840 LINE(130,90)-(130,130),PSET
850 LINE(130,130)-(150,130),PSET
860 LINE(150,130)-(150,90),PSET
870 LINE(160,90)-(180,110),PSET,
B
880 LINE(165,10)-(165,40),PSET
890 CIRCLE(230,20),14
900 PAINT (230,20)
910 LINE(160,110)-(180,130),PSET
,B
920 LINE (161,90)-(161,130),PSET
930 PLAY"V31O1L8CCEEEGGGO2CCEEGG
O3CCEEGGO4CL200CGECGECGECGECGECG
ECGECGECGECGECGECGECL1CP1"
940 PMODE4,1:PCLS4:SCREEN1,1
950 LINE(10,20)-(20,150),PSET,BF
960 LINE(20,20)-(50,30),PSET,BF
970 LINE(20,140)-(50,150),PSET,B
F
980 LINE(40,30)-(50,40),PSET,BF
990 LINE(40,130)-(50,140),PSET,B
F
1000 LINE(60,20)-(70,150),PSET,
BF
1010 LINE(90,20)-(100,150),PSET,
BF
1020 LINE(70,20)-(90,30),PSET,BF
1030 LINE (70,140)-(90,150),PSET
,BF
1040 LINE(110,75)-(140,80),PSET,
BF
1050 LINE(140,80)-(110,100),PSET
,B
1060 LINE(115,92)-(135,92),PSET
1070 LINE(115,94)-(135,94),PSET
1080 LINE(115,96)-(135,96),PSET
1090 LINE(115,98)-(135,93),PSET
1100 LINE(115,90)-(135,90),PSET
1110 LINE(115,88)-(135,88),PSET
1120 LINE(115,86)-(135,86),PSET
1130 LINE(150,20)-(190,30),PSET,
BF
1140 LINE(150,30)-(160,150),PSET
,BF
1150 LINE(160,140)-(190,150),PSE
T,BF
1160 LINE(180,130)-(190,140),PSE
T,BF
1170 LINE(180,30)-(190,40),PSET,
BF
1180 LINE(230,20)-(240,150),PSET
,BF
1190 LINE(210,20)-(230,30),PSET,
BF
1200 LINE(210,140)-(230,150),PSE
T,BF
1210 LINE(200,20)-(210,150),PSET
,BF
1220 PLAY"V31O2L20CDEFGAO3CDEFGA
O4CDEFGAO5CCCEEEGGGEEEL1CO1L4CO2
C01CP1"
1230 '******CO-CO KEYBOARD******
**
1240 PMODE4,1:PCLS:SCREEN1,1
1250 LINE(30,30)-(220,160),PSET,
B
1260 LINE(40,40)-(210,70),PSET,B
F
1270 LINE(30,80)-(220,80),PSET
1280 LINE(40,100)-(210,150),PSET
,B
1290 LINE(50,110)-(200,140),PSET
,B
1300 LINE(70,140)-(190,150),PSET
,BF
1310 LINE(50,120)-(200,120),PSET
1320 LINE(50,130)-(200,130),PSET
1330 LINE(60,110)-(60,140),PSET
```

continued on page 51

HAMSAT

16/32K ECB
Ham Radio Utility

by Dr Thomas Clarke & M. Garth

This program appeared in April 1985 edition of CoCo and we're re-printing it here for the benefit of those who wanted the program but couldn't find it on the April tape - another april fools joke, perhaps?

Anyway, we're terribly sorry it wasn't on the tape in the first place and are proud to re-print it here in all it's glory.

- ed.

ONE OF THE problems in writing programs for Radio Amateurs is that not all Amateurs have the same, or similar computers. Assembly Language programs have to be written for particular **processors, creating all sorts of problems.**

There are advantages in writing programs in BASIC. Simple changes can often be made to enable the program run in most computers.

With Orbital prediction programs for Amateur Satellites, there are a number of options available to us. Generally, however, it boils down to one of three options.

Option one is to program around the fact that the satellite has a stable orbit of a particular time increment. We can program the computer to advance the satellite by its equatorial crossings. This **method is rather crude and presents some problems in providing the aquisition angles that are really necessary.**

Most Amateur satellites are in polar orbits that are for all intents and purposes circular. Consequently, reasonable accuracy can be obtained by writing a program that flies the satellite in a circular orbit. This is option two.

The third option is to program the computer to calculate the orbit, regardless of whether it it circular or eliptical, and to **provide the necessary aquisition angles. The programs to be presented here this month, and next, are written for option two and three.**

"HAMSAT" was written in Microsoft BASIC, for CoCo, based on a paper, "Calculation of Range and Azimuth of a Satellite from a Ground Station", written by the late Mr. L.Algate.

The program outputs Time (UTC), Azimuth and Elevation Angles and Range in Kms of circular orbit satellites. This includes the Russian RS series as well as Amsat "Oscar" spacecraft.

The necessary data to be included into the program are:

a. The satellite orbit time in minutes. Lines 2720, 2800, 2880, 2960 and 3040.

b. The satellite inclination in degrees. Lines 2730, 2810, 2890, 2970 and 3040.

This data must be updated regularly if accuracy is to be maintained.

The program will ask for the following data:

1. Date - day, month, year.
2. Station location (town),
3. Station latitude
4. Station longitude
5. Start orbit number
6. Stop orbit number
7. EQX west longitude (decimal)
8. Time UTC (decimal).

The program calculates 120 subsatellite points (latitude and longitude) of the orbit and compares each position with the latitude and longitude of the orbit and compares each position with the latitude of the receiving amateur station, calculating range, azimuth angle and elevation angle of the subsatellite point.

Slant range is used a delimeter (set in Lines 2770, 2820, 2900, 2980 and 3060). If range is greater than the delimeter the spacecraft is assumed to be not within aquisition range and calculation of the azimuth and elevation angles is aborted.

Each orbit is regarded as individual and the program stores the equator crossing time and longitude by actually computing past the equator, a time delimeter is used to detect EQX, (contained in the same lines as the range delimeter). The program then advances the orbit count by one, **zeros** all counters and starts a new orbit.

Elapsed time is counted and updates in UTC are maintained as the orbit progresses. The last orbit (stop orbit) EQX and time are printed for use as data for further computation.

There are a number of changes that can be made by Amateur stations. It is not ideal to have to input your station co-ordinates and location everytime you want to run the program.

Lines 610, 620 and 630 can be deleted and the station location inserted in line 1340 (G$), station latitude inserted in line 800 (B4), and station longitude inserted in line 810 (B5).

There is a POKE to 1200 baud output to the printer (line 720), a poke to change screen colour (line 100), speedup pokes for mathematical computations (lines 730, 1040, 2400, 2480 and 2540) and slowdown pokes for printing (line 1810, 2350, 2420, 2450 and 2500). It will work on 16K CoCo with ECB. It has not been tried on a 4K machine.

The program is fairly slow in computing the orbital points. CoCo's are super fast mathematical calculators and when it is doing its number crunching, a "Please wait" message appears on the screen, just to let you know something is happening even though the silence is deafening.

```
0 GOTO100
1 '******** HAMSAT **************
3 SAVE"213:3":END
100 POKE359,57:SCREEN0,1
110 CLS
120 PRINT:PRINT:PRINT:PRINT
130 PRINT"**********************
*********"
140 PRINT"*            'HAMSAT'
      *"
150 PRINT"*           M.J.GARTH
      *"
160 PRINT"*             VK2ZLX
      *"
170 PRINT"*              NOWRA
      *"
180 PRINT"*              N.S.W.
      *"
190 PRINT"**********************
*********"
200 FOR I=0 TO 2000:NEXT
210 PRINT"**********************
*********"
220 PRINT"*  THE PROGRAM CALCULA
TES 120 *"
230 PRINT"*  POINTS ON THE ORBIT
 OF THE *"
240 PRINT"*   SPACECRAFT AND COM
PARES   *"
250 PRINT"*EACH POINT WITH THE P
OSITION *"
260 PRINT"*   OF THE RECEIVING S
TATION  *"
270 PRINT"*  ALL INPUTS MUST BE
DECIMAL *"
280 PRINT"* I.E. TIME MUST BE IN
SERTED  *"
290 PRINT"*     AS 12.5 VICE 12.
30.00   *"
300 PRINT"*  STATION CO-ORDINATE
S ARE   *"
310 PRINT"* CONTAINED IN LINES
800-810 *"
320 PRINT"* AND ARE SET FOR NOWR
A N.S.W.*"
330 PRINT"*REQUIRES 16K RAM PLUS
 PRINTER*"
340 PRINT"*   HIT Y <YES> TO CON
TINUE   *"
350 PRINT"**********************
*********"
360 INPUT "CONTINUE   Y/N";A$
370 IF A$="N" THEN 360 ELSE 380
380 CLS
390 PRINT:PRINT:PRINT
400 PRINT" *********************
*********"
410 PRINT" * ORBITAL DATA ARE CO
NTAINED *"
420 PRINT" *  IN SUBROUTINES FRO
M LINE  *"
430 PRINT" *    2700 THRU LINE 3
100     *"
440 PRINT" * THESE DATA MUST BE
UPDATED *"
450 PRINT" *   REGULARLY IF PRED
ICTION  *"
460 PRINT" *ACCURACY IS TO BE MA
INTAINED*"
470 PRINT" *********************
*********"
480 INPUT "CONTINUE Y/N";A$
490 IF A$="N" THEN 480 ELSE 500
500 CLS:GOSUB 2560:CLS
510 INPUT "DAY,MONTH,YEAR";D,M,Y
520 J4=D:J5=M:J6=Y
530 IF Y/4=INT(Y/4)THEN 540 ELSE
 F9=0:GO TO 550
540 F9=1
550 FOR I=1 TO M:READ D9:NEXT I
560 DATA 0,31,59,90,120,151,181,
212,243,273,304,334,365
570 J=D+D9
580 IF M>2 THEN 600
590 GO TO 610
600 J=J+F9
610 INPUT "STATION LOCATION ";G$
620 INPUT "STATION LATITUDE=";B4
630 INPUT "STATION LONGITUDE=";B
5
640 INPUT "START ORBIT";N
650 INPUT "STOP ORBIT ";N2
660 INPUT "EQX WEST LONG";E
670 INPUT "TIME UTC";T
680 H5=INT(T):H6=(T-H5)*60:H7=IN
T(H6)
690 H8=(H6-H7)*60:H9=INT(H8)
700 CLS
710 PRINT
720 POKE149,0:POKE150,41:' 1200
BAUD
730 POKE 65495,0:' SPEEDUP
740 P=3.14159
750 R=6378.1:' EARTH RADIUS KMS
760 R9=398580:' GRAV CONSTANT
770 T2=1439.93:' EARTH ROTATION
MINUTES
780 T7=0:T8=0:T9=0
790 C4=57.29577951
800 A1=B4/C4:' STAT LAT IN RADS
810 A2=B5/C4:' STAT LONG IN RADS
820 Q=O2*60:Q1=P*2:Q2=((Q/Q1)^2)
830 Q3=((Q2*R9)^.33333)
840 Q4=Q3-R:O1=Q4:O4=(O2*360)/T2
850 O1=Q4
860 O4=(O2*360)/T2
870 ' ORBIT TIME CONSTANTS
880 N8=O2/120:T6=O2/60:N9=T6/120
890 C1=COS(O3/C4)
900 C2=P*2/O2
910 C3=P*2/T2
920 C5=O3/C4
930 ' CONSTANTS FOR OBSERVERS PO
SITION
940 K1=COS(A1)*COS(A2)
950 K2=COS(A1)*SIN(A2)
960 K3=SIN(A1)
970 K4=SIN(A1)*COS(A2)
980 K5=SIN(A1)*SIN(A2)
990 K6=COS(A1)*(-1)
1000 K7=SIN(A2)*(-1)
1010 K8=COS(A2)
1020 GOSUB 1750
1030 PRINT:PRINT"PLEASE WAIT FOR
 DATA OUTPUT"
1040 POKE 65495,0
1050 ' CALC SUB SATELLITE LAT LO
NG & 4 QUAD ATN
1060 C7=SIN(C2*A3)
1070 C8=COS(C2*A3)
1080 IF C8<>0 THEN 1100
1090 C8=1.000E-04
1100 C6=C7/C8
1110 Y=C7
1120 X=C8*(1/C1)
1130 IF X<>0 THEN 1160
1140 L=P
1150 GOTO 1180
1160 L=ATN(Y/X)
1170 IF X<0 THEN 1220
1180 IF Y<0 THEN 1200
1190 GOTO 1230
1200 IF X=0 THEN 1220
1210 L=L+P
1220 L=L+P
1230 Q5=C3*A3
1240 Q6=L-Q5
1250 IF B9=1 THEN 1280
1260 Q7=Q6*C4
1270 GO TO 1290
1280 Q7=Q6*C4-180
1290 L0=Q7
1300 IF L0>0 THEN 1320
1310 L0=L0+360
1320 L0=L0-E
1330 IF L0>0 THEN 1350
1340 L0=360+L0
1350 IF L0<360 THEN 1370
1360 L0=360-L0
1370 L6=L0*P/180:' SUB SATELLITE
 LONG
1380 L1=SIN(C2*A3)*SIN(C5)
1390 L2=SQR(1-(L1^2))
1400 L3=ATN(L1/L2)
1410 L4=L3*C4:' SUB SATELLITE LA
T
1420 H=INT(T):'CONV TIME TO H/M/
S
1430 H1=(T-H)*60
1440 H2=INT(H1)
1450 H3=(H1-H2)*60
1460 H4=INT(H3)
1470 IF A3<N7 THEN 1490:' OVER H
ORIZ CHECK
1480 GOTO 1510
1490 B=360-L0:' STORE EQX DEGREE
S
1500 B1=H:B2=H2:B3=H4:' STORE EQ
X TIME
1510 GOSUB 2020
1520 IF A3<5 THEN 1600
1530 IF R3>N6 THEN 1600
1540 IF T7=1 THEN 1590:' PRINT C
OUNTERS
1550 IF T8=0 THEN 2350
1560 IF T8=1 THEN 2420
1570 IF T8=2 THEN 2450
1580 GOTO 1600
1590 T7=0
1600 A3=A3+N8:T=T+N9:' SET TIME
1610 IF T>24 THEN 1630
1620 GOTO 1650
1630 T=T-24:' IF OVER 24 HRS
1640 J=J+1:' ADD ONE DAY
1650 IF A3>N7+1 THEN 1670:' IS I
T OVER EQUATOR
1660 GO TO 1060
1670 N=N+1: ' ADD ONE ORBIT
1680 E=B:' MAKE EQX CORRECT
1690 A3=0:T8=0:T=T-(N9*2):'ZERO
COUNT AND RESET TIME
1700 GOTO 2500
1710 IF N=N2 THEN 1730: 'CHECK I
F STOP ORBIT
1720 GOTO 1060
1730 END
1740 REM PRINT HEADER ROUTINE
1750 R1=R+O1:K=O2*60
1760 V=((P*2)*(R1/K))*3600
1770 A=(1-((-R/R1)^2))
1780 F=SQR(A):F1=F*R1/R
1790 F2=ATN(F1):O6=F2*C4
1800 O7=SIN(F2)*R1
1810 POKE 65494,0:' SLOW DOWN
1820 PRINT#-2
```

```
1830 PRINT#-2,TAB(15);"ORBITAL D
ATA FOR ";C$
1840 PRINT#-2,TAB(18);"DATA REFE
RENCE   ";G$
1850 PRINT#-2,TAB(20);"DATE=";J4
;J5;J6
1860 PRINT#-2,TAB(20);"DAY OF YE
AR=";J
1870 PRINT#-2,TAB(20);"START ORB
IT=";N
1880 PRINT#-2,TAB(20);"STOP ORBI
T=";N2
1890 PRINT#-2,TAB(20);"EQX DEGRE
ES=";E
1900 PRINT#-2,TAB(20);"EQX TIME
UTC=";H5;H7;H9
1910 PRINT#-2,TAB(20);"ALTITUDE=
";O1;"KMS"
1920 PRINT#-2,TAB(20);"PERIOD=";
O2;"MINS"
1930 PRINT#-2,TAB(20);"INCLINATI
ON=";O3;"DEGS"
1940 PRINT#-2,TAB(20);"INCRIMENT
=";O4;"DEGS"
1950 PRINT#-2,TAB(20);"VELOCITY=
";V;"KMS/HR"
1960 PRINT#-2,TAB(20);"RANGE AT
AOS=";O7;"KMS"
1970 PRINT#-2,TAB(20);"TLM BEACO
N=";D$
1980 PRINT#-2,TAB(20);"DOWNLINK
FREQ=";E$
1990 PRINT#-2,TAB(20);"UPLINK FR
EQ =";F$
2000 RETURN
2010 REM CALCULATE RANGE
2020 K9=R*K1:J1=R*K2:J2=R*K3
2030 X1=R1*(COS(L3)*COS(L6))
2040 Y1=R1*(COS(L3)*SIN(L6))
2050 Z1=R1*SIN(L3)
2060 X2=X1-K9:Y2=Y1-J1:Z2=Z1-J2
2070 R3=SQR((X2^2)+(Y2^2)+(Z2^2)
)
2080 IF R3<N6 THEN 2110
2090 RETURN
2100 REM CALCULATE AZ EL ANGLE
2110 R4=X2/R3:R5=Y2/R3
2120 R6=Z2/R3:E1=K1*R4
2130 E2=K2*R5:E3=K3*R6
2140 E4=E1+E2+E3
2150 E5=SQR(1-E4^2):E6=E4/E5
2160 E7=ATN(E6)*C4
2170 M=K4*R4:M1=K5*R5
2180 M2=K6*R6:M3=M+M1+M2
2190 M4=M3/E5:M5=M4*(-1)
2200 M6=(K7*R4)+(K8*R5)
2210 M7=M6/E5:M8=M7/M5
2220 M9=ATN(M8)
2230 IF M5<0 THEN 2280
2240 IF M7<0 THEN 2260
2250 GOTO 2320
2260 M9=(P*2)-ABS(M9)
2270 GOTO 2320
2280 IF M7<0 THEN 2310
2290 M9=P-ABS(M9)
2300 GOTO 2320
2310 M9=M9+P
2320 M9=M9*C4
2330 RETURN
2340 REM PRINT ROUTINES
2350 POKE 65494,0
2360 PRINT#-2
2370 PRINT#-2," DAY";TAB(8);"ORB
IT"
2380 PRINT#-2,J;TAB(8);N
2390 T8=1
2400 POKE 65495,0
2410 GOTO 1600
2420 POKE 65494,0
2430 PRINT#-2,"TIME UTC";TAB(22)
;"ELEVATION";TAB(37);"AZIMUTH";T
AB(50);"RANGE"
2440 PRINT#-2
2450 POKE 65494,0
2460 PRINT#-2,H;H2;H4;TAB(22);IN
T(E7*1000)/1000;TAB(36);INT(M9*1
000)/1000;TAB(48);INT(R3*1000)/1
000
2470 T8=2:T7=1
2480 POKE 65495,0
2490 GOTO 1600
2500 POKE 65494,0
2510 PRINT#-2
2520 PRINT#-2,"  TIME UTC";TAB(1
5);"EQX W LONG";TAB(27);"ORBIT"
2530 PRINT#-2,B1;B2;B3;TAB(14);E
;TAB(27);N
2540 POKE 65495,0
2550 GOTO 1710
2560 PRINT"**********************
*********"
2570 PRINT"*  ENTRY NO 1 AMSAT O
SCAR 8  *"
2580 PRINT"*   ENTRY NO 2 RUSSIA
N RS5   *"
2590 PRINT"*   ENTRY NO 3 RUSSIA
N RS6   *"
2600 PRINT"*   ENTRY NO 4 RUSSIA
N RS7   *"
2610 PRINT"*   ENTRY NO 5 RUSSIA
N RS8   *"
2620 PRINT"**********************
*********"
2630 INPUT"SATELLITE SELECTION =
";B9
2640 IF B9=1 THEN 2710
2650 IF B9=2 THEN 2790
2660 IF B9=3 THEN 2870
2670 IF B9=4 THEN 2950
2680 IF B9=5 THEN 3030
2690 GO TO 2560
2700 REM SATELLITE ORBITAL PARAM
ETERS
2710 C$="AMSAT OSCAR 8"
2720 O2=103.13
2730 O3=98.8119
2740 D$ ="29.71 MHZ"
2750 E$ ="29.410-29.450 MHZ"
2760 F$ ="145.91-145.95 MHZ"
2770 N6=4250:N7=103.5
2780 RETURN
2790 C$="RUSSIAN RS5"
2800 O2=119.498
2810 O3=82.960
2820 N6=5250:N7=120
2830 D$="29.331 MHZ"
2840 E$="29.410-29.450 MHZ"
2850 F$="145.91-145.95 MHZ"
2860 RETURN
2870 C$="RUSSIAN RS6"
2880 O2=118.660
2890 O3=82.959
2900 N6=5250:N7=119.6
2910 D$="29.411 MHZ"
2920 E$="29.41-29.45 MHZ"
2930 F$="145.91-145.95 MHZ"
2940 RETURN
2950 C$="RUSSIAN RS7"
2960 O2=119.139
2970 O3=82.958
2980 N6=5250:N7=119.6
2990 D$="29.341 MHZ"
3000 E$="29.46-29.5 MHZ"
3010 F$="145.96-146.0 MHZ"
3020 RETURN
3030 C$="RUSSIAN RS8"
3040 O2=119.707
3050 O3=82.953
3060 N6=5250:N7=120.2
3070 D$="29.461 MHZ"
3080 E$="29.46-29.5 MHZ"
3090 F$="145.96-146.0 MHZ"
3100 RETURN
3110 END
```

continued from page 48

```
1340 LINE(70,110)-(70,140),PSET
1350 LINE(80,110)-(80,140),PSET
1360 LINE(90,110)-(90,140),PSET
1370 LINE(100,110)-(100,140),PSE
T
1380 LINE(110,110)-(110,140),PSE
T
1390 LINE(120,110)-(120,140),PSE
T
1400 LINE(130,110)-(130,140),PSE
T
1410 LINE(140,110)-(140,140),PSE
T
1420 LINE(150,110)-(150,140),PSE
T
1430 LINE(160,110)-(160,140),PSE
T
1440 LINE(170,110)-(170,140),PSE
T
1450 LINE(180,110)-(180,140),PSE
T
1460 LINE(190,110)-(190,140),PSE
T
1470 PLAY"P1"
1480 PLAY"V31O4L50CEGCEGCEGCEGCC
CCCCCGEGCGECGECGECEGCECGCECGCECG
CCCL2O1CP10"
1490 '***MONITER TV SET***
1500 PMODE4,1:PCLS:SCREEN1,1
1510 LINE(30,30)-(230,150),PSET,
B
1520 LINE(40,40)-(180,140),PSET,
B
1530 LINE(190,40)-(220,140),PSET
,B
1540 LINE(193,50)-(197,70),PSET,
B
1550 LINE(198,45)-(202,75),PSET,
B
1560 LINE( 203,40)-(207,80),PSET
,B
1570 LINE (208,45)-(212,75),PSET
,B
1580 LINE (213,50)-(217,70),PSET
,B
1590 LINE (190,90)-(220,90),PSET
1600 LINE(190,100)-(220,100),PSE
T
1610 CIRCLE(195,95),2,1
1620 CIRCLE(205,95),2,1
1630 CIRCLE(215,95),2,1
1640 LINE(50,50)-(170,130),PSET,
BF
1650 LINE (195,105)-(215,135), P
SET,BF
1660 PLAY"P1P1"
1670 GOTO80
```

Here we go, one more time!

PENTOMINOES

32K ECB (16K modifyable)
GAME

by Bob Delbourgo

PENTOMINOES CONSISTS OF 5 squares with four common edges. There are twelve basic shapes if one discounts configurations that are obtained by rotating or by reflection (turning over).

Two players may pick and place pentominoes within a grid so as to occupy as much space and thus prevent the opponent's entry.

Full instructions are in the program, and good luck!

The Listing:

```
 1 'pentominoes - Bob Delbourgo - OCT. 1985
2 GOTO10
3 SAVE"205K:3":END'1
4 GOSUB117
5 '16K COCO OWNERS SHOULD DELETE LINES 11-21
10 CLS8:CLEAR700:DIMA(4),C(4)
11 PRINT@40,"PENTO";:PRINT@46,"MIN";:PRINT@51,"  ";:PRINT@80,"OS";:PRINT@84,"BY";:PRINT@139," ";:PRINT@141,"D";:PRINT@169,"BOB";:PRINT@173,"ELB";:PRINT@177,"OURG";:PRINT@203," ";:PRINT@206,"  ";:PRINT@212,"O";
12 PRINT@293," ";:PRINT@299," ";:PRINT@309,"DI";:PRINT@315,"S";:PRINT@325,"DO ";:PRINT@331,"YOU";:PRINT@335,"NEED";:PRINT@342,"RE";:PRINT@347,"N";:PRINT@359," ";:PRINT@363," ";:PRINT@369," ";:PRINT@375,"C";:PRINT@377,"TIO";
13 PRINT@430," ";:PRINT@432," ";:PRINT@462,"Y/N";
14 I$=INKEY$:IFI$="N"THEN22ELSEIFI$="Y"THEN15ELSER=RND(1000):GOTO14
15 CLS7:PRINT@32,"PENTOMINOES CONSIST OF 5 SQUARESWITH FOUR COMMON EDGES. THERE  ARE TWELVE BASIC SHAPES IF ONE  DISCOUNTS CONFIGURATIONS THAT  ARE OBTAINED BY ROTATION OR BY  REFLECTION (TURNING OVER)."
16 PRINT@256,"TO SEE THOSE SHAPES TRY A GAME. SINCE THE PENTOMINOES OCCUPY 60 (=12X5) SQUARES THEY CAN FILL A CHECKERBOARD, BARRING 4 SQUARES.";:PRINT@416,"OR ELSE NINE PENTOMINOES COULD  FILL A TRIPLY SCALED PENTOMINO.":GOSUB117:CLS6
17 PRINT@32,"THERE ARE AT LEAST 3 WAYS THAT  ONE CAN PLAY PENTOMINOES:-":PRINT@128,"ONE MAY TRY TO FILL AN 8X8 GRID WITH THE 12 BASIC PENTOMINOES  WITH 4 SQUARES FILLED IN, OR":PRINT@256,"ONE MAY FILL UP A TRIPLE-SIZE  PENTOMINO WITH 9 OF THEM, OR"
18 PRINT@352,"TWO PLAYERS MAY PICK AND PLACE  PENTOMINOES WITHIN A GRID SO AS TO OCCUPY AS MUCH SPACE AND THUSPREVENT THE OPPONENT'S ENTRY.":GOSUB117
19 CLS5:PRINT@32,"TAKE YOUR PICK AND REMEMBER...":PRINT@96,"TO MOVE THE PENTO CURSOR ON THE LEFT SIDE OF THE SCREEN, PRESS  THE  U, D, R, L  KEYS.":PRINT@224,"TO MOVE THE BOARD CURSOR ON THE RIGHT USE THE ARROW KEYS."
20 PRINT@320,"TURN A PENTOMINO CLOCKWISE BY  KEYING IN C, ANTICLOCKWISE BY  KEYING IN A. TURN IT UPSIDE DOWNBY PRESSING M. PRESS I TO INSERTA PENTOMINO. Q TO QUIT IF STUCK.";
21 PLAY"O3T2V20L8CEL4GP16L16CP16EL4GO4L8CO3BL4AP16L16FP16EL4DO4L8DCO3L4BP16L16BP16AL4GL8ABL4O4C":GOSUB117
22 CLSRND(7)+1:PRINT@32,"CHOOSE :-"
23 PRINT@96,"(1) CHECKERBOARD (8X8) PUZZLES":PRINT"(2) PENTOMINO TRIPLICATION":PRINT"(3) 2-PLAYER SQUARE (NXN) GAME"
24 I$=INKEY$:IFVAL(I$)<1ORVAL(I$)>3THEN24
25 ONVAL(I$)GOTO26,34,38
26 DIMB(11,11):BS=8
27 PRINT@224,"<S>YMMETRICAL OR <R>ANDOM";:INPUTI$:IFI$="S"THEN30
28 IFI$="R"THEN33
29 R=RND(100):GOTO27
30 PRINT@288,"X-COORD. (1-4)";:INPUTX0:X0=INT(X0):IFX0<1ORX0>4THEN30
31 PRINT@320,"Y-COORD. (1-4)";:INPUTY0:Y0=INT(Y0):IFY0<1ORY0>4THEN31
32 GOTO40
33 PRINT@224,"SETTING IT UP ..."
:R1=RND(4):S1=RND(4):R2=RND(4)+4:S2=RND(4):R3=RND(4):S3=RND(4)+4:R4=RND(4)+4:S4=RND(4)+4:GOTO40
34 PRINT@224,"PENTOMINO # (1-12)";:INPUTN:N=INT(N):IFN<1ORN>12THEN34
35 IFN=1THENDIMB(18,9)ELSEDIMB(18,12)
36 NN=3
37 PRINT@288,"THE OBJECT OF THE EXERCISE IS TOFILL THE TRIPLY SCALED AREA WITHany nine PENTOMINOES.";I$="Q":GOTO40
38 PRINT@224,"SIZE OF BOARD (7-9)";:INPUTBS:BS=INT(BS):IFBS<7ORBS>9THEN38
39 PRINT@288,"NOW ENTER YOUR PENTOMINOES ONE  AT A TIME AND ALTERNATELY. THE  LAST TO ENTER SUCCESSFULLY WINS THE GAME ...":FORT=1TO2000:NEXT:DIMB(BS+3,BS+3):I$="P":GOTO40
40 DIMA$(12):SU$="R5EU2HL4HU2ER5BR4D6F2R2E2U6BR10L4G2D4F2R4BR10L4H2U4E2R4BR10L6D4R4L4D4R6BR4R5EU2HL4HU2ER5BR10L5GD2FR4FD2GL5"
41 A$(1)="0000000000111110000000000"
42 A$(2)="0000000000011110100000000"
43 A$(3)="0000000000111100010000000"
44 A$(4)="0000001100001110000000000"
45 A$(5)="0000001000011100010000000"
46 A$(6)="0000001000011100100000000"
47 A$(7)="0000001000011100001000000"
48 A$(8)="0000000111001000010000000"
49 A$(9)="0000000000011100110000000"
50 A$(10)="0000000100011100010000000"
51 A$(11)="0000001010011100000000000"
52 A$(12)="0000001100001100001000000":PMODE4,1:PCLS1
53 H$="111111111111":KE$="U8D4R2E4G4F4BR10L6U4R4L4U4R6BR4F3E3G3D5BR7R5EU2HL4HU2ER5":TU$="O4V25T255L255DGEFDEAAAABFGE":R$="U3HFER2F":P$(1)="U8":P$(2)="R3L6E6UHL3G2"
54 IFI$="R"ORI$="S"ORI$="P"THENFORI=2TOBS+1:FORJ=2TOBS+1:B(I,J)=1:NEXTJ,I
55 COLOR0:IFI$="S"THENBX=X0+1:BY=Y0+1:GOSUB116:BX=10-Y0:BY=X0+1:GOSUB116:BX=Y0+1:BY=10-X0:GOSUB116:BX=10-X0:BY=10-Y0:GOSUB116:GOTO6056 IFI$="R"THENBX=R1+1:BY=S1+1:GOSUB116:BX=R2+1:BY=S2+1:GOSUB116:BX=R3+1:BY=S3+1:GOSUB116:BX=R4+1:BY=S4+1:GOSUB116:GOTO60
57 FORI=1TO5:FORJ=1TO4
```

```
58 IFMID$(A$(N),I+5*J,1)="1"THEN
FORP=0TO2:FORQ=0TO2:B(3*I+P-1,3*
J+Q-1)=1:XL=21*I+7*P+120:YL=21*J
+7*Q-13:LINE(XL,YL)-(XL+4,YL+4),
PSET:LINE(XL+4,YL)-(XL,YL+4),PSE
T:NEXTQ,P
59 NEXTJ,I
60 SCREEN1,1:GOSUB119:LINE(0,0)-
(133,175),PRESET,B:LINE(0,175)-(
133,191),PRESET,B:LINE(133,0)-(2
56,191),PRESET,B:LINE(133,98)-(2
56,175),PRESET,B
61 DRAW"C0BM144,188;U8G2E2F2H2BR
10D8H2F2E2BE2BR4R6H2F2G2E2BR6E2G
2F2H2R6BR24BD4;XKE$;":DRAW"BM14,
180;D7FR4EU7BF8U8R6FD2GL5R2F4BR8
U8R4F2D4G2L4BR20L6U8BR24BD8;XKE$
;"
62 IFI$="Q"THEN63ELSEFORI=1TOBS:
FORJ=1TOBS:IFB(I+1,J+1)=2THENNEX
TJ,I ELSELINE(134+7*I,1+7*J)-(13
8+7*I,5+7*J),PRESET:LINE(138+7*I
,1+7*J)-(134+7*I,5+7*J),PRESET:N
EXTJ,I
63 DRAW"C0BM144,104;G4F4BR6H2U4E
2R2F2D4G2L2R2BU2F4BU2BR2E4H4;BR8
BD4D3FR2EU3D3FBR3U4BU2UBF2BR2R4L
2U2D6FRE":DRAW"BM144,118;G4F4BR4
U8F3E3D8BR4E4H4BF8U4BU2UBD7BR4;X
R$;BF3;XR$;BF3BRR2EU2HL2GD2FBR7;
XR$;"
64 DRAW"BM144,132G4F4BR4U5E3F3DL
6R6D4BR4E4H4BF8U4R3FD3BU5BR3R4L2
U2D6FREBFBR3U4BU2UBR12BUG4F4BR8L
3H2U4E2R3BR4F4G4BR10HU7BF4BR2R2F
D2GL2HU2EBR11L3GD2FR3BR5U7D4RE2G
2F3":DRAW"BM144,146;G4F4BR3R6L3U
8L3R6BR3F4G4;BR8U4R3FD3BR3R3EHL2
HER3BR3BFR4HL2GD2FR3BR5;XR$;
65 DRAW"BM195,149;R4L2U2D6FRE":X
C=22:YC=22:HC=1:IFI$="P"THENLINE
(133,160)-(256,175),PSET,BF:DRAW
"BM140,172C1U8R4FD2GL5BF4BR12L6U
8BF8BR2U5E3F3DL6R6D4BR6U5H3F3E3B
R9L6D4R4L4D4R6BR4U8R5FD2GL5R2F4;
BR12U8BR4D8BE2L8BU4R8":COLOR0
66 BX=5:BY=5:PP=1:IFI$="Q"THENBX
=8
67 W=0:XB=7*BX+127:YB=7*BY-6:IFI
$="P"THENCOLOR0:LINE(240,160)-(2
56,175),PSET,BF:DRAW"C1BM244,172
;XP$(PP);":COLOR1
68 PUT(XB,YB)-(XB+4,YB+4),A,NOT:
PUT(XC,YC)-(XC+4,YC+4),C,NOT:FOR
T=1TO40:NEXTT:PUT(XB,YB)-(XB+4,Y
B+4),A,NOT:PUT(XC,YC)-(XC+4,YC+4
),C,NOT
69 K$=INKEY$:IFK$=""THENFORT=1TO
30:NEXTT:GOTO68
70 K=INSTR("QIACMRLUD"+CHR$(9)+C
HR$(8)+CHR$(94)+CHR$(10),K$):IFK
=0THEN68
71 ONK GOTO72,74,82,79,85,87,92,
97,101,105,107,109,111
72 IFI$="P"THENPRINT@448,"player
 #"PP" loses!!":GOSUB117
73 RUN
74 GOSUB130:IFW=1THENSOUND100,2:
GOTO67
75 U=1:GOSUB126
76 NN=NN+1:COLOR1:LINE(XC-15,YC-
15)-(XC+20,YC+20),PSET,BF:MID$(H
$,HC,1)="0":A$(HC)=STRING$(15,"0
"):IFNN=12THEN113
77 PP=PP+1:IFPP=3THENPP=1
78 GOTO67
79 GOSUB118
80 FORJ=21TO1STEP-5:A1$=A1$+MID$
(A$(HC),J,1):A2$=A2$+MID$(A$(HC)
,J+1,1):A3$=A3$+MID$(A$(HC),J+2,
1):A4$=A4$+MID$(A$(HC),J+3,1):A5
$=A5$+MID$(A$(HC),J+4,1):NEXTJ:A
$(HC)=A1$+A2$+A3$+A4$+A5$
81 COLOR1:LINE(XC-15,YC-15)-(XC+
20,YC+20),PSET,BF:X=XC-15:Y=YC-1
5:U=0:GOSUB127:GOTO67
82 GOSUB118
83 FORJ=5TO25STEP5:A1$=A1$+MID$(
A$(HC),J,1):A2$=A2$+MID$(A$(HC),
J-1,1):A3$=A3$+MID$(A$(HC),J-2,1
):A4$=A4$+MID$(A$(HC),J-3,1):A5$
=A5$+MID$(A$(HC),J-4,1):NEXTJ:A$
(HC)=A1$+A2$+A3$+A4$+A5$
84 COLOR1:LINE(XC-15,YC-15)-(XC+
20,YC+20),PSET,BF:X=XC-15:Y=YC-1
5:U=0:GOSUB127:GOTO67
85 A1$=LEFT$(A$(HC),5):A5$=RIGHT
$(A$(HC),5):A2$=MID$(A$(HC),6,5)
:A4$=MID$(A$(HC),16,5):A$(HC)=A5
$+A4$+MID$(A$(HC),11,5)+A2$+A1$
86 COLOR1:LINE(XC-15,YC-15)-(XC+
20,YC+20),PSET,BF:X=XC-15:Y=YC-1
5:U=0:GOSUB127:GOTO67
87 XC=XC+42:IFXC>106THENXC=XC-12
6:YC=YC+42
88 IFYC>148THENYC=22
89 HC=HC+1:IFHC>12THENHC=1
90 IFMID$(H$,HC,1)="0"THEN87
91 GOTO67
92 XC=XC-42:IFXC<0THENXC=XC+126:
YC=YC-42
93 IFYC<0THENYC=YC+168
94 HC=HC-1:IFHC=0THENHC=12
95 IFMID$(H$,HC,1)="0"THEN92
96 GOTO67
97 YC=YC-42:HC=HC-3:IFYC<0THENYC
=YC+168:XC=XC-42:HC=HC+11
98 IFXC<0THENXC=XC+126:HC=12
99 IFMID$(H$,HC,1)="0"THEN97
100 GOTO67
101 YC=YC+42:HC=HC+3:IFYC>148THE
NYC=YC-168:XC=XC+42:HC=HC-11
102 IFXC>106THENXC=XC-126:YC=22:
HC=1
103 IFMID$(H$,HC,1)="0"THEN101
104 GOTO67
105 BX=BX+1:IFB(BX,BY)=0THENSOUN
D1,1:BX=BX-1
106 GOTO67
107 BX=BX-1:IFB(BX,BY)=0THENSOUN
D1,1:BX=BX+1
108 GOTO67
109 BY=BY-1:IFB(BX,BY)=0THENSOUN
D1,1:BY=BY+1
110 GOTO67
111 BY=BY+1:IFB(BX,BY)=0THENSOUN
D1,1:BY=BY-1
112 GOTO67
113 LINE(133,160)-(256,175),PRES
ET,BF
114 DRAW"BM148,172;C1;XSU$;":FOR
T=1TO30:NEXT:DRAW"BM148,172;C0;X
SU$;"
115 FORI=1TO30:NEXT:IFINKEY$=""T
HEN114ELSERUN
116 LINE(7*BX+126,7*BY-7)-(7*BX+
132,7*BY-1),PSET,B:B(BX,BY)=2:RE
TURN
117 IFINKEY$=""THEN117ELSERETURN
118 A1$="":A2$="":A3$="":A4$="":
A5$="":RETURN
119 X=21:Y=21
120 FORI=1TO12:IFMID$(H$,I,1)="0
"THEN125
121 FORJ=0TO24:JJ=INT(J/5)
122 IFMID$(A$(I),J+1,1)="1"THENL
INE(X+7*J-35*JJ-14,Y+7*JJ-14)-(X
+7*J-35*JJ-8,Y+7*JJ-8),PRESET,B
123 NEXTJ
124 X=X+42:IFX>105THENX=X-126:Y=
Y+42
125 NEXTI:RETURN
126 X=XB-15:Y=YB-15
127 FORJ=0TO24:JJ=INT(J/5)
128 IFMID$(A$(HC),J+1,1)="1"THEN
XL=X+7*J-35*JJ:YL=Y+7*JJ:LINE(XL
,YL)-(XL+6,YL+6),PRESET,B:COLOR1
:LINE(XL+1,YL+1)-(XL+5,YL+5),PSE
T,BF:COLOR0:IFU=1THENB(BX-2+J-5*
JJ,BY-2+JJ)=2:PLAYTU$
129 NEXTJ:RETURN
130 W=0:FORI=1TO5:FORJ=0TO4:IFMI
D$(A$(HC),I+5*J,1)="0"THEN132
131 XX=BX-3+I:YY=BY-2+J:IFB(XX,Y
Y)=0ORB(XX,YY)=2THENW=1:I=6:J=5
132 NEXTJ,I:RETURN
```

CORRECTION

CoCo 3 Color Chart (February 1987 Rainbow Magazine.)

'Color Chart for the COCO3' by Rick Adams & Dale Lear in the February 1987 magazine, looked good, and I keyed in the program with great anticipation to see 64 colours on screen at once, but to my dismay found it would not work correctly.

An examination of line 280 in the machine language listing revealed the problem, it was set-up for American N.T.S.C. TV systems.

Rather than fiddle with the machine language section and make a 'hash' of it, I chose to modify the basic section.

Change the following lines:

```
190 HPRINT(4,20),"COLOR COMPU
 TER 3 - COLOR CHART"
240 HLINE(X*80+10,5)-(X*80+40
  ,133),PSET,B
260 HPRINT(X*10+5,Y+1),X*16+Y
270 HLINE(X*80+10,Y*8+13)-(X*
  80+40,Y*8+13),PSET
280 HPAINT(X*80+20,Y*8+10),8+
  X,1
```

The program will now run correctly on Australian 625 line P.A.L. tv systems. Enjoy the 64 colours on screen simultaneously, and in passing, they look superb on the CM-8 RGB Monitor.

SMARTER DATA STRUCTURES in FORTH

Languages

by John Redmond

LAST MONTH WE looked at the simplest ways Forth has for storing data - variables and constants and the use of CREATE to label a block of memory to be used as an array.

It's now time to develop further the use of CREATE to construct new and smarter memory objects.

Remember that, to set aside the array, we coded CREATE ECOSYSTEM 24 ALLOT but we did not make any real decisions about how we would use the ECOSYSTEM.

If we type ECOSYSTEM from the keyboard, the address of the first of the 24 bytes (the zeroth byte) is put on the stack. If we want the address of the seventh byte, we can now type in 7 + and the incremented address is now on the stack. Now, if we want the value of the byte at that address, we type in C@ . and the value is typed to the screen in the current number base (usually DECIMAL or HEX).

This sequence of Forth words can be in a program, too, and perhaps the same sequence occurs several times over in the same program. This wastes memory - and it is not elegant.

Pascal and C programmers would point out that such a sequence of instructions should be relegated to a procedure or function. And Forth programmers would agree. The crudest way would be to define a new word, such as BYTE:

```
: BYTE ( address, offset--byte) + C@ ;
```

This word expects on the stack the base address and the offset (note the bracketed stack comment in the definition). It adds these values to get the address (pointer) of the byte value required, gets this byte and returns it on the stack.

Now, in the program, we can code:

```
ECOSYSTEM 7 BYTE
```

... instead of ...

```
ECOSYSTEM 7 + C@
```

and achieve a small (trivial?) improvement in readability and code size. But surely Forth can do better than that! And perhaps we might have wanted a 16-bit value rather than an 8-bit value?

To develop our approach further, we need to introduce a unique Forth concept - the definition of defining words.

This is a potentially mind-blowing concept in that it amounts to temporal indirection. (Remember that spatial indirection is the use of pointers to memory objects rather than the objects themselves.)

We literally put off looking at them.) The idea of defining words is best understood with examples. These will all involve (directly or indirectly) the use of the sequence:

```
CREATE (defining code) DOES> (action code).
```

The simplest example is the word VARIABLE, which can be defined as

```
: VARIABLE CREATE 2 ALLOT DOES> ;
```

Note that this defining code ALLOTs 2 bytes, while the action does nothing (there is nothing between DOES> and ;). When we use the construct VARIABLE ECOSYSTEM, VARIABLE takes over temporary control of the input stream and uses CREATE to fetch the next word in the stream (i.e. ECOSYSTEM) to construct a header which can be found in the Forth dictionary.

Then two bytes are ALLOTed and the defining code is finished by DOES>, which plays some clever tricks with the value in the cfa (see last month) of ECOSYSTEM.

This code returns the address of the first of the two ALLOTed bytes when ECOSYSTEM is invoked. Nothing more happens, because VARIABLE defines such a simple data structure.

We now have a general understanding of how VARIABLE works; so let's go a small step further. Remember that CONSTANT returns on the stack the CONTENTS of a memory location, rather than just its address. Its definition is:

```
: CONSTANT CREATE , DOES> @ ;
```

As we saw last month, it is used in the following way: 116 CONSTANT POINTER. When this short command is input, the following events occur:

1. the value of 116 is put on the stack.
2. CONSTANT is executed and, in sequence,
   a. a header is made with the name POINTER (this is done by CREATE);
   b. the value of 116 is inserted after the header; and
   c. DOES> pokes special code (don't worry what it is!) into the cfa of POINTER.

It should be understood that all this is done at COMPILE time. When POINTER is actually used (i.e., at RUN time), two other things happen:

1. the address of memory directly after the cfa of POINTER (i.e., its parameter field address) is put on the stack.
2. @ then takes over and replaces this address with the value (116) at this address.

Think about simple (and obvious) extensions to these - like 2CONSTANT, which returns a double-length number at run time,

```
: 2CONSTANT CREATE , , DOES> 2@ ; or
```

CCONSTANT, which returns a byte value:

```
: CCONSTANT CREATE C, DOES> C@ ;
```

Now we can return to the problem of arrays. Remember that we want to submerge (hide) the way in which the address of an element is calculated. We might try:

```
: ARRAY CREATE ALLOT DOES>
+ ;
```

Now ARRAY is a defining word, just like CONSTANT, and is used as 24 ARRAY ECOSYSTEM. Note that it expects on the stack a number (e.g., 24) and that, after CREATE has done its work, ALLOT uses this number to decide how many BYTES are allocated for the private (local) use of ARRAY. That's all that happens at compile time, but what about at run time?

Well, ECOSYSTEM expects a number on the stack which gives it the offset into the array (as before). This is added, as before, to the base address given by ECOSYSTEM, by the + word in the definition of ARRAY. Now we can get the address of the seventh byte of ECOSYSTEM by simply coding:

```
7 ECOSYSTEM.
```

By now, you should be starting to see the economy and elegance of Forth. To summarize, in words, what we have done: We have defined a defining word (ARRAY) which is used to define another class of words (e.g., ECOSYSTEM) which, at run time, use code which is present only in the original defining word (ARRAY). Read it again!

To be fair to you, gentle reader, Charles Moore (the inventor of Forth) has admitted that it took him about three years to really master this concept of temporal indirection. It will take you no more than thirty minutes!

The definitions so far are pretty trivial. What, for instance, do we do if the elements of the array are 16-bit integers, 32-bit double integers or the 5-byte floating-point numbers used by Color Basic?

Well, it's simple in principle - we just multiply the array index by 2, 4 or 5 before adding to the base address. Fine, but we want to code this just once, in the defining word. How?

Forth is a language that is really going places and it has so much development available to it. Much of this development is in the area of style. This is partly subjective, I know, but in my (subjective) opinion most Forth code is terrible, just as most Pascal code and (particularly) C code is terrible. It may work, but it's unreadable. So let's look at a syntax for array words.

Remember that it must look right! For a start, we'll define:

```
: CELLS(number--#bytes,
type) 2* 1 ;
```

(The stack comment means that CELLS expects a number on the stack and that it returns two numbers - the number to bytes and a type number on top.) What is a cell? It is Forth jargon for a 16-bit (2-byte) memory location. If we want an array with cells, and we want to use CELLS, we need an upgraded ARRAY definition:

```
: ARRAY CREATE C, ALLOT
DOES> 1+ ;
```

This will construct an array with an 8-bit value in the very first byte (put there by the C,) and words defined by it will return the address of the NEXT byte after this first byte (see the 1+ ?). So what's the point of the first byte? Absolutely no point - as we've defined ARRAY - but we can at least use CELLS and ARRAY together:

```
21 CELLS ARRAY ECOSYSTEM
```

But what about that byte? Remember that it is a type number. It can be used to tell the program using ECOSYSTEM how to use the data in it. (Note that we have hidden from the program just how we calculate the element address.) If the program has been given information about the significance of the 1 (e.g., by using smart words which can interpret it), it will know how to fetch the value from, or put a value into, the array.

But ECOSYSTEM, as defined, does not give back the information about its data type; so let's try again:

```
: ARRAY CREATE C, ALLOT
DOES> COUNT ;
```

Now, when we define ECOSYSTEM as above, we get an ECOSYSTEM which returns on the address of its zeroth element with a value on top which indicates the data type. NOW smart words, like VALUE, SET and SHOW, can be used to fetch, change and display any of the array elements, while protecting the user or her/his code from any need to understand how the data are stored.

This is the direction in which the fourth and fifth generation languages are moving, but there are costs - large code size and slow execution.

My own inclination is to be slow at hurrying that way: I prefer to concentrate on good syntax and small, effective code. This forces me (quite cheerfully) to stay with Forth and I do this knowing that there is no data structure or algorithm that cannot be implemented (and implemented very well) in the language.

This will be the direction of future articles: algorithms, code factoring and data structures. All the heady material of the text books, and all expressed easily, efficiently and portably in high-level Forth.

What's factoring? It's the dividing up of the code statements (words in Forth) into well-chosen procedures and functions (words in Forth) to give code that is readable and maintainable - and reusable in other programs.

As a simple example of wrong factoring, look at

```
21 CELLS ARRAY ECOSYSTEM
```

It doesn't look right - and it isn't. The idea that I was developing was that there would be four different descriptor words (BYTES, CELLS, LONGS and FLOATS) which would pass some values to ARRAY so that it could incorporate into the array the type byte plus the correct amount of memory.

But all these words use ARRAY; so why keep ARRAY out of their definitions? Let's define:

```
: BYTES 0 ARRAY ;
: CELLS 2* 1 ARRAY ;
: LONGS 4 * 2 ARRAY ;
: FLOATS 5 * 3 ARRAY ;
```

There are more elegant, more advanced, ways of doing this, but they can wait. It's worth noting that our change of factoring has improved the syntax to ...

```
21 CELLS ECOSYSTEM.
```

No other language is so readable (read any Prolog lately?). Think about how you would define VALUE, SET and SHOW. Until next month.

■

ALL MY LOVING

32K ECB
MUSIC+ ADDITIONS

by Steve Youngberry

STEVE YOUNGBERRY has supplied us with a selection of songs from the Beatles Era. Those people with a copy of "Music+" (see July '86 edition of Australian Rainbow Magazine) will be the lucky ones to benefit from Mr. Youngberry's work.

The selection we have for you are three of the Beatles songs; "All my Loving", "When I'm 64" and "If I Fell in Love with You".

I'm sure you'll enjoy them all!

Listing One: All my Loving

Listing Two: When I'm 64

Listing Three: If I Fell

Listing One:

```
COL: LEN ,V1  ,V2  ,V3  ,V4
  1: 4   ,A3# ,0   ,0   ,0
  2: 4   ,A3  ,0   ,0   ,0
  3: 4   ,G3  ,G2  ,0   ,0
  4: 4   ,G3  ,D3  ,A2# ,0
  5: 4   ,A3  ,D3  ,A2# ,0
  6: 4   ,A3# ,D3  ,A2# ,0
  7: 4   ,C4  ,E2  ,0   ,0
  8: 4   ,D4  ,C3  ,A2# ,0
  9: 4   ,D4  ,C2  ,0   ,0
 10: 4   ,E4  ,C3  ,A2# ,0
 11: 4   ,F4  ,F2  ,0   ,0
 12: 4   ,F4  ,C3  ,A2  ,0
 13: 4   ,F4  ,C3  ,A2  ,0
 14: 4   ,E4  ,C3  ,A2  ,0
 15: 4   ,D4  ,D2  ,0   ,0
 16: 4   ,A3  ,F2  ,0   ,0
 17: 4   ,A3  ,A2  ,0   ,0
 18: 4   ,D4  ,D3  ,A2  ,F2
 19: 4   ,D4  ,A2# ,0   ,0
 20: 4   ,D4  ,F3  ,D3  ,0
 21: 4   ,D4  ,F3  ,D3  ,0
 22: 4   ,C4  ,F3  ,D3  ,0
 23: 4   ,A3# ,G2  ,0   ,0
 24: 4   ,A3# ,D3  ,A2# ,0
 25: 4   ,G3  ,D3  ,A2# ,0
 26: 4   ,F3  ,D3  ,A2# ,0
 27: 4   ,G3  ,D2# ,0   ,0
 28: 4   ,G3  ,G2  ,0   ,0
 29: 4   ,G3  ,A2# ,0   ,0
 30: 4   ,G3  ,G2  ,0   ,0
 31: 4   ,G3  ,D2# ,0   ,0
 32: 4   ,G3  ,A2# ,G2  ,0
 33: 4   ,A3# ,A2# ,G2  ,0
 34: 4   ,A3  ,A2# ,G2  ,0
 35: 4   ,G3  ,G2  ,0   ,0
 36: 4   ,G3  ,D3  ,A2# ,0
 37: 4   ,A3  ,D3  ,A2# ,0
 38: 4   ,A3# ,D3  ,A2# ,0
 39: 4   ,C4  ,E2  ,0   ,0
 40: 4   ,C4  ,C3  ,A2# ,0
 41: 4   ,D4  ,C2  ,0   ,0
 42: 4   ,E4  ,C3  ,A2# ,0
 43: 4   ,F4  ,F2  ,0   ,0
 44: 4   ,F4  ,C3  ,A2# ,0
 45: 4   ,F4  ,C3  ,A2# ,0
 46: 4   ,E4  ,C3  ,A2# ,0
 47: 4   ,D4  ,D2  ,0   ,0
 48: 4   ,A3  ,F2  ,0   ,0
 49: 4   ,A3  ,A2  ,0   ,0
 50: 4   ,D4  ,D3  ,A2  ,F2
 51: 4   ,D4  ,D3  ,A2  ,F2
 52: 4   ,D4  ,F2  ,0   ,0
 53: 4   ,D4  ,D3  ,A2# ,0
 54: 4   ,D4  ,D3  ,A2# ,0
 55: 4   ,C4  ,D3  ,A2# ,0
 56: 4   ,A3# ,E2  ,0   ,0
 57: 4   ,A3  ,C3  ,A2# ,0
 58: 4   ,A3  ,C2  ,0   ,0
 59: 4   ,G3  ,C3  ,A2# ,0
 60: 4   ,F3  ,F2  ,0   ,0
 61: 4   ,F3  ,C3  ,A2  ,0
 62: 4   ,F3  ,C2  ,0   ,0
 63: 4   ,F3  ,C3  ,A2  ,0

COL: LEN ,V1  ,V2  ,V3  ,V4
 64: 4   ,F3  ,F2  ,0   ,0
 65: 4   ,F3  ,C3  ,A2  ,0
 66: 4   ,F4  ,0   ,0   ,0
 67: 4   ,E4  ,0   ,0   ,0
 68: 4   ,D4  ,D2  ,0   ,0
 69: 4   ,A3  ,F2  ,0   ,0
 70: 4   ,A3  ,A2  ,0   ,0
 71: 4   ,A3  ,D3  ,0   ,0
 72: 4   ,A3  ,E2  ,0   ,0
 73: 4   ,G3  ,C3# ,A2  ,0
 74: 4   ,F3  ,C3# ,A2  ,0
 75: 4   ,G3  ,C3# ,A2  ,0
 76: 4   ,A3  ,F2  ,0   ,0
 77: 4   ,A3  ,C3  ,A2  ,0
 78: 4   ,A3  ,C2  ,0   ,0
 79: 4   ,A3  ,C3  ,A2  ,0
 80: 4   ,A3  ,F2  ,0   ,0
 81: 2   ,A3  ,C3# ,A2  ,0
 82: 4   ,F4  ,0   ,0   ,0
 83: 4   ,E4  ,0   ,0   ,0
 84: 4   ,D4  ,D2  ,0   ,0
 85: 4   ,A3  ,F2  ,0   ,0
 86: 4   ,A3  ,A2  ,0   ,0
 87: 4   ,A3  ,D3  ,0   ,0
 88: 4   ,A3  ,E2  ,0   ,0
 89: 4   ,G3  ,C3# ,A2  ,0
 90: 4   ,F3  ,C3# ,A2  ,0
 91: 4   ,G3  ,C3# ,A2  ,0
 92: 4   ,A3  ,F2  ,0   ,0
 93: 4   ,A3  ,C3  ,A2  ,0
 94: 4   ,A3  ,C2  ,0   ,0
 95: 4   ,A3  ,C3  ,A2  ,0
 96: 4   ,A3  ,F2  ,0   ,0
 97: 2   ,A3  ,C3# ,A2  ,0
 98: 4   ,A3# ,0   ,0   ,0
 99: 4   ,A3  ,0   ,0   ,0
100: 4   ,G3  ,G2  ,0   ,0
101: 4   ,G3  ,D3  ,A2# ,0
102: 4   ,A3  ,D3  ,A2# ,0
103: 4   ,A3# ,D3  ,A2# ,0
104: 4   ,C4  ,E2  ,0   ,0
105: 4   ,D4  ,C3  ,A2# ,0
106: 4   ,D4  ,C2  ,0   ,0
107: 4   ,E4  ,C3  ,A2# ,0
108: 4   ,F4  ,F2  ,0   ,0
109: 4   ,F4  ,C3  ,A2  ,0
110: 4   ,F4  ,C3  ,A2  ,0
111: 4   ,E4  ,C3  ,A2  ,0
112: 4   ,D4  ,D2  ,0   ,0
113: 4   ,A3  ,F2  ,0   ,0
114: 4   ,A3  ,A2  ,0   ,0
115: 4   ,D4  ,D3  ,A2  ,F2
116: 4   ,D4  ,A2# ,0   ,0
117: 4   ,D4  ,F3  ,D3  ,0
118: 4   ,D4  ,F3  ,D3  ,0
119: 4   ,C4  ,F3  ,D3  ,0
120: 4   ,A3# ,G2  ,0   ,0
121: 4   ,A3# ,D3  ,A2# ,0
122: 4   ,G3  ,D3  ,A2# ,0
123: 4   ,F3  ,D3  ,A2# ,0
124: 4   ,G3  ,D2# ,0   ,0
125: 4   ,G3  ,G2  ,0   ,0
126: 4   ,G3  ,A2# ,0   ,0

COL: LEN ,V1  ,V2  ,V3  ,V4
127: 4   ,G3  ,G2  ,0   ,0
128: 4   ,G3  ,D2# ,0   ,0
129: 4   ,G3  ,A2# ,G2  ,0
130: 4   ,A3# ,A2# ,G2  ,0
131: 4   ,A3  ,A2# ,G2  ,0
132: 4   ,G3  ,G2  ,0   ,0
133: 4   ,G3  ,D3  ,A2# ,0
134: 4   ,A3  ,D3  ,A2# ,0
135: 4   ,A3# ,D3  ,A2# ,0
136: 4   ,C4  ,E2  ,0   ,0
137: 4   ,C4  ,C3  ,A2# ,0
138: 4   ,D4  ,C2  ,0   ,0
139: 4   ,E4  ,C3  ,A2# ,0
140: 4   ,F4  ,F2  ,0   ,0
141: 4   ,F4  ,C3  ,A2# ,0
142: 4   ,F4  ,C3  ,A2# ,0
143: 4   ,E4  ,C3  ,A2# ,0
144: 4   ,D4  ,D2  ,0   ,0
145: 4   ,A3  ,F2  ,0   ,0
146: 4   ,A3  ,A2  ,0   ,0
147: 4   ,D4  ,D3  ,A2  ,F2
```

```
148: 4   ,D4  ,D3  ,A2  ,F2
149: 4   ,D4  ,F2  ,0   ,0
150: 4   ,D4  ,D3  ,A2# ,0
151: 4   ,D4  ,D3  ,A2# ,0
152: 4   ,C4  ,D3  ,A2# ,0
153: 4   ,A3# ,E2  ,0   ,0
154: 4   ,A3  ,C3  ,A2# ,0
155: 4   ,A3  ,C2  ,0   ,0
156: 4   ,G3  ,C3  ,A2# ,0
157: 4   ,F3  ,F2  ,0   ,0
158: 4   ,F3  ,C3  ,A2  ,0
159: 4   ,F3  ,C2  ,0   ,0
160: 4   ,F3  ,C3  ,A2  ,0
161: 4   ,F3  ,F2  ,0   ,0
162: 4   ,F3  ,C3  ,A2  ,0
163: 4   ,F4  ,0   ,0   ,0
164: 4   ,E4  ,0   ,0   ,0
165: 4   ,D4  ,D2  ,0   ,0
166: 4   ,A3  ,F2  ,0   ,0
167: 4   ,A3  ,A2  ,0   ,0
168: 4   ,A3  ,D3  ,0   ,0
169: 4   ,A3  ,E2  ,0   ,0
170: 4   ,G3  ,C3# ,A2  ,0
171: 4   ,F3  ,C3# ,A2  ,0
172: 4   ,G3  ,C3# ,A2  ,0
173: 4   ,A3  ,F2  ,0   ,0
174: 4   ,A3  ,C3  ,A2  ,0
175: 4   ,A3  ,C2  ,0   ,0
176: 4   ,A3  ,C3  ,A2  ,0
177: 4   ,A3  ,F2  ,0   ,0
178: 2   ,A3  ,C3# ,A2  ,0
179: 4   ,F4  ,0   ,0   ,0
180: 4   ,E4  ,0   ,0   ,0
181: 4   ,D4  ,D2  ,0   ,0
182: 4   ,A3  ,F2  ,0   ,0
183: 4   ,A3  ,A2  ,0   ,0
184: 4   ,A3  ,D3  ,0   ,0
185: 4   ,A3  ,E2  ,0   ,0
186: 4   ,G3  ,C3# ,A2  ,0
187: 4   ,F3  ,C3# ,A2  ,0
188: 4   ,G3  ,C3# ,A2  ,0
189: 4   ,A3  ,F2  ,0   ,0

COL: LEN ,V1  ,V2  ,V3  ,V4
190: 4   ,A3  ,C3  ,A2  ,0
191: 4   ,A3  ,C2  ,0   ,0
192: 4   ,A3  ,C3  ,A2  ,0
193: 4   ,A3  ,F2  ,0   ,0
194: 1   ,A3  ,C3# ,A2  ,0
195: 2   ,A3  ,C3# ,A2  ,0
```

Listing Two:

```
COL: LEN ,V1  ,V2  ,V3  ,V4
  1: 6   ,D4  ,A3# ,F3  ,0
  2: 8   ,C4# ,A3# ,F3  ,0
  3: 8   ,D4  ,A3# ,F3  ,0
  4: 8   ,F4  ,A3# ,D3  ,0
  5: 4   ,F4  ,A3# ,D3  ,0
  6: 4   ,D4  ,A3# ,F3  ,0
  7: 6   ,F4  ,A3# ,D3  ,0
  8: 8   ,G4  ,A3# ,D3  ,0
  9: 8   ,F4  ,A3# ,D3  ,0
 10: 3   ,A4# ,D4  ,F3  ,0
 11: 4   ,A4# ,D4  ,G3  ,0
 12: 4   ,A4# ,D4  ,A3# ,F3
 13: 4   ,D5  ,D4  ,A3# ,F3
 14: 4   ,A4# ,D4  ,A3# ,F3
 15: 4   ,A4# ,D4  ,A3# ,F3
 16: 4   ,G4  ,D4  ,A3# ,F3
 17: 4   ,C5  ,D4# ,A3  ,F3
 18: 6   ,F4  ,D4# ,A3  ,F3
 19: 8   ,G4  ,D4# ,A3  ,F3
 20: 6   ,A4  ,D4# ,A3  ,F3
 21: 8   ,G4  ,D4# ,A3  ,F3
 22: 4   ,F4  ,D4# ,A3  ,F3
 23: 6   ,A4  ,D4# ,A3  ,F3
 24: 8   ,A4# ,0   ,A3  ,F3
 25: 6   ,B4  ,0   ,A3  ,F3
 26: 8   ,C5  ,0   ,A3  ,F3
 27: 6   ,A4  ,D4# ,A3  ,F3
 28: 8   ,A4# ,0   ,A3  ,F3
 29: 6   ,B4  ,0   ,A3  ,F3
 30: 8   ,C5  ,0   ,A3  ,F3
 31: 6   ,A4  ,D4# ,C4  ,F3
 32: 4   ,G4# ,D4# ,C4  ,F3
 33: 8   ,G4  ,D4# ,C4  ,A3
 34: 4   ,G4  ,D4# ,C4  ,A3
 35: 4   ,G4  ,D4# ,0   ,C3
 36: 4   ,A4  ,D4# ,F3  ,0
 37: 4   ,A4# ,0   ,G3  ,0
 38: 4   ,B4  ,0   ,G3# ,0
 39: 4   ,C5  ,0   ,A3  ,0
 40: 8   ,D5  ,F4  ,A3# ,0
 41: 8   ,C5# ,E4  ,A3# ,0
 42: 8   ,C5  ,D4# ,A3# ,0
 43: 8   ,A4# ,D4  ,A3# ,0
 44: 4   ,A4# ,D4  ,A3# ,0
 45: 4   ,0   ,0   ,A3# ,0
 46: 6   ,D4  ,A3# ,F3  ,0
 47: 8   ,C4# ,A3# ,F3  ,0
 48: 8   ,D4  ,A3# ,F3  ,0
 49: 8   ,F4  ,A3# ,D3  ,0
 50: 4   ,F4  ,A3# ,D3  ,0
 51: 4   ,D4  ,A3# ,F3  ,0
 52: 6   ,F4  ,A3# ,D3  ,0
 53: 8   ,G4  ,A3# ,D3  ,0
 54: 8   ,F4  ,A3# ,D3  ,0
 55: 3   ,A4# ,D4  ,F3  ,0
 56: 4   ,A4# ,D4  ,G3  ,0
 57: 6   ,D5  ,D4  ,G3# ,F3
 58: 4   ,D5  ,D4  ,G3# ,F3
 59: 6   ,C5  ,D4  ,G3# ,F3
 60: 4   ,C5  ,D4  ,G3# ,F3
 61: 4   ,G4  ,D4  ,A3# ,F3
 62: 4   ,A4# ,0   ,0   ,0
 63: 4   ,A4# ,A3# ,G3  ,D3#

COL: LEN ,V1  ,V2  ,V3  ,V4
 64: 4   ,A4# ,C4  ,G3  ,D3#
 65: 4   ,A4# ,D4  ,G3  ,D3#
 66: 6   ,A4# ,D4# ,G3  ,D3#
 67: 8   ,G4  ,D4# ,G3  ,D3#
 68: 8   ,A4# ,D4# ,G3  ,D3#
 69: 8   ,C5# ,F4# ,A3# ,F3
 70: 4   ,C5# ,F4# ,A3# ,E3
 71: 4   ,C5  ,0   ,A3# ,E3
 72: 6   ,A4# ,F4  ,D4  ,F3
 73: 8   ,G4  ,F4  ,D4  ,F3
 74: 8   ,A4# ,F4  ,D4  ,F3
 75: 8   ,A4  ,F4  ,B3  ,G3
 76: 8   ,A4  ,F4  ,B3  ,G3
 77: 4   ,G4  ,F4  ,B3  ,G3
 78: 4   ,D5  ,E4  ,A3# ,C3
 79: 4   ,D5  ,E4  ,A3# ,C3
 80: 4   ,D5  ,D4# ,A3  ,F3
 81: 4   ,D5  ,D4# ,A3  ,F3
 82: 6   ,A4# ,D4  ,A3# ,0
 83: 8   ,A4# ,D4  ,F3  ,0
 84: 6   ,A4# ,D4  ,G3  ,0
 85: 8   ,A4# ,D4  ,F3  ,0
 86: 4   ,A4# ,D4  ,A3# ,0
```

Listing Three:

```
COL: LEN ,V1  ,V2  ,V3  ,V4
  1: 6   ,G3  ,0   ,0   ,0
  2: 6   ,A3  ,0   ,0   ,0
  3: 4   ,A3# ,D3  ,C2  ,0
  4: 4   ,A3# ,D3  ,F2  ,0
  5: 4   ,C4  ,D3# ,C2  ,0
  6: 4   ,C4  ,D3# ,G2  ,0
  7: 4   ,A3  ,F3  ,D2  ,0
  8: 2   ,A3  ,F3  ,A2  ,0
  9: 4   ,G3  ,E3  ,A2# ,C2#
 10: 4   ,A3# ,D3# ,0   ,C2
 11: 4   ,A3# ,D3# ,E2  ,0
 12: 4   ,A3# ,D3# ,G2  ,0
 13: 4   ,A3# ,D3# ,G2  ,0
 14: 8   ,A3# ,D3# ,A2# ,0
 15: 8   ,G3  ,A2# ,0   ,0
 16: 4   ,A3# ,D3# ,F2  ,0
 17: 4   ,A3  ,D3# ,F2  ,0
 18: 4   ,G3  ,D3# ,F2  ,0
 19: 6   ,G3  ,D3# ,F2  ,0
 20: 6   ,A3  ,D3# ,F2  ,0
 21: 4   ,A3# ,D3  ,C2  ,0
 22: 4   ,A3# ,D3  ,F2  ,0
 23: 4   ,C4  ,D3# ,C2  ,0
 24: 4   ,C4  ,D3# ,G2  ,0
 25: 4   ,A3  ,F3  ,D2  ,0
 26: 2   ,A3  ,F3  ,A2  ,0
 27: 4   ,G3  ,E3  ,A2# ,C2#
 28: 4   ,A3# ,D3# ,0   ,C2
 29: 4   ,A3# ,D3# ,E2  ,0
 30: 4   ,A3# ,D3# ,G2  ,0
 31: 4   ,A3# ,D3# ,A2# ,0
 32: 4   ,A3# ,G3  ,A2# ,0
 33: 4   ,A3# ,D3# ,F2  ,0
 34: 4   ,A3  ,D3# ,F2  ,0
 35: 4   ,G3  ,D3# ,F2  ,0
 36: 4   ,A3  ,D3# ,F2  ,0
 37: 2   ,A3# ,D3  ,0   ,0
 38: 6   ,G3# ,0   ,0   ,0
 39: 6   ,A3# ,0   ,0   ,0
 40: 4   ,C4  ,G3# ,D3  ,A2#
 41: 4   ,D4  ,G3# ,D3  ,A2#
 42: 4   ,D4# ,G3# ,D3  ,A2#
 43: 4   ,F4  ,G3# ,D3  ,A2#
 44: 4   ,D4# ,G3  ,0   ,0
 45: 4   ,D4# ,G3  ,A2# ,0
 46: 4   ,D4# ,G3  ,D3# ,0
 47: 4   ,A3# ,0   ,0   ,0
 48: 4   ,A3# ,F3# ,0   ,0
 49: 4   ,A3# ,F3# ,A2# ,0
 50: 4   ,A3# ,F3# ,D3# ,0
 51: 8   ,F3# ,F3# ,D3# ,0
 52: 8   ,G3# ,F3# ,D3# ,0
 53: 4   ,A3# ,F3# ,D3# ,A2#
 54: 4   ,C4  ,F3# ,D3# ,A2#
 55: 4   ,C4# ,F3# ,D3# ,A2#
 56: 4   ,D4# ,F3# ,D3# ,A2#
 57: 4   ,D4  ,F3  ,D3# ,A2#
 58: 4   ,D4  ,F3  ,F2  ,0
 59: 4   ,D4  ,F3  ,D3  ,0
 60: 8   ,C4  ,D3  ,0   ,0
 61: 8   ,D4  ,D3  ,0   ,0
 62: 4   ,D4# ,A3  ,0   ,0
 63: 4   ,D4# ,A3  ,F2  ,0

COL: LEN ,V1  ,V2  ,V3  ,V4
 64: 4   ,D4# ,A3  ,D3# ,0
 65: 8   ,G3  ,D3# ,0   ,0
 66: 8   ,A3  ,D3# ,0   ,0
 67: 4   ,A3# ,D3  ,C2  ,0
 68: 4   ,A3# ,D3  ,F2  ,0
 69: 4   ,C4  ,D3# ,C2  ,0
 70: 4   ,C4  ,D3# ,G2  ,0
```

continued on page 60

MACHINE LANGUAGE PROGRAMMING

LESSON THREE
32K CoCo with EDTASM+

by Malcolm Patrick

This is part three in a four part series of tutorials started from March 1987. To get the most of this tutorial, Malcomb refers heavily to a book from Tandy, "TRS-80 Colour Computer Assembly Language Programming" (Cat no. 62-2077).

LOAD IN THE listing from page 4 of the Radio Shack book and save it in ASCII with the name of "DAYONE" for future reference.

Executing or Running the Program

First load the program "DAYONE" into the assembler and INSERT in line 90 "ORG" for the opcode and "$3FF0" as the operand. This means ORIGINATE the program at hexadecimal location 3FF0 or decimal 16368.

Next change line 260 from " LOOP JMP LOOP " to "(4 spaces)RTS(5 spaces)" in the opcode.

This tells the computer to give you back control (or return from subroutine) after it has performed the above instructions.

This is how DAYTHREE now looks:

```
00090          ORG     $3FF0
 ($3F00 FOR 16K) starting
 address
00100 *  bubble sort
 REM EXEC &H3F00 NOT EXEC
 &H3FF0
00110 BUBSRT   CLR     PASSNO
 SET # PASS TO 0
00120 BUB010   LDX     #$400
 POINT TO SCREEN
00130          LDY     #0
 SET CHANGE FLAG TO 0
00140 BUB020   LDA     ,X+
 GET FIRST ENTRY
00150          CMPA    ,X
 test next
00160          BLS     BUB030
 go if a<=b
00170          LDB     ,X
 get second entry
00180          STB     -1,X
 swap b to a
00190          STA     ,X
 swap a to b
00200          LDY     #1
 set change
00210 BUB030   CMPX    #$400+511
 test for screen end
00220          BNE     BUB020
 go if not one pass
00230          INC     PASSNO
 increment pass #
00240          CMPY    #0
 test change flag
00250          BNE     BUB010
 go if change occurred
00260          RTS
 return to Basic
00270 PASSNO   FCB     0
 pass #
00280          END
```

Absolute Address

Now assemble the listing by typing 'A/IM/WE/AO'. This means 'Assemble In Memory Wait on any Errors at the Absolute Origin of the listing'.

By using the Shift @ keys you will be able to stop the assembling process and check the first line.

Here you will see that the OBJECT ADDRESS is now 3FF0. I have said that the Object Address can sometimes be different, by using /AO it will always be the same.

Saving in Binary

To save the assembled source code to tape with the name of "NAMTHREE" type A NAMTHREE (enter). This will save to tape in a binary form the assembled listing and can only be loaded from Basic by the CLOADM command.

Cloading and Executing From BASIC

Quit EDTASM+ or switch off and remove the ROMpack. From BASIC you can load the machine language program called NAMTHREE by CLOADM"NAMTHREE" (enter).

There will be no listing as this program is now poked into memory beginning at location Hexadecimal 3FF0.

EXECute the program by typing:

```
EXEC&H3FF0
```

... meaning Execute from location $3FF0. The program will sort the screen the same as it did in Zbug.

To see the results better fill the screen with some letters and numbers then type ...

```
EXEC&H3FF0:PRINT@230," "
```

Binary, Hexadecimal and Decimal

To make the programming of machine language easier it is a must that you learn how to convert Decimal to Hexadecimal and vice versa, and be able to convert both to binary and back.

This sounds difficult but remember Practice Makes Perfect.

Don't let these three words frighten you. Look at it this way - there are 16 ounces to the pound, three feet to the yard, 60 minutes to an hour, and many more unusual conversions used in the every day use. Morse code is a complex language but once practiced it is easy to understand.

You may ask, "Why do we need to know how to convert decimal to binary then binary to hexadecimal?"

The answer is that the computer always works on a binary system and you will be programming in either decimal or hexadecimal. Look at line 210 of the listing called "DAYONE" - it tells the computer to compare the X register with 400 hexadecimal + 511 decimal. If you add them together you will get decimal 1535 or hexadecimal 5FF which is the last pixel on the screen.

The 6809 microcomputer uses one byte made up of 8 binary digits. Each digit is either on or off, so that you may understand this byte we represent the on with a one and the off with a zero.

One such byte may look like this - '10110010'.

Decimal to Binary to Decimal

To convert this byte to decimal is not so hard (it totals 178 decimal). I will show you how this is added. Each digit position of an 8-bit binary number represents the power of two.

The "bit position" on the right is 2 to the power of 1.

The next is 2 to the first power of two.

The next is 2 to the second power and so on.

So the right most bit is 1, but it is off so the count = 0.

The second bit is 2 and is on, so that count = 2.

The third bit is 4 and is off, so the count = 0.

The fourth bit is 8 and is of, so that count = 0.

The fifth bit is 16 and is on, so that count = 16.

The sixth bit is 32 and is on, so that count = 32.

The seventh bit is 64 and is off, so that count = 0.

The eighth bit is 128 and is on, so that count = 128.

The total decimal count = 178 (178 = 0+2+0+0+16+32+0+128)

To convert it back to a binary digit is just the reverse.

Does 128 go into 178? Yes, the first digit on left = (1) & 50 left over.

Does 64 go into 50? No, the second digit from left = (0) & 50 left over.

Does 32 go into 50? Yes, the next digit = (1) & 18 left over.

Does 16 go into 18? Yes, the next digit = (1) & 2 left over.

Does 8 go into 2? No, the next digit = (0)

Does 4 go into 2? No, the next digit = (0).

Does 2 go into 2? Yes, the next digit = (1).

And there is still one digit left so that must be off (0).

Read from the first answer to the last answer (top to bottom) and your result will be '10110010'

Read page 16 and 17 of your Radio Shack assembly book and take plenty of time to understand this subject. If you don't learn this to a reasonable degree you will only stumble through assembly language programming.

Hexadecimal to Binary to Hexadecimal

The 6809 chip is made up of 8 bits - four of these bits can be arranged in 16 different ways.

Decimal	Binary	Hexadecimal
0	0000 0000	0
1	0000 0001	1
2	0000 0010	2
3	0000 0011	3
4	0000 0100	4
5	0000 0101	5
6	0000 0110	6
7	0000 0111	7
8	0000 1000	8
9	0000 1001	9
10	0000 1010	A
(note the change in alphabet)		
11	0000 1011	B
12	0000 1100	C
13	0000 1101	D
14	0000 1110	E
15	0000 1111	F
16	0001 0000	10

If you wanted to continue on until we had all bits switched on (11111111) you would find there are 256 different combinations.

Note that "F" equals "1111", "E" equals "1110" and "A" equals "1010".

Therefore "FF" must equal "11111111" and "AF" will equal "10101111" and "E4" will equal "11100100".

Learn how to count in hex (hexadecimal). Each night instead of counting sheep count in hex!

To do this you say each digit separately; One-One, One-Two, One-Three, ... up to One-A, One-B, One-C.

Continue until F-One-Two, F-One-Three.

Now learn to count backwards; One-One-F, One-One-E, One-One-D, and so on.

Read page 18 to 20 of your Radio Shack assembly book. Study it well.

Look at listing DAYTHREE.

In line 120 you have "#$400". This is telling you to point to screen location HEX 400.

How do we know this? Because the "$" after the "#" (sharp or hash sign) tells us. If it was decimal there would not be a dollar sign there.

HEX 400 = decimal 1024.

We could change the operand in line 120 to read "#1024".

How have we come by this number - "1024"?

If we convert hex400 to binary it would look like this.

```
0100 0000 0000
  =    =    =
  4    0    0
```

Now to convert the binary to decimal.

```
0 1 0 0     0 0 0 0     0 0 0 0
= = = =     = = = =     = = = =
2 1 5 2     1 6 3 1     8 4 2 1
0 0 1 5     2 4 2 6
4 2 2 6     8
8 4
```

Questionnaire for Lesson Three

Q1: If we wanted to start executing the program called DAYTHREE at $7F00, what would we put in line 90?

A1: .....

Q2: What would line 210 in DAYTHREE look like if it was in decimal?

A2: .....

Q3: What is binary 0111 0010 in decimal?

A3: .....

Q4: How would you save DAYTHREE in binary?

A4: .....

Q5: How would you say hex101F? (Write it down as you would say it.)

A5: .....

Q6: Is "$1010" hex, binary, or decimal? How can you tell?

A6: .....

Q7: Is "#1010" hex, binary, or decimal? How can you tell?

A7: .....

Q8: Is "#$1010" hex, binary, or decimal? How can you tell?

A8: .....

Q9: If you saw this - "$FACE" - in the operand would it be a call to the label, a hexadecimal address, a decimal address or something else? How can you tell?

A9: .....

Q10: If you saw this - "FACE" in the operand would it be a call to the label, a binary address, a decimal address or something else? How can you tell?

A10:
.....

Q11: Convert these binary numbers into decimal:

a. 01011010 b. 01101101
c. 00101011 d. 10001000
e. 00100010

A11: a: ..... b: .....

c: ..... d: .....

e: .....

```
Q12:  Convert these binary
numbers into hex:
  a. 01011010  b. 01101101
  c. 00101011  d. 10001000
  e. 00100010

A12: a: ........  b: ........

     c: ........  d: ........

     e: ........
Q13: Convert these numbers into
decimal:
  a. $12A0   b. $A012  c. $0004
  d. $0F0F   e. $F0F0

A13: a: ........  b: ........

     c: ........  d: ........

     e: ........
Q14: Convert these numbers into
hex:
  a. #32400  b. #65495  c. #0009
  d. #3854   e. #65535

A14: a: ........  b: ........

     c: ........  d: ........

     e: ........
Q15:  What address does the
"/AO" (as in "A/IM/AO") change?

A15: ..........................
Q16:  What does the command
"/WE" do?

A16: ..........................
Q17:  What is the end of screen
address? (Tell me in decimal and
hex!)

A17: ..........................
Q18: In line 210 of "DAYTHREE",
could you change any part (add
to or delete) and still have the
same answer? If so, what would
you do?

A19: ..........................
```

Answers for Lesson Three

```
Q1:  If we wanted to start
executing the program called
DAYTHREE at $7F00 what would we
put in line 90?
A1: A1 ORG  $7F00

Q2:  What would line 210 in
DAYTHREE look like if it was in
decimal?
A2: BUB030  CMPX  #1535

Q3: What is binary 0111 0010 in
decimal?
A3: 114

Q4: How would you save DAYTHREE
in binary?
A4: A DAYTHREE

Q5: How would you say hex101F?
(Write it down as you would say
it.)
A5 ONE ZERO ONE F

Q6: Is "$1010" hex, binary, or
decimal? How can you tell?
A6:  Hexadecimal.  How?  The
Dollar sign before the number.

Q7: Is "#1010" hex, binary, or
decimal? How can you tell?
A7:  Decimal.  How?  The sharp
sign and no dollar sign.

Q8: Is "#$1010" hex, binary, or
decimal? How can you tell?
A8:  Hexadecimal.  How?  The
dollar sign is present.

Q9: If you saw this - "$FACE" -
in the operand would it be a
call to the label, a hexadecimal
address, a decimal address or
something else? How can you
tell?
A9:  Hexadecimal.  How? By the
dollar sign.

Q10:  Convert these binary
numbers into decimal:
  a. 01011010  b. 01101101
  c. 00101011  d. 10001000
  e. 00100010
A10: a.#90   b.#109  c.#35
     d.#136  e. #34

Q11:  Convert these binary
numbers into hexadecimal:
  a. 01011010  b. 01101101
  c. 00101011  d. 10001000
  e. 00100010
A11: a.$5A  b.$6E  c.$2B
     d.$88  e.$22

Q12: Convert these numbers into
decimal:
  a. $12A0  b. $A012
  c. $0004  d. $0F0F
  d. $F0F0

A12: a.#4786  b.#40978
     c.#4     d.#3855
     e.#61680

Q13: Convert these numbers into
hex:
  a. #32400  b. #65495
  c. #0009   d. #3854
  e. #65535
A13: a. $7E90  b. $FFD7
     c. $9     d. $F0E
     e. $FFFF

Q14:  What address does the
"/AO" (as in "A/IM/AO") change?
A14:  The address in the Object
code (Object Address).

Q15:  What does the command
"/WE" do?
A15: Wait on any Errors.

Q16:  What is the end of screen
address? Tell me in decimal and
hex.
A16: Dec: #1535   Hex:$5FF

Q17:  In line 210 of "DAYTHREE",
could you change any part (add
to or delete) and still have the
same answer? If so what would
you do?
A17:  Change the operand to
#1535 or to $5FF.

Q18:  If you saw this - "FACE" -
in the operand would it be a
call to the label, a binary
address, a decimal address or
something else? How can you
tell?
A18:  It would be a call to the
Label. It has no Dollar sign
before it.
```

■

ALL MY LOVING

continued from page 57

```
71: 4   ,A3  ,F3  ,F2  ,0
72: 2   ,A3  ,F3  ,A2  ,0
73: 4   ,G3  ,E3  ,D2  ,C2#
74: 4   ,A3# ,D3# ,C2  ,0
75: 4   ,A3# ,D3# ,D2# ,0
76: 4   ,A3# ,D3# ,G2  ,0
77: 8   ,A3# ,D3# ,D2  ,0
78: 8   ,G3  ,0   ,D2  ,0
79: 4   ,A3# ,D3# ,F2  ,0
80: 4   ,A3  ,D3# ,F2  ,0
81: 4   ,G3  ,D3# ,F2  ,0
82: 4   ,A3  ,D3# ,F2  ,0
83: 4   ,A3# ,D3  ,C2  ,0
84: 4   ,A3# ,D3  ,F2  ,0
85: 4   ,C4  ,D3# ,C2  ,0
86: 4   ,C4  ,D3# ,G2  ,0
87: 4   ,A3  ,F3  ,D2  ,0
88: 2   ,A3  ,F3  ,A2  ,0
89: 4   ,G3  ,E3  ,D2  ,C2#
90: 4   ,A3# ,D3# ,C2  ,0
91: 4   ,A3# ,D3# ,E2  ,0
92: 4   ,A3# ,D3# ,G2  ,0
93: 8   ,A3# ,D3# ,A2# ,0
94: 8   ,G3  ,A2# ,0   ,0
95: 4   ,A3# ,D3# ,F2  ,0
96: 4   ,A3  ,D3# ,F2  ,0
97: 4   ,G3  ,D3# ,F2  ,0
98: 4   ,A3  ,D3# ,F2  ,0
99: 4   ,A3# ,D3  ,0   ,0
100: 4  ,A3# ,D3  ,F2  ,0
101: 8  ,A3# ,D3  ,D2  ,0
102: 8  ,F3  ,D2  ,0   ,0
103: 8  ,F3  ,D2  ,0   ,0
104: 4  ,F3# ,D3# ,D2  ,F2#
105: 4  ,G3# ,D3# ,D2  ,F2#
106: 4  ,A3# ,D3# ,D2  ,F2#
107: 4  ,C4  ,D3# ,D2  ,F2#
108: 4  ,D4  ,A3# ,F3  ,0
109: 4  ,D4  ,A3# ,F3  ,F2
110: 1  ,D4  ,A3# ,F3  ,D3
```

■

ECHO SONG

APPLICATION

16K ECB by Craig Stewart

ECHOSONG is a small experiment using a short sound program that allows the user to input songs and play them so that they sound like they're being played in an echo-chamber.

Although extremely simple in operation (simply play a sound and then play the same sound at a lesser volume afterwards) I found it quite pleasant to listen to - it gives songs a 'grand' touch.

Within the program, I have put several songs in data statements (I really apologize for the songs, but I am not overstocked with music books - I am sure someone else could come up with something much better).

To enter your own songs, put them in data statements with commas between every note and end the song with a "ZZ".

The Listing:

```
0 GOTO5
1 '********** ECHOSONG *********
   ****** CRAIG STEWART ********
2 SAVE"146E:3":END'10
5 CLS3:PRINT@198,"ECHO SONG EXPE
RIMENT";
10 DIM A$(70,3)
15 PLAY"L2":POKE65495,0
20 L(1)=38:L(2)=64
25 READ A$:IF A$="ZZ" THEN CLS:E
ND ELSE PLAY"V31"+A$:IF A$>="A"
AND A$<="G" THENPLAY"V10"+A$:GOT
O 25
30 GOTO 25
35 DATA O2L35,D,G,L25,B,L35,B,A,
L25,G,L35,A,G,E,L10,G
40 DATA L35,G,G,L25,B,L35,G,B,L2
503,D,L3503,C,O2,B,L10,A,L3503,D
,C,O2L25,B,L35,B,A,L25,G,L35,A,B
,O3,D,L10,C,L3502,E,E,L25,D,L35,
F,G,L25,A,L35,B,A,L10,G
45 DATA P1L35,C,E,L25,G,G,G,G,L1
5,G,L3503,C,O2,L25,G,L35,E,L40,F
,L25,G,G,F,D,L10,C,L35,C,E,L25,G
,G,G,G,L15,G,L3503,C,O2L25,G,L35
,E,L40,F,L25,G,G,F,D,L10,C
50 DATA O3L35,C,O2L25,B,A,F,A,F,
L20,G,L35,A,L25,G,O3L35,C,O2,B,L
25,A,F,A,O3L20,C,O2L10,G,L35,C,E
,L25,G,G,G,G,L15,G,L3503,C,L2502
,G,L35,E,F,L25,G,G,F,D,L10,C
55 DATA ZZ
```

Zooming 'round memory with...

TAPE ZAPPER

16K ECB UTILITY

by Justin Lipton

THE IDEA of Tape Zapper is to load a machine language program into memory and alter it. You can push the 'N' button to go to the next block of memory, or push 'B' to go back into memory.

If you want to alter the contents of memory use the arrow keys to move your cursor around. Pressing 'C' will allow you to change that part of memory.

At anytime pressing 'F' will end the program and pressing 'S' will save the entire program, modifications and all.

Mind you, not all ML programs will work entirely with this program.

The Listing:

```
0 GOTO10
1 '******** TAPE ZAPPER ********
2 '****** JUSTIN LIPTON ********
3 SAVE"109A:3":END
10 CLS
20 PRINT"TAPE ZAPPER"
30 INPUT"NAME OF M.L. PROGRAM";A
$
40 CLOADM A$
50 CLS
60 ST=PEEK(487)*256+PEEK(488)
70 ED=PEEK(126)*256+PEEK(127)-1
80 EX=PEEK(157)*256+PEEK(158)
90 PRINT @ 352,"START ADDRESS;"S
T:PRINT @ 384,"END ADDRESS;"ED:P
RINT @ 416,"EXECUTE ADDRESS;"EX
100 L=ST:P=96
110 FOR A=L TO L+254
120 B=PEEK(A):G$=CHR$(B)
130 PRINT @ P,G$;:P=P+1
140 PRINT @ 13,"LOCATION;"A
150 IF A>=ED THEN PRINT @ 0,"END
...";:L=ED-255:GOTO 180
160 IF A+255=ST THENPRINT @ 0, "
START....";:L=ST:GOTO 180
170 NEXT A
180 T$=INKEY$
190 IF T$="S" THEN 380
200 IF T$="E" THEN END
210 IF T$="R" THEN RUN
220 IF P<97 THEN240
230 IF T$=CHR$(8) THEN P=P-1:A=A
-1:PRINT @ P,CHR$(143);:B=PEEK(A
+1):G$=CHR$(B):PRINT @ P+1,G$;:G
OTO 180
240 IF P>349 THEN 260
250 IF T$=CHR$(9) THEN P=P+1:A=A
+1:PRINT @ P,CHR$(143);:B=PEEK(A
-1):G$=CHR$(B):PRINT @ P-1,G$;:G
OTO 180
260 IF T$="C" THEN 360
270 IF T$="N" THEN L=L+255:P=96:
CLS:GOTO 110
280 IF T$="B" THEN L=L-255:P=96:
CLS:GOTO 110
290 IF P<128 THEN 310
300 IF T$="^" THEN B=PEEK(A):PRI
NT @ P,CHR$(B);:P=P-32:A=A-32
310 IF P>319 THEN 330
320 IF T$=CHR$(10)THEN B=PEEK(A)
:PRINT @ P,CHR$(B);:P=P+32:A=A+3
2
330 PRINT @ P,CHR$(143+64);:PRIN
T @ P,CHR$(143);
340 PRINT @ 13,"LOCATION;"A
350 GOTO 180
360 C$=INKEY$:IF C$="" THEN 360
370 C=ASC(C$):POKE A,C:GOTO 180
380 CLS:PRINT"GET TAPE READY."
390 PRINT"PRESS PLAY AND RECORD.
"
400 LINE INPUT"NEW NAME OF PROGR
AM-";O$
410 CSAVEM O$,ST,ED,EX:CLS
420 GOTO 180
```

The Things we do!

As many have noticed, some time ago we introduced a recommended Standard for the way programmers start their programs.

We recommend that the following proceedure be adopted for all your programming:

```
1 '****Program Name************
   ********Author**************
   *********Date***************
2 GOTO 10
3 (C)SAVE "Program Name"
4 - 9 Further REM's re program
10 Start of program
```

With the introduction of the CoCo 3, we will be prefixing all program file names for the CoCo 3 with the numeral 3.

So on a disk from us a program in the directory which DIR's as "3DONKEY/BIN" would be a program for the CoCo 3, and a program which DIR's as "DONKEY/BIN" would also work on the CoCo 2's & 1's.

The same principle will apply to tape file names.

We ask that at least with the programs you submit to us you adhere to this convention.

SUBMARINE BATTLE

16K ECB + joystick
GAME

by Justin Lipton

SUBMARINE BATTLE IS A game which was written in PMODE 3. It involves an aircraft carrier (which you control with the right joystick) and two submarines beneath you.

The object is to try and score points by dropping depth charges (by holding your finger on the fire button) and hitting the submarines below.

You also avoid blowing up by missles fired by the submarines. You have only five depth chrges and one carrier - good luck captain.

The Listing:

```
0 CLEAR100:POKE65495,0
1 DIMV(35),W(36),X(41),Y(6),Z(5)
:GOTO2
2 '**** SUBMARINE BATTLE *******
   ******* JUSTIN LIPTON *******
3 SAVE"134B:3":END'1
6 Q$-"1;2;3;4;5;6;7;8;9;10;11;12
"
10 PMODE3,1:PCLS3
20 DRAW"BM100,100;C2R10E5U5H5L10
U3L8D3L6U8L4D2R2D6L2G5D5F5R20"
30 PAINT(105,95),4,2
40 DRAW"C2":LINE(77,90)-(114,90)
,PSET:LINE(77,93)-(115,93),PSET
50 FOR A= 77 TO 112 STEP 9
60 DRAW"C1":LINE(A,90)-(A+3,93),
PSET,BF
70 NEXTA
75 GET(70,76)-(120,101),V,G
80 DRAW"BM 200,100;C2L15G3D3F3R5
0E3U3H3L15U8L3D2R1D6L4U3L13D3"
90 PAINT(205,105),4,2
100 DRAW"C2":LINE(182,103)-(238,
106),PSET,B
110 FORA=182TO235STEP9
120 DRAW"C1":LINE(A,103)-(A+3,10
6),PSET,BF
130 NEXTA
135 GET(177,91)-(244,110),W,G
140 DRAW"BM 50,50;C2F5R70E5L20U5
L10D5L3U10L4D10L15U3R8U3L13D5L15
"
150 PAINT(60,51),2,2
180 DRAW"C2":LINE(20,20)-(150,60
),PSET,B
190 PAINT(25,25),1,2
200 DRAW"C4":LINE(52,52)-(129,52
),PSET
210 DRAW"BM89,53;C3G2BR5BU2F2"
215 GET(45,40)-(136,56),X,G
220 DRAW"BM 200,30;C2G2D5R5U5H2D
4C4D2"
225 GET(194,26)-(206,40),Y,G
230 DRAW"BM 30,150;C2L5D4R5U4"
240 PAINT(29,153),4,2
245 GET(22,145)-(32,157),Z,G
247 GOTO 3000
250 PCLS3:SCREEN1,0
251 DRAW"C1":LINE(0,0)-(256,60),
PSET,BF
255 CIRCLE(0,0),20,2,1,0,.25
256 PAINT(3,3),2,2
270 A=125:C=A+91:B=5:D=B+50:E=6:
F=180:G=F+67:I=6:R=61:S=R+12:MEN
=1:DC=5:SC=0:MISS=0:BULL=0
280 H=JOYSTK(0)
285 IFA>157THEN295
290 IFH=63THENA=A+5:C=A+91
295 IFA<5THEN310
300 IFH=0THENA=A-4:C=A+91
310 PUT(A,44)-(C,60),X,PSET
320 PUT(B,120)-(D,145),V,PSET
330 B=B+E:D=B+50
335 IFB<10THENE=6
340 IFB=>200THENE=-4
350 PUT(F,170)-(G,189),W,PSET
360 F=F+I:G=F+67
370 IFG>250THENI=-4
375 IFMISS=1THENMISS=1:IFPPOINT(
L+6,M-2)=2THEN MEN=MEN-1:DRAW"C3
":LINE(L,M)-(L+12,M+14),PSET,BF:
MISS=0:GOTO1000
380 IFG<71THENI=6
385 IFBULL=1THEN395
390 IFPEEK(65280)=254ORPEEK(6528
0)=126THENSOUND1,1:BULL=1:ST=1:D
C=DC-1
391 IFDC=-1THEN A$="OUT OF AMMUN
ITION":GOTO 3000
395 IFBULL=1ANDST=1THENT=A+41:ST
=0:R=61:S=R+12
400 IFBULL=1THENPUT(T,R)-(T+10,S
),Z,PSET
405 IFBULL=1ANDPPOINT(T+10,S-2)(
>3THENBULL=0:DRAW"C3":LINE(T,R)-
(T+10,S),PSET,BF:GOTO 2000
410 IFBULL=1THENR=R+5:S=R+12:PLA
Y"T75AE"
420 IFS>208THENBULL=0
425 IFMISS=1THEN440
430 IFRND(10)>8THENMISS=1:SO=1
440 IFMISS=1ANDSO=1THENL=B+19:SO
=0:M=112
450 IFMISS=1THENPUT(L,M)-(L+12,M
+14),Y,PSET
460 IFMISS=1THENM=M-3:PLAY"T1000
4G"
470 IFMISS=1ANDM=58THENDRAW"C3":
LINE(L,M+3)-(L+12,M+14),PSET,BF:
MISS=0
800 GOTO280
1000 FORWW=1TO7:PLAY"O5L242V15":
PLAYQ$:NEXTWW:PUT(A+1,45)-(C-1,5
9),X,PSET:FOREE=1TO460:NEXTEE
1002 FORFF=1TO3:FORXX=1TO10:PLAY
"L190O1V28CV29DV20EV10F#V21G#V16
GV3":NEXT XX,FF
1003 PLAY"T2L20O1V31BV28AV24GV20
FV16DV10C#V5C"
1005 IFMEN=0THEN A$="YOU GOT BLO
WN UP":GOTO 3000
1010 GOTO280
2000 FORWW=1TO7:PLAY"O5L242V15":
PLAYQ$:NEXTWW
2005 FOREE=1TO460:NEXTEE
2007 IFR<145THENPUT(B+1,121)-(D-
1,144),V,PSET:SC=SC+5:ELSEPUT(F+
1,171)-(G-1,188),W,PSET:SC=SC+10
2010 PLAY"T2P4V15L16O4CEGL8O5CL1
6O4AL4O5C"
2020 GOTO280
3000 CLS
3001 IF SC>HS THEN HS=SC
3002 PRINT @ 8, A$;
3005 PRINT @ 75,"SUB BATTLE";
3007 PRINT @143, "BY";
3010 PRINT @ 202,"JUSTIN LIPTON"
;
3015 PRINT @ 330,"LAST SCORE;"SC
3020 PRINT @ 362,"HIGH SCORE;"HS
3040 PRINT @ 423,"PRESS FIRE TO
BEGIN";
3045 SCREEN 0,1
3046 FOR EE=1TO460:NEXTEE
3050 PP=PEEK(65280)
3060 IF PP=254 OR PP=126 THEN GO
TO 250 ELSE GOTO 3050
```

■

Hi everyone! I felt that since I had some room in the corner of the page here, I might give you a short program. I'm not sure what good it is, but it looks kinda pretty!

```
10 X=X+.2
20 PRINTTAB(SIN(X)*10+10);
"HELLO THERE!"
30 GOTO 10
```

■

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

Best of CoCoOz #5 ADVENTURES

ADV 32K ..... S. RAYNER
QUEST ..... TONY PARFITT
LABYRINTH ..... JAMES REDMOND
ADV + ... SEAN LOVE
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 STEPHENSON
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**

GOLDSOFT

P.O. BOX 1742, SOUTHPORT. QLD. 4215 Phone (075) 510 015

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 ( )

MicoOz Tape (MC-10 Programs for your Mico):
12 Months $75.00 ( )
6 Months $42.00 ( )
1 Month $10.00 ( )

The CoCo3 Tape/Disk:
1 - Tape: $10.00 Disk: $16.00 ( )
2 - Tape: $10.00 Disk: $16.00 ( )

"Say the 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:	-	( )

Special offer: Buy any 2 "Best of" and pay only $28.00!

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: .....

5-87

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

user CONTACTS

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	ARIH 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	KEVIN WINTERS	046.66.8068
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 48 7031
PARKES	DAVID SMALL	068 62 2682
PORT MACQUARIE	RON LALOR	065 83 8223
SPRINGWOOD	DAVID SEAMONS	047 51 2107
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 1831
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
NORTH	STEVEN EISENBERG	08 250 6214
MORPHETTVALE	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 BAMES	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
CHURCHILL	GEOFF SPOVART	051 22 1389
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	GEORGE FRANCIS	051 34 5175
SALE	BRYAN McHUGH	051 44 4792
SHEPPARTON	ROSS FARRAR	058 25 1007
SMYTHESDALE	TONY PATTERSON	053 42 8815
SVAN HILL	BARRIE GERRAND	050.32.2838
TONGALA	TONY HILLIS	058 59 2251
TRARALGON	LEIGH DAVES	051 74 5552
WONTHAGGI	LOIS O'MEARA	056 72 1593
YARRAWONGA	KEN SPONG	057 44 1468

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	FRED BISSELING	064 52 3263

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:

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
WAGGA WAGGA	CES JENKINSON	069 25 2263

CHRISTIAN USERS' GROUP:

COLLIE	RAYMOND L. ISAAC	097 34 1578

300 BAUD BULLETIN BOARDS

SYDNEY:

INFOCENTRE	02 344 9511
TANDY ACCESS	02 625 8071
THE COCOCONNECTION	02 618 3591
HUB	002 49 4405

VIDEOTEX SYSTEMS

VIATEL	01955
MOUSETEX	059 42 5528
VTX 4000	03 741 3295

TANDY INFO ON VIATEL

GOLDLINK	VIATEL #642#
BLAXLAND COMPUTER SERVICES	VIATEL #64263#
PARIS RADIO	VIATEL #64268#
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

SOFTGOLD MAGAZINE
Registered by Australia Post —
Publication No. QBG 4009
AUSTRALIAN CoCo
Publication No. QBG 4007
P.O. BOX 1742
SOUTHPORT, QLD. Australia. 4215

GOLDLINK

COM. STATION 692

ON

Telecom VIATEL®
AUSTRALIA'S NATIONAL VIDEOTEX SERVICE

BULLETIN BOARDS
TANDY COMPUTERS
ATARI COMPUTERS
COMMODORE COMPUTERS
FAST FLORIST
REVIEWS
HARDWARE & SOFTWARE
SHOP

YES WE EVEN TALK BACK !!