Registered by Australian Post — Publication No. QBG 4009

NEW GUINEA K5.95

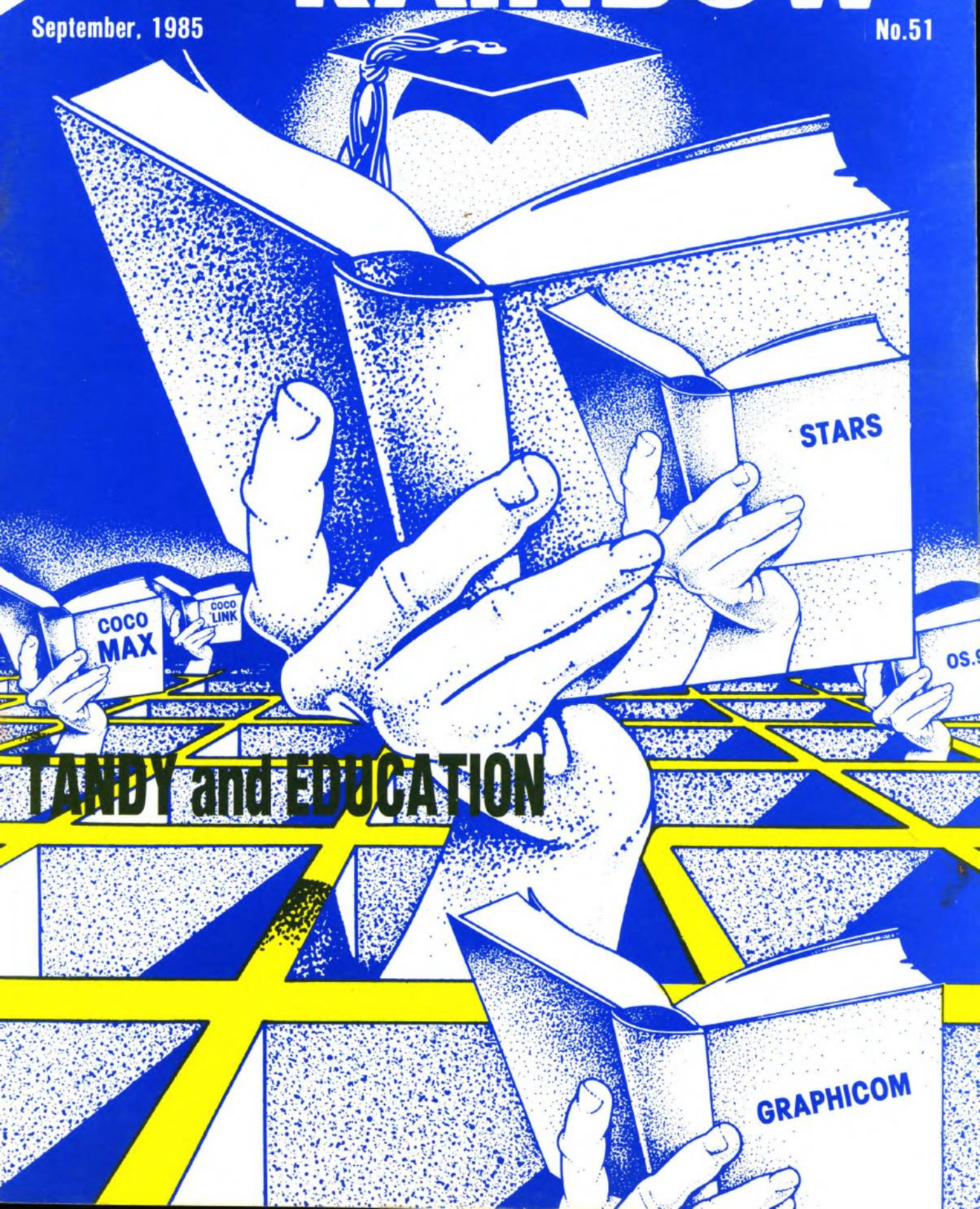
$4.50

for your TANDY Color Computer

AUSTRALIAN RAINBOW

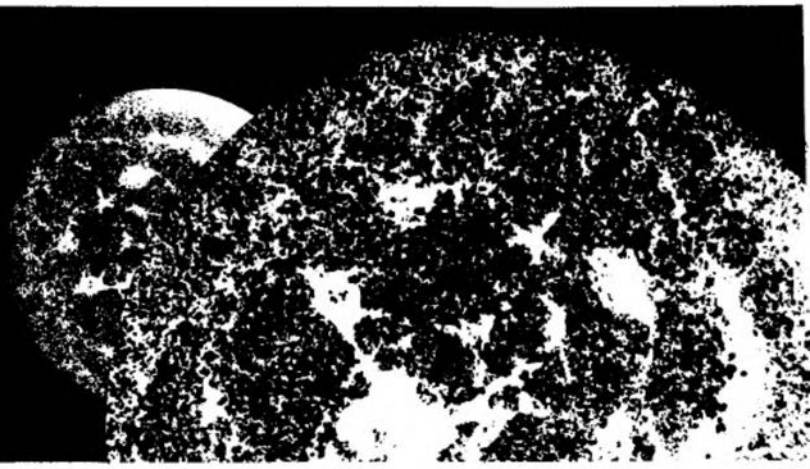
September, 1985

No.51



TANDY and EDUCATION

AUSTRALIA LEADS THE WORLD IN INFORMATION TECHNOLOGY WITH...



★★★★

a sophisticated easy to use DATABASE

As a teacher you have probably wondered how you, let alone your students are going to cope with the information explosion. You understand that human knowledge is expanding so fast that not even the largest book publishers can keep up...text books in most of the sciences are out of date even before they come off the printing press.

Now for the first time you as a teacher can utilise the world's most advanced educational on-line database of knowledge. Scholastic Text and Retrieval System is the world's first on-line database not only incorporating the full text of a major dictionary (The Macquarie Dictionary) but as well, totally dedicated to providing a means for teachers and students alike to manage and control the information explosion.

In seconds with STARS you are able to retrieve information on your computer in the library or classroom, that normally would have taken hours if not days to research. As well as this, the whole system is able to be updated daily if need be...something you could never hope to do with books.

★★★★

STARS — a world first with the Macquarie Dictionary

Computing 2000 Corporation makes world history in information technology — STARS the world's first and only database incorporating the whole of a dictionary (Macquarie Dictionary) stored on computer. Millions of subjects at your fingertips with greater subject coverage than the expensive reference books. This revolutionary NEW service is available to subscribers on an annual basis, and is combatable with most home and office computers. All that is required to hook up to STARS is a modem (a device which allows computers to communicate over the telephone). Modems can be purchased at a moderate cost from any local computer store.

★★★★

Unbelievable results with the incredible information manager "STARS"

"I used STARS to find the names of different cattle breeds...it was fantastic, within seconds my computer was telling everything I wanted to know...it would have taken me hours to find the same information in the library". — SCHOOL STUDENT.

"I created a list on my floppy disk of all the articles on aboriginal languages and replayed it on the classroom computer...it brought to life the subject to all the students with a degree of motivation I couldn't have hoped for in any other media". — TEACHER.

...STARS, an invaluable classroom resource that never ceases to amaze teachers and students alike.

"The thing I like about STARS is it opens up the use of the computer to almost any subject area. As well it gives students practical experience in operating a real on-line database...something they will be required to do in the workplace of the near future". — SCHOOL LIBRARIAN.

SPECIAL OFFER

Imagine! Sitting at your computer in Australia and within seconds communicating via satelite to London, Paris, Tokyo, or New York for less than a local postage charge.

Imagine spanning the four corners of the world with the computer to computer communication and having you and your students learn first hand from the other students what it's like to live in other countries with greater ease than visiting a school in your next suburb

★★★★

FREE REGISTRATION TO STARS SUBSCRIBERS

Now C.T.C. makes what was only a dream, a reality. Through your registration to STARS you can now have access to the world's largest private international Data Communication Network. Turn your computer not only into an advanced high-tech information centre but as well a sophisticated international communication device.

★★★★

STARS — world's first

KNOWLEDGE AT YOUR FINGERTIPS IN SECONDS FOR USE IN THE HOME, OFFICE OR SCHOOL

Manage the information explosion with the world's most advanced on-line computer system placing knowledge at your fingertips. Save hours on research for business or school.

AVERAGE SEARCH TIME ONLY 2-3 SECONDS. YOU PAY ONLY FOR THE TIME CONNECTED.

CTC

Computing 2000 Corporation Pty Limited

44 Hunter Street, Sydney, NSW

STARS

SCHOLASTIC TEXT AND RETRIEVAL SYSTEM

SPECIAL OFFER*

$49.95 registration fee

$14.00 an hour connect time

SPECIAL OFFER THROUGH TANDY STORES ONLY!

$29.95 for 6 months registration

Hourly rate unchanged

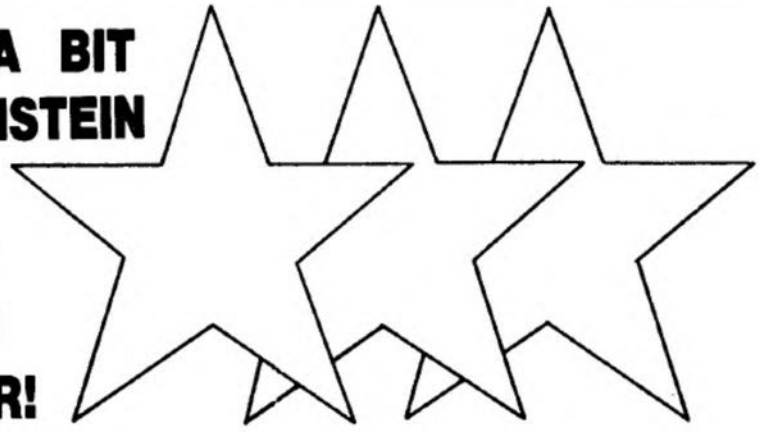
For use by researchers, copywriters, secretaries, executives, students, teachers, scientists, etc.

* Special offer only available on Bankcard, Mastercard or Visa.

* For users outside capital cities an additional $6.00 an hour for local phone call access.

* Special offer available to schools only on account.

ADD A BIT OF EINSTEIN TO YOUR BRAIN POWER!



BONUS OFFER

Included as a special bonus in your registration to STARS is free registration to the International Computer to Computer Network.

MAKE YOUR OWN MINI-DATABASES ON ALMOST ANY SUBJECT.

No extra charge for printing out or storing information on your computer's floppy disk.

SAVE MONEY ON ON-LINE CHARGES

Information stored on your floppy disk can be replayed at any time at no charge.

See STARS at a TANDY store near you!

registration form

Mr (Please print)
Mrs
Miss
Ms .....
Surname Given Names

Address .....

City / State .....

Home Tel.: ..... Bus. Tel.: .....

Date of Birth / / Occupation .....

Indicate the primary use of STARS with a tick:

☐Father ☐Mother ☐Son(s) ☐Daughter(s) ☐Others .....

How many people in the family will be using the Service? .....

Specify type of personal computer you will be using .....

Payment method: (Tick one) ☐Bankcard ☐Mastercard ☐Visa ☐School Account ☐Company Account ($6 invoice fee applies)

Please debit my account for sum of $49.95 and $14 an hour or part thereof being the hourly charge to be debited monthly.

Account No. ..... Expiry Date / /

Authorisation Signature: .....

Mail to PO BOX 314, GPO, NSW, 2000. Prices as at 1st January, 1985. Prices subject to change.

AUSTRALIAN

RAINBOW

index

UPPER CASE - ARTICLE ONLY
UPPER / lower case - Program
Programs are available already typed
up on Rainbow on Tape

AUSTRALIAN RAINBOW Publisher and Editor Graham Morphett. Co- editor Kevin Mischewski. Assistant Editor Sonya Young. With grateful assistance from Brian Dougan, Richard and Judy, Bob Thomson, Paul Humphries, Alex Hartmann, Michael Horn, Jim and Sheryl Bentick, Annette Morphett. Cover Art Jim Bentick.
ADVERTISING DEADLINES: The 7th of the preceeding month of publication. All advertising is arranged through ToToAdvertising, PO Box 8730, Gold Coast Mail Centre, Qld, 4217.
OS 9: Kevin Holmes is the contact for os 9 information. He also has access to OS 9 Software from the US. His address is: 39 Pearson St., Narara, NSW, 2250.
All programmes in this issue of AUSTRALIAN RAINBOW are available on Rainbow on Tape.

Telephone: 075 51 0015 Voice; 075 32 6370 CoCoLink.
Printed by: Australian Rainbow Magazine PO Box 1742, Southport, Qld, 4215.
Founder GREG WILSON.

This is a special issue for several reasons.

The first is that this is our Education issue.

The use of the Colour Computer as a tool in schools has grown dramatically over the last year.

There are a number of factors involved in this growth. They include:

1. The support Tandy has given present education users;
2. The competitive price of the colour computer - dollar for dollar, you can get more hands on computers more quickly when you buy CoCos for your school;
3. The breadth of external support for the CoCo;
4. The growing dissatisfaction with many of the existing brands;
5. The realisation that CoCo can not only work in the classroom, but also in the office and library, making it the total answer to the school's computing needs.
6. The physical strength of a CoCo. Many CoCos live in schools for years without any maintainence!

We've been keeping some great programs specially for this magazine, so if you are involved in education, whether you teach, administer a school, or just want to help your own children at home, there's something in here for you!

The second reason this issue is special, is that Tandy have decided to stock our magazine in all their stores and this issue is first cab off the rank!

So we welcome you Tandy, and your staff.

Many of you wont have seen our magazine before, so I want to spend some time explaining what we are about.

Firstly, all programs are donated by readers.

Next, we support the concept of learning through a User Group. Although we accept the entertainment value of the various computers we support, we do not see that our own primary function is to entertain.

Rather, we want to show people the potential of Tandy's computers, and show the great changes owning a Tandy computer can bring to people's lives.

User Groups are important because most owners will need one to one support sometime. User Groups do what even the most dedicated Tandy shop can't do - they give individual instruction for as long as is necessary.

In fact there are now over 110 User Groups in Australia, the character of which is considerably different to those which have been formed to "support" other computers.

For example, it is most unusual to find our support groups pirating software. I know it happens, but where such instances have been brought to the attention of the appropriate people, prosecutions have already taken place. But the grand majority of the groups just want to learn and to teach others.

As a Tandy computer user, it is most comforting to know that (usually local) help is always at the end of the telephone.

Our magazine is the medium through which the majority of interested Color Computer users find out about new products for their computer.

In Australia, there are a number of companies whose primary target is to support the Tandy range with companion products. Our recently published "Network Quarterly" (available gratis in Tandy shops only) describes a number of products from eight such suppliers.

We supply a number of items ourselves. In fact our founder Greg Wilson, was the original supplier of third party software in Australia.

However that side of things was too complex for a magazine to continue with, and Greg passed that part of the business to Software Spectrum, who continue to supply the majority of third party material.

None the less, there are a number of products which have sprung up around the magazine - our speciality is Australian software! Our products include:

* Australian Rainbow on Tape - the programs in each Australian Rainbow Magazine ready to run. $12.00 monthly.
* American Rainbow on Tape - the programs in the American Rainbow Magazine ready to run. $12.00 monthly.
* Australian CoCo Magazine - our companion magazine for new users of Colour Computers, MC-10 computers, Tandy 100's, 200's, 1000's and 2000's. $3.45 monthly or $31.00 annually.
* CoCoOz - The CoCo programs in each Australian CoCo Magazine ready to run. There is a minimum of eight programs on each tape $8.00 monthly or $75.00 annually.
* MiCoOz - The MC-10 programs in each Australian CoCo Magazine ready to run. $8.00 monthly or $75.00 annually.
* Books - for CoCo, Byte (easy) $5.95, Help (medium) $9.95 and Facts (machine language) $11.95. For MC-10, MiCo Help $9.95.
* The Best of CoCoOz #1 - Education programs.
  The Best of CoCoOz #2 - Part 1 (for 16K & 16 K ECB) & Part 11 (for 32K ECB) - Games programs.
  Prices - $10.00 / tape, $21.95 / disk.
* CoCoLink - the Tandy computer user's own Bulletin Board. Phone number is 075-51-0015, price $29.00 annually.
* Speech Pack Speller - two programs on tape to assist teachers with children who experience motivational problems with regard to spelling. The Tandy Speech / Sound Pack is required, price of the tape is $39.95.
* Othello - the board game, now on tape for $14.95.
* The CoCoConnection - connect your CoCo to the outside world! Just plug CoCoConnection to CoCo's ROM port and control robots, model trains, scientific and classroom experiments, burglar alarms, most automatic processes. Fully supported with solid documentation, further hardware (eg, A to D converters) and on-going articles in

Continued on P 66 ...

LETTERS

Hi Guys and Gals.
This is my first Note to Australian Rainbow. I thought this tip might be useful.
A POKE 359,255 disables keyboard output to the monitor. This can have several uses.

PASSWORD PROGRAM

```
10 PRINT"PLEASE ENTER PASSWORD"
20 POKE 359,255
30 INPUT PW$:IFPW$="WALDO"THEN50
40 POKE359,126:PRINT"ACCESS DENIED!!
": GOTO10
50 POKE359,126:PRINT"HELLO WALDO"
60 PRINT"ACCESS TO PROGRAM GRANTED"
```

The poke 359,126 returns screen output to normal.
NB If BREAK key is pressed during execution and location 359 contains value 255, a POKE 359,126 will be required to enable screen output.
R.WALDON
CAMP HILL QLD.

*

Dear Graham,
Once again, congratulations to all the gang for such a fine job on the mags. It is still a refreshing surprise to find such a caring group of people running what must be a demanding and, at times, frustrating business.
Unfortunately I find myself in the position where I must add my vote to those who do not like the direction that RAINBOW is heading. Each month there appears to be less and less that I can cope with even though I have been an avid fan of RAINBOW for several years now and try hard, a lot of the stuff is just too 'Hi Tech'. So please try to remember us slow learners. (Who don't like the predominance of games & etc. in Australian CoCo)
Enough Griping and on to the real reason for this letter.....
Like you, I am continually amazed at the seemingly never-ending applications and possibilities of 'Old Faithfull' CoCo. So here is one for the bright boys to think about.
Having had some contact with the deaf community and reading about CoCo Connection I got to wondering if it may be possible to set up CoCo and Connection with modem to monitor alarms, 'phone, door, baby's room & etc. etc. in a deaf persons home. (Might be a whole new area for Mr. 'T' to tap into).

J.DOHERTY
EMERALD VIC.

Dear Jim,
Aussie Rainbow still draws a majority of its material from the US. If we get more Australian input, we might be able to do something about your concerns, but I'm glad there is material to challenge you in Rainbow - you'd get bored pretty quickly if we just fed you stuff you already know!
You read our minds as far as the CoCoConnection is concerned. In fact our other main interest in producing it is to assist the disabled.
Graham.

*

Dear Graham,
Do you have any information on EARS that you can tell me?
What's the best book or books for a total beginner of Machine Language / Assembly Language?

J. FLETCHER
GRAFTON N.S.W.

Dear J,
Tandy have a book which should be a big help called "TRS 80 Color Computer Assembly Language Programming" by William Barden, Jr. CAT No 62-2077 ($9.95).
Expect some news on EARS soon. We were prepared to resell the product in Australia because no one else had the money and it sounded so good, but it now appears that someone will be distributing the program here in the near future.
Graham.

*

Dear Graham,
The system that I use is a TRS-80 Color Computer with 64K main bank ram, and 64K reserve page switched ram. The operating system is OS-9. I run two 5.25" 35 track standard Tandy (TEAC) disk drives, and two 150 track 8" Olivetti disk drives. (under OS-9 only) The printer is a DMP 200.
The modem is a simple 300 baud acoustic coupler (soon to be updated,)with the serial interface being home built (like most things around here) 300 baud RS232 card, at X'FF78.
I run a computer club with mostly Color Computer owners in the group, and one COMMODORE owner (also a CoCo owner).
The address for the club is :-
Peter May,
61 Nein Street,
Scarborough, 4020.
Phone (07) 203 6723 after 6PM.
The club is situated at the above address and meets the first and third Saturday of every month at 7.30pm
It would be appreciated if you could place this information in the magazine, for the use of all color computer owners. The meet contact is myself.

Peter MAY
SCARBOROUGH QLD.

Peter,
Your comments on the use of your 128K system would be appreciated, as is news of your club. We knew you were there we just didn't know where to point people!
Graham

*

Dear Graham,
If anyone reading your magazine uses the coco composing program or any other composer program that can be run without the editor, have they tried entering:

```
POKE 65497,0:EXEC:POKE 65496,0
```

This creates an interesting sound effect.

B. CHENOWETH
MONTMORENCY VIC.

To all at Australian Rainbow Magazine,
We would like to express our thanks and appreciation at receiving the Greg Wilson award to everyone who is involved with Australian Rainbow, and not least of all to the readers of the magazine. However we do believe that many other people in the CoCo Community deserve to have their excellent programming recognised and it is our sincere wish that they, in their turn, will be furure recipients of the prize.
We feel a great debt to Greg wilson who encouraged us in our early attempts at programming by submitting our material for publication. We feel quite sure that he would have been thrilled to see the way Graham and his colleagues have carried on the good work and made the magazine such a thriving success. The culmination of all their effort and dedication to the CoCo community has of course been CoCoConf and we gather that it succeeded splendidly. With luck and some planning, we hope to be there next year and we shall be delighted to meet the next person (or persons?) to be graced with the Greg Wilson award. For this year the 'plaque' takes pride of place among the ornaments on our sideboard and we do not cease to admire it (even if Tasmania has been left off the bottom!).
Thanking you all once again,
Bob, Daniel and Tino Delbourgo
Sandy Bay. Tas.

*

Dear Graham,
Greetings to all at Australian Rainbow and CoCo. Thank you for the pleasure they bring each month. I too discovered GREG and came to know his feelings and ideals as if he was in person. This is carried on by the gang who picked up the reins so ably.
Apart from the wealth of information and quality of programs (not to mention the professional presentation) you give us all a feeling of 'being there'. This is unique and worth building on.
Could I suggest a column such as 'CoCo BREAK' perhaps at the end of the Magazine, or even 'BEDTIME CoCo', where the month's trials and tribs / events / exasperations etc, would give a forum for humour, which we in the user family would look forward to (literally!) each issue.
I know costs dictate trimming the fat, but ... the contents are there already in amoungst the other 'lines', yes we look between the lines too!!
Well E.G.B.D.F. girls too (must REM Martha (who's great)) that's a nested loop isn't it?
I wish you all the best,
Clive Winsall,
Aspendale. Vic.

Clive,
I've left off the last part of your letter because of space, but thanks for your kind words.
There are frustrations in any enterprise, this one as much or more than many, because it has always been undercapitalised. I'm afraid we really didn't know what we were taking on.
But we have been fortunate, because when things have looked grim, each time our readers have rallied with help and support. Right now, many of the things we have planned since the start are just starting to happen, so we are enjoying a very satisfying period.

Continued on P 66 ...

OodlesOf Games For 4K

A Mind Boggler

See if you can solve this Picstick puzzle. Play against the computer, alternating turns until all 15 sticks are taken. The object is to force the computer to take the last stick. Each turn (your's and the computer's) is limited to one, two or three sticks. Of course, when you figure out the correct pattern you will always win. But, how long will that take?

The listing: PICSTICK

```
 1 CLS:PRINT"PICK UP STICKS":PRIN
 T
 2 PRINT" YOU MAY PICK UP 1,2, OR
  3"
 3 PRINT" STICKS EACH TIME. THE O
 BJECT"
 4 PRINT" IS TO MAKE THE COMPUTER
  PICK"
 5 PRINT" THE LAST ONE. PRESS ENT
 ER"
 6 PRINT" WHEN YOU ARE READY."
 7 INPUT A$
 8 CLS(Ø)
 1Ø FOR I=1 TO 15
 2Ø FOR J=16 TO 31
 3Ø SET(I*4,J,4)
 4Ø NEXT J
 5Ø NEXT I
 6Ø SW=1 :T=15
 1ØØ PRINT@Ø,"ENTER NO. (1,2, OR
 3)"
 11Ø INPUT A$
 12Ø A=VAL(A$)
 13Ø IF A<1 OR A>3 THEN 1ØØ
 14Ø T=T-A
 145 PRINT" STICKS CHOSEN="A" REM
 AIN="T
 15Ø IF T>Ø THEN 18Ø
 16Ø IF SW=1 THEN PRINT@Ø,"YOU LO
 SE" ELSE PRINT@Ø,"YOU WIN"
 17Ø GO TO 1ØØØ
 18Ø SW = -SW
 19Ø B=(15-(T+A))+1:B1=(B+A)-1
 2ØØ FOR I=B TO B1
 21Ø FOR J=16 TO 31
 22Ø RESET(I*4,J)
 23Ø NEXT J
 24Ø NEXT I
 25Ø IF SW=1 THEN 1ØØ
 26Ø ST=T-1
 27Ø MC=Ø
 28Ø IF MC=3 THEN 36Ø
 29Ø MC =MC+1
 3ØØ A=(ST-MC)/4
 31Ø A=ABS(A)
 32Ø A=A-INT(A)
 33Ø IF A<>Ø THEN 28Ø
 34Ø A=MC
 35Ø GOTO 14Ø
 36Ø A=1
 37Ø GOTO 14Ø
 1ØØØ PRINT@32,"END GAME"
 1Ø1Ø PRINT"RESTART = ENTER"
 1Ø2Ø INPUT A$:GOTO 8
```

Brian Baxter
Arlington, TX

Where Are the Walls?

The Maze Game begins with a man (in the upper left corner of a blank screen) and a blue dot (in the lower right corner of the screen) indicating an exit. Your goal is to find your way out of the dark maze, however, the walls are invisible. They only appear as you run into them.

The listing: MAZEGAME

```
 1Ø REM MAZE
 2Ø DIM A$(16)
 3Ø MV=Ø:F=3
 4Ø GOSUB 7ØØ
 6Ø FOR I=1 TO 16
 7Ø READ A$(I)
 8Ø NEXT I
 9Ø READ L,E:X=INT(L/1ØØ):Y=L-(X*
 1ØØ)
 95 EX=INT(E/1ØØ):EY=E-(EX*1ØØ)
 1ØØ CLS(Ø):MX=X:MY=Y
 1Ø1 SET(EY*2,(EX-1)*2,3)
 1Ø5 GOSUB 4ØØ
 12Ø B$=INKEY$
 123 IF EX=X AND EY=Y THEN 8ØØ
 125 IF B$="" THEN 12Ø
 13Ø MX=X:MY=Y
 14Ø IF B$="U" AND X>1 THEN MX=X-
 1
 15Ø IF B$="D" AND X<16 THEN MX=X
 +1
 16Ø IF B$="L" AND Y>1 THEN MY=Y-
 1
 17Ø IF B$="R" AND Y<32 THEN MY=Y
 +1
 18Ø IF MID$(A$(MX),MY,1)="W"THEN
  GOSUB2ØØ ELSE GOSUB3ØØ
 19Ø GOTO 12Ø
 2ØØ REM WALL
 21Ø II=(MX-1)*2:IJ=(MY-1)*2
 22Ø FOR I=II TO II+1
 23Ø FOR J=IJ TO IJ+1
 24Ø SET(J,I,8):NEXT J
 25Ø NEXT I:RETURN
 3ØØ REM RESET/SET POSITION
 31Ø II=(X-1)*2:IJ=(Y-1)*2
 32Ø FOR I=II TO II+1
 33Ø FOR J=IJ TO IJ+1
 34Ø RESET(J,I)
 35Ø NEXT J:NEXT I
 4ØØ II=(MX-1)*2:IJ=(MY-1)*2
 41Ø FOR I=II TO II+1
 42Ø FOR J=IJ TO IJ+1
 43Ø SET(J,I,5)
 44Ø NEXT J:NEXT I
 445 X=MX:Y=MY:MV=MV+1
 45Ø RETURN
 5ØØ DATA WWWWWWWWWWWWWWWWWWWWWW
 WWWWWWWWW
 51Ø DATA WSWSSSWSSWSSSWSSSWWFSWS
 WSSSWSSSW
 52Ø DATA WSWSWSWSWWSWSWSWSSSWSWS
 WSWSSSWSW
 53Ø DATA WSSSWSWSSSSWSWSWWWSWSWS
 WSSWWWWSW
 54Ø DATA WWWWWSWSWSWSSWSSSWSWSWS
 WWSSSSWSW
 55Ø DATA WSSSSSSSWSWSSSWWSWSWSSS
 WSWWSSWSW
 56Ø DATA WSWSWWWWWSWSWSSSSSSSSWS
 SSSSSWSSW
 57Ø DATA WSWSWSSSWSSSSWWWWSSWSSW
 WWWWWWSWW
 58Ø DATA WSWSSSWSWWWWSSSSSWSWSWS
 SSSSSWSSW
 59Ø DATA WSSWWWWSWSSSWFWWWWWWSWS
 WSWWSSWSW
 6ØØ DATA WWFWSSSSWSWSSWSSSSSWSWS
 WSSSWSSSW
 61Ø DATA WWWWSWWWWSSWSWSWWWSWSWW
 WWWSWWWSW
 62Ø DATA WSSSSWSSSWSWSWSSSWSWSWS
 SSSSWSSSW
 63Ø DATA WSWWWSSWSWSWSSSWSWSWSWW
 SWWWWWWWW
 64Ø DATA WFSSSSWWSSSWSWWWSWSWSFW
 SSSSSSSSW
 65Ø DATA WWWWWWWWWWWWWWWWWWWWWW
 WWWWWWWSW
 66Ø DATA Ø2Ø2,1631
 7ØØ CLS:PRINT"MAZE (VERISION 1)"
 71Ø PRINT:PRINT"FIND YOUR WAY OU
 T"
 72Ø PRINT"U=UP D=DOWN L=LEFT R=R
 IGHT"
 725 PRINT"EXIT IS BLUE SQUARE"
 73Ø PRINT"PRESS 'ENTER' TO START
 "
 74Ø INPUT B$
 75Ø RETURN
 8ØØ CLS:PRINT"YOU MADE IT OUT IN
 "MV"MOVES."
```

Brian Baxter
Arlington, TX

Watch Out, Danny Sullivan!

Indy4K is a road race in which you control a super fast car with your right joystick. Stay on the road for as many miles as you can.

If your CoCo can not take the "Vitamin E" POKE, delete it from Line 0.

The listing: INDY4K

```
 Ø POKE65495,Ø:CLS:PRINT@1Ø6,"KAM
 AKAZIE KAR":PRINT@175,"BY":PRINT
 @238,"PAUL":PRINT@269,"WAGORN":P
 RINT@334,"FOR":PRINT@397,"APOLLO
 ":PRINT@428,"SOFTWARE"
 1 SOUND 1ØØ,4:SOUND14Ø,5:SOUND1Ø
 Ø,1:SOUND14Ø,1:SOUND1ØØ,1:SOUND1
 1Ø,1:SOUND12Ø,1:SOUND13Ø,1:SOUND
 14Ø,1:SOUND13Ø,2:SOUND12Ø,4:SOUN
 D11Ø,8:SOUND1ØØ,1Ø
 2 FORPO=5Ø3 TO 23 STEP -64:PRINT
 @PO,"!";
 3 POKE65315,63:FORI=1TO255STEP9:
 POKE65312,255:POKE65312,I:NEXT
 4 PRINT@PO-31,"!";:POKE65315,63:
 FORI=1TO255STEP9:POKE65312,255:P
 OKE65312,I:NEXT:NEXTPO
 5 PRINT@49Ø,"PRESS A KEY";:EXEC4
 1329
 6 C1$=CHR$(132+96)+CHR$(143+48)+
 CHR$(136+96):C2$=CHR$(132+96)+CH
 R$(14Ø+48)+CHR$(136+96):WL$=CHR$
 (194):CLS:P=15:S$=CHR$(255):M$=C
 HR$(128):R$=CHR$(131+32)+S$+M$+M
 $+M$+M$+WL$+M$+M$+M$+S$+CHR$(131
 +32):RP=12
 7 FORI=1 TO2Ø:PRINTTAB(RP);R$:NE
 XT:PRINT@P,C1$;:PRINT@P+32,C2$;:
 PRINT@14+98,"start";:SOUND1ØØ,1:
```

```
SOUND2ØØ,3:SOUND5Ø,3:SOUND1ØØ,1:
SOUND14Ø,2:SOUND13Ø,3:SOUND11Ø,4
:SOUND1ØØ,5:SOUND9Ø,6:SOUND8Ø,7:
PRINT@Ø,"";
8 CD=RND(3)-2:FORL=1TO 4
9 D=D+1
1Ø IF JOYSTK(Ø)=<3 AND P>=1 THEN
 P=P-1:ELSEIFJOYSTK(Ø)>=6Ø ANDP<
=28 THENP=P+1
11 KI=P+64+1Ø24:IFPEEK(KI)=255OR
PEEK(KI+1)=255ORPEEK(KI+2)=255TH
EN 16
12 PRINTTAB(RP);R$
13 GOSUB15:IFRP+CD>ØANDRP+CD<18T
HENRP=RP+CD:NEXTL:ELSEGOTO8
14 GOTO 8
15 PRINT@P,C1$;:PRINT@P+32,C2$;:
PRINT@479,"":RETURN
16 E1$=CHR$(139+48)+CHR$(14Ø+16)
+CHR$(141+48):E2$=E1$:E3$=CHR$(1
41+16)+CHR$(131+96)+CHR$(139+16)
:E4$=E3$:FORI=1TO5:PRINT@P,E1$;:
PRINT@P+32,E2$;:POKE65315,63:FOR
R=1TO4:POKE65312,255:POKE65312,R
ND(255):NEXT:PRINT@P,E3$;:PRINT@
P+32,E4$;
17 SOUND2ØØ,1:NEXTI
18 SOUND1Ø,4:SOUND4Ø,2:SOUND1,1:
SOUND1Ø,1:SOUND4Ø,4:SOUND2Ø,1:PR
INT@23Ø,"MILELAGE:"D/2Ø"MILES."
19 PRINT@48Ø+RP+1,"PRESS A KEY";
:POKE65314,14:EXEC41329:CLEAR:GO
TO6
2Ø FORI=1TO 4:MOTORON:FORK=1TO5Ø
Ø:NEXT:SOUND1,4:FORK=1TO5ØØ:NEXT
:CSAVE"KAM KAR":NEXT
```

Paul Wagorn
Carp, Ontario

Screen Cleaning

Use the arrow keys in Cleanup to travel the screen and clean up all of the characters. But, don't touch the outer walls or any of the graphics.

The listing: CLEANUP

```
 5 CLEAR 5ØØ:CLS
1Ø PRINT:PRINT"(PRESS ANY KEY TO
 START)"
15 A$=INKEY$:IF A$=""THEN 15
2Ø CLS:BL=Ø:SC=Ø:INPUT"STARTING
LEVEL(1-5Ø)";S:IF S<1 OR S>5Ø TH
EN 2Ø
25 CLS(Ø):PRINT"SCORE=";SC
3Ø FOR X=Ø TO S*3-1
35 P=RND(375)+95:IF PEEK(P+1536Ø
)<33 THEN PRINT@P,CHR$(RND(63)+1
28);ELSE35
4Ø NEXT X
45 FOR X=Ø TO S*3-1
5Ø P=RND(375)+97:IF X/3=INT(X/3)
 THEN IF PEEK(P+1536Ø)<33 THEN P
RINT@P,CHR$(RND(94)+32);ELSE 5Ø
55 NEXT X
6Ø FOR X=Ø TO 63:SET(X,3,5):SET(
X,31,5):NEXT X:FOR X=4 TO 31:SET
(Ø,X,5):SET(63,X,5):NEXT X
65 X=5:Y=5:U=Ø:H=Ø:A$=INKEY$:FOR
 D=1 TO 2 STEP Ø:IF INKEY$<>""TH
EN D=2:NEXT D ELSE NEXT D
7Ø K$=INKEY$
75 IF K$=CHR$(94) THEN U=-1:H=Ø
ELSE IF K$=CHR$(1Ø) THEN U=1:H=Ø
 ELSE IF K$=CHR$(8) THEN U=Ø:H=-
1 ELSE IF K$=CHR$(9) THEN U=Ø:H=
1
8Ø X=X+H:Y=Y+U:IF POINT(X,Y)>1 T
HEN 1ØØ
85 IF POINT(X,Y)=-1 THEN SC=SC+1
:BL=BL+1:PRINT@7,SC;
9Ø SET(X,Y,1):IF BL<5 THEN 7Ø
95 CLS:PRINT"LEVEL";S;"COMPLETED
...BONUS:";S*5:SC=SC+S*5:BL=Ø:S=
S+1:FOR D=1 TO 1ØØØ:NEXT D:GOTO
25
1ØØ SOUND 45,1:CLS:PRINT"YOU MAD
E IT TO LEVEL "S;"WITH A SCORE O
F "SC
```

Stephan J. Elms
Fort Ann, NY

Hunt and Peck

Typanic will test your knowledge of the CoCo's keyboard, as well as your reflexes. Shoot down falling characters before they hit the ground by typing the appropriate key on the keyboard. If you miss five times, the game is over and the CoCo displays your score.

The listing: TYPANIC

```
1Ø CLS:PRINT@43,"TYPANIC":PRINT:
PRINTTAB(7);"LEVEL:":PRINTTAB(7)
;"(1) BEGINNER":PRINTTAB(7);"(2)
 INTERMEDIATE":PRINTTAB(7);"(3)
EXPERT":PRINTTAB(7);"(4) PRETTY
DARN HARD"
2Ø A$=INKEY$:IF A$="" THEN 2Ø
3Ø A=VAL(A$):IF A>4 OR A<1 THEN
2Ø
4Ø POKE282,1:CLS:T=3ØØ-5Ø*VAL(A$
)
5Ø PO=2+RND(28):C=32+RND(58)
6Ø C$=CHR$(C):Y=PO
7Ø A$=INKEY$
8Ø PRINT@Y,C$;:Y=Y+32
9Ø IF Y>448 THEN 13Ø
1ØØ IF A$=C$ THEN 15Ø
11Ø FOR DL=1 TO T:NEXTDL:PRINT@Y
-32," ";
12Ø IF A$="" THEN 7Ø ELSE 7Ø
13Ø SOUND 1,1:W=W+1:IF W=5THEN17
Ø
14Ø GOTO 5Ø
15Ø T=T-A:SOUND 2ØØ,1:PRINT@Y-32
," ";:SC=SC+1:PRINT@Ø,SC:IFT<4 T
HEN T=5
16Ø GOTO5Ø
17Ø CLSØ:FOR S=2ØØ TO 1 STEP -6
18Ø SOUND S,1:NEXTS
19Ø PRINT@16Ø,"YOUR SCORE IS "SC
;
2ØØ PRINT@224,"PLAY AGAIN (Y/N)?
";:A$=INKEY$:IFA$=""THEN2ØØELSEI
FA$="Y"THENRUN ELSEEND
```

Michael Sims
Nanuet, NY

Memory Jogger

Letter Memory is a simple but challenging game that test your memory. The object of the game is to remember a letter string that keeps growing larger. When you run the program it will briefly display a letter which you are to remember. You must type the letter and the same letter will appear with another letter added. The computer will keep adding more and more to the string, one letter at a time, until you forget part of the string (or, if you type it incorrectly).

The listing: LETRMMRY

```
1Ø '
2Ø ' LETTER MEMORY
3Ø ' BY BEN JOHNSON
4Ø '
5Ø SC=Ø
6Ø A$="ABCDEFGHIJKLMNOPQRSTUVWXY
Z"
7Ø CLS
8Ø FOR T=1 TO 255
9Ø GOSUB 14Ø
1ØØ INPUT F$
11Ø IF F$<>S$ THEN 22Ø
12Ø SC=SC+T:IF T/5=INT(T/5) THEN
 SC=SC+5
13Ø NEXT T
14Ø R=RND(26)
15Ø D$=MID$(A$,R,1)
16Ø S$=S$+D$
17Ø D$=""
18Ø PRINTS$
19Ø FOR H=1 TO 5ØØ:NEXT H
2ØØ CLS
21Ø RETURN
22Ø CLS3:PRINT"? ";F$:PRINT"SORR
Y, YOU LOST TRACK."
23Ø PRINT"SCORE: "SC
24Ø PRINT"CURRENT STRING:"
25Ø PRINTS$
26Ø GOSUB31Ø
27Ø FOR L=1 TO E:PRINT@159+NM(L)
,"-":NEXTL
28Ø PRINT:PRINT"TAP <ENTER> TO P
LAY AGAIN..."
29Ø A$=INKEY$:IF A$<>CHR$(13) TH
EN 29Ø
3ØØ RUN
31Ø FOR J=1 TO LEN(S$)
32Ø IF MID$(S$,J,1)<>MID$(F$,J,1
) THEN E=E+1:NM(E)=J:NEXT ELSE N
EXT
33Ø FOR J=1 TO LEN(F$)
34Ø IF MID$(F$,J,1)<>MID$(S$,J,1
) THEN E=E+1:NM(E)=J:NEXT ELSE N
EXT
35Ø RETURN
```

Ben Johnson
Charlestown, Wv

EDUCATION PAGE

Most things we attempt in the classroom we understand. We know within limits what will happen when we have a writing lesson, or when we decide to teach fractions.

We are still however, finding a place for the computer in the classroom.

Over the past few years, computers have been used in the school for a number of tasks, and with varying equipment.

People have used them to teach BASIC, LOGO, rote Maths, rote Spelling and sometimes, Word Processing. There are occasionally other uses which are applied to a school's computer, usually by an exceptional teacher with personal skills in computing.

The difficulty is still that there is a definite lack of vision in the Curriculum, of the computer and its place in society - let alone its place in the school.

It is time we addressed ourselves therefore to a co-ordinated approach, one might even say, - for better or worse, because at least then we'd have a starting point!

The realities of today's schools seem to be:

1. There are not enough computers;
2. Therefore computers tend to be grouped into an area known as "THE computer room";
3. There are not enough teachers either prepared to learn, or who already possess the skills to "do the job properly";
4. We have a vague impression there may be some value in ensuring that the kids are familiar with the computer.

I'd like you to take your mind's crayon and cross all thoughts along these lines out of your head. We need to find a fresh approach. Hitherto, what has been done has been experimental - let's get on with it!

Tenet 1. The computer is a tool.

Tenet 2. Because of Tenet 1, it follows in education, that we should be able to achieve some educational aim when we utilise a computer.

I suggest that computer classrooms be broken up and each class in the school be given a computer. This wont please network salesmen, but it achieves three things:

1. It reduces the aura surrounding the computer room, and the associated feeling kids get that the computer is something very special - this is bull! The computer is just another tool some folk use.
2. It gets the computer into the classroom where it can be utilised to illustrate points which are being made across subject areas.
3. Teachers and kids get to be familiar with the computer more quickly, and the reliance on just one or two staff members for assistance with the computer starts to diminish.

I feel a bit stupid saying the following, but I am assured it is necessary to state the obvious.

Ask yourself what the purpose is behind what you intend to do in the classroom. Then ask if it is really appropriate to use the computer to assist in the task you have in mind. Then plan the method you will use to impliment your aims, and after you have run the lesson - evaluate what you have done and if you can't identify any benefit, or if you perceive that you could achieve the same or a better effect without the computer, then DON'T use it.

One of the real abilities of the computer is it's ability to assist in teaching the skill of reason.

Skills which the computer can assist with might include the following:

Grade 1:
Sorting according to prespecified attributes;
Interpreting visual information;
Identifying and using the required option; &
Guessing and checking answers.

Grade 2:
Searching for relevant information;
Identifying & completing patterns;
Determining reasonableness of results; &
Acting out situations.

Grade 3:
Reading tables, graphs and maps; &
Drawing sketches & diagrams.

Further suggestions for years 4 through 10 be found in the book "Problem Solving", (P 21) by Barry Salmon and Neville Grace and available from the Curriculum Services Branch of the Queensland Education Department.

Another book to read is the "Handbook of Primary Education and Computing" by D.W.W. Ellingham; Castle House Publications. This book is based on English practice and has many good ideas.

There are programs for the Colour Computer which assist in achieving these aims, and more are on the way thanks to the Australian Software Development Council, which has been specifically set up to develop software to meet the educational and Curriculum objectives of Australian schools.

*

Tandy has finally begun the 900 Series of software.

This system allows you to shop for many of those excellent programs you see advertised in this magazine, even though they come from other suppliers - so even if you live in Armidale, (which really is a nice place - despite what they say about it!), you now have direct access to programs like Telewriter 64 and the VIP series

through your Tandy shop.

I understand that Software Spectrum alone has upwards of 40 items available in this way. Great going Tandy - this can only ensure the continuing success of CoCo!

*

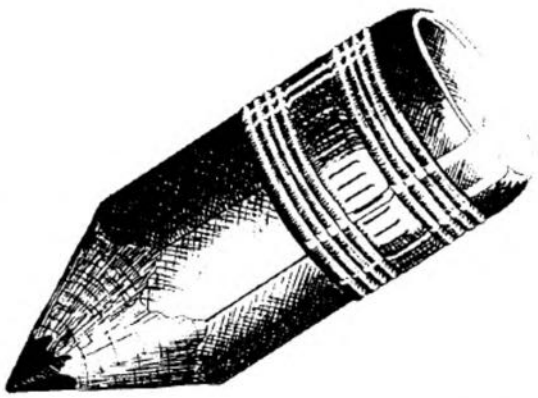
The latest Education Communique from the flying pens of Karel Davey and Leo Wilson (in that order) has just been released.

Detailed is news of a most interesting Computer Course at Cloyton High School, where the local Tandy store works with the school to assist students to gain first hand practical and real experience of the use of the computer in the outside world.

A partial list of the 900 Series software available from Tandy is also included, and there is news of the Tandy School Security System.

ORGANIZATION | **16K** | the RAINBOW

CoCo Testmaker



By Lynn C. Sherman and Walter Baldassaro

All students have one thing in common: They hate to take tests. Teachers likewise have one thing in common: They hate writing and correcting those test papers. The CoCo may not be ready to help in the laborious task of grading tests, but it is most helpful in writing test papers. CoCo Testmaker is not limited to use by teachers. Concerned parents will find it a perfect way to help their children study, check what they have learned and, in turn, learn how to take written tests. When it comes to formatting the test, typing the lines for essay-type answers and shuffling the choices in matching lists, CoCo Testmaker gets an A+!

When using this program the teacher has a six-entry menu from which to work. This allows for greater flexibility, and makes it possible to make two test using the same questions but in a different section order. This helps cut down on the temptation of wandering student eyes. Each group of questions is automatically numbered. The test paper heading is placed on the top right-hand corner of the paper to aid in sorting through stacks of test papers.

Prior to typing the questions in, each section of the program prints out a line of instructions for the student. True and false questions ask for one of these words to be circled, as does the correct choice in the multiple-chioce questions. Fill in the blank entries are self-explanatory. Number four on the menu, "Answer Questions", allows for both objective and subjective testing. Up to four lines can be automatically printed under the question, thereby eliminating the time-consuming typing task normally required of teachers compiling tests.

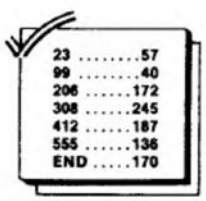
Matching tests have always been popular with teachers and students. By matching statements in one column with information in another column, the students can use comparison of choices and the process of elimination. For the teacher it is a quick grading type of test, which is important when there are up to 200 papers to be corrected for the next day. *CoCo Testmaker* permits the teacher to enter, for example, states in one column and capitals in the other. Shuffling the information is done by lines 600-695. Different patterns of jumbling this information can be accomplished by editing this section of the program.

CoCo Testmaker is designed to be very "teacher friendly." The goal was to cut down on the time it took to write and type up test papers. With the advent of plain paper copy machines as the way tests are mass printed, it was only natural that the CoCo would replace the typewriter/mimeograph machine connection. Automatic numbering, columning, shuffling and the printing of lines for students to write on have cut the time of this task in half.

As mentioned earlier, *CoCo Testmaker* can be used as a home study aid as well. Teaching the youngest students how to take these types of written tests will give them a definite advantage throughout their entire school years. Reviewing studied homework via a homemade test should help any student, no matter what the grade level. Taking written tests is just like everything else in life — practice makes perfect!

This program was written for 16K, but can be expanded in several areas if the user has a larger memory machine. By altering lines 420-425, additional lines can be printed to allow for longer essay answers. As indicated in Line 546, only 10 matching entries can be made. If more are desired, additions would have to be made to the 600-622 series, and to the 650-695 series. It should be noted that due to the placement of the columns on the page, there are limits to the number of characters in each column. There can be a maximum of 27 characters in Column A and 45 in Column B.

On a non-programming topic, the authors offer a test-scoring technique suggestion. Take an extra copy of the test to be graded and cut out the incorrect answers on the objective questions. By laying this cut-out test sheet on the test to be graded, only the wrong answers have to be marked for scoring. This quick method of scoring has a number of modifications available, and cuts down the time it takes the teacher to read the subjective answers and total up the final grade.



The listing: TESTMAKR

```
1 REM ----------------------COCO
TESTER -------------------------
-----------
2 REM ///     A TEST MAKING PROGR
AM BY W. BALDASSARO & L.C. SHERM
AN       ///
3 REM ///                        C
OPYRIGHT 1985
       ///
4 REM FD-DD+C-CC+B-BB+A-AA+ / FD
-DD+C-CC+B-BB+A-AA+ / FD-DD+C-CC
+B-BB+A-AA+
5 CLS
6 CLEAR 2ØØØ
8 GOTO25
9 CLS : INPUT "DATE OF TEST";O$
1Ø PRINT#-2,"
NAME-----------------------":PR
INT#-2,""
2Ø PRINT#-2,"
HOMEROOM-------------------":PR
INT#-2,""
22 PRINT#-2,"
DATE   "+O$:PRINT#-2,""
23 CLS : GOTO 4Ø
25 PRINT "  YOUR COMPUTER IS NOW
 YOUR     TEACHER'S AID AND WILL
 ASSIST   YOU IN MAKING UP A TES
T.  MAKE  SURE THE PAPER IN YOUR
 PRINTER  IS SET TO THE TOP OF T
HE PAGE.                  "
26 PRINT "  PRESS ANY KEY TO BRI
NG UP THE MENU OF TYPES OF QUEST
IONS      OFFERED."ZZZ$
27 INPUT ZZ$
32 CLS
4Ø PRINT "WHICH WOULD YOU PREFER
"
5Ø INPUT"  1 PRINT HEADING ON
             PAPER
           2 MULTIPLE CHOICE
             3 FILL IN THE BLANK
             4 ANSWER QUESTIONS
             5 TRUE FALSE TEST
             6 MATCHING TEST ";N
7Ø IF N <1 OR N >6 THEN5Ø
8Ø ON N GOSUB 9, 2ØØ, 3ØØ, 4ØØ,
  9Ø, 53Ø
9Ø PRINT#-2,"CIRCLE TRUE IF THE
STATEMENT IS TRUE--CIRCLE FALSE
IF THE STATEMENT IS FALSE":PRINT
#-2,""
95 CLS
96 J=Ø
98 PRINT "TRUE OR FALSE QUESTION
S"
99 PRINT"AFTER TWO LINES RETURN
TO ARROW (^)"
1ØØ PRINT"                    ^"
11Ø.INPUT"";A$
12Ø B$="TRUE OR FALSE "
151 Q$="  "
156 J1=1
157 J=J1+J
16Ø PRINT#-2,J;B$;Q$;A$:PRINT#-2
,""
166 GOTO 1ØØ
18Ø END
2ØØ PRINT#-2,"CIRCLE THE BEST AN
D MOST CORRECT ANSWER FOR EACH S
TATEMENT "
2Ø1 J=Ø
2Ø2 CLS
2Ø4 PRINT "TYPE STATEMENT OR QUE
STION      WHEN FINISHED SELECTI
ONS WILL   APPEAR A-B-C-D
           THEN TYPE IN YOUR SEL
ECTIONS.   WHEN FINISHED ALL STA
TEMENTS OR QUESTIONS IN A SECTIO
N PRESS    <BREAK-RUN> FOR MENU
AGAIN""
2Ø5 PRINT"THIS SECTION WILL ACCE
PT TWO    LINES  DO NOT RETURN P
AST  FIRST ARROW(^) THEN  SPACE
TO NEXT   ARROW (^)":PRINT"

     ^      ^"
2Ø6 INPUT"";C$
2Ø7 CLS
2Ø8 J1=1:J=J1+J
21Ø PRINT#-2,"":PRINT#-2,J;TAB(5
)C$:PRINT#-2,""
215 C2$="A.":C3$="B.":C4$="C.":C
5$="D.":C6$="E."
22Ø INPUT "A";A$:INPUT "B";B$:IN
PUT "C";C$:INPUT "D";D$
225 INPUT "DO YOU NEED E, Y/N ";
EE$
23Ø IF EE$="Y" THEN 25Ø ELSE 24Ø
24Ø PRINT#-2,TAB(1Ø)C2$+A$:PRINT
#-2,TAB(1Ø)C3$+B$:PRINT#-2,TAB(1
Ø)C4$+C$:PRINT#-2,TAB(1Ø)C5$+D$
245 GOTO 2Ø5
25Ø INPUT "E";E$
255 PRINT#-2,TAB(1Ø)C2$+A$:PRINT
#-2,TAB(1Ø)C3$+B$:PRINT#-2,TAB(1
Ø)C4$+C$:PRINT#-2,TAB(1Ø)C5$+D$:
PRINT#-2,TAB(1Ø)C6$+E$
26Ø GOTO 2Ø5
3ØØ CLS
3Ø5 PRINT#-2,"FILL IN THE BLANKS
 OF EACH STATEMENTS.":PRINT#-2,"
"
3Ø7 PRINT"THIS SECTION WILL ACCE
PT TWO    LINES  DO NOT RETURN P
AST  ARROW ^ THEN PRESS ENTER TO
 CONTINUE   LINE"
3Ø8 PRINT"STATEMENT OR QUESTION

                    ^      "

31Ø INPUT"";F$
312 J1=1
314 J=J1+J
315 PRINT#-2,J;F$: PRINT#-2,""
316 INPUT "PRESS 1 FOR SPACE IF
NEEDED";A : ON A GOSUB 34Ø
33Ø GOTO 3Ø8
34Ø INPUT" ";AA$
35Ø PRINT#-2,TAB(3)AA$:PRINT#-2,
""
355 GOTO 3Ø8
4ØØ CLS
4Ø2 PRINT#-2,"ANSWER QUESTIONS O
N LINES PROVIDED":PRINT#-2,""
4Ø4 PRINT"THIS SECTION WILL ACCE
PT TWO    LINES  DO NOT RETURN P
AST FIRST ^  SPACE TO NEXT ARROW
 ^"
4Ø5 PRINT"

                    ^      ^";I
NPUT"";Y$
41Ø YY$="----------------------
------------------------------
------------------"
412 J1=1
414 J=J1+J
415 PRINT#-2,J;Y$:PRINT#-2,""
42Ø INPUT "HOW MANY LINES 1-2-3-
4 ";H
425 ON H GOSUB 43Ø, 44Ø, 45Ø,46Ø
43Ø PRINT#-2,YY$:PRINT#-2,""
435 GOTO 4Ø5
44Ø PRINT#-2,YY$:PRINT#-2,"":PRI
NT#-2,YY$:PRINT#-2,""
445 GOTO 4Ø5
45Ø PRINT#-2,YY$:PRINT#-2,"":PRI
NT#-2,YY$:PRINT#-2,"":PRINT#-2,Y
Y$:PRINT#-2,""
455 GOTO 4Ø5
46Ø PRINT#-2,YY$:PRINT#-2,"":PRI
NT#-2,YY$:PRINT#-2,"":PRINT#-2,Y
Y$:PRINT#-2,"":PRINT#-2,YY$:PRIN
T#-2,""
465 GOTO4Ø5
466 END
53Ø PRINT#-2,"MATCH COLUMN A WIT
H COLUMN B.   PLACE NUMBER OF CO
LUMN B IN SPACE PROVIDED IN
                                CO
LUMN A":PRINT#-2,""
545 CLS
546 PRINT"THIS SECTION WILL RAND
OM 1Ø      ITEMS IN COLUMN A WIT
H 1Ø ITEMS IN COLUMN B.":PRINT,"
":PRINT,""
555 PRINT"INPUT COLUMN A AND COL
UMN B     RANDOMING IS AUTOMATIC
.":PRINT,""
558 PRINT"COLUMN A. CAN BE UP TO
                    27 CHARACTER
S LONG.   COLUMN B. CAN BE UP TO
                    45 CHARACTER
S LONG."
56Ø L2$="---"
565 PRINT#-2,"COLUMN A.
             COLUMN B.":PRINT#-2
,""
6ØØ INPUT"A.";Z6$:INPUT"B.";Q$
6Ø4 INPUT "A.";Z5$:INPUT"B.";Q3$
6Ø8 INPUT "A.";Z4$:INPUT"B.";Q2$
61Ø INPUT "A.";Z3$:INPUT"B.";Q4$
612 INPUT "A.";Z2$:INPUT"B.";Q5$
614 INPUT "A.";T2$:INPUT"B.";P2$
616 INPUT "A.";T3$:INPUT"B.";P3$
618 INPUT "A.";T4$:INPUT"B.";P4$
62Ø INPUT "A.";T5$:INPUT"B.";P5$
622 INPUT "A.";T6$:INPUT"B.";P6$
63Ø J=Ø :J1=1 :J=J1+J
65Ø PRINT#-2,L2$+Z5$;TAB(3Ø)J;Q2
$:PRINT#-2,"":J=J1+J
655 PRINT#-2,L2$+Z6$;TAB(3Ø)J;Q3
$:PRINT#-2,"":J=J1+J
66Ø PRINT#-2,L2$+Z3$;TAB(3Ø)J;P4
$:PRINT#-2,"":J=J1+J
665 PRINT#-2,L2$+Z2$;TAB(3Ø)J;Q$
:PRINT#-2,"":J=J1+J
67Ø PRINT#-2,L2$+Z4$;TAB(3Ø)J;Q5
$:PRINT#-2,"":J=J1+J
675 PRINT#-2,L2$+T2$;TAB(3Ø)J;P5
$:PRINT#-2,"":J=J1+J
68Ø PRINT#-2,L2$+T5$;TAB(3Ø)J;P3
$:PRINT#-2,"":J=J1+J
685 PRINT#-2,L2$+T4$;TAB(3Ø)J;P6
$:PRINT#-2,"":J=J1+J
69Ø PRINT#-2,L2$+T6$;TAB(3Ø)J;Q4
$:PRINT#-2,"":J=J1+J
695 PRINT#-2,L2$+T3$;TAB(3Ø)J;P2
$:PRINT#-2,"":J=J1+J
7ØØ END
```

A Caterpillar's Alphabet

By Mike Knolhoff

As a teacher I am able to see and evaluate educational software quite frequently. This gives me the opportunity to get ideas for programs I can write for my own children or for my classes.

I recently saw an interesting program that was written for a more expensive computer. The program was designed to help preschoolers or primary school children learn the alphabet. A sequence of five letters was displayed in order on the screen with the middle letter left blank. The student was to figure out which letter went in the blank and type that letter. The letter typed would then fill in the blank, and a right or wrong cue would be given.

Each time the student typed the correct letter, a caterpillar segment would appear on the screen until the caterpillar was complete. At that time the caterpillar would walk around and off the screen as a visual reward for a job well done. Tapping any key would then restart the program.

The program required no reading skills on the part of the student, and all keyboard input from the student consisted of single keystrokes. The program was written in machine language and sold for about $30.

The listing that follows is my version of this program. Thanks to the powerful graphics commands Tandy gave the CoCo, it is written entirely in BASIC. Admittedly, the graphics are not as refined and the animation not as smooth as its inspiration program, but it does function in much the same manner (and is a great deal more economical). I believe parents will find it very beneficial for their 4-6 year old children.

You need 16K Extended BASIC for the program. It uses the speed-up POKE in Line 690, so if your computer cannot handle this POKE you will have to edit that line to remove it. It will run on disk as well as cassette.

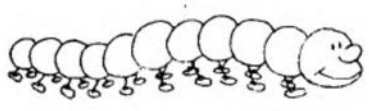
The following is a brief line description of the program:

Line	Description
10-20	Program set up
30-80	Title screen
90-250	DRAW and GET caterpillar segments
260-340	Put letter sequence on the screen
350-370	Receive keyboard input
380-390	Display answer in blank spot
400	Branch to 570 on correct answer
410-420	Wrong answer cue
430	Second try for first wrong answer
440-490	Flash correct answer
500-550	Erase sequence of letters
560	Return to start of loop
570-670	Put caterpillar segment on screen
680-840	Animation routine moves caterpillar left to right and off screen
850-1070	Animation routine moves caterpillar right to left and off screen
1080-1090	Await key press to begin again
1100-1380	Draw strings for letters of the alphabet

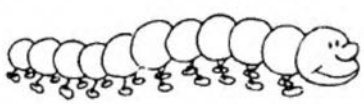
```
140 .....214
390 .....128
590 .....187
770 .....213
960 .....229
1150 .....106
END .....212
```

The listing: LETRSKIP

```
1 '****************************
2 '       CATERPILLAR
3 '       LETTER SKIP
4 '****************************
5 '             BY
6 '       MIKE KNOLHOFF
7 '       STERLING, IL.
8 '          MAY 1984
9 '****************************
10 C=0:DIMH1(14),B(14),H2(14),BL
(14)
20 CLS2:GOSUB1110
30 PRINT@201," CATERPILLAR ";:PR
INT@265," LETTER SKIP ";
40 A$="T303L4CP32L8CP128L4DP32L8
DP128EP128GP128EP128L4.C"
50 PLAYA$
60 PLAY"P8L4CP32L8CP128L4DP32L8D
P128L4.EP128CP8"
70 PLAYA$
80 PLAY"P8L3AP32DP128L8FP128L4.E
P128L1C"
90 PMODE3,1:PCLS:SCREEN0,1
100 CIRCLE(20,20),8,6,.9:PAINT(2
0,20),6,6
110 CIRCLE(24,16),2,8
120 DRAW"BM26,22C8L6H2"
130 DRAW"BM18,14C8U4E4BR4G4D4"
140 GET(12,2)-(28,32),H1,G
150 PCLS
160 CIRCLE(20,20),8,6,.9:PAINT(2
0,20),6,6
170 CIRCLE(16,16),2,8
180 DRAW"BM14,22C8R6E2"
190 DRAW"BM18,14C8U4H4BR4F4D4"
200 GET(12,2)-(28,32),H2,G
210 PCLS
220 CIRCLE(20,20),8,6,.9:PAINT(2
0,20),6,6
```



```
23Ø DRAW"BM18,26C8NG4BR4F4"
24Ø CIRCLE(2Ø,2Ø),4,8,.9
25Ø GET(12,2)-(28,32),B,G
26Ø PCLS
27Ø SCREEN1,1
28Ø R=RND(-TIMER)
29Ø R=RND(26)
3ØØ IF R<3THEN31ØELSEDRAW"BMØ,2Ø
C7S16"+L$(R-2)
31Ø IF R<2THEN32ØELSEDRAW"BM5Ø,2
ØC7S16"+L$(R-1)
32Ø DRAW"BM92,64C8S16R12"
33Ø IF R>25THEN35ØELSEDRAW"BM15Ø
,2ØC7S16"+L$(R+1)
34Ø IF R>24THEN35ØELSEDRAW"BM2ØØ
,2ØC7S16"+L$(R+2)
35Ø I$=INKEY$:IFI$=""THEN35Ø
36Ø A=ASC(I$)-64
37Ø IFA<1OR A>26THEN35Ø
38Ø DRAW"BM1ØØ,2ØC8S16"+L$(A)
39Ø FORK=1TO5ØØ:NEXTK
4ØØ IFA=R THEN57Ø
41Ø FORX=1TO5Ø:PLAY"V3ØT255L255O
1CP25Ø":NEXTX
42Ø W=W+1:DRAW"BM1ØØ,2ØC5S16"+L$
(A)
43Ø IFW=1THEN35Ø
44Ø W=Ø:FORX=1TO5
45Ø DRAW"BM1ØØ,2ØC8S16"+L$(R)
46Ø FORK=1TO1ØØ:NEXTK
47Ø DRAW"BM1ØØ,2ØC5S16"+L$(R)
48Ø FORK=1TO1ØØ:NEXTK
49Ø NEXTX
5ØØ IFR<3THEN51ØELSEDRAW"BMØ,2ØC
5S16"+L$(R-2)
51Ø IFR<2THEN52ØELSEDRAW"BM5Ø,2Ø
C5S16"+L$(R-1)
52Ø DRAW"BM1ØØ,2ØC5S16"+L$(R):DR
AW"BM92,64C5R12"
53Ø IFR>25THEN54ØELSEDRAW"BM15Ø,
2ØC5S16"+L$(R+1)
54Ø IFR>24THEN56ØELSEDRAW"BM2ØØ,
2ØC5S16"+L$(R+2)
55Ø FORK=1TO3ØØ:NEXT
56Ø GOTO29Ø
57Ø W=Ø
58Ø C=C+1
59Ø ON C GOTO6ØØ,61Ø,62Ø,63Ø,64Ø
,65Ø,66Ø
6ØØ PUT(96,1ØØ)-(112,13Ø),H1,PSE
T:GOTO665
61Ø PUT(8Ø,1ØØ)-(96,13Ø),B,PSET:
GOTO665
62Ø PUT(64,1ØØ)-(8Ø,13Ø),B,PSET:
GOTO665
63Ø PUT(48,1ØØ)-(64,13Ø),B,PSET:
GOTO665
64Ø PUT(32,1ØØ)-(48,13Ø),B,PSET:
GOTO665
65Ø PUT(16,1ØØ)-(32,13Ø),B,PSET:
GOTO665
66Ø PUT(Ø,1ØØ)-(16,13Ø),B,PSET
665 FORX=1TO5:SCREEN1,Ø:FORK=1TO
15Ø:NEXTK:SCREEN1,1:FORK=1TO15Ø:
NEXTK:NEXTX
67Ø PLAY"T255L255V3ØO2CDEFGABO3C
DEFGABO4CDEFGABAGFEDCO3BAGFEDCO3
BAGFEDCO2BAGFEDC"
68Ø IPC<7THEN5ØØ
69Ø POKE65495,Ø:SCREEN1,Ø:X1=112
:X2=128:M$="V31O1T255L255CP255C"
7ØØ FORK=1TO8
71Ø PUT(X1,1ØØ)-(X2,13Ø),H1,PSET
72Ø PUT(X1-16,1ØØ)-(X2-16,13Ø),B
,PSET
73Ø PUT(X1-112,1ØØ)-(X2-112,13Ø)
,BL,PSET
74Ø PLAYM$:FORD=1TO5Ø:NEXTD
75Ø X1=X1+16:X2=X2+16
76Ø NEXTK
77Ø PUT(224,1ØØ)-(24Ø,13Ø),B,PSE
T
78Ø PUT(128,1ØØ)-(144,13Ø),BL,PS
ET:PLAYM$:FORD=1TO12Ø:NEXTD
79Ø X1=144:X2=16Ø
8ØØ FORK=1TO6
8Ø5 PUT(224,1ØØ)-(24Ø,13Ø),BL,PS
ET
81Ø PUT(X1,1ØØ)-(X2,13Ø),BL,PSET
:PLAYM$
815 PUT(224,1ØØ)-(24Ø,13Ø),B,PSE
T
82Ø FORD=1TO8Ø:NEXTD
83Ø X1=X1+16:X2=X2+16:NEXTK
835 PUT(224,1ØØ)-(24Ø,13Ø),BL,PS
ET
84Ø FORD=1TO8ØØ:NEXTD
85Ø PUT(224,15Ø)-(24Ø,18Ø),H2,PS
ET:PLAYM$:FORD=1TO12Ø:NEXTD
86Ø X1=2Ø8:X2=224:FORK=1TO6
87Ø PUT(X1,15Ø)-(X2,18Ø),H2,PSET
88Ø PUT(X1+16,15Ø)-(X2+16,18Ø),B
,PSET:PLAYM$
89Ø X1=X1-16:X2=X2-16
9ØØ FORD=1TO8Ø:NEXTD
91Ø NEXTK
92Ø X1=112:X2=128
93Ø FORK=1TO8
94Ø PUT(X1,15Ø)-(X2,18Ø),H2,PSET
95Ø PUT(X1+16,15Ø)-(X2+16,18Ø),B
,PSET
96Ø PUT(X1+112,15Ø)-(X2+112,18Ø)
,BL,PSET:PLAYM$
97Ø FORD=1TO5Ø:NEXTD
98Ø X1=X1-16:X2=X2-16:NEXTK
99Ø PUT(96,15Ø)-(112,18Ø),BL,PSE
T
1ØØØ PUT(Ø,15Ø)-(16,18Ø),B,PSET:
PLAYM$
1Ø1Ø FORD=1TO12Ø:NEXTD
1Ø2Ø X1=8Ø:X2=96
1Ø3Ø FORK=1TO6
1Ø35 PUT(Ø,15Ø)-(16,18Ø),BL,PSET
1Ø4Ø PUT(X1,15Ø)-(X2,18Ø),BL,PSE
T:PLAYM$
1Ø45 PUT(Ø,15Ø)-(16,18Ø),B,PSET
1Ø5Ø FORD=1TO8Ø:NEXTD
1Ø6Ø X1=X1-16:X2=X2-16
1Ø7Ø NEXTK
1Ø75 PUT(Ø,15Ø)-(16,18Ø),BL,PSET
1Ø8Ø I$=INKEY$:IFI$=""THEN1Ø8Ø
1Ø9Ø POKE65494,Ø:C=Ø:SCREEN1,1:G
OTO5ØØ
11ØØ END
111Ø DIML$(26)
112Ø L$(1)="BD1ØU5NR8U1E4F4D6BR4
BU1Ø"
113Ø L$(2)="D1ØR6E2U1H2NL6E2U1H2
L6BR12"
114Ø L$(3)="BR3NR5G3D4F3R5BR3BU1
Ø"
115Ø L$(4)="D1ØR5E3U4H3L5BR12"
116Ø L$(5)="NR8D5NR6D5R8BR4BU1Ø"
117Ø L$(6)="NR8D5NR6D5BR12BU1Ø"
118Ø L$(7)="BR8L6G2D6F2R4E2U3L4B
R8BU5"
119Ø L$(8)="D1ØU5R8D5U1ØBR4"
12ØØ L$(9)="BR1R6L3D1ØL3R6BR5BU1
Ø"
121Ø L$(1Ø)="BR2R6L3D8G2L1H2U2BR
12BU6"
122Ø L$(11)="D1ØBR8L1H5NL2E5R1BR
4"
123Ø L$(12)="D1ØR8BR4BU1Ø"
124Ø L$(13)="ND1ØF4E4ND1ØBR4"
125Ø L$(14)="ND1ØD1F8D1U1ØBR4"
126Ø L$(15)="BR2G2D6F2R4E2U6H2L4
BR1Ø"
127Ø L$(16)="ND1ØR6F2D1G2L6BR12B
U5"
128Ø L$(17)="BR2G2D6F2R4E1NH3NF1
E1U6H2L4BR1Ø"
129Ø L$(18)="ND1ØR6F2D1G2L6R1F5B
R4BU1Ø"
13ØØ L$(19)="BR8L6G2D1F2R4F2D1G2
L6BR12BU1Ø"
131Ø L$(2Ø)="R4ND1ØR4BR4"
132Ø L$(21)="D8F2R4E2U8BR4"
133Ø L$(22)="D6F4E4U6BR4"
134Ø L$(23)="D1ØE4NU1F4U1ØBR4"
135Ø L$(24)="D1F8D1BL8U1E8U1BR4"
136Ø L$(25)="D1F4ND5E4U1BR4"
137Ø L$(26)="R8D1G8D1R8BR4BU1Ø"
138Ø RETURN
```



64K WORD GAME

HOMOPHONE

by Bob Horne

Homophones are words that sound alike but have different spelling and different meanings. For example, THERE and THEIR. Such pairs of words are often referred to as HOMONYMS.

This program uses the same machine language routine as "Antonyms" to print upper and lower case letters to the graphics screen. Normally, 42 characters per line are printed, and 24 lines are printed before scrolling.

I have utilised 2 additional features of this program here. In line 70, the POKEs alter the number of characters per line to 21. Line 130 sets things back to 42 characters per line. POKEs to 243 and 244 set the position for printing on the screen. For example:

```
POKE243,10:POKE244,6:PRINT"HELLO"
```

will begin printing 11 spaces across on the 7th line down.

The second feature of this program is a POKE to 246. This location is normally set to 6. In "Homophones", I set this to 12. EXEC31930, which normally clears the whole screen, will now only clear the text beneath the graphics reward area. Another example of this would be if you wanted to emulate the APPLE II hi-res Graphics screen, page 1, the

POKE246 with a value from 26. This enables graphics on the screen with 4 lines of text beneath.

"Homophones" has many subroutines. Actually, the main loop is only 12 lines long. A lot of string manipulation takes place, hence the large CLEAR number in line 20.

The following is a summary of how the program works:

10 - 60: Dimensions the arrays, DRAW and GET the graphic rewards, POKE in the machine language routine and read the main data.

70 - 120: The title screen.

130 - 150: Asks if you need instructions. After fist time through the program, it returns always to this screen for the next user.

160 - 200: Type in your name - but keep it down to 12 letters.

210 - 250: Set the variables and draw a box for the graphic reward.

260 - 370: This is the main loop. In line 310, the POKE282,0 prints lower case letters to the screen during input and POKE282,255 returns things to normal. ST is a variable that tells if the answer if right at first try or not.

380 - 450: This is the review section. In line 430 the correct answer is converted from lower case to upper case for emphasis.

460 - 480: This ends your turn and it is time for the next person to take over.

490: A short pause.

500 - 520: Wait for the user to read the screen.

530: Clears the text on the screen.

540 - 560: This is the "correct answer" routine - whether it was the first attempt or not.

650 - 750: This is the "wrong answer" routine. If ST=1 then this is the second time it was wrong.

760 - 780: This is the scoreboard routine.

790 - 840: This is a formatting routine that ensures that no word - splitting occurs at the ends of lines.

850 - 900: Instructions

910 - 970: This routine is used on the title screen to scroll individual letters in from the left and then up the screen.

980 - 1040: Draws a "Smiley" and a "Saddy".

1050 - 1210: Data for the machine language routine.

1220 - 1570: The data used by the main program. Note the format like this:

HOMOPHONE, HOMOPHONE, THE FIRST PART OF THE SENTENCE TO THE MISSING WORD, THE REST OF THE SENTENCE, THE CORRECT ANSWER

I have included 36 examples. If you have a different number of examples, then alter the value if ZZ in line 30 accordingly.

(This program crashes when used with the disk operating system plugged in - how about a fix from one of you! G.)

THE LISTING:

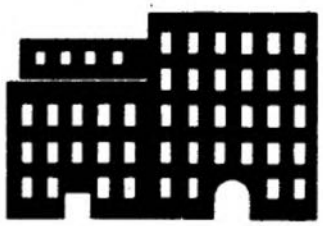
```
1 REM***************************
2 REM*      HOMOPHONES         *
3 REM*          BY             *
4 REM*       BOB HORNE         *
5 REM*     IPSWICH, QLD.       *
6 REM***************************
10 CLS:PRINT@128,"NOW LOADING MA
CHINE LANGUAGE.":PRINT:PRINT"STA
ND BY."
20 CLEAR400,31918
30 NU=10:ZZ=36:DIMA1$(ZZ),A2$(ZZ
),Q1$(ZZ),Q3$(ZZ),A$(ZZ),Z(ZZ),F
(19),F1(19):SW=42
40 GOSUB980
50 FORX=31919TO32767:READA$:POKE
X,VAL("&H"+A$):NEXTX
60 FORX=1TO ZZ:READA1$(X),A2$(X)
,Q1$(X),Q3$(X),A$(X):NEXTX
70 POKE32021,216:POKE32066,21:PO
KE32151,21
80 EXEC31919
90 H=5:V=10:KK$="HOMOPHONES":GOS
UB910
100 H=9:V=12:KK$="BY":GOSUB910
110 H=5:V=14:KK$="BOB  HORNE":GO
SUB910
120 GOSUB490
130 POKE32021,248:POKE32066,42:P
OKE32151,42:GOSUB530:Q=RND(-TIME
R)
140 POKE246,6:GOSUB530:POKE243,5
:POKE244,10:PRINT"Do you want in
structions ?":POKE243,5:POKE244,
12:PRINT"Type 'Y' or 'N'"
150 IN$=INKEY$:IF IN$="" THEN150
 ELSE IF IN$="Y" THENGOSUB850 EL
SE IF IN$="N" THEN160 ELSE150
160 GOSUB530
170 POKE246,12:GOSUB530:POKE244,
5:PRINT"Please type your name an
d press the      <ENTER> key...
..":INPUT"-)";NM$
180 IF LEN(NM$))12 THEN NM$=LEFT
$(NM$,12):POKE244,10:PRINT"You h
ave a long name. I will call you
 ":PRINTNM$:GOSUB490
190 GOSUB530:H=0:V=5:JK$="I'm pl
eased to meet you, "+NM$+"."+" W
e'll have some fun together.":GO
SUB790
200 GOSUB490
210 GOSUB530
220 Q2$=" ..... "
230 FORI=1TO ZZ:Z(I)=0:NEXTI
240 T=0:R=0:ST=0
250 COLOR0:LINE(0,0)-(255,32),PS
ET,B
260 FORI=1TO NU
270 Q=RND(ZZ):IF Z(Q)=1 OR Z(Q)=
2 THEN270 ELSEZ(Q)=1
280 T=T+1
290 PRINT"Question"I":":PRINTA1$
(Q)" , "A2$(Q)
300 H=0:V=4:JK$=Q1$(Q)+Q2$+Q3$(Q
):GOSUB790
310 POKE244,7:PRINTSTRING$(42,45
);:PRINT"Type a word and press E
NTER.":POKE282,0:INPUT"-) ";R$:P
OKE282,255
320 IF R$=A$(Q) THEN330 ELSE350
330 R=R+.5:IF ST=0 THENR=R+.5
340 GOSUB540:GOTO360
350 GOSUB650:Z(Q)=2:IF ST=0 THEN
 ST=1:GOSUB530:GOTO290
360 ST=0:GOSUB760:GOSUB500
370 NEXTI
380 POKE246,6:GOSUB530:PRINT"Her
e are the problems you missed":G
OSUB500
390 IF R=T THENFORX=1TO5:PLAY"V3
103T255;1;2;3;4;5;6;7;8;9;10;11;
12;04;1;2;3;4;5;6;7;8;9;10;11;12
;03;12;11;10;9;8;7;6;5;4;3;2;1":
NEXTX:PRINT"Congratulations ";NM
$:PRINT"You didn't miss any!!":G
OSUB500:GOTO460
400 H=0:V=0
410 FORI=1TO ZZ
420 IF Z(I)<>2 THEN450
430 IF Z(I)=2 THEN WA$="":FORX=1
TOLEN(A$(I)):X$=MID$(A$(I),X,1):
X$=CHR$(ASC(X$)-32):WA$=WA$+X$:N
```

```
EXTX:JK$="* "+Q1$(I)+" "+WA$+" "
+Q3$(I):GOSUB790
440 V=V+2
450 NEXTI:GOSUB500
460 GOSUB530:POKE244,10:PRINT"An
other turn (Y/N) "
470 IN$=INKEY$:IF IN$="" THEN470
 ELSEIF IN$="Y" THEN GOSUB530:GO
TO140 ELSE IF IN$="N" THEN GOSUB
530:END ELSE470
480 GOSUB530:END
490 FORX=1TO500:NEXTX:RETURN
500 PRINT:PRINT"Press (ENTER) to
 continue.";:PLAY"T100L10002;4;8
;12"
510 IN$=INKEY$:IF IN$="" THEN510
 ELSE IF IN$()CHR$(13) THEN510
520 GOSUB530:RETURN
530 EXEC31930:RETURN
540 IF ST=1 THENPRINT"It's right
 this time. GOOD WORK.":GOSUB490
:RETURN
550 PLAY"V31O3T200L10CC#DE-EFF#G
GG#AA#B04CC#DE-EFF#T100IL4CL200O
4BA#AG#GF#FEE-DCO3BA#AG#GF#FEE-D
CO2BA#AG#GF#FEE-DC#CO2BA#AG#GF#F
EE-DC#CO1BA"
560 PRINT:X=RND(6):ON X GOSUB590
,600,610,620,630,640
570 IF ST=0 THEN PUT(I*24-17,2)-
(7+I*24,30),F,PSET
580 GOSUB490:RETURN
590 PRINT"You're hot!!":RETURN
600 PRINT"Terrific!!!!":RETURN
610 PRINT"Spot on!!!!!":RETURN
620 PRINT"Great going!":RETURN
630 PRINT"Bonza "N$"!!":RETURN
640 PRINT"Super "N$"!!":RETURN
650 PLAY"O3V30L20T2B-B-B-L5V25E-
V20FO2B-"
660 IF ST=1 THENPRINT"Sorry, sti
ll wrong. The correct answer is
";A$(Q):RETURN
670 PRINT:X=RND(5):ON X GOSUB710
,720,730,740,750
680 PUT(I*24-17,2)-(7+I*24,30),F
1,PRESET:GOSUB490
690 GOSUB530
700 RETURN
710 PRINT"No. Try harder.":RETUR
N
720 PRINT"That's not right!! ";N
M$:RETURN
730 PRINT"Wrong! Wrong! Wrong!":
RETURN
740 PRINT"What!!!!!!":RETURN
750 PRINT"Not again!!!":RETURN
760 P$="s, ":IF T=1 THEN P$=", "
770 PRINTSTRING$(42,45);:PRINT"Y
ou have"R"correct out of"T"probl
em";P$:PRINT"for a score of"INT(
R*100/T+.5)"percent.":PRINTSTRIN
G$(42,45);
780 RETURN
790 IF LEN(JK$)<=SW THEN830
800 FORX=SW TO 0 STEP-1:IF MID$(
JK$,X,1)=" " THEN820
810 NEXTX:GOTO830
820 L$=LEFT$(JK$,X):W$=L$:GOSUB8
40:JK$=RIGHT$(JK$,(LEN(JK$))-X):
GOTO790
830 W$=JK$:POKE243,H:POKE244,V:P
RINTW$:RETURN
840 POKE243,H:POKE244,V:PRINTW$:
V=V+1:H=0:RETURN
850 GOSUB530:POKE243,15:PRINT"HO
MOPHONES.":PRINT:PRINT"CURRENT i
s a flow of water, air or
 electricity.":PRINT"CURRANT is
a dried grape."
860 PRINT:PRINT" Because these t
wo words SOUND the same  but ar
e SPELT differently and have
    different meanings, they are
 called       HOMOPHONES."
870 PRINT:PRINT" While the progr
am is running just follow the di
rections on the screen."
880 POKE243,10:POKE244,22:PRINT"
Press (ENTER) to start."
890 IN$=INKEY$:IF IN$="" THEN890
 ELSE IF IN$()CHR$(13) THEN890
900 RETURN
910 FORX1=1TOLEN(KK$):JK$=MID$(K
K$,X1,1):IF JK$=CHR$(32) THEN H=
H+1:NEXTX1 ELSEGOSUB920:NEXTX1:R
ETURN
920 POKE243,1:POKE244,21:FORX2=1
TO H-1:POKE243,X2:PRINT" ";JK$;:
SO$=STR$(RND(12)):PLAY"O1T80V20;
XSO$;":NEXTX2
930 FORX=21TO V STEP-1
940 POKE243,H:POKE244,X:PRINTJK$
:POKE243,H:PRINT" "
950 NEXTX:SO$=STR$(RND(12)):PLAY
"O2V31;XSO$;"
960 H=H+1
970 RETURN
980 PMODE4,1:PCLS1:COLOR0:CIRCLE
(128,95),11,,1.3
990 CIRCLE(124,92),3:CIRCLE(132,
92),3
1000 DRAW"BM126,100NH2R5E2M131,1
02L5M124,98":DRAW"BM117,89L1D6R1
BM139,89RD6L"
1010 GET(116,80)-(140,108),F,G
1020 DRAW"C1":LINE(124,98)-(133,
102),PSET,BF:DRAW"C0BM126,100NG2
R5F2"
1030 GET(116,80)-(140,108),F1,G
1040 RETURN
1050 DATACC,06,00,97,F5,DD,F6,88
,18,DD,F8,0F,F3,0F,F4,CC,FF,FF,9
E,F6,ED,81,ED,81,9C,F8,25,F8,30,
8C,0A,BF,01,6B,30,8C,39,BF,01,68
,39,0D,6F,27,03,7E,8C,F1,0F,70
1060 DATA34,14,CC,01,00,34,06,BD
,A1,C1,26,12,6A,E4,26,F7,63,61,2
7,03,86,80,8C,86,20,17,00,A7,20,
E9,A7,E4,86,20,17,00,9E,35,06,35
,14,32,62,39,0D,6F,27,01,39,34,1
6
1070 DATA86,F8,B7,FF,22,96,F5,44
,8A,80,8E,FF,C6,A7,1A,A7,1D,A7,1
F,44,27,0C,25,04,A7,81,20,F7,A7,
01,30,02,20,F1,A6,E4,81,20,25,49
,8D,66,DC,F3,4C,81,2A,25,39,4F,9
7
1080 DATA89,5C,34,04,DB,F6,D0,F8
,35,04,25,2B,5A,34,66,9E,F6,33,8
9,01,00,20,0E,37,26,ED,81,10,AF,
81,37,26,ED,81,10,AF,81,11,93,F8
,25,ED,CC,FF,FF,ED,81,ED,81,9C
1090 DATAF8,26,F8,35,66,DD,F3,35
,16,32,62,39,81,0D,26,04,D6,F4,2
0,B8,81,08,26,EF,DC,F3,4A,2A,06,
86,2A,4A,5A,2B,E4,DD,F3,86,20,8D
,02,20,DC,34,36,34,06,96,F3,C6
1100 DATA06,3D,CB,02,54,49,54,49
,54,49,A7,60,96,F4,9B,F6,1F,01,A
6,60,31,8C,51,A6,A6,A7,60,43,A7,
61,31,8C,AF,A6,62,C6,05,D7,89,3D
,31,AB,CC,08,06,34,06,5F,20,06,8
6
1110 DATA08,A7,E4,E6,A0,59,A6,84
,24,04,A4,63,20,02,AA,62,A7,84,3
0,88,20,6A,E4,26,EC,30,89,FF,00,
6A,61,27,10,A6,62,44,24,04,30,01
,86,80,A7,62,43,A7,63,20,CE,32
1120 DATA64,35,86,80,08,20,02,40
,04,10,01,00,00,00,00,00,00,00,F
A,00,00,00,E0,00,E0,00,28,FE,00,
FE,28,24,54,D6,54,48,C6,C8,10,26
,C6,6C,92,6A,04,0A,00,00,E0,E0,0
0
1130 DATA38,44,82,00,00,00,00,82
,44,38,10,38,7C,38,10,10,10,7C,1
0,10,1A,1C,00,00,00,10,10,10,10,
10,06,06,00,00,00,06,08,10,20,C0
,7C,8A,92,A2,7C,00,42,FE,02,00
1140 DATA4E,92,92,92,62,44,82,92
,92,6C,10,30,50,FE,10,E4,A2,A2,A
2,9C,7C,92,92,92,0C,86,88,90,A0,
C0,6C,92,92,92,6C,60,92,92,92,7C
,00,6C,6C,00,00,00,DA,DC,00,00,1
0
1150 DATA28,44,82,00,28,28,28,28
```

```
,28,00,82,44,28,10,40,80,9A,60,0
0,4C,92,9A,82,7C,3E,48,88,48,3E,
82,FE,92,92,6C,7C,82,82,82,44,82
,FE,82,82,7C,FE,92,92,92,82,FE
1160 DATA90,90,90,80,7C,82,82,92
,9E,FE,10,10,10,FE,00,82,FE,82,0
0,0C,02,02,02,FC,FE,10,28,44,82,
FE,02,02,02,02,FE,40,30,40,FE,FE
,40,20,10,FE,7C,82,82,82,7C,FE,9
0
1170 DATA90,90,60,7C,82,8A,84,7A
,FE,90,98,94,62,44,A2,92,8A,44,8
0,80,FE,80,80,FC,02,02,02,FC,E0,
18,06,18,E0,FE,04,18,04,FE,C6,28
,10,28,C6,C0,20,1E,20,C0,86,8A
1180 DATA92,A2,C2,FE,82,82,00,00
,C0,20,10,08,06,00,00,82,82,FE,2
0,40,80,40,20,01,01,01,01,01,00,
80,40,20,00,1C,22,22,3C,02,FE,12
,22,22,1C,1C,22,22,22,04,1C,22,2
2
1190 DATA12,FE,1C,2A,2A,2A,18,00
,10,7E,90,40,18,25,25,25,1E,FE,2
0,20,1E,00,00,00,BE,00,00,04,02,
22,BC,00,FE,08,14,22,00,00,00,FE
,00,00,1E,20,1E,20,1E,3E,10,20
1200 DATA20,1E,1C,22,22,22,1C,3F
,24,24,24,18,18,24,24,24,3F,3E,1
0,20,20,00,12,2A,2A,2A,04,20,7C,
22,04,00,3C,02,02,04,3E,38,04,02
,04,38,3C,02,0C,02,3C,22,24,3E,1
2
1210 DATA22,38,05,05,05,3E,22,26
,2A,32,22,10,6C,82,82,00,00,00,F
F,00,00,00,82,82,6C,10,30,40,20,
10,60,AA,55,AA,55,AA,FF,FF,FF,FF
,FF
1220 DATArains,reins,It seems th
at it often,at the weekend.,rain
s
1230 DATAsaw,sore,The boy has a,
on his foot.,sore
1240 DATAsaw,sore,He used a,to c
ut the piece of wood.,saw
1250 DATApiece,peace,I found a,o
f glass on the field.,piece
1260 DATApiece,peace,After many
years of war,was welcome.,peace
1270 DATAsail,sale,Woolworths su
permarket had a,last Tuesday.,sa
le
1280 DATAsail,sale,The boat lost
 its,in the strong wind.,sail
1290 DATAstares,stairs,He is rud
e when he,at other people.,stare
s
1300 DATAstares,stairs,Look at t
he boy on the,.,stairs
1310 DATAtheir,there,My book is,
on the table.,there
1320 DATAtheir,there,Those are,b
ooks on the table.,their
1330 DATAbare,bear,That is a sav
age,near the tree.,bear
1340 DATAbare,bear,His head is,.
,bare
1350 DATAwear,where,What shall I
,to the party?,wear
1360 DATAwear,where,James asked
his father,they were going for t
he holidays.,where
1370 DATAfair,fare,Did you pay t
he bus,to the driver?,fare
1380 DATAfair,fare,It is not,if
you take the ball now.,fair
1390 DATAfair,fare,Shane has,hai
r.,fair
1400 DATApassed,past,The witch,b
y on her broom.,passed
1410 DATApassed,past,We went,the
 football field.,past
1420 DATAtire,tyre,Father bought
 a new,for the car.,tyre
1430 DATAtire,tyre,When doing ex
ercises a sick person may,quickl
y.,tire
1440 DATAscent,sent,We,the parce
l two weeks ago.,sent
1450 DATAscent,sent,The,bottle b
roke when it fell.,scent
1460 DATAcent,scent,I gave one,t
o the mission funds.,cent
1470 DATAfarther,father,Matthew'
s,gave him a new bicycle.,father
1480 DATAfarther,father,On the w
alkathon I walked,than John.,far
ther
1490 DATApraise,prays,Mark recei
ved much,for his neat work.,prai
se
1500 DATApraise,prays,The boy,to
 God every night.,prays
1510 DATAplain,plane,The kangaro
o hopped across the,.,plain
1520 DATAplain,plane,The,took of
f at twelve o'clock.,plane
1530 DATAmedal,meddle,You should
 not,in other people's affairs.,
meddle
1540 DATAmedal,meddle,The brave
soldier received a,.,medal
1550 DATAcurrant,current,The,bun
s came in late today.,currant
1560 DATAcurrant,current,The ele
ctric,from the battery is very w
eak.,current
1570 DATArains,reins,The horse r
ider had hold of the,.,reins
```

32K

WINDOWS



by Ross Eldridge and Graham Morphett

Some hi-res text screens look pretty good. In fact, the one used by Tom Horne in his Homophones program this month looks excellent!

CoCo can make programs which use these or other screens look and perform better by allowing you to use windows.

The accompanying program is designed to be loaded after you've loaded and run Tom's "Homophones".

Don't RESET, just load and RUN "Homophones". When the program asks if you require instructions, press BREAK and CLOAD "WINDOWS".

The Listing:

```
1 '**********WINDOW**********
  *********ROSS ELDRIDGE********
2 GOTO10
3 SAVE"WINDOW:2":DIR2:STOP
10 PMODE4,1:COLOR0,3:PCLS
20 PRINT"THIS IS A TEST TO SEE H
OW "
30 PRINT"AN EFFECT SIMILAR TO WI
NDOWING"
40 PRINT"MAY BE ACCOMPLISHED ON
THE TANDY"
50 PRINT"COLOR COMPUTER."
60 PRINTSTRING$(15,13)
70 A$=INKEY$:IF A$="" THEN 70
80 PCOPY1 TO 5
90 COLOR0:LINE(30,0)-(150,40),PS
ET,BF
92 PCOPY 1 TO 6
100 A$=INKEY$:IF A$=""THEN100
110 IF A$="1" THEN PCOPY 5 TO 1
115 IF A$="2" THEN PCOPY 6 TO 1
120 GOTO 100
```

BRIAN'S BITS

THE BEST QUAILTY HARDWARE ADDITIONS FOR YOUR COLOR COMPUTER AVAILABLE IN AUSTRALIA

Rainbow Bits is no more, but I will continue to bring you the best range of quality hardware for the colour computer including modems.

SPECIAL OFFER Save 33%

I have a few only automatic modems with the Rainbow Bits package. To save the trouble of repackaging, these are available at a once only clearance price of only $200.00 + p&p

Disk Controller (Gold Contacts), compact size, fully Tandy compatible, but with many extra features (eg Auto line numbering, Monitor, RAM, OS-9 and FLEX commands). $180.00 + p&p

Disk Drives. Only the best quality CHINNON or TEAC with controller & cable with instruction manual. P.O.A.

Video Amp. Colour or Mono, with audio amp. Plugs in, no soldering. $ 25.00

Postage & Packing $ 5.00.
Registered Mail $10.00.

Phone Brian 07-30-2072, or write to 17 Penley St., THE GAP. QLD. 4061.

Blaxland Computer Services PTY. LTD.

Specialists in Color Computer nardware and software and a wide range of components

Now Authorised TANDY DEALER #9254

64K RAM UP GRADE KIT.....Posted $89.00

HARDWARE

PACKAGE DEALS

128K UPGRADES (suit grey and early white) — $220

SOFTWARE

C-10 Cassettes $1.50 each.

DISKETTES S.S D.D. 40T per 10.....$30
VIDEO MONITOR DRIVER (with sound output).....$32
MAILING LABLES per 1000 from.....$16.00
CASSETTE LABELS per 100.....$4.00
PRINTER PAPER 70 g.s.m. 9½" x 11" per 1000 $22.00
PRINTER PACKAGE: BMC-80 PRINTER, SERIAL ADAPTOR SCREE DUMP PROGRAMME.....$499.00
(adaptor plugs directly to CO-CO and 9600 Baud selectable)
DISK DRIVES
COMPLETE UNITS (ready to run)
1 x 40T D.S.D.D.....$600.00
2 x 40T D.S.D.D.....$1100.00
1 x 40T double case and power supply.....$740

Amber Monitor 35mhz $195

Bankcard welcomehere

Model 1000 Specialists.

POSTAGE AND FREIGHT EXTRA (where applicable)

76A MURPHY ST, BLAXLAND. 2774

PHONE : (047) 39 3903 P.O.BOX 125 BLAXLAND. 2774

SMALL SCHOOL LIBRARY BORROWING SYSTEM

by Ken Stewart

(Ken supplies this program on disk along with further operating information and a demonstration file. However we are pleased to be able to show you this program and to give you a listing for it. G.)

Minimum system required:

- . Tandy Colour Computer 16k or 64k
- . 1 Disk Drive
- . 80 Column Printer optional

-Instructions-

1. Switch on computer system (T.V., computer, drive, printer). You will see the start up message on the screen.
2. For 16K, type POKE 25,6:NEW (ENTER)
3. Insert the disk in the drive and close the door.
4. Type RUN "L (ENTER). The drive will start, and the title screen will appear. Press ENTER and the program will start.

A demonstration file of borrowers is included on the disk. To see this file, press (2). Press (2) again to see the whole file. Note the screen layout.

When the whole file has been displayed, you will return to the "See File" menu. Pressing '0' at any time will return you to this menu.

An individual borrower's record can be seen by pressing (1). Then type in the name you wish to check. Type surname, then a comma, and then first name WITHOUT ANY SPACE. Press (ENTER). If you have made a mistake or the name can't be found you will be prompted to try again.

HARDCOPY: If you have a printer, select (3) (printout). Press (ENTER) when the printer is ready. The whole file will be printed out.

CHANGING BOOKS: Only borrowers currently on the file can borrow books. When the name is entered, books currently on loan are displayed.

If books are to be returned and none others borrowed, press (ENTER) for a blank entry. When finished, press (3) to go to the Main Menu.

OVERDUES: The program will display all borrowers who have overdue books, on screen or printer. From the Main Menu, press (4). At the prompt, type in the date in numerals, with a leading zero if necessary. A 2 digit numeral is required. Select (S) or (P) and the overdue listing will begin.

END: Press (5) from the Main Menu to end the program. Caution: Do not press (BREAK) or Reset to stop the program. If (BREAK) is accidentally pressed, type CONT (ENTER) to continue the program.

DELETING: To build your own file of borrowers, first delete the demonstration file. Press (3) from the Main Menu. Press (2) to Delete, and you will be prompted whether to delete each record in turn.

The quick way to delete the demonstration file is to exit the program, then type KILL "NAMES/BKS" (ENTER).

BUILDING YOUR OWN FILE: The program builds a file of borrowers, their books, and the due date for returns, called "NAMES/BKS". This is arranged alphabetically, and any future entries are also put in alphabetical order.

To build your own file, from the Main Menu press (1). You will be prompted to enter the borrowers surname, first name, and book details. When you have entered your list of names, type (N) for "no more data" and wait while the computer puts the file in alphabetical order. This file can be added to at any time in the future.

NOTE: A maximum of 22 characters can be used for the borrower's name, 12 for call number, 4 for accession number, and 8 for due date. Two books may be borrowed at a time.

USING THE PROGRAM

This program is suitable for a small school or a single class. It is based on the "call no., accession no.", borrowing system. Up to 35 borrowers may use it at a time, borrowing up to 2 books at a time. For more than 35 borrowers, either COPY "L/BAS" onto another disk and use a separate disk for each class, or else see program details for details of lines to change.

Older children use this program with ease with little supervision; children in Year 3 can use it with help. Otherwise, children could fill in ordinary borrowing cards which can be later entered by a proficient operator.

A listing of borrowings can be printed daily or weekly, and overdues can be checked and listed at any time.

The program is of course not so much a library management system, as a computer awareness program. Children should be shown how computers keep records, search and update information, and report to the user, as in the commercial world.

PROGRAM DETAILS:

The 3 main routines in the program are adapted directly from sample programs in the Disk Manual.

There is a "speed up poke" in lines 590 and 830. If your computer will not handle these, delete these lines.

The program occupies about 9500 bytes; when inputting new data an extra 2500 is cleared. To fit in 16k you should POKE 25,6:NEW (ENTER) before loading. (Other memory can be saved by DEL - 100, DEL 3820 -, and removing all REM lines).

The size of the Names/Bks file is determined in line 510.

If you want more than 35 records, change the value of ARRAY$ and CLEAR in this line.

Each record is 70 characters long, so for 100 borrowers you would need to clear at least 7000.

If you would like a copy of this program, please forward a blank disk + $10, or $20 and I will send one to you with demonstration file and instructions. Please contact me for help or if you have comments or suggestions, at:

Greycliffe State School
M/S 1396
Biloela. QLD. 4715.
079-95-3153.

The Listing:

```
1 '*****SMALL LIBRARY SYSTEM*****
  ***********KEN STEWART********
  ***********24/7/85***********
2 GOTO10
3 SAVE"L:2":DIR
10 ' START UP AND TITLE MODULE
20 CLS0:FOR H=14 TO 49:SET(H,6,3
):SET(H,16,3):NEXT H
30 FOR V=6 TO 16:SET(14,V,3):SET
(49,V,3):NEXT V
40 PRINT@172," LIBRARY ";:PRINT@
204,"BORROWING";:PRINT@296,"BY K
EN STEWART, ";:PRINT@328,"GREYCL
IFFE S.S.,";:PRINT@360,"M.S.1396
,BILOELA";:PRINT@392,"PHONE 079-
953153";
50 PRINT@453,"PRESS (ENTER) TO S
TART";
60 AN$=INKEY$:IF AN$="" THEN 60
70 IF AN$=CHR$(13) THEN 120
80 GOTO 50
110 ' MAIN MENU
120 CLS3:PRINT@74,"MAIN MENU";:S
OUND 200,1
130 PRINT@105,"SELECTIONS:";
140 PRINT@168,"(1) INPUT DATA";
150 PRINT@200,"(2) SEE FILE  ";
160 PRINT@232,"(3) CHANGE BKS";
170 PRINT@264,"(4) OVERDUES  ";
180 PRINT@296,"(5) END       ";
190 PRINT@360," ENTER CHOICE ";
200 CH$=INKEY$:IF CH$="" THEN 20
0
210 ON VAL(CH$) GOSUB 500,1000,1
500,2000,2500
220 GOTO 120
495 'INPUT FILE
500 CLS:SOUND100,1
510 CLEAR2500:DIM ARRAY$(35)
520 OPEN "D",#1,"NAMES/BKS",70
530 FIELD #1,70 AS INFO$
540 IF LOF(1)=0 THEN I=1:GOTO 60
0
550 FOR I=1 TO LOF(1)
560 GET #1,I
570 ARRAY$(I)=INFO$
580 NEXT I
590 POKE 65495,0
600 CLS2:SOUND50,1
610 PRINT@75,"INPUT DATA";
620 FOR WAIT=1 TO 1000:NEXT
630 CLS:PRINT:PRINT
640 INPUT "SURNAME   ";S$
650 INPUT "FIRST NAME";F$
655 PRINT"          FIRST ITEM"
660 INPUT "CALL NO.  ";C1$
670 INPUT "ACCESS.NO.";A1$
680 INPUT "DUE DATE (DD/MM/YY)";
D1$
682 PRINT"          SECOND ITEM"
684 INPUT "CALL NO.  ";C2$
686 INPUT "ACCESS.NO.";A2$
688 INPUT "DUE DATE (DD/MM/YY)";
D2$
689 GOSUB 910
690 ARRAY$(I)=LEFT$(S$+","+F$+"
              ",22)+C1$+A1$+D1$+
C2$+A2$+D2$
700 PRINT@426,"MORE DATA (Y/N)?"
710 AN$=INKEY$
720 IF AN$="Y" THEN I=I+1:GOTO
630
730 IF AN$="N" THEN 750
740 GOTO 710
750 CLS4:PRINT@198,"SORTING- PLE
ASE WAIT";:FOR J=1 TO I
760 FOR K=J TO I
770 IF ARRAY$(J) (ARRAY$(K) THEN
810
780 TEMP$=ARRAY$(J)
790 ARRAY$(J)=ARRAY$(K)
800 ARRAY$(K)=TEMP$
810 NEXT K
820 NEXT J
830 POKE 65494,0
850 FOR N=1 TO I
860 LSET INFO$=ARRAY$(N)
870 PUT #1,N
880 NEXT N
890 CLOSE #1
900 GOTO 120
910 IF LEN(C1$)(12 THEN 950
915 IF LEN(C1$))12 THEN 955
920 IF LEN(A1$)(4  THEN 960
925 IF LEN(A1$))4  THEN 965
930 IF LEN(D1$)(8  THEN 970
932 IF LEN(D1$))8  THEN 972
934 IF LEN(C2$)(12 THEN 974
936 IF LEN(C2$))12 THEN 976
938 IF LEN(A2$)(4  THEN 978
940 IF LEN(A2$))4  THEN 980
942 IF LEN(D2$)(8  THEN 982
944 IF LEN(D2$))8  THEN 984
946 GOTO 990
950 C1$=C1$+" ":GOTO 910
955 C1$=LEFT$(C1$,12):GOTO 915
960 A1$=A1$+" ":GOTO 920
965 A1$=LEFT$(A1$,4):GOTO 925
970 D1$=D1$+" ":GOTO 930
972 D1$=LEFT$(D1$,8):GOTO 932
974 C2$=C2$+" ":GOTO 934
976 C2$=LEFT$(C2$,12):GOTO 936
978 A2$=A2$+" ":GOTO 938
980 A2$=LEFT$(A2$,4):GOTO 940
982 D2$=D2$+" ":GOTO 942
984 D2$=LEFT$(D2$,8):GOTO 944
990 RETURN
995 'SEE FILE
1000 CLS8:SOUND 50,1:PRINT@104,"
(1) SEE ONE NAME";:  PRINT@136,"
(2) SEE ALL FILE";:  PRINT@168,"
(3) PRINT   OUT ";:  PRINT@200,"
(4) QUIT        ";
1002 AN$=INKEY$:IF AN$=""THEN 10
02
1003 ON VAL(AN$) GOSUB 1010,1400
,1450,120
1004 GOTO 1000
1005 '  SEE ONE NAME
1010 GOSUB 3825
1020 CLS8:SOUND100,1:PRINT@96,"E
NTER NAME- (LAST,FIRST)        "
;
1025 LINE INPUT NM$
1030 N1$=NM$
1035 IF LEN(NM$)(22 THEN 1350
1040 IF LEN(NM$))22 THEN 1355
1045 FIRST=1
1050 MID=INT((LOF(1)+1)/2)
1055 LAST=LOF(1)
1060 CNT=0
1065 GET #1,LAST
1070 IF NAMES$=NM$ THEN 1100
1075 GET #1,MID
1080 IF CNT)(LOF(1)+1)/2 THEN 13
00
1085 IF NAMES$(NM$ THEN 1200
1090 IF NAMES$)NM$ THEN 1250
1100 GOSUB 3000
1130 PRINT@420,"PRESS (ENTER) TO
 CONTINUE",,,"    ELSE PRESS (Q
) TO QUIT"
1135 AN$=INKEY$:IF AN$="" THEN
1135
1140 IF AN$="Q" THEN CLOSE #1:RE
TURN
1142 IF AN$=CHR$(13) THEN 1020
1150 GOTO 1130
1200 FIRST=MID
```

```
1210 MID=(MID+LAST)/2
1220 CNT=CNT+1
1230 GOTO 1075
1250 LAST=MID
1260 MID=(MID+FIRST)/2
1270 CNT=CNT+1
1280 GOTO 1075
1300 CLS:SOUND1,10
1310 PRINT@100,N1$:PRINT@138,"NO
T FOUND":PRINT:PRINT"TO TRY AGAI
N PRESS (ENTER)      TO QUIT PRE
SS (Q)"
1320 AN$=INKEY$:IF AN$="" THEN
1320
1322 IF AN$="Q" THEN CLOSE #1:RE
TURN
1340 IF AN$=CHR$(13) THEN 1020
1345 GOTO 1300
1350 NM$=NM$+" ":GOTO 1035
1355 NM$=LEFT$(NM$,22):GOTO 1040
1395 ' SEE ALL FILE
1400 GOSUB 3825
1407 FOR I=1 TO LOF(1)
1408 GET #1,I
1410 GOSUB 3000
1427 PRINT:INPUT"PRESS (ENTER) F
OR NEXT NAME";AN$
1430 NEXT I
1445 CLOSE#1:RETURN
1449 ' LIST FILE TO PRINTER
1450 GOSUB 3825
1457 R=0
1460 CLS:SOUND150,1
1470 INPUT"PRESS (ENTER) WHEN PR
INTER READY";AN$
1475 PRINT#-2,TAB(10);"NAME";TAB
(35);"CALL NO.";TAB(55);"ACCESS.
 NO.";TAB(70);"DUE DATE";CHR$(13
)
1480 R=R+1:GET #1,R
1482 PRINT#-2,TAB(5);NAME$;TAB(3
5);C1$;TAB(55);A1$;TAB(70);D1$
1484 PRINT#-2,TAB(35);C2$;TAB(55
);A2$;TAB(70);D2$;CHR$(13)
1486 IF LOF(1)<>R THEN 1480
1492 CLOSE#1:RETURN
1495 'EDIT FILE
1500 CLS5:SOUND200,1:PRINT@74,"C
HANGE BOOKS";
1505 PRINT@106,"SELECTIONS:-";
1510 PRINT@168,"(1) BORROW BOOKS
";
1515 PRINT@200,"(2) DELETE FILE
";
1520 PRINT@232,"(3) QUIT
";
1525 PRINT@298," 1,2, OR 3 ?";
1530 AN$=INKEY$:IF AN$="" THEN
1530
1535 ON VAL(AN$) GOSUB 1560,1900
,120
1540 GOTO 1500
1555 ' BORROWING ROUTINE
1560 GOSUB 3500
1635 GOSUB 3000
1775 PRINT" CHANGE FIRST BOOK? (
Y/N)";
1780 AN$=INKEY$:IF AN$=""THEN 17
80
1782 IF AN$="Y" THEN 1788
1784 IF AN$="N" THEN 1798
1786 GOTO 1780
1788 CLS:PRINT:PRINT:SOUND100,1
1790 PRINT"PRESS (ENTER) AT EACH
 PROMPT            IF NOT BORROW
ING"
1792 INPUT"CALL NO.  ";X1$
1794 INPUT"ACCESS NO.";Y1$
1796 INPUT"DUE DATE  ";Z1$
1797 GOTO 1800
1798 X1$=C1$:Y1$=A1$:Z1$=D1$
1800 SOUND100,1:PRINT:PRINT"CHAN
GE SECOND BOOK? (Y/N)";
1805 AN$=INKEY$:IF AN$=""THEN 18
05
1810 IF AN$="Y" THEN 1820
1812 IF AN$="N" THEN 1832
1814 GOTO 1805
1820 SOUND100,1:CLS:PRINT:PRINT"
          SECOND BOOK"
1822 PRINT"PRESS (ENTER) AT EACH
 PROMPT       IF NOT BORROWING"
1824 INPUT"CALL NO.   ";X2$
1826 INPUT"ACCESS NO. ";Y2$
1828 INPUT"DUE DATE   ";Z2$
1830 GOTO 1834
1832 X2$=C2$:Y2$=A2$:Z2$=D2$
1834 GOSUB 3200
1835 SOUND150,1:PRINT:PRINT"
 IS THIS CORRECT? (Y/N)"
1840 AN$=INKEY$:IF AN$=""THEN 18
40
1842 IF AN$="Y" THEN 1850
1844 IF AN$="N" THEN 1775
1846 GOTO 1840
1850 GOSUB 910
1855 LSET NAME$=NM$
1860 LSET C1$=X1$
1862 LSET A1$=Y1$
1870 LSET D1$=Z1$
1875 LSET C2$=X2$
1880 LSET A2$=Y2$
1885 LSET D2$=Z2$
1890 PUT #1,MID
1892 CLOSE #1
1895 RETURN
1899 ' DELETE ROUTINE
1900 GOSUB 3825
1910 OPEN "O",#2,"TEMP/FIL",70
1915 FIELD #2,22 AS TNAME$,12 AS
T1$,4 AS T2$,8 AS T3$,12 AS T4$,
4 AS T5$,8 AS T6$
1917 J=0
1920 FOR I=1 TO LOF(1)
1925 GET #1,I
1930 GOSUB 3000
1935 PRINT@420,"DELETE THIS RECO
RD? (Y/N)"
1940 AN$=INKEY$:IF AN$="" THEN 1
940
1945 IF AN$="Y" THEN 1980
1950 IF AN$="N" THEN 1960
1955 GOTO 1940
1960 LSET TNAME$=NAME$:LSET T1$=
C1$:LSET T2$=A1$:LSET T3$=D1$:LS
ET T4$=C2$:LSET T5$=A2$:LSET T6$
=D2$
1965 J=J+1
1970 PUT #2,J
1980 NEXT I
1985 CLOSE
1987 KILL "NAMES/BKS"
1988 RENAME "TEMP/FIL" TO "NAMES
/BKS"
1990 RETURN
1995 'LIST OVERDUE BOOKS-SORT BY
 DATE
2000 CLS0
2010 SOUND50,1:PRINT@72,"OVERDUE
 BOOKS";
2015 PRINT@128,"ENTER TODAY'S DA
TE"
2020 INPUT "DAY (01 TO 31)";DD$
2025 INPUT "MTH.(01 TO 12)";MM$
2028 CLS
2030 PRINT@41,"overdue books":PR
INT:PRINT"   LIST ON sCREEN OR p
RINTER?"
2035 PRINT"       PRESS (Q) TO Q
UIT"
2040 AN$=INKEY$:IF AN$=""THEN 20
40
2045 IF AN$="S" THEN 2100
2050 IF AN$="P" THEN 2300
2055 IF AN$="Q" THEN 110
2060 GOTO 2030
2095 ' SCREEN LIST ROUTINE
2100 GOSUB 3825
2108 OD=0
2110 FOR I=1 TO LOF(1)
2115 GET #1,I:CLS
2116 P$=LEFT$(D1$,2):Q$=MID$(D1$
,4,2)
2117 R$=LEFT$(D2$,2):S$=MID$(D2$
,4,2)
2120 IF VAL(P$)<=0 THEN 2145
2125 IF VAL(Q$)<=0 THEN 2145
```

```
2130 IF VAL(MM$)<VAL(Q$) THEN 21
45
2132 IF VAL(MM$)>VAL(Q$) THEN 21
40
2135 IF VAL(DD$)<VAL(P$) THEN 21
45
2140 GOSUB 2200
2145 IF VAL(R$)<=0 THEN 2275
2150 IF VAL(S$)<=0 THEN 2275
2155 IF VAL(MM$)<VAL(S$) THEN 22
75
2157 IF VAL(MM$)>VAL(S$) THEN 21
65
2160 IF VAL(DD$)<VAL(R$) THEN 22
75
2165 GOSUB 2220
2170 GOTO 2275
2200 OD=OD+1:CLS:PRINT@43,"overd
ue":SOUND1,1:PRINT@100,NAME$:PRI
NT@172,"BOOKS OUT":PRINT@192,"CA
LL NO.":PRINT@204,"ACCESS NO.":P
RINT@218,"DUE"
2210 PRINT@256,C1$:PRINT@270,A1$
:PRINT@278,D1$
2212 PRINT@419,"PRESS <ENTER> TO
 CONTINUE"
2213 AN$=INKEY$:IF AN$=CHR$(13)
THEN 2215
2214 GOTO 2212
2215 RETURN
2220 OD=OD+1:PRINT@43,"overdue":
SOUND1,1:PRINT@100,NAME$:PRINT@1
72,"BOOKS OUT":PRINT@192,"CALL N
O.":PRINT@204,"ACCESS NO.":PRINT
@218,"DUE"
2230 PRINT@320,C2$:PRINT@334,A2$
:PRINT@342,D2$
2232 PRINT@419,"PRESS <ENTER> TO
 CONTINUE"
2233 AN$=INKEY$:IF AN$=CHR$(13)
THEN 2235
2234 GOTO 2232
2235 RETURN
2275 NEXT I
2280 CLOSE #1:CLS:PRINT@360,OD;"
BOOKS OVERDUE":GOTO 2030
2295 ' PRINTER LIST ROUTINE
2300 GOSUB 3825
2315 CLS:SOUND 50,1:PRINT@96,"PR
ESS <ENTER> WHEN PRINTER READY"
2317 INPUT ZZ
2318 PRINT#-2,TAB(25);"**OVERDUE
  BOOKS AT ";DD$;"/";MM$;"**";CH
R$(13)
2319 PRINT#-2,TAB(10);"NAME";TAB
(35);"CALL NO.";TAB(55);"ACCESS.
 NO.";TAB(70);"DUE DATE";CHR$(13
)
2320 FOR I=1 TO LOF(1)
2330 GET #1,I
2340 P$=LEFT$(D1$,2):Q$=MID$(D1$
,4,2)
2350 R$=LEFT$(D2$,2):S$=MID$(D2$
,4,2)
2352 IF VAL(P$)<=0 THEN 2380
2356 IF VAL(Q$)<=0 THEN 2380
2360 IF VAL(MM$)<VAL(Q$) THEN 23
80
2365 IF VAL(MM$)>VAL(Q$) THEN 23
75
2370 IF VAL(DD$)<VAL(P$) THEN 23
80
2375 GOSUB 2430
2380 IF VAL(R$)<=0 THEN 2420
2385 IF VAL(S$)<=0 THEN 2420
2390 IF VAL(MM$)<VAL(S$) THEN 24
20
2395 IF VAL(MM$)>VAL(S$) THEN 24
05
2400 IF VAL(DD$)<VAL(R$) THEN 24
20
2405 GOSUB 2450
2420 NEXT I
2425 CLOSE #1:GOTO 2030
2430 PRINT#-2,TAB(5);NAME$;TAB(3
5);C1$;TAB(55);A1$;TAB(70);D1$
2440 RETURN
2450 PRINT#-2,TAB(5);NAME$;TAB(3
5);C2$;TAB(55);A2$;TAB(70);D2$
2460 RETURN
2495 'END
2500 CLS:SOUND1,10:PRINT:PRINT"T
YPE 'RUN'<ENTER>  TO RESTART ":E
ND
2999 ' SCREEN LIST OF BORROWING
3000 CLS:SOUND100,1
3010 PRINT@100,NAME$:PRINT@172,"
BOOKS OUT":PRINT@192,"CALL NO.":
PRINT@204,"ACCESS NO.":PRINT@218
,"DUE"
3020 PRINT@256,C1$:PRINT@270,A1$
:PRINT@278,D1$:PRINT@320,C2$:PRI
NT@334,A2$:PRINT@342,D2$
3030 RETURN
3200 CLS:PRINT@100,NAME$:PRINT@1
72,"BOOKS OUT":PRINT@192,"CALL N
O.":PRINT@204,"ACCESS NO.":PRINT
@218,"DUE:"
3210 PRINT@256,X1$:PRINT@270,Y1$
:PRINT@278,Z1$:PRINT@320,X2$:PRI
NT@334,Y2$:PRINT@342,Z2$
3220 RETURN
3495 ' BORROW BOOKS ROUTINE
3500 GOSUB 3825
3520 CLS8:SOUND100,1:PRINT@96,"E
NTER NAME- (LAST,FIRST)       "
;
3525 LINE INPUT NM$
3530 N1$=NM$
3535 IF LEN(NM$)<22 THEN 3800
3540 IF LEN(NM$)>22 THEN 3820
3545 FIRST=1:MID=INT((LOF(1)+1)/
2):LAST=LOF(1)
3550 CNT=0
3560 GET #1,LAST
3570 IF NAME$=NM$ THEN MID=LAST:
GOTO 3600
3575 GET #1,MID
3580 IF CNT>(LOF(1)+1)/2 THEN 37
00
3585 IF NAME$<NM$ THEN 3650
3590 IF NAME$>NM$ THEN 3680
3600 GOSUB 3000
3610 PRINT@420,"PRESS <ENTER> TO
 BORROW,",,,"      ELSE PRESS <Q
> TO QUIT"
3620 AN$=INKEY$:IF AN$=""THEN 36
20
3630 IF AN$="Q" THEN CLOSE #1:GO
TO 120
3640 IF AN$=CHR$(13) THEN RETURN
3650 FIRST=MID:MID=(MID+LAST)/2
3660 CNT=CNT+1
3670 GOTO3575
3680 LAST=MID:MID=(MID+FIRST)/2
3690 CNT=CNT+1:GOTO 3575
3700 CLS:SOUND1,10:PRINT@100,N1$
:PRINT@138,"NOT FOUND":PRINT:PRI
NT"TO TRY AGAIN PRESS <ENTER>
   TO QUIT PRESS <Q>"
3710 AN$=INKEY$:IF AN$=""THEN 37
10
3720 IF AN$="Q" THEN CLOSE #1:GO
TO 120
3730 IF AN$=CHR$(13) THEN 3520
3800 NM$=NM$+" ":GOTO 3535
3820 NM$=LEFT$(NM$,22):GOTO 3540
3825 OPEN "D",#1,"NAMES/BKS",70
3830 FIELD #1,22 AS NAME$,12 AS
C1$,4 AS A1$, 8 AS D1$,12 AS C2$
,4 AS A2$,8 AS D2$
3835 RETURN
3838 CLOSE: END
3840 '************************
3850 '*       LIBRARY        *
3860 '*    BY KEN STEWART    *
3870 '*  COPYRIGHT (C) 1985  *
3880 '************************
3890 'FOR DETAILS OR HELP PLEASE
     PHONE 079-953153
     OR WRITE TO THE AUTHOR.
3900 'ADDRESS:-
3910 ' GREYCLIFFE STATE SCHOOL,
     M/S 1396,
     BILOELA.Q.4715.
3920 '
3930 '
```

Scan It And Understand It With Rapid Reading

By T.C. Taulli

Do you want to read Rainbow magazine in half the time? Does it take you an arduous effort to finish a book? Do you want to read War and Peace but just don't have the time? If you answer "yes" to these questions then Rapid Reading is just the program for you.

The average person reads around 225-250 words per minute (WPM). This program can increase your reading rate to 600 wpm. Just put forth a concentrated effort and you'll be reading War and Peace at an incredible pace.

The program uses 16K ECB. After running, the title screen will prompt for the WPM desired. Each sentence of the story will be printed from left to right. This makes it impossible to go back over what you've already read, which is important for high reading speeds.

While using this program try to do a couple of things:

1) Increase your span of recognition by reading in phrases. Reading word for word decreases your reading speed.

2) Eradicate subvocalization by not saying the words in your mind as this is a major factor in being an average reader. Let the words go from the eye to the brain. Don't say them!

The program uses an original story called "One Amazing Program," by Thomas M. Taulli. If you want to use another story or nonfiction piece then follow these directions:

1) Delete the current story by typing DEL 430-.
2) Start at Line 430 and type DATA followed by a sentence with no more than 32 characters.
3) After entering the material, type DATA -1 as your end-of-file marker.

Now it's all up to you.

200	237	1200	53
440	156	1370	203
600	223	1510	154
750	245	1670	250
900	182	1810	230
1060	12	END	207

The listing: RAPOREAD

```
1Ø CLEAR5ØØ:DIMA$(2ØØ)
2Ø CLS3
3Ø FOR X=Ø TO 4:PRINT@X*32,STRIN
G$(32,149);:NEXT
4Ø PRINT@39,"<<RAPID  READING>>"
;
5Ø PRINT@1Ø8,"(C) 1985";
6Ø PRINT@73,"BY T.C. TAULLI";
7Ø PLAY"V25;T5;O2;A;B;G;E;F;E;G;
E;F;G;B"
8Ø PRINT@486,"<<<PRESS ANY KEY>>
>";
9Ø K$=INKEY$:IFK$<>""THEN11Ø
1ØØ GOTO9Ø
11Ø CLS
12Ø CLS:LINEINPUT"HOW MANY WORDS
 PER MINUTE?      (1ØØ-6ØØ WPM)
";W$
13Ø W=VAL(W$):IFW=ØORW>6ØØORW<1Ø
ØTHEN12Ø
14Ø CLS4:PRINT@267,"GET READY";:
GOSUB 32Ø
15Ø J=W:CLS4:GOSUB 35Ø
16Ø IF W=>345 THEN Z=1619Ø2/W-31
.59ELSEZ=164134/W-38.Ø4
17Ø CLS3:FOR X=1 TO 5ØØ:NEXT
18Ø FOR H2=1 TO X
19Ø J=Ø:O=Z:IFLEFT$(W$,1)="6"THE
NI=O/49ØØELSEI=O/32
2ØØ O=INT(I):U=O
21Ø PRINT@224,"":PRINT@288,""
22Ø PRINT@256,A$(H2)
23Ø FOR Y=1 TO Z STEP H
24Ø IFY>U THEN U=U+O:J=J+1:PRINT
@255+J," ";
25Ø NEXT Y
26Ø NEXT H2

27Ø CLS4:PRINT"SESSION OVER WITH
.":PRINT"WOULD YOU LIKE TO RESTA
RT THE   PROGRAM?(Y/N)"
28Ø K$=INKEY$:IFK$="Y"THENRUN
29Ø IFK$="N"THEN31Ø
3ØØ GOTO28Ø
31Ø CLS:PRINT"BYE!!!  BYE!!!":EN
D
32Ø X=1
33Ø READA$(X):IFA$(X)="-1"THEN X
=X-1:RETURN
34Ø X=X+1:GOTO33Ø
35Ø IFLEFT$(W$,1)="1"THENH=3.1
36Ø IFLEFT$(W$,1)="2"THENH=3.15
37Ø IFLEFT$(W$,1)="3"THENH=4
38Ø IFLEFT$(W$,1)="4"THENH=4.85
39Ø IFLEFT$(W$,1)="5"THENH=6.5
4ØØ IFLEFT$(W$,1)="6"THENH=8.6
41Ø RETURN
42Ø DATA"IT WASN'T A VERY BIG AD
VER-"
43Ø DATA"TISEMENT, ONLY A COUPLE
 OF"
44Ø DATA"INCHES IN LENGTH. BUT I
T CAUGHT"
45Ø DATA"MY EYE. IT SAID,'EXPAND
 YOUR"
46Ø DATA"THINKING. AMAZING PROGR
AM FOR"
47Ø DATA"YOUR HOME COMPUTER. ONL
Y $19.95"
48Ø DATA"PPD.'"
49Ø DATA"   NOW, I AM JUST AN AV
ERAGE"
5ØØ DATA"GUY. MADE IT THROUGH HI
GH SCHOOL"
51Ø DATA"WITHOUT TOO MUCH TROUBL
E. I TOOK"
52Ø DATA"A FEW CLASSES AT THE LO
CAL"
53Ø DATA"COMMUNITY COLLEGE BEFOR
E I GOT"
54Ø DATA"TIRED OF NIGHT SCHOOL.
I DO PLAN"
55Ø DATA"TO GO BACK SOMEDAY TO G
ET MY"

56Ø DATA"DEGREE. REALLY!"
57Ø DATA"   LAST CHRISTMAS, MY P
ARENTS"
58Ø DATA"BOUGHT ME AN INEXPENSIV
E"
59Ø DATA"COMPUTER AT THE LOCAL E
LECTRON-"
6ØØ DATA"ICS STORE. YOU KNOW THE
 ONE."
61Ø DATA"THEY'RE THE ONES THAT S
END OUT"
62Ø DATA"AD MAILERS ALL THE TIME
. WELL,"
63Ø DATA"THE COMPUTER WAS FUN FO
```

```
R AWHILE"
64Ø DATA"IN THAT I GOT UP OVER A
 MILLION"
65Ø DATA"IN A PAC GAME. BUT HOW
MANY"
66Ø DATA"POWER PILLS CAN YOU EAT
 BEFORE"
67Ø DATA"YOU GET TERMINAL INDIGE
STION?"
68Ø DATA"THAT'S WHY THE AD FOR T
HE"
69Ø DATA"PROGRAM LOOKED GOOD. I
ALSO MADE"
7ØØ DATA"SURE MY PARENTS KNEW I
WAS SEND-"
71Ø DATA"ING FOR A NEW PROGRAM.
NEVER"
72Ø DATA"HURTS TO LET THEM KNOW
I'M USING"
73Ø DATA"THE THINGS THEY BUY ME.
"
74Ø DATA"   WHEN THE PACKAGE FIN
ALLY CAME"
75Ø DATA"IN THE MAIL, I THREW IT
 INTO MY"
76Ø DATA"DESK AS MY FRIEND, JEFF
, AND I"
77Ø DATA"WERE GOING BOWLING. ABO
UT A WEEK"
78Ø DATA"LATER, I WAS LOOKING FO
R A"
79Ø DATA"PENCIL AND I SAW THE PA
CKAGE."
8ØØ DATA"SINCE NOTHING WAS ON TV
, I TOOK"
81Ø DATA"OUT THE CASSETTE AND PU
T IT INTO"
82Ø DATA"THE CASSETTE RECORDER O
F MY COM-"
83Ø DATA"PUTER. I TURNED IT ON A
ND LOADED"
84Ø DATA"IT IN. THE ONE-PAGE INS
TRUCTIONS"
85Ø DATA"SAID TO USE EARPHONES W
HEN USING"
86Ø DATA"THE PROGRAM. I PLUGGED
MINE INTO"
87Ø DATA"THE TV MONITOR."
88Ø DATA"   THE INSTRUCTIONS SAI
D THAT"
89Ø DATA"SINCE PEOPLE ARE ALWAYS
 THINKING"
9ØØ DATA"AND THAT OUR BRAINS ARE
 LIKE"
91Ø DATA"COMPUTERS, THE BEST WAY
 TO THINK"
92Ø DATA"BETTER IS TO USE YOUR C
OMPUTER"
93Ø DATA"AS A BRAIN HELPER. A BI
NARY"
94Ø DATA"BRAIN. THE COMPUTER COU
LD HELP"
95Ø DATA"YOU THINK ON TWO TO EIG
HT"
96Ø DATA"DIFFERENT TOPICS AT A T
IME. WHEN"
97Ø DATA"I READ THAT, I JUST KNE
W THAT I"
98Ø DATA"HAD THROWN AWAY $19.95"
99Ø DATA"   I REACHED TO TURN OF
F THE"
1ØØØ DATA"COMPUTER, AND I KNEW I
 WAS IN"
1Ø1Ø DATA"TROUBLE."
1Ø2Ø DATA"   MY HAND WOULDN'T MO
VE!
1Ø3Ø DATA"   I TRIED AGAIN. MY H
AND BEGAN"
1Ø4Ø DATA"TO MOVE SLUGGISHLY OFF
 THE KEY-"
1Ø5Ø DATA"BOARD. I WATCHED WITH
A GROWING"
1Ø6Ø DATA"SENSE OF PANIC AS MY H
AND MOVED"
1Ø7Ø DATA"IN SLOW MOTION TO THE
REAR OF"
1Ø8Ø DATA"THE COMPUTER. FINALLY,
 MY INDEX"
1Ø9Ø DATA"FINGER REACHED THE ON/
OFF BUTTON"
11ØØ DATA"AND PRESSED DOWN. I EX
PLODED OFF"
111Ø DATA"MY CHAIR, THREW OFF TH
E HEAD-"
112Ø DATA"PHONES, AND I RACED OU
T THE BACK"
113Ø DATA"DOOR. I WAS OUT OF BRE
ATH WHEN"
114Ø DATA"I SCREECHED TO A HALT
IN FRONT"
115Ø DATA"OF THE BACKFENCE."
116Ø DATA"   I GASPED FOR BREATH
 FROM MY"
117Ø DATA"RUN. I JUST COULDN'T B
ELIEVE IT!"
118Ø DATA"WHAT HAD HAPPENED? I L
OOKED AT"
119Ø DATA"THE YARD. NOTHING WAS
OUT OF THE"
12ØØ DATA"ORDINARY. I COULD HEAR
 JEFF'S"
121Ø DATA"DOG BARKING DOWN THE B
LOCK. THE"
122Ø DATA"GRASS STILL NEEDED MOW
ING."
123Ø DATA"   I KNEW I HAD TO GO
BACK TO MY"
124Ø DATA"ROOM AND TO THE COMPUT
ER IF I"
125Ø DATA"WANTED ANSWERS. THE ON
LY PROBLEM"
126Ø DATA"WAS THAT I DIDN'T WANT
 TO GO"
127Ø DATA"BACK. I STOOD THERE FO
R TEN"
128Ø DATA"MINUTES DEBATING WITH
MYSELF."
129Ø DATA"I WENT BACK RELUNCTANT
LY."
13ØØ DATA"   MY ROOM HADN'T CHAN
GED. I"
131Ø DATA"PICKED UP THE ONE-PAGE
 INSTRUC-"
132Ø DATA"TION AND THIS TIME REA
D IT CARE-"
133Ø DATA"FULLY. IT SAID THE FIR
ST PART OF"
134Ø DATA"THE PROGRAM CONTAINED
A SELF-"
135Ø DATA"HYPNOSIS SECTION. 'THI
S ALLOWS"
136Ø DATA"THE BODY TO BE VERY RE
LAXED AND"
137Ø DATA"BRINGS TO THE MIND AN
ALTERED"
138Ø DATA"SENSE OF TIME WHICH SP
EEDS UP"
139Ø DATA"THINKING. IT ONLY SEEM
S THAT"
14ØØ DATA"EVERYTHING AROUND YOU
MOVES IN"
141Ø DATA"SLOW MOTION.' I SHUDDE
RED AS I"
142Ø DATA"REMEMBERED MY HAND MOV
ING EVER"
143Ø DATA"SO SLOWLY TO TURN OFF
THE COM-"
144Ø DATA"PUTER."
145Ø DATA"   THE INSTRUCTIONS AL
SO SAID TO"
146Ø DATA"VISUALIZE IN YOUR MIND
 A CHANNEL"
147Ø DATA"SELECTOR. LIKE ONE ON
A TV SET."
148Ø DATA"CHANNELS 1 THROUGH 9.
CHANNEL 1"
149Ø DATA"WAS FOR REGULAR RECEPT
ION. JUST"
15ØØ DATA"TURN THE SELECTOR IN Y
OUR MIND"
151Ø DATA"TO THIS CHANNEL FOR NO
RMAL TIME."
152Ø DATA"THE OTHER CHANNELS WER
E FOR"
153Ø DATA"ANYTHING YOU WANTED TO
 THINK"
154Ø DATA"ABOUT. FOR EXAMPLE, CH
ANNEL 2"
155Ø DATA"COULD BE USED TO THINK
 ABOUT"
156Ø DATA"LAST NIGHT'S BALL GAME
. CHANNEL"
157Ø DATA"3 FOR STUDYING HISTORY
 MATERIAL"
158Ø DATA"FOR YOUR NEXT TEST. CH
ANNEL 4"
159Ø DATA"COULD BE ABOUT MAKING
PLANS FOR"
16ØØ DATA"NEXT WEEKEND. AND SO O
N."
161Ø DATA"   WHEN YOU CHANGED CH
ANNELS,"
162Ø DATA"YOU BECOME IMMEDIATELY
 AWARE OF"
163Ø DATA"WHAT YOU HAD BEEN THIN
KING ABOUT"
164Ø DATA"ON THAT CHANNEL. THE C
OMPUTER"
165Ø DATA"PROGRAM HELPED THE BRA
IN KEEP"
166Ø DATA"TRACK OF THE DIFFERENT
 TOPICS."
167Ø DATA"ONE HOUR UNDER ALTERED
 TIME WAS"
168Ø DATA"EQUAL TO TWO OR THREE
HOURS OF"
169Ø DATA"NORMAL TIME. AND THAT'
S FOR EACH"
17ØØ DATA"CHANNEL!"
171Ø DATA"   NOW, I AM NO GENIUS
, BUT EVEN"
172Ø DATA"I COULD SEE WHAT I COU
LD DO WITH"
173Ø DATA"THIS PROGRAM. THE FIRS
T THING I"
174Ø DATA"DID WAS..."
175Ø DATA"   'MR. WILSON? IT'S 1
ØAM. I"
176Ø DATA"WILL HOLD YOUR CALLS A
S USUAL"
177Ø DATA"FOR THE NEXT TWO HOURS
. IS THERE"
178Ø DATA"ANYTHING YOU WANT ME T
O DO"
179Ø DATA"BESIDES THE WALTER'S A
CCOUNT?'"
18ØØ DATA"WILSON PUT DOWN THE FI
NANCIAL"
181Ø DATA"REPORTS AND PUSHED A K
EY ON THE"
182Ø DATA"COMMANDER. 'YES. PLEAS
E CALL A"
183Ø DATA"2PM MEETING WITH THE S
ECTIONS'"
184Ø DATA"HEADS. IT WILL BE A ST
RATEGY"
185Ø DATA"MEETING ON THE TAKEOVE
R BID."
186Ø DATA"THANKS.'"
187Ø DATA"   WILSON LEANED BACK
INTO THE"
188Ø DATA"PLUSH EXECUTIVE CHAIR.
 HE LOOKED"
189Ø DATA"OUT THROUGH THE FLOOR-
TO-CEILING"
19ØØ DATA"WINDOWS THAT MADE UP T
HE ENTIRE"
191Ø DATA"NORTH WALL OF HIS LARG
E OFFICE."
192Ø DATA"THE SAN GABRIEL MOUNTA
INS HAD"
193Ø DATA"SNOW ON THEM. THEY LOO
KED CLOSE"
194Ø DATA"BY IN THE CRISP JANUAR
Y AIR."
195Ø DATA"   HE REACHED OVER TO
THE"
196Ø DATA"COMMAND MODULE ON HIS
DESK AND"
197Ø DATA"PRESSED A BUTTON. A DR
AWER SLID"
198Ø DATA"OPEN AND HE PICKED OUT
 AN"
199Ø DATA"EXPENSIVE HEADSET. AS
HE PUT IT"
2ØØØ DATA"ON, HE WHISPERED,'ONE
AMAZING"
2Ø1Ø DATA"PROGRAM.'"
2Ø2Ø DATA-1
```

SCHOOL IS IN THE HEART OF A CHILD

4K

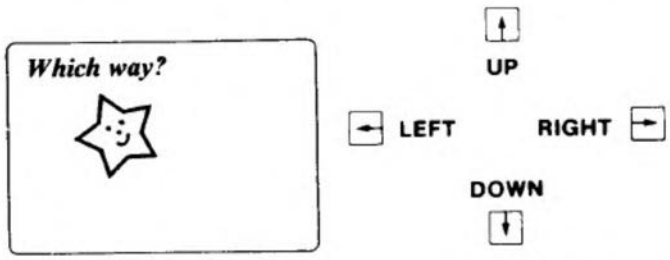
the RAINBOW

Help Wandering Star In The Right Direction

By Bob Albrecht and Ramon Zamora

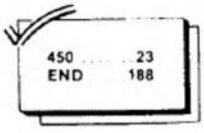
Would you like to help Wandering Star move towards cosmic dust? Well, you can. Use the arrow keys.

— Press → to tell her to move right.
— Press ← to tell her to move left.
— Press ↑ to tell her to move up.
— Press ↓ to tell her to move down.
— Press any other key and she will move randomly.



Wandering Star waits for you to press a key. Press any key and she will move. Press an arrow key to tell her which way to go.

Sometimes she will understand you, sometimes she won't. You see, there is cosmic noise in the galaxy, so sometimes she can't hear you because of the noise. (Parent, you can adjust the amount of cosmic noise from zero percent to 100 percent.)



Listing 1: OASIS

```
100 REM**WANDERING STAR SCH 15-1
110 CLS
199 '
200 REM**GOURMET OASIS
210 MENU$ = "........,,@"
220 FOR K=1 TO 100
230 : GD = RND(10)
240 : GD$ = MID$(MENU$,GD,1)
250 : PRINT @RND(510), GD$;
260 NEXT K
299 '
300 REM**WANDERING STAR APPEARS
310 WS$ = "*"
320 R = 7: RN = R
330 C = 16: CN = C
340 SP = 32*RN + CN
350 PRINT @SP, WS$;
360 FOR ZZ=1 TO 1000: NEXT ZZ
399 '
400 REM**SHE MIGHT GET YOUR HELP
410 PCN = 50  '% COSMIC NOISE
420 IF RND(100)-1<PCN THEN 510
430 K$=INKEY$:IF K$="" THEN 430
440 KC = ASC(K$)
450 IF KC=9  THEN CN=C+1:GOTO610
460 IF KC=8  THEN CN=C-1:GOTO610
470 IF KC=10 THEN RN=R+1:GOTO610
480 IF KC=94 THEN RN=R-1:GOTO610
499 '
500 REM**TOO MUCH NOISE--MOVE RA
NDOMLY
510 W = RND(4)
520 IF W=1 THEN CN = C + 1
530 IF W=2 THEN CN = C - 1
540 IF W=3 THEN RN = R + 1
550 IF W=4 THEN RN = R - 1
599 '
600 REM**KEEP HER IN THE OASIS
610 IF CN<0  THEN CN = 0
620 IF CN>31 THEN CN = 31
630 IF RN<0  THEN RN = 0
640 IF RN>15 THEN RN = 0
650 IF 32*RN+CN=511 THEN 510
699 '
700 REM**WANDERING STAR WANDERS
710 PRINT @SP, " ";
720 R=RN: C=CN: SP=32*R+C
730 PRINT @SP, WS$;
740 SOUND RND(255), 1
750 FOR ZZ=1 TO 20: NEXT ZZ
799 '
800 REM**GOTO DIRECTION SELECTOR
810 GOTO 410
```

Line 410 determines the amount of cosmic noise. We have set it to 50 percent. Therefore, about 50 percent of the time Line 420 will send CoCo to block 500 and Wandering Star will move randomly.

— For entirely random movement, set PCN=100. She will ignore key presses and move entirely at random.
— For complete keyboard control, set PCN=0. Now Wandering Star will move *only* when someone presses a key. If you press an arrow key, she will go in the direction of the arrow. If you press a key other than an arrow key, she will move randomly.

Try various settings between PCN=0 and PCN=100 until you find the one just right for your child or yourself.

Perhaps your child would like to see Wandering Star in the Alphabet Soup Oasis. Easy! Just change Line 210 as follows:

```
210 MENU$ = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

How would you change block 200 so the oasis has brightly colored gourmet shapes? (Hint: Put colored graphics characters in random places on the screen.)

More Interesting Patterns

We have shown you programs to do arithmetic and geometric sequences. Now we will look at ways to generate patterns such as the following:

1) 11, 111, 1111, 11111 and so on.
2) 99, 999, 9999, 99999 and so on.
3) 32, 332, 3332, 33332 and so on.
4) 34, 334, 3334, 33334 and so on.

Interesting things happen when we compute the square of each number in one of the above patterns.

In the meantime, however, how do we get the CoCo to generate the patterns? In particular, how do we write a program to generate any pattern of this type, using as few "get started" numbers as possible?

For pattern numbers 1 and 2, it's easy.

Pattern 1

First number:	11
Second number:	111 = 10*11 + 1
Third number:	1111 = 10*111 + 1

And so on.

Aha! Next number = 10*Previous number + 1
Or, in BASIC: S = 10*S + 1

Pattern 2

First number: 99
Second number: 999 = 10*99 + 9
Third number: 9999 = 10*999 + 9
And so on.

Next number = 10*Previous number + 9
In BASIC: S = 10*S + 9

So, from the evidence in working with patterns 1 and 2, it looks as if we need two numbers to define a pattern of this type. We will call them 'S' and 'B.'

S = Starting number
B = Number to add on after multiplying the old value of 'S' by 10.

Then, the next value of 'S' is computed like this.

S = 10*S + B

But, alas, it doesn't work for pattern numbers 3 and 4. Oh well, back to the old drawing board.

Here is a way to get Pattern 3:

First number: 32
Second number: 332 = 10*(32 + 1) + 2
Third number: 3332 = 10*(332 + 1) + 2
And so on.

Next number: S = 10*(S + 1) + 2

Will it work for Pattern 4? Almost. We have to make a slight change.

First number: 34
Second number: 334 = 10*(34 - 1) + 4
Third number: 3334 = 10*(334 - 1) + 4
And so on.

Next number: S = 10*(S - 1) + 4

Insight! Hang on while we take the big jump. (Don't be afraid to experiment; it's fun!)

Pattern 1 11, 111, 1111, etc.
S = 10*(S + 0) + 1
Pattern 2 99, 999, 9999, etc.
S = 10*(S + 0) + 9
Pattern 3 32, 332, 3332, etc.
S = 10*(S + 1) + 2
Pattern 4 34, 334, 3334, etc.
S = 10*(S + (-1) + 4

Now we have it. To get the next number, do these things:

1) Add something to the previous number. This something *might* be a negative number.
2) Multiply the result by 10.
3) Add something to that result.

In BASIC: S = 10*(S + A) + B

The following table shows the values of 'S,' 'A' and 'B' for our four patterns.

Pattern	S	A	B
1) 11, 111, 1111, ...	11	0	1
2) 99, 999, 9999, ...	99	0	9
3) 32, 332, 3332, ...	32	1	2
4) 34, 334, 3334, ...	34	-1	4

Your turn! Show the values of 'S,' 'A' and 'B' for each of the following patterns.

Pattern	S	A	B
5) 43, 433, 4333, ...			
6) 98, 998, 9998, ...			
7) 37, 337, 3337, ...			

Here is a program to generate patterns defined by 'S,' 'A' and 'B.' The DATA statements contain values of 'S,' 'A' and 'B' for patterns 1 through 4.

Listing 2: PATTERNS

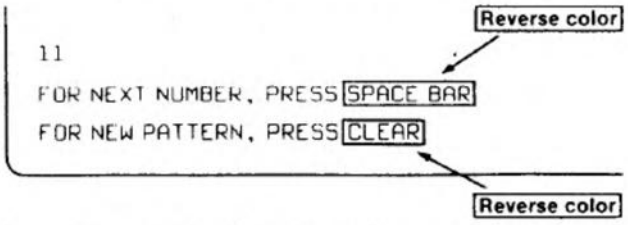
```
1ØØ REM**NMBR PATTERNS SCH 15-2
11Ø CLS
12Ø PRINT "TRY SOME NUMBER PATTE
RNS.": PRINT
13Ø PRINT "PRESS THE spacebar TO
 BEGIN."
14Ø IF INKEY$="" THEN 14Ø
199 '
2ØØ REM**READ STARTING NUMBERS
21Ø CLS
22Ø READ S, A, B
23Ø IF S = 1E37 THEN PRINT "I'M
OUT OF PATTERNS" : END
299 '
3ØØ REM**SHOW 'LATEST' NUMBER
31Ø PRINT @448, S
32Ø PRINT
399 '
4ØØ REM**COMPUTE NEXT NUMBER
41Ø S = 1Ø*(S+A) + B
499 '
5ØØ REM**WHAT TO DO NEXT
51Ø PRINT @48Ø, "FOR NEXT NUMBER
, PRESS spacebar"
52Ø PRINT "FOR NEW PATTERN, PRES
S clear";
53Ø K$=INKEY$:IF K$="" THEN 53Ø
54Ø IF K$=" " THEN 31Ø
55Ø IF K$=CHR$(12) THEN 11Ø
ELSE 53Ø
599 '
9ØØ REM**VALUES OF S, A, B
91Ø DATA 11, Ø, 1
92Ø DATA 99, Ø, 9
93Ø DATA 32, 1, 2
94Ø DATA 34, -1, 4
95Ø DATA 1E37, 1E37, 1E37
```

Try the program using our DATA statements. It begins like this:

```
TRY SOME NUMBER PATTERNS
PRESS THE SPACE BAR TO BEGIN
```

This is in reverse color.

Press the space bar and the first pattern begins.



To continue with this pattern, keep pressing the space bar. To get the next pattern, press the CLEAR key. If all the starting numbers have been used, you will see the following message.

```
I'M OUT OF PATTERNS
OK
```

Add your own DATA statements with values of 'S,' 'A' and 'B.' Put the flags (1E37) in the last (and only the last) DATA statement.

EDUCATION OVERVIEW

Developing Effective Computer Literacy Methods

By Michael Plog, Ph.D.

Nine hundred and ninety hours — a little less than 1,000 hours. That is the amount of time the average elementary student spends in school each year for instructional purposes, minus such things as lunch and recess.

The Association for Supervision and Curriculum Development (ASCD) conducted this survey of 1,500 elementary schools across the country. The study will be replicated at regular intervals in future years in order to establish trend data. Also, similar studies are planned for middle and high schools.

The ASCD study found that the average student spends five and one-half hours per day in school, for a school year of 180 days. Sixty percent of each day is spent on instruction in language arts, mathematics, social studies and science.

It breaks down to about 100 minutes a day for language arts, 52 minutes for mathematics, 34 minutes for social studies and 28 minutes for science. Now, five and one-half hours is 330 minutes. If we add the time spent on the basic four subjects, we get 214 minutes per day. What about the other 116 minutes?

Well, other subjects consume part of that time: health gets an average of 22 minutes per day, physical education gets 15 minutes, music, 14 minutes and art, 13 minutes. There are many other subjects schools handle during the course of a year, including drug education, foreign languages, consumer education, environment and the list goes on. Most of these subjects are dealt with (by most schools) by integrating them into the regular curriculum. For example, global education may be studied in social studies; consumer issues studied in mathematics.

In addition to collecting information from schools for this study, the ASCD also asked experts in subject matter areas to recommend ideal time allocations. As you might predict, the subject matter specialists believe far more time is necessary in their areas of expertise than would be possible to fit in the school day.

Add to this the demand from public segments to include time for topics of great significant social concerns, and it becomes clear that all this takes more time than we can possibly expect elementary students to spend in schools.

These results were reported by principals. The actual time a single student may spend in any particular area, such as language arts, will vary greatly with individual teacher's competence in an area, or even preference for particular subjects.

Above was reported the average times students spend on the different subjects. The *range* of times, however, varied among the schools surveyed. One-fourth of the respondents to the survey said only five hours per week was spent on language arts.

You may have noticed that instruction in computers has not been mentioned so far. The researchers conducting the study did not list computer instruction as one of the topics to be reported by the schools. As a result, we do not have any firm information about the amount of time elementary students spend on computers.

This survey was distributed to elementary schools, where computer instruction may be less popular than with secondary schools. It is possible that much of the computer time is integrated into the rest of the curriculum. That is, students spend time doing a computer lesson about science, or a lesson dealing with history or reading, etc. Of course, many schools do not know what to do with computers, only using them for student games and "toys" for interested instructors. About 20 percent of the principals responding to the ASCD survey indicated their teachers could get help from a computer coordinator or specialists.

The question of what to do with computers is still being hotly debated in schools across the country. Many school administrators appear to be unsure of how to best use computers. As a result, more states are setting guidelines for computer use in school districts. The people at the state level are also unsure of how to best use computers. About two dozen states have set requirements for computer instruction through state legislatures and state boards of education.

Texas and Tennessee both will require schools to teach computer skills to seventh and eighth grade students beginning next school term. Texas mandates a separate computer literacy course in the junior high school; in Tennessee, a three-week unit can be covered in any subject area. However, both states are providing a state curriculum.

Objectives of the computer unit are defined by the state education agency, and all course materials and software

are state approved. In Texas, 2,000 teachers must demonstrate that they are qualified to teach computer courses by passing a test by September. The state education agency offers a five-day training program, and also has materials available through regional service centers.

Colorado, New York and Minnesota are doing things differently. Instead of having one uniform system required of all students, these three states are providing examples, but no specific guidance or direction. In Minnesota, all students must be exposed to computer technology by the time they graduate. The state recommends activities for various age groups, but does not set a curriculum — that is left up to individual schools. In New York, the state is providing $17.5 million to buy software and hardware, but will not mandate a particular curriculum.

In the District of Columbia, all teachers must take a course in computer literacy and software evaluation in order to obtain a teaching certificate. Massachusetts requires competency in educational technology for all teachers, and New Hampshire requires all middle school teachers to take one computer science course. In Montana and Vermont, teachers in certain subject areas must be able to use computers in their classes, but are not required to take a course or pass a test.

Utah is one state in a category by itself. By 1988, computer science instructors will have to show their competency by majoring or minoring in computer science, completing a state approved in-service training program or passing a proficiency test. Those teachers in other areas do not have to do anything. Clearly, Utah does not want to have computers integrated into the entire curriculum, but only wants a computer course in schools.

The idea of requiring teachers to take a test proving competency in computer abilities can be attractive or can be a disaster. An official in Colorado thinks little of computer competency testing: "The state doesn't require an endorsement for someone to use a 16 millimeter projector, and that's a teaching tool. The computer is a teaching tool, too. I don't see any difference between the two."

It is a positive feature to have independent confirmation of a teacher's ability in a subject matter area. For public education, the community has a right to know teachers possess the proper skills related to their work.

The normal certification process provides this to some degree. Teachers of mathematics, for example, have to be graduated from a recognized center of higher learning and must have taken certain mathematics courses while at a college or university. In the field of computers, however, there is no subject matter specialty in many colleges.

Besides, what area of computer knowledge should be tested? I happen to enjoy programming my Color Computer in BASIC. I know other people who never program their own computer. They purchase commercial programs and just love to tear off the cover of the Color Computer to play with the insides. Do these differences of preference make either of us less literate? What about the person who is interested in the history of computing? Should that be part of computer literacy?

The problem is that computer literacy includes all aspects of computers — and none of them! You do not have to program a computer in order to be computer literate. You do not have to recognize a resistor to be computer literate, nor do you have to know when Tandy introduced the Color Computer to be literate.

It seems the central point of computer literacy is possession of the skills and abilities to make a computer work for you. There are many ways teachers can help students become computer literate. Knowing how to program in BASIC may be less helpful for students than how to operate a word processor program and a database program. Knowing how to replace a memory chip may be less important to students than knowledge of a spreadsheet package.

Thus, the state education people in Texas, Tennessee and Utah could be taking the wrong approach. A state curriculum is less helpful for computer literacy than a locally developed curriculum. The best curriculum of all is an understanding of the needs of a student, and then designing a course of study to fit those needs.

Notice that "student" is singular — teachers should work with individual students on individual needs and educational experiences. I agree with a Minnesota state employee: "The computer should be a part of everyday life. Kids should not have to go to special people to learn about it."

We all need to learn; each of us has needs not necessarily shared by others. Write if you wish to take issue with me. My address is 829 Evergreen, Chatham, IL 62629.

EDUCATION NOTES

16K ECB

The Rainy Day Account

By Steve Blyn

You are never too young to learn the value of saving your money. One never knows when a rainy day emergency may come along. Similarly, one never knows when that big ticket item that you have been wanting for so long finally goes on sale.

Many school systems throughout the country have banking programs for elementary school age children. This helps to reinforce the importance of saving.

Parents and grandparents often open accounts for youngsters. This may represent a substantial amount of money. The child should not necessarily be aware of this account. We can, however, easily open a small account with the child for the purposes of instruction as well as savings. This account can help instill positive values and experiences into the child's environment.

A larger than life approach is often quite successful with younger children.

(Steve Blyn teaches both exceptional and gifted children, holds two master's degrees and has won awards for the design of programs to aid the handicapped. He and his wife, Cheryl, own Computer Island.)

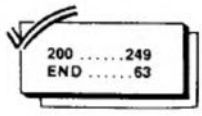
This implies making a big deal out of what is often commonplace to us. Visiting a bank and signing your name to an application for an account is a big deal to many youngsters. (Signing for your first library card is in the category of important events.)

This month's program illustrates a simple bank account book that shows deposits and withdrawals. The child's task is to compute his balance after each transaction.

The child should first be familiarized by an adult with the meaning of the key words. They are withdrawal, deposit and balance. These words may be likened to positive and negative numbers if this is appropriate for your children. A deposit of $5.00 is similar to +5 and a withdrawal of $3.00 is similar to -3.

The balance would be +2, which is the sum of these two signed numbers. An introduction to signed numbers can be approached in this manner. Some children find this approach easy to understand; others would be better served by using play money to figure out each transaction in a tangible manner.

The reasons we use a bank for our savings should also be discussed. Mention of the safety factor and the interest that may be earned can be explained by the adult. Children will not initially believe that withdrawals can be made at any time. The fact that the money is still his even though it is in the bank should be stressed.



The listing: BANKACCT

```
10 REM"BANK ACCOUNTS"
20 REM"STEVE BLYN,COMPUTER ISLAN
D,NY,1985
30 CLS:PRINT"WHAT IS YOUR NAME";
40 INPUT NA$
50 N$=LEFT$(NA$,10)
60 FOR N=1 TO 10
70 B=50
80 READ A$(N): NEXT N
90 CLS
100 PRINT@0,N$;"'S BANK ACCOUNT"
110 PRINT STRING$(32,255);
120 PRINT"DATE  WITHDRAWAL DEPOS
IT BALANCE" ;
130 PRINT"----- ---------- -----
-- -------";
140 PRINT"01/01     0          50
    50"
150 FORR=1TO10:PRINT@160+T,A$(R)
160 T=T+32
170 NEXT R
180 DATA 01/25,02/10,03/06,04/14
,05/22,07/15,09/30,10/19,11/08,1
2/17
190 FOR T=1 TO 10
200 D=RND(20)
210 R=RND(4)
220 W=0
230 IF R=4 THEN W=RND(20):D=0
240 PRINT@168+L,W;
250 PRINT@178+L,D;
260 PRINT@185+L,"";
270 INPUT G
280 B=B+D-W
290 IF G=B THEN PRINT@492,"CORRE
CT   ";:PRINT@185+L," ";:SOUND14
0,3:SOUND150,3:CT=CT+1
300 IF G<>B THEN PRINT@492,"SORR
Y";B;:PRINT@186+L,B:SOUND10,5
310 L=L+32
320 PRINT@26,"*=";CT;
330 NEXT T
340 SOUND200,5:PRINT@483,"PRESS
ENTER TO GO AGAIN.";
350 EN$=INKEY$
360 IF EN$=CHR$(13) THEN RUN ELS
E 350
```

SCHOOL IS IN THE HEART OF A CHILD

16K

Play And Learn Together— Wonderment Is Contagious!

By Bob Albrecht and Ramon Zamora

Since you haven't said you want more Wandering Star, this time we will give you an assortment of programs from *TRS-80 Color BASIC* by Bob Albrecht.

We begin with *Intergalactic Broadcasting*. We suggest you type it in using *your* name. Then, encourage your child to change it to her or his name.

```
100 CLS
105 '
110 PRINT "INTERGALACTIC"
120 SOUND 89, 20
125 '
130 PRINT "BROADCASTING"
140 SOUND 125, 20
145 '
150 PRINT "COMPANY"
160 SOUND 147, 20
165 '
170 PRINT "BRINGS TO YOU..."
180 SOUND 176, 40
185 '
200 CLS
210 PRINT "LUCY ";
220 SOUND 89, 1
230 GOTO 210
```

Of course, you may want to put *your* name in Line 210.

You see, the way to make a computer wonderful to a child is to play with the child on the computer. Play together, learn together. Wonderment is contagious!

We think your small child might like this one. Encourage him or her to type in his or her name.

```
100 REM**GRAND FINALE SCH 16-2
110 CLS
199 '
200 REM**FOR WHOM?
210 INPUT "YOUR NAME"; N$
299 '
300 REM**CRESCENDO FOR N$
310 CLS
320 FOR T=1 TO 255
330 : PRINT N$;
340 : SOUND T, 1
350 NEXT T
399 '
400 REM**LONG TIME DELAY
410 FOR K=1 TO 2500: NEXT K
499 '
500 REM**DO IT AGAIN
510 GOTO 110
```

Almost every kid we know is on a team or has a friend on a team. So we suggest this program called *Go, Team, Go!*

```
1ØØ REM**GO TEAM GO! SCH 16-3
199 '
2ØØ REM**'GO' ON A BLUE SCREEN
21Ø CLS 3: PRINT "GO";
22Ø GOSUB 91Ø <-- Use time delay subroutine
299 '
3ØØ REM**'TEAM' ON ORANGE SCREEN
31Ø CLS 8: PRINT "TEAM";
32Ø GOSUB 91Ø <-- Use time delay subroutine
399 '
4ØØ REM**'GO!' ON MAGENTA SCREEN
41Ø CLS 7: PRINT "GO!";
42Ø GOSUB 91Ø <-- Use time delay subroutine
43Ø GOSUB 91Ø     twice for longer delay
499 '
5ØØ REM**KEEP IT GOING
51Ø GOTO 21Ø
599 '
6ØØ REM**TIME DELAY SUBROUTINE
91Ø FOR K=1 TO 5ØØ: NEXT K
92Ø RETURN
```

Adults love to kid adults. Kids love to kid adults. Adults love to kid kids. And so on. Imagine this: You are the last one to go to bed tonight. Before you go, put a message on the screen for the early risers.

```
1ØØ REM**MESSAGE BLINKER SCH 16-
4
199 '
2ØØ REM**GET MESSAGE & PLACE
21Ø CLS
22Ø INPUT "YOUR MESSAGE"; M$
23Ø INPUT "WHERE SHALL I BLINK I
T"; P
299 '
3ØØ REM**BLINK MESSAGE ON
31Ø CLS: PRINT @P, M$;
32Ø Z = 5ØØ
33Ø GOSUB 91Ø
399 '
4ØØ REM**BLINK MESSAGE OFF
41Ø CLS 2
42Ø Z = 3ØØ
43Ø GOSUB 91Ø
499 '
5ØØ REM**DO IT AGAIN
51Ø GOTO 31Ø
599 '
9ØØ REM**TIME DELAY SUBROUTINE
91Ø FOR K=1 TO Z: NEXT K
92Ø RETURN
```

Can you figure out how to use the following program to paint many (or few) colored stripes on the screen?

```
1ØØ REM**STRIPE 'PAINTBRUSH' SCH
 16-5
11Ø CLS Ø
199 '
2ØØ REM**DIALOG WITH PAINTER
21Ø PRINT @Ø, CHR$(3Ø): PRINT @Ø
,;
22Ø INPUT "DOWN,L,R,CLR"; DOWN,
L, R, CLR
299 '
3ØØ REM**PAINT HORIZONTAL STRIPE
31Ø FOR OVER=L TO R
32Ø : SET(OVER, DOWN, CLR)
33Ø NEXT OVER
399 '
4ØØ REM**DONE. SOUND OFF.
41Ø SOUND 89, 1Ø
499 '
5ØØ REM**GO BACK FOR MORE
51Ø GOTO 21Ø
```

The real wonderment of computers is to make them do what you want them to do. Everyone who reads this magazine can learn to read and understand CoCo BASIC programs, if only the people who write for the magazine have compassion for you, the beginner. If you learn to read and understand BASIC programs written by others, you will soon learn to express yourself in the language built in to every home computer.

Now try to read and understand this program. Replace the DATA statements with locations of your stars.

```
1ØØ REM**CONSTELLATION SCH 16-6
11Ø CLS Ø
199 '
2ØØ REM**NS IS NUMBER OF STARS
21Ø READ NS
299 '
3ØØ REM**TURN ON NS STARS
31Ø FOR STAR=1 TO NS
32Ø : READ OVER, DOWN
33Ø : SET(OVER, DOWN, 8)
34Ø NEXT STAR
399 '
4ØØ REM**DO NOTHING LOOP
41Ø GOTO 41Ø
499 '
5ØØ REM**STAR DATA
91Ø DATA 7
92Ø DATA  6, 12, 18, 1Ø
93Ø DATA 26, 12, 34, 14
94Ø DATA 38, 2Ø, 54, 2Ø
95Ø DATA 56, 14
```

Values of OVER and DOWN for seven stars

A mandala is a symmetric pattern; nice to look at. A giant snowflake is beautifully symmetric about its center. Snowflakes are great mandalas but melt too soon. Use this program to put an ever changing mandala on the screen.

```
1ØØ REM**MANDALA, EVER CHANGING
SCH 16-7
11Ø CLS Ø
199 '
2ØØ REM**HORIZONTAL & VERTICAL O
FFSET
21Ø H = RND(32) - 1
22Ø V = RND(16) - 1
299 '
3ØØ REM**RANDOM COLOR
31Ø KOLOR = RND(8)
399 '
4ØØ REM**TURN ON FOUR BLIPS
41Ø SET(31 - H, 15 - V, KOLOR)
42Ø SET(31 - H, 16 + V, KOLOR)
43Ø SET(32 + H, 15 - V, KOLOR)
44Ø SET(32 + H, 16 + V, KOLOR)
499 '
5ØØ REM**DELAY, THEN DO MORE
51Ø Z = 1Ø
52Ø FOR K=1 TO Z: NEXT K
53Ø GOTO 21Ø
```

RUN the program. The computer turns on four lights at a time, symmetric with the center of the screen. If you don't see this happen, increase the time delay by changing Line 510 to:

```
510 Z = 500
```

and RUN the program again. If you want the mandala to change more rapidly, delete lines 510 and 520, or change Line 510 to `510 Z = 1`.

Experiment! Try these variations:

Variation 1: Change only Line 210, as follows:
```
210 H = RND(RND(32)) - 1
```
Variation 2: Change only Line 220, as follows:
```
220 V = RND(RND(16)) - 1
```
Variation 3: Change both lines 210 and 220, as follows:
```
210 H = RND(RND(32)) - 1
220 V = RND(RND(16)) - 1
```
Variation 4: Change either Line 210 or Line 220, or both, as follows:
```
210 H = RND(RND(RND(32))) - 1
220 V = RND(RND(RND(16))) - 1
```
Variation 5: Change either Line 210, or Line 220, or both:
```
210 H = 32 - RND(RND(32))
220 V = 16 - RND(RND(16))
```
Variation 6: Change Line 310:
```
310 KOLOR = RND(RND(8))
```
Variation 7: Anything suggested by the above variations.

Experiment! The best variations are your variations.

Here is a simple number guessing game. The number of stars tells you how close you are to the CoCo's secret number. Can you guess the number in seven guesses (every time)?

The listing: STARS

```
1ØØ REM**STARS - A GUESSING GAME
 SCH 16-8
199 '
2ØØ REM**TELL HOW TO PLAY
21Ø CLS
22Ø PRINT "WELCOME TO MY GALAXY.
 I'LL"
23Ø PRINT "THINK OF NUMBER, 1 TO
 1ØØ."
24Ø PRINT "YOU GUESS MY NUMBER.
IF YOU"
25Ø PRINT "MISS, I'LL PRINT SOME
 STARS."
26Ø PRINT "THE CLOSER YOU ARE, T
HE MORE"
27Ø PRINT "STARS YOU WILL SEE."
28Ø PRINT "IF YOU SEE 7 STARS (*
******),"
29Ø PRINT "YOU ARE VERY, VERY CL
OSE!"
299 '
3ØØ REM**COCO 'THINKS' OF A NUMB
ER
31Ø X = RND(1ØØ)
399 '
4ØØ REM**GET GUESS, G
41Ø PRINT @48Ø,;
42Ø INPUT "YOUR GUESS"; G
499 '
5ØØ REM**D IS DISTANCE FROM X
51Ø D = ABS(X - G)
599 '
6ØØ REM**CHECK FOR A WIN
61Ø IF D=Ø  THEN 81Ø
699 '
7ØØ REM**NO WIN. PRINT HINT.
71Ø PRINT @464, "*";
72Ø IF D<64 THEN PRINT " *";
73Ø IF D<32 THEN PRINT " *";
74Ø IF D<16 THEN PRINT " *";
75Ø IF D<8  THEN PRINT " *";
76Ø IF D<4  THEN PRINT " *";
77Ø IF D<2  THEN PRINT " *";
78Ø PRINT: GOTO 41Ø
799 '
8ØØ REM**WINNER!
81Ø CLS
82Ø FOR K=1 TO 1ØØ
83Ø : PRINT @RND(51Ø), "*";
84Ø NEXT K
85Ø PRINT @48Ø, "YOU GOT IT, MY
NUMBER WAS" X
899 '
9ØØ REM**PLAY AGAIN?
91Ø PRINT "TO PLAY AGAIN, PRESS
ANY KEY"
92Ø IF INKEY$="" THEN 92Ø ELSE 2
1Ø
```

Plain and Fancy Patterns, the Easy Way

Our next number pattern program will generate any of the sequences shown in previous episodes, and lots more. Each sequence is defined by four numbers: 'S,' 'M,' 'A' and 'B.' 'S' is the first number in the pattern. To get the next number:

1) Add 'A' to the preceding number
2) Multiply the result of Step 1 by 'M'
3) Add 'B' to the result of Step 2 in BASIC: `S = M*(S + A) + B`

The following table shows the values of 'S,' 'M,' 'A' and 'B' for some of our previous patterns.

Pattern	S	M	A	B
1,2,3,4, ...	1	1	1	0
2,5,8,11, ...	2	1	3	0
1,2,4,8, ...	1	2	0	0
11,111,1111, ...	11	10	0	1
32,332,3332, ...	32	10	1	2
34,334,3334, ...	34	10	-1	4

It's your turn. Show the values of 'S,' 'M,' 'A' and 'B' for each of the following patterns.

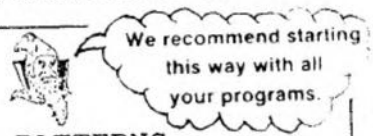
Pattern	S	M	A	B
2,4,6,8, ...				
1,3,5,7, ...				
1,10,100,1000, ...				
3,6,12,24, ...				
1,-2,4,-8, ...				
99,999,9999, ...				
43,433,4333, ...				
98,998,9998, ...				
37,337,3337, ...				
12,102,1002, ...				

Relax for a while. Do something physical. Jog, stretch, dance, play tennis. Then browse through our previous *Number Patterns* programs. Now, refreshed in mind and body, do the next exercise.

Exercise

Write the program *Number Patterns* which generates patterns defined by 'S,' 'M,' 'A' and 'B.' Read their values from `DATA` statements. Write `DATA` statements for the patterns you want to see.

We will help you get started by showing an outline of the program using `REM` statements. All you have to do is write the statements that follow each `REM` statement.




```
1ØØ REM**NUMBER PATTERNS
2ØØ REM**READ STARTING NUMBERS
3ØØ REM**SHOW 'LATEST' NUMBER
4ØØ REM**COMPUTE NEXT NUMBER
5ØØ REM**WHAT TO DO NEXT
9ØØ REM**DATA: VALUES OF S,M,A,B
```

DragonSmoke

Our newsletter, *DragonSmoke*, is growing slowly. We began in January 1985 with two pages copied on our trusty Canon PC copy machine, then grew to eight pages in February, 16 in March, and 20 in April. Here are two ways to sample *DragonSmoke*.

copy machine, then grew to eight pages in February, 16 in March, and 20 in April. Here are two ways to sample *DragonSmoke*.

— Send $1 and we will send you the first four issues, January, February, March and April.

— Or, send $1 and ask for the latest issue. Our address: *DragonSmoke*, P.O. Box 7627, Menlo Park, CA 94026.

DragonSmoke is a beginner's periodical covering computers, role playing games, play-by-mail games and tennis.

EDUCATION OVERVIEW

Integrating Computers Into Classroom Instruction

By Michael Plog, Ph.D.

When educators speak of computer uses in schools, they generally think of computer assisted instruction (CAI) or computer management of instruction (CMI). When used as CMI, the computer may never be touched by students, but used as a recording device by teachers. Some teachers have even written programs to calculate grades and print report cards.

When used as CAI, students usually have direct access to the computer. If students are sitting at a computer keyboard, they are generally learning something through the electronic medium. The computer is transformed into a smart workbook. Student responses are used by the program to determine what problems to present next, or what to assign the student.

In math classes, for example, students are practicing addition facts in front of a screen instead of using paper and pencil. Sometimes the computer lesson presents new facts to students instead of drilling on facts already learned.

There is nothing wrong with this type of computer use in schools. Such learning can be very helpful for students by using the computer to drill or even introduce material, and no one would doubt the benefits of having the computer perform calculations for teachers.

In neither case, however, is the computer fully integrated into a lesson. Most CAI deals with lower order learning skills, such as practice, drill, etc. Some simulations attempt to deal with higher order thinking skills, such as synthesis, analysis of knowledge and testing hypotheses generated by students. This month, I would like to present a different type of classroom computer use, and ask your assistance about such suggestions.

For lack of a better term, we are calling this use of the computer "integrating with the lesson." This means the computer is used in a unit of study in various places to help students with higher order learning skills.

For example, let us consider a social studies lesson. (This example is presented only because I taught social studies, and am not all that familiar with other subject areas.) Intentionally, the lesson is one of the dullest I can think of: a civics or government lesson about local elections. The class is to be divided into several groups, each group attending to a single campaign.

The purpose of the lesson will be to have students learn about local election processes, understand factors that influence local campaigns, and conduct an analysis of election victories and losses. The class will not be spending full time every day on the lesson, but will be dealing with other activities associated with local democratic processes. This class activity will take a few months to be completed.

The role of the teacher in this lesson is to coordinate and guide student activities. The teacher will do very little lecturing, but spend most of the time working with the groups of students. The activities done by the students are the key part of this lesson, including their use of the computer.

Each group of students will have to write letters to tne candidates in the various races being examined. The letters will explain what the class is doing, ask for interviews with the candidates, and ask for updates on news releases and public appearances. The letters do not have to be the same from each group, but all groups should be aware of other letters. A word processing package should be used to write the letters.

Throughout the campaign, each group of students should identify as many issues as possible discussed by each candidate. The position of each candidate on every issue should be recorded. A database package or word processing package could be used to store issues and positions. This file will need updating often, as candidates make speeches, public appearances and news releases come out.

The students should keep a record of each candidate's appearance before civic groups, presentations before public bodies, news conferences, etc.

One important learning activity for the class would be to conduct a pre-election public opinion poll of the races being examined. There could be three or four such polls during the campaign. Each group would contribute questions to the poll, but there should be one poll from the entire class.

The poll could be taken of students in the school and their parents (not just in the civics class). Results of polls should be released to candidates and the newspapers — with an explanation that the results are unscientific and probably not accurate in predicting the final vote.

A spreadsheet or database package could be used to store the poll results. A BASIC program could be written (possibly by students in the computer class, not the students in the civics class) to calculate frequencies and percentages of responses to questions. If preferred, a spreadsheet package could be used to calculate frequencies and percentages. Some of the graphics packages available on the market would be an excellent way of obtaining figures and charts of the poll results. Naturally, a word processing package would be used to write the results of the poll.

After the election, students could obtain voting results by precinct for all races studied in the class. Precinct results can be stored on a spreadsheet or database. Comparisons of actual results with the pre-election polls could be made to determine how close the sample matched the final vote. The precinct results have greater use for analysis of the election, which is the purpose of the whole exercise.

Students can compare all races examined by the groups of the class to see what patterns exist. Each group would have to share its files with all other groups. In order to provide an analysis of the elections, the teacher may want to direct the classroom to positions on issues held by candidates, speaking engagements, editorial support from news media and even precinct voting patterns. A BASIC program or word processing package can be used to merge important information from the various files created by the students.

Throughout this entire exercise, students are not taught anything about the computer itself; they never sit down in front of a keyboard for a lesson. The computer is integrated into the lesson, which would be impossible to complete in the same way without electronic assistance. Some students might learn how to type during the unit on local elections; some will learn their way around a database package; some will discover previously unknown secrets of a spreadsheet program; and some may never turn on the computer. Someone in each group will have to operate the computer: store and retrieve information, type reports and put pieces of data together in a coherent form. That does not mean all members of the group will have to be turned into computer operators.

The role of the computer is to organize information needed by the students in order to solve problems and assist with higher order thinking skills. In the truest sense of the term, the computer is a classroom tool, very similar to a 16mm movie projector or card catalog in the library. Students are never assigned a computer task for the purpose of that task itself, but only to help them in a learning experience.

After all, the purpose of this lesson is not to learn about commercial packages or BASIC programming; it is to learn about local elections. Without the electronic aid, students would have a much more difficult time sifting through the information on hand, and might not ever be able to conduct the same type of analysis that is possible with the computer.

The example of a civics class lesson is given for illustrative purposes only. The same type of activities could be done in many different classrooms with many different subjects.

As far as I know, there is no school where such a curriculum is in use. In all probability, there are individual classrooms where teachers are using the computer in the manner described above, but doing so in isolation from other teachers in the building. In a way, that is a desirable condition; teaching is an art, and individual teachers are expected to create individualistic lessons for their students.

If you, or anyone you know, is using the computer to conduct lessons similar to the one described, please write to me. I would like to know about the activities and start a file on such units of instruction. My address is 829 Evergreen, Chatham, IL 62629.

32K ECB

ANYONE FOR KENSINGTON?

by Bob Delbourgo

It is very rare that a truely original and exciting game hits the market. But this can be said quite unequivocally for the new game KENSINGTON. Ever since we were given it as a gift two Christmases ago I have tried in vain to develop an airtight strategy for winning! This proves (or at any rate suggests) that there is as much to Kensington as there is to GO or CHESS.

The aim of the game is to be the first to form a hexagon of your own pieces around a circle (providing that the colour of the circle is not that of your opponent). The directions / rules are given in the program if you are hazy about them. Even if you know the rules of play read through them if only to discover how to shift the cursor, manoevure the pieces and make your moves. Remember that when you form a triangle in your colour for the first time, you may displace one of your opponent's pieces to any vacant vertex on the board; with a completed square, you are entitled to shift two of your opponent's pieces.

The program is for two players as I was incapable of finding the optimal strategy and thus could not figure out how to get the CoCo to play properly. It will not fit into 16K as there are far too many DATA statements that are needed to specify the locations of the 72 vertices and their interconnections. (Incidentally, do be very careful when you type in the DATA as errors will be difficult to spot afterwards). All these need dimensioning and occupy loads of valuable memory. The relationship between the various vertices are held in strings P$( ) and the sets that appear in squares or triangles are embodied in the strings S$( ) and T$( ). This will help you to understand the listing I hope.

Oh, by the way, I have called the program "ALBERT" because Kensington has nine letters and also because the idea of the game was conceived around the Albert Memorial in Kensington Gardens where we often used to take Daniel and Tino for a stroll when they were infants.

THE LISTING:

```
1 '**********ALBERT*************
  ********BOB DELBOURGO********
2 GOTO10
3 SAVE"ALBERT:2":STOP
10 CLS0:CLEAR1200:DIMX(72),Y(72)
,P$(72),S$(30),SS$(30),T$(24),TT
$(24):DI$="UERFDGLH":TU$(1)="V10
03L4C02GL16GP255L8AP255L16F":TU$
(2)="V1003L8CL6FL16EL8DC02B":MU$
(1)="V10L25503C04E05AGGGGGG":MU$
(2)="V10L25503C02G01ECCCCCCC"
11 P$=STRING$(72,"0"):PI=3.14159
26535:FORI=1TO6:X(I)=22*COS((2*I
-5)*PI/6)+128:Y(I)=22*SIN((2*I-5
)*PI/6)+96
12 X(I+6)=X(I)+60:Y(I+6)=Y(I):X(
I+24)=X(I)-60:Y(I+24)=Y(I):X(I+1
2)=X(I)+30:Y(I+12)=Y(I)+51:X(I+1
8)=X(I)-30:Y(I+18)=Y(I)+51:X(I+3
0)=X(I)-30:Y(I+30)=Y(I)-51:X(I+3
6)=X(I)+30:Y(I+36)=Y(I)-51:NEXTI
13 X(43)=X(1):Y(43)=Y(42)-18:X(4
4)=X(37)-11:Y(44)=Y(37)-18:X(45)
=X(44)+22:Y(45)=Y(44):X(46)=X(38
)+11:Y(46)=Y(38)-18:X(47)=X(38)+
22:Y(47)=Y(38)
14 X(48)=X(47):Y(48)=Y(47)+22:X(
49)=X(8)+11:Y(49)=Y(8)-18:X(50)=
X(8)+22:Y(50)=Y(8):X(51)=X(50):Y
(51)=Y(9):X(52)=X(49):Y(52)=Y(9)
+18
15 X(53)=X(14)+22:Y(53)=Y(14):X(
54)=X(53):Y(54)=Y(15):X(55)=X(10
):Y(55)=Y(15)+18:X(56)=X(16)+11:
```

```
Y(56)=Y(16)+18:X(57)=X(16)-11:Y(
57)=Y(56)
16 X(58)=X(4):Y(58)=Y(55):X(59)=
X(22)+11:Y(59)=Y(57):X(60)=X(22)
-11:Y(60)=Y(59):X(61)=X(23)-11:Y
(61)=Y(55):X(62)=X(23)-22:Y(62)=
Y(54)
17 X(63)=X(62):Y(63)=Y(53):X(64)
=X(29)-11:Y(64)=Y(52):X(65)=X(29
)-22:Y(65)=Y(29):X(66)=X(65):Y(6
6)=Y(30):X(67)=X(64):Y(67)=Y(49)
18 X(68)=X(63):Y(68)=Y(35):X(69)
=X(68):Y(69)=Y(36):X(70)=X(25):Y
(70)=Y(46):X(71)=X(26):Y(71)=Y(4
4):X(72)=X(6):Y(72)=Y(71)
19 PRINT@261,CHR$(191)+CHR$(191)
;:PRINT@293,STRING$(2,207);:PRIN
T@324,STRING$(4,191);:POKE1316,1
97:POKE1319,202:POKE1346,207:POK
E1353,207:PRINT@354,STRING$(8,20
7);:PRINT@385,STRING$(10,179);:P
RINT@416,STRING$(12,179);
20 FORI=99TO323STEP32:POKEI+1024
,143:POKEI+1029,143:NEXTI:FORI=5
TO69STEP32:POKE1024+I,133:POKEI+
1025,138:NEXTI:POKE1188,158:POKE
1156,159:POKE1157,158:POKE1125,1
51:POKE1126,155:POKE1158,157:POK
E1159,159:POKE1191,157:POKE1221,
143:POKE1253,143:POKE1254,14
21 DATA 108,140,172,204,236,268,
300,332,333,111,143,175,207,239,
271,303,335,336,337,305,273,241,
209,208,341,340,339,307,275,243,
211,212,213,245,277,276
22 DATA 343,311,279,247,215,248,
217,218,188,189,190,125,157,221,
253,285,317,349,350
23 FORI=1TO9:READD:POKE1024+D,20
7:NEXTI:FORI=1TO15:READD:POKE102
4+D,191:NEXTI:FORI=1TO12:READD:P
OKE1024+D,207:NEXTI:FORI=1TO8:RE
ADD:POKE1024+D,191:NEXTI:FORI=1T
O11:READD:POKE1024+D,207:NEXTI
24 PRINT@370,"F R O M";:PRINT@39
6,STRING$(19,139);:PRINT@448,"A
VERSION FOR COCO - R.DELBOURGO";
25 FORI=1TO6:PRINT@428,"k e n s
i n g t o n";:FORT=1TO100:NEXT:P
RINT@428,"K E N S I N G T O N";:
FORT=1TO100:NEXTT,I
26 PLAY"03V20L6CFP100FL18FGAB-04
CP24"
27 PLAY"03L6FL4GP12L16AB-L4A"
28 PRINT@448,"WOULD YOU LIKE INS
TRUCTIONS Y/N?";:I$=INKEY$:IFI$=
"N"THEN39ELSEIFI$="Y"THEN29ELSE2
8
29 PRINT@212,"THIS BRILLIANT NEW"
:PRINT@244,"GAME WAS INVENTED":PR
INT@276,"QUITE RECENTLY BY":PRINT
@108,"TAYLOR AND FORBES":PRINT@1
40,"AND IS PATENTED BY":PRINT@17
2,"WHALE TOYS LIMITED.":PRINT@20
4,STRING$(20,128);
30 PRINT@448,STRING$(32,128);:PR
INT@396,STRING$(20,128);:PRINT@2
36,"HERE IS THE COLOR":PRINT@268
,"COMPUTER VERSION FOR";:PRINT@3
00,"TWO PLAYERS, CALLED":PRINT@3
32,"albert, AND HERE ARE";:PRINT
@364,"THEIR RULES OF PLAY.";
31 GOSUB189:GOSUB191:PRINT@212,"I
N phase 1, EACH OF":PRINT@244,"TH
E PLAYERS "+CHR$(239)+" OR "+CHR
$(255):PRINT@276,"ALTERNATELY PLA
CES":PRINT@108,"HIS/HER 15 PIECE
S AT";:PRINT@140,"THE BOARD VERT
ICES."
32 PRINT@204,"THIS IS ACHIEVED B
Y":PRINT@236,"MOVING THE CURSOR"
:PRINT@268,"IN THE DIRECTIONS":P
RINT@300,"SPECIFIED BY KEYS":PRI
NT@332,"u,e,r,f,d,g,l,h  AND";:P
RINT@364,"THEN PRESSING enter"
33 GOSUB189:PRINT@212,"IN phase 2
, PLAYERS":PRINT@244,"MOVE THEIR
PIECES":PRINT@276,"INTO VACANT AD
JACENT";:PRINT@108,"VERTICES, IN
 TURN.":PRINT@140,STRING$(20,128
);
34 PRINT@172,"MOVE THE CURSOR IN
TO";:PRINT@204,"POSITION VIA u,e
,..":PRINT@236,"THEN PRESS KEY m
":PRINT@268,"FOLLOWED BY ANY KEY
":PRINT@300,"FOR DIRECTION.":PRI
NT@332,"IF IN ERROR, PRESS":PRIN
T@364,"n TO NULLIFY MOVE m";
35 GOSUB189:GOSUB191:PRINT@212,"T
HE GAME IS WON WHEN";:PRINT@244,"
EITHER PLAYER FORMS":PRINT@276,"A
 COMPLETE HEXAGON":PRINT@108,"AR
OUND A CYAN CIRCLE";:PRINT@140,"
OR CIRCLE OF HIS/HER";:PRINT@172
,"OWN COLOUR."
36 PRINT@204,"IF A PLAYER CANNOT
":PRINT@236,"MOVE, THE OPPONENT"
:PRINT@268,"PLAYS AGAIN PROVIDED
";:PRINT@300,"c IS FIRST PRESSED
.":PRINT@332,"DRAWS CAN ONLY BE"
:PRINT@364,"REACHED BY CONSENT."
37 GOSUB189:GOSUB191:PRINT@212,"D
URING ANY phase, IF";:PRINT@244,"
BY THE MOVE A SQUARE";:PRINT@276,
"OF ONE COLOUR IS":PRINT@108,"FI
LLED, 2 PIECES":PRINT@140,"OF TH
E OPPONENT MAY":PRINT@172,"BE SH
IFTED. FOR A":PRINT@204,"TRIANGL
E, 1 PIECE IS";
38 PRINT@236,"TO BE MOVED.":PRIN
T@268,"USE THE CURSOR, m &":PRIN
T@300,"THE DIRECTION KEYS":PRINT
@332,"TO EFFECT THE MOVES.";:PRI
NT@396,"GOOD LUCK NOW!!":POKE147
1,143:GOSUB189
39 PRINT@448," please wait for i
nitialization ";
40 DATA 0,41,0,2,0,6,0,33,0,40,1
2,0,3,0,0,1
41 DATA 2,0,11,13,0,4,0,0,0,3,0,
18,0,20,0,5
42 DATA 6,0,0,4,0,1,27,0,0,1,0,0
,5,0,26,34
43 DATA 0,48,0,8,0,12,0,39,0,49,
50,0,9,0,0,7
44 DATA 8,0,51,52,0,10,0,0,0,9,0
,53,0,14,0,11
45 DATA 12,0,0,10,0,13,3,0,0,7,0
,0,11,0,2,40
46 DATA 0,11,0,14,0,18,0,3,0,10,
53,0,15,0,0,13
47 DATA 14,0,54,55,0,16,0,0,0,15
,0,56,0,57,0,17
48 DATA 18,0,0,16,0,58,21,0,0,13
,0,0,17,0,20,4
49 DATA 0,5,0,20,0,24,0,27,0,4,1
8,0,21,0,0,19
50 DATA 20,0,17,58,59,22,0,0,0,2
1,0,59,0,60,0,23
51 DATA 24,0,0,22,0,61,62,0,0,19
,0,0,23,0,63,28
52 DATA 0,35,0,26,0,30,0,68,0,34
,6,0,27,0,0,25
53 DATA 26,0,5,19,0,28,0,0,0,27,
0,24,0,63,0,29
54 DATA 30,0,0,28,0,64,65,0,0,25
,0,0,29,0,66,67
55 DATA 0,72,0,32,0,36,0,71,0,43
,42,0,33,0,0,31
56 DATA 32,0,41,1,0,34,0,0,0,33,
0,6,0,26,0,35
57 DATA 36,0,0,34,0,25,68,0,0,31
,0,0,35,0,69,70
58 DATA 0,45,0,38,0,42,0,44,0,46
,47,0,39,0,0,37
59 DATA 38,0,48,7,0,40,0,0,0,39,
0,12,0,2,0,41
60 DATA 42,0,0,40,0,1,33,0,0,37,
0,0,41,0,32,43
61 DATA 0,44,0,42,0,32,0,72,0,0,
45,37,0,43,0,0
62 DATA 0,0,0,46,0,37,44,0,0,0,0
,47,0,38,0,45
63 DATA 0,0,0,0,48,0,38,46,47,0,
```

Take This Message, Please!

By Steve Blyn

Functional literacy has been defined as the ability to apply reading, writing, speaking and listening skills to problems and tasks of a practical nature encountered in everyday life.

Computers can offer students many opportunities to attain and maintain these skills. One of the skills included in functional literacy is the ability to receive and write a clear message. This month's article will focus on this skill.

Our program will present a form for taking a telephone message. The student's task is to complete the information necessary to communicate the message from one person to another.

Sounds simple, doesn't it? Just write down the information the caller gave you. Like many other tasks, this is simple only after you have mastered it.

You really have to live with children to appreciate the kinds of errors they can make when taking a telephone message. One common mistake is that children will forget to ask either for the name of the caller or the caller's telephone number. (This, of course, only happens when they answer an important call for you!) "I didn't know I had to ask for his telephone number. I thought you knew everyone's number," says your child.

Children also often omit the important part of a phone message. It takes time and practice as well as some maturity for children to learn this, much the same way reading teachers devote a lot of time to teaching the related skill of recognizing the main ideas of paragraphs and stories.

The features we feel would make a phone message complete are:

1) The date and time of the message
2) To whom the message is directed
3) The name and phone number of the caller
4) A summary of the message

Program lines 180-240 ask the child to record the date and time as well as the caller's name and telephone number. The strings at the end of the lines represent the information that will be saved in the data file.

The actual message gets entered on Line 260. Lines 270-310 question whether the message is correct. We felt it wise to give the child a chance to redo the message if he is not satisfied with it.

This program's foremost purpose is to enable the student to write clear messages to and for others. To accomplish this end, the program also presents a means to save and retrieve these messages. A file for each message will be created for future reading and evaluation.

Provision has been made for either cassette or disk storage. The beginning of the program asks which form of storage you will be using. The device #1 is used for disk and #-1 is used for cassette by the CoCo. Lines 60-90 will let you select either tape or disk storage.

The user may also select to either write a new message or look at a previous message. When selecting filenames for this or any program, there are two rules to remember.

1) Limit your filename to eight letters or less. A file extension of "/DAT" will automatically be included on disk files.

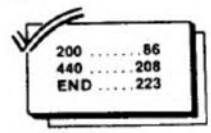
2) Do not call two programs by the same name. A disk drive will overwrite and wipe out the first program if you give another the same name.

The routine on Line 320 saves the message as a file. The routine on lines 360-390 retrieve any file that was previouly saved. Lines 400-430 print out that message on the screen. You can have the messages sent to a printer if the device #-2 is used on these lines or an additional set of similar lines. For example, Line 430 could be changed to have the output on both screen and printer by including:

```
430 PRINT"MESSAGE:";F$:PRINT#-
    2,"MESSAGE:"F$
```

Messages are made by a caller to the student who records and saves it. The message is presumably retrieved by the third person for whom it was intended. This would most probably be the teacher or parent. At this point, the messages taken by the student can be analyzed for completeness and clarity.

The results of this program are best tested by having children answer real telephone calls and take messages. This, however, can also be done by role playing with several children. You can be the pretend caller one time and the answerer/recorder the next time. Sample phone conversation cards of typical phone calls could also be easily made for a single child to practice the skill of recording a message.



The listing: MESSAGES

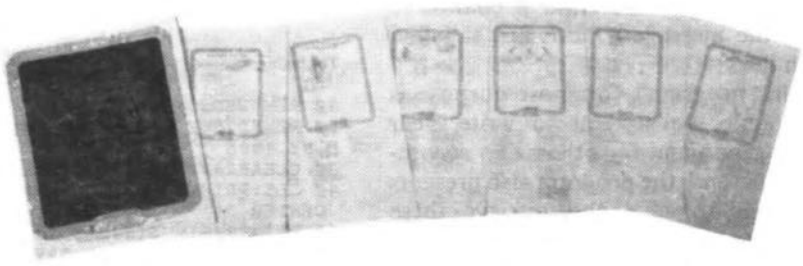
```
1Ø REM"PERSONAL MESSAGES"
2Ø REM"STEVE BLYN,COMPUTER ISLAN
D,NY,1985
3Ø CLEAR1ØØØ
4Ø CLS:PRINT"      YOUR  MESSAGE
 CENTER
5Ø PRINTSTRING$(32,255);
6Ø PRINT"WILL YOU BE SAVING MESS
AGES ON  DISK OR TAPE";:INPUT DV
$
7Ø IF DV$="D" THEN DV=1
8Ø IF DV$="T" THEN DV=-1
9Ø IF DV$<>"D" AND DV$<>"T" THEN
 6Ø
1ØØ PRINT:PRINT"DO YOU WANT TO S
AVE OR LOOK AT A MESSAGE    s  O
R  l  ?"
11Ø EN$=INKEY$
12Ø IF EN$="L" THEN 36Ø
13Ø IF EN$="S" THEN 15Ø
14Ø GOTO 11Ø
15Ø CLS
16Ø PRINT"WHAT IS THE NAME OF TH
IS MESSAGE";:INPUT N$
17Ø CLS
18Ø PRINT@7,"PERSONAL MESSAGES";
19Ø PRINT@64,"TO:";:LINEINPUT A$
2ØØ PRINT@96,"DATE:";:LINEINPUT
B$
21Ø PRINT@115,"TIME:";:LINEINPUT
 C$
22Ø PRINT@128," ":'REMOCE THE EX
TRA IGNORED SIGN
23Ø PRINT@128,"CALLER:";:LINEINP
UT D$
24Ø PRINT@16Ø,D$;"'S #";:LINE IN
PUT E$
25Ø PRINT@224,STRING$(192,".");
26Ø PRINT@224,"MESSAGE:";:LINEIN
PUT F$
27Ø PRINT@417,"IS THIS MESSAGE C
ORRECT(Y/N) ?";
28Ø EN$=INKEY$
29Ø IF EN$="Y" THEN 32Ø
3ØØ IF EN$="N" THEN 25Ø
31Ø GOTO 28Ø
32Ø OPEN "O",#DV,N$
33Ø WRITE#DV,A$,B$,C$,D$,E$,F$
34Ø CLOSE#DV
35Ø GOTO 1ØØ
36Ø CLS:PRINT"WHAT IS THE NAME O
F THIS MESSAGE";:INPUT N$
37Ø OPEN "I",#DV,N$
38Ø INPUT#DV,A$,B$,C$,D$,E$,F$
39Ø CLOSE#DV
4ØØ PRINT"A MESSAGE TO ";A$
41Ø PRINT"CAME ON ";B$;" AT ";C$
42Ø PRINT"FROM ";D$;": #";E$
43Ø PRINT"MESSAGE:";F$
44Ø PRINT:PRINT"ENTER TO GO ON O
R 'E TO END"
45Ø EN$=INKEY$
46Ø IF EN$=CHR$(13) THEN GOTO 1Ø
Ø
47Ø IF EN$="E" THEN CLS:END
48Ø GOTO 45Ø
```

Tandy ELECTRONICS

SAVE! ON

SALE ENDS 30/9/85

EDUCATIONAL SOFTWARE FOR YOUR COLOR COMPUTER



Color Logo Programming Language. Students of all ages will discover the visual rewards of programming and problem-solving with this unique color graphics program. They program a "turtle" to apply mathematical and geometric concepts on the screen. Advanced students can even run several programs at once for animation and games.
Disk Version. 26-2721 Was 139.95 . . . . . **Now 59.95**
Program Pak. 26-2722 Was 69.95 . . . . . **Now 49.95**

Computer Literacy: Computers Past and Present. Introduces the history of computers for grades 4 to 8. Defines the computer, its uses and responses. Access to a computer is not necessary. Includes teacher guide, six spirit masters and wall chart. 26-2755 Was 9.95 . . . . . **Now 3.95**

The Illustrated Computer. A guide to applying machine language programming concepts and techniques for secondary students. The program manual has 3 sections: User Guide, Text and Exercises, and Answer Key. They teach programming through hands-on experience, tutorial-style explanations, exercises and solutions. For 32K Color Computer disk system with Disk Extended Color BASIC. 26-2670 Was 49.95 . . . . . **Now 29.95**

Color Pilot. An easy-to-follow guide to mixing text and high-resolution color graphics for stunning results. Students learn how to create illustrated stories. Manual with sample programs.
Tape Version. 26-2709 **Was 79.95 . . . . . Now 49.95**
Disk Version. 26-2710 **Was 129.95 . . . . . Now 59.95**

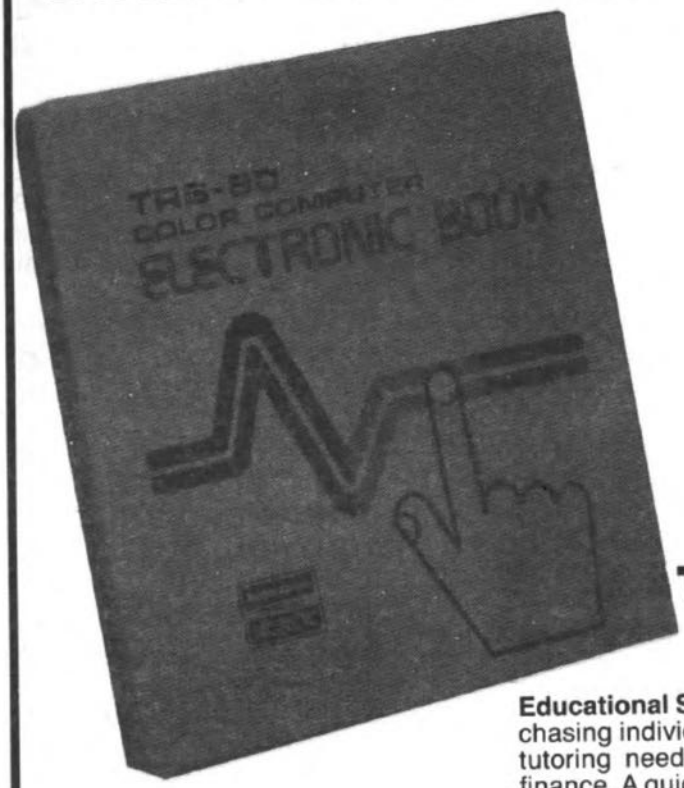
SKILL DEVELOPMENT IN READING AND VOCABULARY

Fun With Reading. Each package includes an illustrated reader, a read-along cassette tape and a computer tape with spelling and vocabulary exercises. For years 4 to 6.
The Hound of The Baskervilles. An exciting story designed to motivate a desire to read. 26-2551 Was 29.95 . . . . . **Now 14.95**
Moby Dick. Follow this adventure by listening, watching and reading. 26-2552 Was 29.95 . . . . . **Now 14.95**
20,000 Leagues Under the Sea. Children are encouraged to learn by the positive feedback from the computer. 26-2553 Was 29.95 . . . . . **Now 14.95**
Vocabulary Tutor 1. 26-2568 Was 9.95 . . . . . **Now 4.95**
Vocabulary Tutor 2. 26-2569 Was 9.95 . . . . . **Now 4.95**

If an item is out of stock your store manager will gladly order the item for you

Bring Rainbow/Coco to your store to take advantage of these offers.

Learn With The Electronic Book

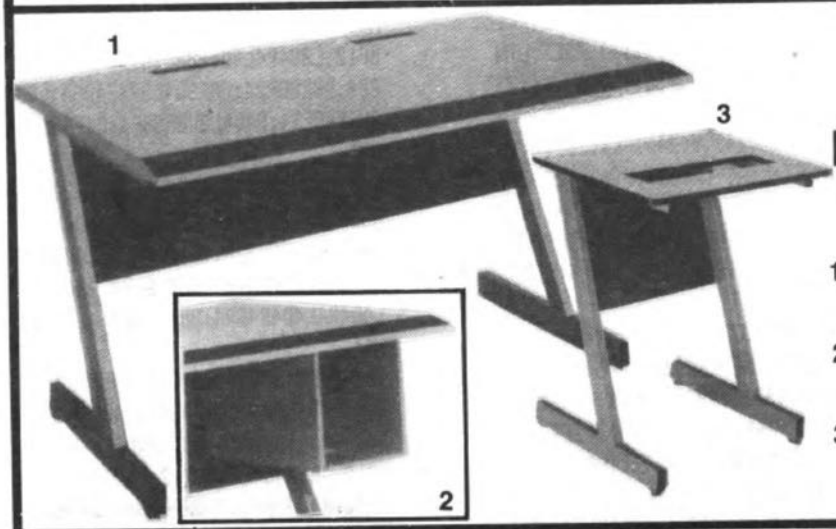


- **The Screen Produces an Instant Reward for Correct Answers**

The Color Computer Electronic Book. Prepare your children for school with an education series that motivates them to learn more! The touch-sensitive pages help children teach themselves. Plugs into joystick port. 26-3141 ..... **29.95**
Maze Master. An introduction to computer use. 26-2541 ..... **24.95**
Shape Maker. 26-2541 ..... **24.95**
The Number Factory. One number goes in the factory, another comes out. Solve the equation. 26-2543 ..... **24.95**
Word Wizard. Select the correct spelling by "shooting" at it. 26-2544 ..... **24.95**

THE EDUCATIONAL SOFTWARE SOURCEBOOK

Educational Software Sourcebook. A thorough guide to locating and purchasing individual software programs. The Sourcebook has anticipated the tutoring needs of teachers as well as student administration and even finance. A quick glance at the index satisfies a particular subject need. Each entry states subject use, program description, user level, its application, hardware requirements and ordering information. 26-7991 ..... **14.95**



LIMITED QUANTITY!

INCREDIBLE SAVINGS ON COMPUTER FURNITURE

1. **System Desk.** Contemporary styling is perfect for home or office. Unassembled. 26-1305
   Was 349.95 ..... **Now 199.95**
2. **Disk Bay.** Attaches underneath desk. Holds two external disk drives. 26-1306
   Was 49.95 ..... **Now 9.95**
3. **Universal Printer Stand.** Includes slotted top. 26-1308 Was 169.95 ..... **Now 69.95**

Sorry No Rainchecks on These Items

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

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

```
0,49,0,7,39,0
64 DATA 0,0,0,50,0,8,0,48,0,0,0,
0,51,0,8,49
65 DATA 50,0,0,0,0,52,9,0,0,51,0
,0,0,53,0,9
66 DATA 0,52,0,0,54,0,14,10,53,0
,0,0,0,55,15,0
67 DATA 0,54,0,0,0,56,0,15,0,55,
0,0,0,0,57,16
68 DATA 0,16,56,0,0,0,0,58,0,17,
0,57,0,59,0,21
69 DATA 0,58,0,0,0,0,60,22,0,22,
59,0,0,0,0,61
70 DATA 0,23,0,60,0,0,0,62,63,0,
23,61,0,0,0,0
71 DATA 0,28,24,0,62,0,0,64,0,29
,0,63,0,0,0,65
72 DATA 66,0,29,64,0,0,0,0,0,67,
30,0,65,0,0,0
73 DATA 0,68,0,30,0,66,0,0,69,0,
35,25,0,67,0,0
74 DATA 0,70,36,0,68,0,0,0,0,71,
0,36,0,69,0,0
75 DATA 0,0,72,31,0,70,0,0,0,0,0
,43,0,31,71,0
76 FORI=1TO72:P$(I)="":FORL=1TO8
:READD:P$(I)=P$(I)+CHR$(D):NEXTL
,I
77 DATA 1,2,40,41,2,3,11,12,3,4,
18,13,4,5,19,20,5,6,26,27,1,6,34
,33
78 DATA 7,12,40,39,10,11,13,14,1
7,18,20,21,19,24,28,27,25,26,34,
35,32,33,41,42
79 DATA 37,42,43,44,37,38,46,45,
38,39,48,47,7,8,49,48,8,9,51,50,
9,10,53,52
80 DATA 14,15,54,53,15,16,56,55,
16,17,58,57,21,22,59,58,22,23,61
,60,23,24,63,62
81 DATA 28,29,64,63,29,30,66,65,
25,30,67,68,35,36,69,68,31,36,70
,71,31,32,43,72
82 FORI=1TO30:S$(I)="":SS$(I)="0
000":FORL=1TO4:READD:S$(I)=S$(I)
+CHR$(D):NEXTL,I
83 DATA 1,33,41,2,12,40,3,11,13,
4,18,20,5,19,27,6,26,34
84 DATA 32,42,43,7,39,48,10,14,5
3,17,21,58,24,28,63,25,35,68
85 DATA 37,44,45,38,46,47,8,49,5
0,9,51,52,15,54,55,16,56,57
86 DATA 22,59,60,23,61,62,24,64,
65,30,66,67,36,69,70,31,71,72
87 FORI=1TO24:T$(I)="":TT$(I)="0
00":FORL=1TO3:READD:T$(I)=T$(I)+
CHR$(D):NEXTL,I
88 PMODE1,1:PCLS:SCREEN1,1:COLOR
2:FORJ=0TO6:FORI=1TO5:LINE(X(I+6
*J),Y(I+6*J))-(X(I+1+6*J),Y(I+1+
6*J)),PSET:NEXTI:LINE(X(6+6*J),Y
(6+6*J))-(X(1+6*J),Y(1+6*J)),PSE
T:NEXTJ
89 LINE(X(1),Y(1))-(X(33),Y(33))
,PSET:LINE-(X(41),Y(41)),PSET:LI
NE-(X(1),Y(1)),PSET:LINE(X(2),Y(
2))-(X(40),Y(40)),PSET:LINE-(X(1
2),Y(12)),PSET:LINE-(X(2),Y(2)),
PSET
90 LINE(X(3),Y(3))-(X(11),Y(11))
,PSET:LINE-(X(13),Y(13)),PSET:LI
NE-(X(3),Y(3)),PSET:LINE(X(4),Y(
4))-(X(18),Y(18)),PSET:LINE-(X(2
0),Y(20)),PSET:LINE-(X(4),Y(4)),
PSET
91 LINE(X(5),Y(5))-(X(19),Y(19))
,PSET:LINE-(X(27),Y(27)),PSET:LI
NE-(X(5),Y(5)),PSET:LINE(X(6),Y(
6))-(X(26),Y(26)),PSET:LINE-(X(3
4),Y(34)),PSET:LINE-(X(6),Y(6)),
PSET
92 LINE(X(32),Y(32))-(X(43),Y(43
)),PSET:LINE-(X(42),Y(42)),PSET:
LINE-(X(32),Y(32)),PSET:LINE(X(7
),Y(7))-(X(39),Y(39)),PSET:LINE-
(X(48),Y(48)),PSET:LINE-(X(7),Y(
7)),PSET
93 LINE(X(10),Y(10))-(X(14),Y(14
)),PSET:LINE-(X(53),Y(53)),PSET:
LINE-(X(10),Y(10)),PSET:LINE(X(1
7),Y(17))-(X(21),Y(21)),PSET:LIN
E-(X(58),Y(58)),PSET:LINE-(X(17)
,Y(17)),PSET
94 LINE(X(24),Y(24))-(X(28),Y(28
)),PSET:LINE-(X(63),Y(63)),PSET:
LINE-(X(24),Y(24)),PSET:LINE(X(2
5),Y(25))-(X(35),Y(35)),PSET:LIN
E-(X(68),Y(68)),PSET:LINE-(X(25)
,Y(25)),PSET
95 LINE(X(37),Y(37))-(X(44),Y(44
)),PSET:LINE-(X(45),Y(45)),PSET:
LINE-(X(37),Y(37)),PSET:LINE(X(3
8),Y(38))-(X(46),Y(46)),PSET:LIN
E-(X(47),Y(47)),PSET:LINE-(X(38)
,Y(38)),PSET
96 LINE(X(8),Y(8))-(X(49),Y(49))
,PSET:LINE-(X(50),Y(50)),PSET:LI
NE-(X(8),Y(8)),PSET:LINE(X(9),Y(
9))-(X(51),Y(51)),PSET:LINE-(X(5
2),Y(52)),PSET:LINE-(X(9),Y(9)),
PSET
97 LINE(X(15),Y(15))-(X(54),Y(54
)),PSET:LINE-(X(55),Y(55)),PSET:
LINE-(X(15),Y(15)),PSET:LINE(X(1
6),Y(16))-(X(56),Y(56)),PSET:LIN
E-(X(57),Y(57)),PSET:LINE-(X(16)
,Y(16)),PSET
98 LINE(X(22),Y(22))-(X(59),Y(59
)),PSET:LINE-(X(60),Y(60)),PSET:
LINE-(X(22),Y(22)),PSET:LINE(X(2
3),Y(23))-(X(61),Y(61)),PSET:LIN
E-(X(62),Y(62)),PSET:LINE-(X(23)
,Y(23)),PSET
99 LINE(X(29),Y(29))-(X(64),Y(64
)),PSET:LINE-(X(65),Y(65)),PSET:
LINE-(X(29),Y(29)),PSET:LINE(X(3
0),Y(30))-(X(66),Y(66)),PSET:LIN
E-(X(67),Y(67)),PSET:LINE-(X(30)
,Y(30)),PSET
100 LINE(X(36),Y(36))-(X(69),Y(6
9)),PSET:LINE-(X(70),Y(70)),PSET
:LINE-(X(36),Y(36)),PSET:LINE(X(
31),Y(31))-(X(71),Y(71)),PSET:LI
NE-(X(72),Y(72)),PSET:LINE-(X(31
),Y(31)),PSET
101 FORI=43TO71:LINE(X(I),Y(I))-
(X(I+1),Y(I+1)),PSET:NEXTI:LINE-
(X(43),Y(43)),PSET
102 FORI=1TO72:FORJ=1TO5:CIRCLE(
X(I),Y(I)),J,1:NEXTJ:CIRCLE(X(I)
,Y(I)),6,2:NEXTI
103 FORI=1TO15STEP2:CIRCLE(128,9
6),I,2:CIRCLE(188,96),I,2:CIRCLE
(68,96),I,2:CIRCLE(158,147),I,4:
CIRCLE(98,147),I,4:CIRCLE(158,45
),I,3:CIRCLE(98,45),I,3:NEXTI
104 DRAW"BM20,20;C2U8D8NE6NR8NF6
ND8NG6NL8NH6":DRAW"BM16,0;D8R8U8
BF12;L8U4R4L4U4R8BL24;D8U4L8U4D8
BG4;D8R8BR24;U8R8D4L8R4F4BG4;L8D
8U4R4BL24;R4D4L8U8R8BF4;R4F4G4L4
U8"
105 DRAW"BM196,6;R8U4L8D8BR20;L8
U8BD8BR12;U4E4F4L8R8D4BR8;U4H4F4
E4":DRAW"BM208,12;G4F4BR8L4U4R4L
4U4R4BR8;D8U4H4D8BR10;U8L2R4BR8;
L4D4R4L4D4R4BR4;U8R4D4L4F4;BR4E4
H4"
106 DRAW"BM208,24;D8E2F2U8BR4;D8
U4R4U4D8BE8;L4D4R4L4D4R4BR4;U8F4
U4D8BR10;U8BR10;D8U4H4D8":DRAW"B
M218,40;R4U4L4D8BR12;L4U8BF8;U8R
4D4L4R4D4BR8;L4U8R4BF8;L4U4R4L4U
4R4"
107 PCOPY1TO3:PCOPY2TO4:L=1:P=2:
W=0
108 FORII=1TO30:P=P+1:IFP=5THENP
=3
109 FORI=1TO4:CIRCLE(248,6),I,P:
NEXTI:GOSUB127
110 PP=P:GOSUB152:IFII<11THEN112
111 GOSUB160:IFW=1THEN119
112 IFII<5THEN114
113 GOSUB164
```

```
114 NEXTII
115 P=P+1:CC=0:IFP=5THENP=3
116 FORI=1TO4:CIRCLE(248,6),I,P:
NEXTI:GOSUB136
117 IFCC=1THEN115ELSEGOSUB160:IF
W=1THEN119
118 GOSUB164:GOTO115
119 GOSUB193:COLOR1:LINE(0,172)-
(60,192),PSET,BF:COLOR2
120 FORN=1TO6:CIRCLE(6,184),N,P:
NEXTN:DRAW"BM16,180;D6F2E2U2D2F2
E2U6BR4;D8BR4;U8F8U8BR12;L8D4R8D
4L8"
121 PLAY"O3L4GFL16ECFDL8GP100L8F
L4EDL16CL4C"
122 FORN=2TO4:DRAW"C"+STR$(N)+"B
M196,180;G4F4BR4U8R8D4L8R4F4BR4E
4H4BR12;L4D4R4L4D4R4BR4;U8R8D4L4
F4BE8;D8L4U8BF8;U8F4U4D8"
123 IFINKEY$="R"THEN125
124 NEXTN:GOTO122
125 N=5:NEXTN:PCLS1:PCOPY3TO1:PC
OPY4TO2:PMODE1,1
126 P$=STRING$(72,"0"):FORI=1TO3
0:SS$(I)=STRING$(4,"0"):NEXTI:FO
RI=1TO24:TT$(I)=STRING$(3,"0"):N
EXTI:W=0:P=2:GOTO108
127 REM all subroutines begin he
re
128 GOSUB192
129 I$=INKEY$:IFI$=""THEN128
130 M=INSTR(1,DI$+CHR$(13),I$):I
FM=0THEN128
131 IFM<9THEN134
132 IFMID$(P$,L,1)<>"0"THENSOUND
250,1:GOTO128
133 FORJ=1TO6:CIRCLE(X(L),Y(L)),
J,P:NEXTJ:PLAYMU$(P-2):MID$(P$,L
,1)=CHR$(46+P):RETURN
134 IFMID$(P$(L),M,1)=CHR$(0)THE
NSOUND100,2:GOTO128
135 L=ASC(MID$(P$(L),M,1)):GOTO1
27
136 GOSUB192
137 I$=INKEY$:IFI$=""THEN136ELSE
M=INSTR(1,DI$+"MC",I$):IFM=0THEN
136
138 IFM=9THEN141ELSEIFM=10THENCC
=1:RETURN
139 IFMID$(P$(L),M,1)=CHR$(0)THE
NSOUND100,1:GOTO136
140 L=ASC(MID$(P$(L),M,1)):GOTO1
36
141 IFPPOINT(X(L),Y(L))<>P+4 THE
NSOUND200,1:GOTO136
142 L1=L:C1=PPOINT(X(L)-6,Y(L)):
CIRCLE(X(L),Y(L)),6,1
143 I$=INKEY$:PLAY"V30L25503GP8"
:IFI$=""THEN143ELSEM=INSTR(1,DI$
+"N",I$):IFM=0THEN143
144 IFM=9THENCIRCLE(X(L1),Y(L1))
,6,C1:GOTO136
145 IFMID$(P$(L),M,1)=CHR$(0)THE
NSOUND100,1:GOTO143
146 L2=ASC(MID$(P$(L),M,1)):IFMI
D$(P$,L2,1)<>"0" THENSOUND200,1:
GOTO143
147 MID$(P$,L1,1)="0":MID$(P$,L2
,1)=CHR$(46+P)
148 GOSUB156:L=L2:PP=P:GOSUB152
149 FORI=1TO5:CIRCLE(X(L1),Y(L1)
),I,1:NEXTI:CIRCLE(X(L1),Y(L1)),
6,2
150 FORI=1TO6:CIRCLE(X(L2),Y(L2)
),I,P:NEXTI
151 PLAYMU$(P-2):RETURN
152 FORK=1TO30:KK=INSTR(1,S$(K),
CHR$(L)):IFKK THENMID$(SS$(K),KK
,1)=CHR$(46+PP)
153 NEXTK
154 FORK=1TO24:KK=INSTR(1,T$(K),
CHR$(L)):IFKK THENMID$(TT$(K),KK
,1)=CHR$(46+PP)
155 NEXTK:RETURN
156 FORK=1TO30:KK=INSTR(1,S$(K),
CHR$(L1)):IFKK THENMID$(SS$(K),K
K,1)="0"
157 NEXTK
158 FORK=1TO24:KK=INSTR(1,T$(K),
CHR$(L1)):IFKK THENMID$(TT$(K),K
K,1)="0"
159 NEXTK:RETURN
160 W=0:ST$=STRING$(6,46+P):IFLE
FT$(P$,6)=ST$ORMID$(P$,7,6)=ST$O
RMID$(P$,25,6)=ST$THENW=1
161 IFP=3THENIFMID$(P$,31,6)=ST$
ORMID$(P$,37,6)=ST$THENW=1
162 IFP=4THENIFMID$(P$,19,6)=ST$
ORMID$(P$,13,6)=ST$THENW=1
163 RETURN
164 GOSUB193:DRAW"BM202,180;L4D8
R4BR4;U8D4R4U4D8BR8;L4U4R4L4U4R4
BR8;L4D8R4BR4;U8BR4G4F4BR4;U8D8B
R4;U8F4U4D8BE8;L4D8R4U4L2"
165 FORK=1TO30:KK=INSTR(1,S$(K),
CHR$(L)):IFKK THENIFSS$(K)=STRIN
G$(4,46+P)THENJJ=2:GOSUB169:K=31
:GOTO168
166 NEXTK
167 FORK=1TO24:KK=INSTR(1,T$(K),
CHR$(L)):IFKK THENIFTT$(K)=STRIN
G$(3,46+P)THENJJ=1:GOSUB169:K=31
168 NEXTK:GOSUB193:RETURN
169 GOSUB193:DRAW"BM200,192;R4U4
L4U4R4BR4;D8U4R4U4D8BR4;U8D8BR4;
U4R4L4U4R4BR4;R2D8U8R2"
170 IFP=3THENQ=4ELSEQ=3
171 FORJ=1TOJJ:FORN=1TO4:CIRCLE(
228+12*J,186),N,Q:NEXTN,J
172 I$="":DRAW"BM2,172;L2D8R2BR4
U8F2E2D8BR4R2U8L2BR6;R4D8L4U8BR8
;D6F2E2U6BR8;L4D4R4L4D4R4":DRAW"
BM2,184;L2D8R2BR4U8D2F4D2U8BR4R2
D8L2BR6BU8;D8R4U8BR4;D8R4BR4BU8;
D8R4BR4;U8D8BR4;U4R4L4U4R4BR4;D4
R4U4D8L4":FORJ=1TOJJ:PLAYTU$(J)
173 GOSUB192
174 I$=INKEY$:IFI$=""THEN173ELSE
M=INSTR(1,DI$+"M",I$):IFM=0THEN1
73
175 IFM=9THEN177
176 IFMID$(P$(L),M,1)=CHR$(0)THE
NSOUND100,1:GOTO173 ELSEL=ASC(MI
D$(P$(L),M,1)):GOTO173
177 IFPPOINT(X(L),Y(L))<>Q+4 THE
NSOUND200,1:GOTO173
178 L1=L:C1=C:CIRCLE(X(L1),Y(L1)
),6,1
179 C=PPOINT(X(L)-6,Y(L)):CIRCLE
(X(L),Y(L)),6,1:PLAY"V10L25503G"
:CIRCLE(X(L),Y(L)),6,C
180 I$=INKEY$:IFI$=""THEN179ELSE
M=INSTR(1,DI$+"N"+CHR$(13),I$):I
FM=0THEN179
181 IFM=9THENCIRCLE(X(L1),Y(L1))
,6,C1:GOTO173
182 IFM=10THEN185
183 IFMID$(P$(L),M,1)=CHR$(0)THE
NSOUND100,1:GOTO179
184 L=ASC(MID$(P$(L),M,1)):GOTO1
79
185 IFL=L1 THENSOUND200,1:GOTO17
9 ELSEFORN=1TO6:CIRCLE(X(L),Y(L)
),N,Q:NEXTN:FORN=1TO4:CIRCLE(X(L
1),Y(L1)),N,1:NEXTN:CIRCLE(X(L1)
,Y(L1)),6,2
186 FORN=1TO4:CIRCLE(228+12*J,18
6),N,1:NEXTN
187 MID$(P$,L1,1)="0":MID$(P$,L,
1)=CHR$(46+Q):PP=Q:GOSUB152:GOSU
B156
188 NEXTJ:GOSUB193:RETURN
189 PRINT@497,"any inkey";:FORN=
1TO50:NEXTN:PRINT@497,"ANY INKEY
";
190 IFINKEY$=""THENFORN=1TO50:NE
XTN:GOTO189ELSERETURN
191 FORN=12TO396STEP32:PRINT@N,S
TRING$(20,128);:NEXTN:RETURN
192 C=PPOINT(X(L)-6,Y(L)):CIRCLE
(X(L),Y(L)),6,1:PLAY"V1L25501C":
CIRCLE(X(L),Y(L)),6,C:RETURN
193 COLOR1:LINE(196,180)-(256,19
2),PSET,BF:COLOR2:RETURN
```

Soccer Instructor

By Vincent H. Sheridan

A text and graphics program, Soccer Instructor helps newcomers to the game of soccer understand the field markings and their effect on the game. The program requires 32K or 64KECB and a cassette player. I have coached minor league soccer for nine years, and wrote this program to show my family that CoCo could be used for more than playing games.

The title page is followed by an introductory note on the program after which the field is drawn in PMODE4 for the first time, off screen, and shown following the SCREEN command. The program then switches to a menu listing seven field features.

Selection of a feature will cause the field to be redrawn quickly by means of the PCOPY command. The feature is emphasized by flashing, by PSETting and PRESETting lines or PAINTing first in the foreground colour and then in the background colour. After five flashes, the text screen is shown to describe the field feature and its effect on the game.

Soccer Instructor will be of use to beginning soccer players and coaches alike for a better understanding of the game.

Program Structure

10-20	Credit
30	Reserves eight graphics pages
50-240	Title page
250-280	Introductory text
290-310	INKEY$ routine for branch to Line 10000 for field graphics subroutine
320-430	Menu
440	Branch to exit program
450	Branch to repeat program
1000-7190	Subroutines for field features
10000-10260	Draws initial soccer field

240	236	4000	98
340	54	6090	27
1010	95	7015	205
1111	184	7170	217
2140	222	10140	63
3050	114	END	223

The listing: SOCCER

```
10 REM**************************
       *    SOCCER FIELD        *
       * AN ILLUSTRATED GUIDE   *
       * FOR NEW SOCCER PLAYERS *
       *     BY V.A.SHERIDAN    *
       *  COPYRIGHT (C) 1985    *
       **************************
20 REM**************************
30 PCLEAR8
40 CLS
50 FOR X=33TO62
60 PRINT@X,CHR$(128)
70 NEXTX
80 FORX=449TO478
90 PRINT@X,CHR$(128)
100 NEXTX
110 FORY=65TO417STEP32
120 PRINT@Y,CHR$(128)
130 NEXTY
140 FORY=80TO462STEP32
150 PRINT@Y,CHR$(128)
160 NEXTY
170 FORZ=94TO446STEP32
180 PRINT@Z,CHR$(128)
190 NEXTZ
200 PRINT@131,"S O C C E R";
210 PRINT@164,"F I E L D";
220 PRINT@338,"A GUIDE FOR";
230 PRINT@370,"NEW PLAYERS";
240 FORT=0TO3000:NEXTT
250 CLS
260 PRINT:PRINT"**********SOCCER
 FIELD********** THE GAME OF SOC
CER IS PLAYED ONA SPECIALLY MARK
ED FIELD.IT IS  IMPORTANT THAT A
 NEW PLAYER     LEARNS WHAT THES
E MARKINGS ARE  FOR AND HOW THEY
 EFFECT THE GAME";
270 PRINT".I HOPE THIS PROGRAM W
ILL PROVE TO BE USEFUL.
        A PICTURE OF THE SOCCE
R FIELD   FOLLOWS,YOU WILL THEN
SEE A LISTOF FIELD FEATURES.PRES
SING THE  NUMBER KEY WILL TELL Y
OU MORE."
280 PRINT:PRINT"PRESS ANY KEY TO
 SEE THE FIELD."
290 I$=INKEY$
300 IF I$=""THEN290
310 IFI$<>""THENCLS:GOSUB10000
320 PRINT"          soccer field
        ":PRINT"(1)SIZE OF THE
 FIELD.          (2)THE GOAL.
                 (3)THE GOAL AR
EA.              (4)THE PENALTY
 SPOT.           (5)THE PENALTY
 AREA.           ";
330 PRINT"(6)THE CENTRE CIRCLE.
        (7)THE CORNERS."
340 PRINT:PRINT"PICK A SUBJECT A
ND PRESS THE    NUMBER KEY,OR PR
ESS 'E' TO END  THE PROGRAM,OR P
RESS 'R' TO     REPEAT THE PROGR
AM."
350 K$=INKEY$
360 IFK$=""THEN350
370 IFK$="1"THENSOUND200,2:GOSUB
1000
380 IFK$="2"THENSOUND200,2:GOSUB
2000
390 IFK$="3"THENSOUND200,2:GOSUB
3000
400 IFK$="4"THENSOUND200,2:GOSUB
4000
410 IFK$="5"THENSOUND200,2:GOSUB
5000
420 IFK$="6"THENSOUND200,2:GOSUB
6000
430 IFK$="7"THENSOUND200,2:GOSUB
7000
440 IFK$="E"THENCLS:PRINT@193,"N
OW YOU KNOW THE FIELD!!":PRINT@2
57,"HAVE A GOOD SOCCER SEASON!!"
:FORT=1 TO10000:NEXTT:CLS:END
450 IFK$="R"THENCLS:GOTO10
460 CLS:GOTO320
1000 FORN=1TO5
1010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
1020 FORT=1TO500:NEXTT
1030 LINE(4,16)-(252,176),PRESET
,B:SOUND150,4:FORT=1TO500:NEXTT
1040 NEXTN
1050 PCLS
1060 CLS
1070 SCREEN0,0
```

```
1080 PRINT"*********SIZE OF FIEL
D**********"
1090 PRINT" THE FIELD IS 50 TO 1
00 YARDS   WIDE,AND 100 TO 130 Y
ARDS LONG. THE LINES AT THE GOAL
 ENDS OF   THE FIELD ARE CALLED
GOAL LINES,AND THE LINES DOWN TH
E SIDES OF THE FIELD ARE CALLED
SIDELINES."
1100 PRINT"IF THE ATTACKING TEAM
 KICKS THE BALL OVER THE GOAL LI
NE THE     DEFENDING TEAM IS AWA
RDED A GOALKICK."
1105 PRINT:PRINT"PRESS ANY KEY T
O CONTINUE."
1106 B$=INKEY$
1107 IF B$=""THEN1106ELSE1110
1110 CLS:PRINT:PRINT"IF THE DEFE
NDING TEAM TOUCHES   THE BALL BE
FORE IT CROSSES THE  GOAL LINE T
HE ATTACKING TEAM IS AWARDED A C
ORNER KICK.              IF A PLAY
ER CAUSES THE BALL TOCROSS THE S
IDELINE,THE OPPOSING TEAM IS AWA
RDED A THROW-IN."
1111 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
1120 A$=INKEY$
1130 IFA$=""THEN1120ELSE RETURN
2000 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
2010 FORN=1TO5
2020 LINE(0,84)-(4,88),PRESET,BF
2030 LINE(0,104)-(4,108),PRESET,
BF
2040 LINE(0,84)-(0,108),PRESET
2050 LINE(256,84)-(256,108),PRES
ET
2060 LINE(256,104)-(252,108),PRE
SET,BF
2070 LINE(256,84)-(252,88),PRESE
T,BF
2080 FORT=1TO500:NEXTT
2090 LINE(0,84)-(4,88),PSET,BF
2100 LINE(0,104)-(4,108),PSET,BF
2110 LINE(0,84)-(0,108),PSET
2120 LINE(256,84)-(256,108),PSET
2130 LINE(256,104)-(252,108),PSE
T,BF
2140 LINE(256,84)-(252,88),PSET,
BF
2150 SOUND150,6
2160 NEXTN
2170 PCLS1
2180 CLS
2190 PRINT"***********THE GOAL*
**********"
2200 PRINT" THE GOALS ARE MADE O
F TWO      UPRIGHT GOALPOSTS AND
 A CROSSBAR.THEY CAN ONLY BE OF
WOOD OR    METAL,AND ARE PAINTED
 WHITE.     THE GOAL IS 8 YARDS(
24 FEET)   WIDE,AND 8 FEET HIGH.
           ";
2210 PRINT" THE USE OF NETS IS O
PTIONAL,   BUT CAN BE ENFORCED B
Y THE LOCALRULING BODY.
            FOR A GOAL TO BE SCO
RED,THE    BALL MUST BE COMPLETE
LY BEHIND  THE GOALINE."
2220 PRINT" PRESS ANY KEY TO CON
TINUE."
2230 I$=INKEY$
2240 IF I$=""THEN2230ELSERETURN
3000 FORN=1TO5
3010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
3020 PAINT(10,96),0,0
3030 PAINT(246,96),0,0
3040 SOUND150,4:FORT=1TO500:NEXT
T
3050 NEXTN
3060 PCLS1
3070 CLS
3080 PRINT"*********THE GOAL ARE
A**********"
3090 PRINT" THE GOAL AREA IS DIR
ECTLY IN    FRONT OF THE GOAL.IT
IS 6 YARDS DEEP,AND EXTENDS 6 YA
RDS EITHER SIDE OF THE GOAL,GIVI
NG IT A    TOTAL WIDTH OF 20 YAR
DS.        ";
3100 PRINT" WHILE IN THE GOAL AR
EA,THE     GOALKEEPER CAN ONLY B
E CHARGED  IF HE IS HOLDING THE
BALL,OR IF HE IS OBSTRUCTING AN
OPPOSING   PLAYER.":PRINT:PRINT"
 PRESS ANY KEY TO CONTINUE."
3110 I$=INKEY$
3120 IF I$=""THEN3110ELSE3130
3130 CLS:PRINT:PRINT" WHEN A GOA
L KICK IS TO BE TAKENTHE BALL MA
Y BE PLACED ANYWHERE IN THAT HAL
F OF THE GOAL AREA   NEARER TO W
HERE IT CROSSED THE  GOAL LINE."
3140 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
3150 K$=INKEY$
3160 IFK$=""THEN3150ELSERETURN
4000 FORN=1TO5
4010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
4015 FORT=1TO500:NEXTT
4020 CIRCLE(28,96),2,1
4030 CIRCLE(228,96),2,1
4040 SOUND150,6
4050 NEXTN
4060 PCLS1
4070 CLS
4080 PRINT"********THE PENALTY S
POT********"
4090 PRINT" THE PENALTY SPOT IS
LOCATED 12 YARDS FROM THE CENTRE
 OF THE    GOAL AND AT RIGHT ANG
LES TO IT.  THE BALL IS PLACED O
N THE SPOT WHEN THE ATTACKING TE
AM IS      AWARDED A PENALTY KIC
K."
4100 PRINT" WHILE THE KICK IS BE
ING TAKEN  ALL PLAYERS EXCEPT TH
E KICKER   AND THE GOALKEEPER MU
ST BE ON   THE FIELD,OUTSIDE THE
 PENALTY   AREA,AND AT LEAST 10
YARDS FROM THE BALL."
4105 PRINT" PRESS ANY KEY TO CON
TINUE."
4110 I$=INKEY$
4120 IF I$=""THEN4110ELSE4130
4130 CLS:PRINT:PRINT" THE GOALKE
EPER MUST STAND ON   HIS GOAL LI
NE (WITHOUT MOVING   HIS FEET) U
NTIL THE PENALTY     KICKER HAS
KICKED THE BALL.THE  KICKER CAN
NOT KICK THE BALL    AGAIN UNTIL
 IT HAS BEEN TOUCHED BY ANOTHER
PLAYER."
4140 PRINT" IF THE GOALKEEPER MO
VES HIS    FEET BEFORE THE BALL
IS KICKED, THE GOAL IS ALLOWED I
F SCORED.IFA GOAL IS NOT SCORED
THE KICK   MUST BE RETAKEN."
4150 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
4160 K$=INKEY$
4170 IFK$=""THEN4160ELSERETURN
5000 FORN=1TO5
5010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
5020 PAINT(10,96),0,0:PAINT(22,1
00),0,0
5030 PAINT(246,96),0,0:PAINT(230
,100),0,0
5040 SOUND150,6
5050 NEXTN
5060 PCLS1
5070 CLS
5080 PRINT"********THE PENALTY A
REA********"
5090 PRINT" THE PENALTY AREA IS
44 YARDS   WIDE BY 18 YARDS DEEP
.INSIDE    THIS AREA THE GOALKEE
PER CAN    HANDLE THE BALL.
           ON GOAL KICKS,THE BAL
L MUST BE  OUTSIDE THIS AREA BEF
ORE IT IS  IN PLAY."
5100 PRINT"IF A DEFENDING PLAYER
 COMMITS ANINTENTIONAL OFFENCE I
NSIDE THIS AREA,THE ATTACKING TE
AM IS      AWARDED A PENALTY KIC
K."
5110 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
5120 I$=INKEY$
5130 IFI$=""THEN5120ELSERETURN
6000 FORN=1TO6
6010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
6015 FORT=1TO500:NEXTT
6020 CIRCLE(128,96),20,1
6030 SOUND150,4
6040 NEXTN
6050 PCLS1
6060 CLS
6070 PRINT"*******THE CENTRE CIR
CLE********"
6080 PRINT" THE CENTRE CIRCLE HA
S A RADIUS OF 10 YARDS,AND IS CE
NTRED ON   THE HALFWAY LINE."
6090 PRINT" AT THE START OF EACH
 HALF OF   PLAY,AND AFTER A GOAL
 IS SCORED THE BALL IS PUT INTO
PLAY BY A  KICK-OFF.THE BALL IS
PLACED AT  THE CIRCLE'S CENTRE,A
ND ONE TEAMIS GIVEN THE BALL.";
6100 PRINT"THE OPPOSING  TEAM MU
ST STAY IN THE OTHER HALFOF THE
FIELD AT LEAST 10 YARDS  AWAY FR
OM THE BALL."
6110 PRINT" PRESS ANY KEY TO CON
TINUE."
6120 I$=INKEY$
6130 IFI$=""THEN6120ELSE6140
6140 CLS:PRINT:PRINT" THE BALL M
UST BE KICKED INTO   THE OPPONEN
T'S HALF OF THE FIELD.THE KICKER
 CAN NOT TOUCH THE   BALL A SECO
ND TIME UNTIL IT HAS BEEN TOUCHE
D BY ANOTHER PLAYER."
6150 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
6160 K$=INKEY$
6170 IFK$=""THEN6160ELSERETURN
7000 FOR N=1TO5
7010 PMODE4,5:COLOR0,1:SCREEN1,0
:PCOPY1TO5:PCOPY2TO6:PCOPY3TO7:P
COPY4TO8
7015 FORT=1TO500:NEXTT
7020 PAINT(5,17),0,0
7030 PAINT(251,17),0,0
7040 PAINT(251,175),0,0
7050 PAINT(5,175),0,0
7060 SOUND150,4:FORT=1TO500:NEXT
T
7070 NEXTN
7080 PCLS1
7090 CLS
7100 PRINT"***********THE CORNER
S**********"
7110 PRINT" THE CORNERS OF THE F
IELD ARE   MARKED BY AN ARC OF 1
 YARD      RADIUS,AND A FLAG OF
MINIMUM    HEIGHT 5 FEET."
7120 PRINT" IF A DEFENDING PLAYE
R TOUCHES  THE BALL BEFORE IT PA
SSES OVER  THE GOAL LINE,THE ATT
ACKING TEAMIS AWARDED A CORNER K
ICK.THE    BALL IS PLACED INSIDE
 THE CORNERMARKING BEFORE BEING
KICKED INTOPLAY."
7130 PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
7140 I$=INKEY$
7150 IFI$=""THEN7140ELSE7160
7160 CLS:PRINT:PRINT" THE DEFEND
ING PLAYERS MUST BE  AT LEAST 10
 YARDS AWAY FROM THE BALL UNTIL
IT IS KICKED.A GOAL  MAY BE SCOR
ED DIRECTLY FROM A   CORNER KICK
."
```

Continued on P 65 ...

32K
ECB

The Great Rainbow Simulation Package

Including World War II, Stock Car Racing And A Do-It-Yourself Simulation Generator For Writing Your Very Own Scenarios

One of the many advantages the computer has given us is the ability to make forecasts based upon the entry of facts or variables which we know will affect the outcome. In essence, we are using computers to predict the future.

Given our preoccupation with history and the computer's capacity to predict logical outcomes, wouldn't it be fasinating to see how history would have been altered if a different set of circumstances had occurred?

There are two special Simulations which allow you to do just that - World War II, a serious program which allows you to control major factors affecting that global conflict; and Stock Car, a fun Simulation of the Trans-South Stock Car Race in Darlington, S.C. Also included is an innovative Simulation generator which allows you to create your own scenarios.

What If? — A World War II Simulation

In this the 40th year since the conclusion of World War II, it is appropriate to reflect upon the global conflict which touched more countries, caused more destruction and death, cost more money, and had more far-reaching effects than any war in the history of humankind.

The map of Europe was changed forever because of the maniacal tendencies of Adolf Hitler, the central figure in world history from 1939 to 1945. Under Hitler's leadership, Germany intended to create the mightiest empire the world had ever seen. Initial plans called for the occupation of adjoining territories - Poland, Denmark, Norway, Belgium, Luxembourg and The Netherlands. Then, after overrunning France, Hitler expected to use air assults to force Britain to make "peace". German troops would then defeat Russia, capture the oil fields in the Middle East, and Hitler would implement his grand plan for a "New Order".

We are left to wonder what the final outcome would have been if German troops had followed through at Dunkirk in 1942 when the British forces appeared to be hopelessly trapped at the Belgium port. But for some inexplicable reason, the Germans waited, giving the Allies the chance to make their desperate escape.

What if the United States had entered the war two years earlier? Would the war have been shortened, saving countless millions of lives? What if the United States, still healing its wounds from the first world war, had never even entered the European conflict and instead concentrated its efforts in the Asian theater?

What if Hitler had not started a two-front war by attacking the Soviet Union? What if Russia had sensed Hitler's ambitions and launched an all-out assault on Germany earlier in the war. The Russians with their superior numbers would surely have conquered Germany. What if the Russians, fresh from victory, had implemented their version of a master plan and attempted to conquer Europe for itself?

What if the Allied invasion of North Africa had failed?

The possibilities are as intriguing as they are staggering in their perceived footprints on the modern history of Europe.

In this Simulation, let's presume you are the supreme commander of all of the Allied and Axis forces. Hitler would envy you because you have the power to change the goals of any country, even his. You have control over the goals of all ground, naval and aerial forces, as well as their supplies and even the morale of the troops.

We assume you will use your newly found powers more constructively than Hitler did, but that's really up to you.

The Allied combatants include the American, British and French forces. The Soviets originally sided with Germany but joined the Allies after being invaded by Germany in 1941. The Axis powers consist of the German, Italian and Polish armies. Actually 50 countries joined the Allies, and nine formed the Axis, but we've included only the major powers for programming efficiency.

As the Simulation begins, you are given four options:

I — Initialize the scenario. This enters the variables for historical events as they actually occurred. You should choose this option the first time you run the program — unless you want to assign a new set of variables for every single phase of the Simulation.

G — A graphics screen showing the European Theater. Countries occupied by Axis powers are shown in yellow, the Allies in green. As the war progresses, the colors change.

A — Advance the scenario. To move to the next stage of the war, which moves along in three-month sequences.

V — To view or change conditions. This is the most important option in the Simulation because this is where you can make vital decisions affecting the outcome of the war.

This option includes a submenu which allows you to view all of the tables one after another by pressing the 'A' key, or to check the status of a specific cell by pressing the 'S' key.

COMBATANTS
(2nd cell — Y values)

0 American
1 British
2 Soviet
3 French
4 German
5 Italian
6 Polish

REGIONS
(1st cell — X values)

0 Atlantic Ocean
1 Mediterranean
2 British Isles
3 France
4 Germany
5 Italy
6 U.S.S.R.
7 Slavic countries
8 Scandinavia
9 Poland
10 Spain
11 North Africa
12 Palestine

FORCES
(3rd cell — B values)

0 Ground troops
1 Naval forces
2 Air forces
3 Troop morale
4 Supplies

If you want a sequential listing, the tables are displayed first by regions (see the Region Table). A status report extending through five screens will give the number of a country's ground troops, naval forces, air force squadrons, and supply units in that region, as well as the morale level of the troops.

On each screen, you are given the options of (C)hange, (K)eep and (M)enu. If you wish to make a change, simply enter any number from zero to 255. If you keep things as they are, you advance to the next screen. This process continues until you have either completed your inspection of all 13 regions or have returned to the main menu.

When you want to review or make a change in a specific table, the 13 regions are displayed. Just enter the number of the region in which you want to make a change. Next, the seven participating countries (see Combatants Table) are shown. Enter the number of the appropriate country. Finally, you will see a display of the types of military forces (see Forces Table), as well as supplies and morale. Simply press 'C' and change the number that applies.

For exmple, if you want to review the status of German ground troops in France, type 3 for France on the Region screen, 4 for German on the Combatants screen, and zero for ground troops on the Forces screen. Here you could greatly diminish the number of German troops, improving France's odds of withstanding an attack.

You are then returned to the main menu where you should select the option of (A)dvance scenario. Before any conflict occurs, however, you will be shown the goals of the combatants. To have a significant impact on the outcome of the Simulation, you must change a country's overall goals here. The options include remaining neutral, defending all positions, or invading one of the 13 regions.

Of course, if you don't change the goals of any country or order changes in troop development, the war probably will unfold as it actually happened. Why probably? The laws of propagation — or who wins a battle, how many troops are lost, changes in morale, maintenance of supply lines, etc. — are governed by rules that have a slight dependence on probability. It is not possible to program all weather conditions, human decisions, or the mistakes and lucky breaks over a period of six years into a 32K or even a 64K computer.

Programming Information

A subroutine contained in lines 5620-5681 determines the initial goals for each country. When you choose to change goals, the flag (FL) is reset signaling a change in the program. If you merely change the distribution of forces or supplies from the main menu, the flag remains unchanged so goals stay the same, but with a different outcome. After all, could the Allies have captured North Africa if you moved all the troops to Denmark to begin the invasion in 1942?

New goals are determined by the program commands in lines 5571-5672. The basis for these goals was the determination of the Allies to defend themselves or to recapture occupied territory. The Axis' goals were nothing short of world domination.

The 28000 block of data in the Simulation generator (see accompanying story) contains descriptive data for "who has what where." The 'X' index is the region, containing 13 of them from X=0 (Atlantic Ocean) to X=12 (Palestine). The 'Y' index is the countries' forces, including seven nations — from Y=0 (United States) to Y=6 (Poland). Types of military forces are governed by the 'B' index, from B=0 for ground troops to B=4 for supplies.

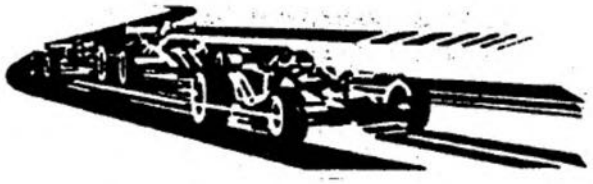
Other Variables

The B(X) array keeps track of who controls what region. 'X' is the same order as Index X (i.e., 0=Atlantic Ocean to 12=Palestine). If the value of B(X) equals zero, the Allies control it. If B(X) equals one, then it is controlled by the Axis.

The A(Y,15) array is a bit more complicated. The first index, 'Y', is the same as Index Y that indicates a country's forces. The second index tells something about that country. For example, A(Y,0) is the loyalty — zero equals Allies, one equals Axis. Remember, the Soviets changed loyalty when they were invaded by the Germans in 1941. A(Y,1) is the current goal of a country: 100 means to defend all positions; 999 means to remain neutral; 1000 means that the country is under foreign domination; zero through 12 means invade a region. A(Y,2) to A(Y,10) are not used. A(Y,11) is the industrial strength of a nation; A(Y,12) is the technological strength; A(Y,13) is the material strength; A(Y,14) is the energy availability; and A(Y,15) is the morale of the country's troops.

Good luck, and don't blame us if Monty and Patton don't get along!

VRRROOOM!
VRRROOOM!



Start your engines, ladies and gentlemen. You're about to experience 500 miles of grueling stock car racing in an exciting Simulation of the Trans-South Stock Car Race - one of the key races on the NASCAR circuit - run in Darlington, S.C.

You'll control the fates of such fierce competitors as Darrell Waltrip, Buddy Baker and Bill Elliott, along with 12 other serious contenders. Elliott actually won this year's race in April before a crowd in excess of 50,000.

There are 367 laps around the 1.3 mile track. At speeds of approximately 140 to 157 miles per, it takes about three hours to complete the race.

Since three hours is an eternity when you are sitting in front of a keyboad, we think you'll want to get involved. Consider yourself the crew chief of all 15 cars, with the ability to determine car speeds, location on track, amount of fuel, the condition of the various parts of the cars, and when pit stops are necessary.

The main menu will display four options:

I - Initialize the scenario, entering all variables for the race as it actually happened. If you do not initialize the program, you will have to enter new values for every phase of the Simulation.

G - A graphics screen showing the leaders' scoreboard. As the race evolves, the positions of the drivers change.

A - Advance the scenario. Watch the race which progresses at 30-second intervals.

V - View or change conditions. This is where you come in as crew chief. When you select this option, a submenu is displayed giving you a choice of viewing all cells (or tables) sequentially or changing a specific cell.

LOCATION
(1st cell — A values)

1 Front straightaway
2 Turn one
3 Turn two
4 Back stretch
5 Turn three
6 Turn four
7 In pits

DRIVERS
(2nd cell — Y values)

0 Bill Elliott/Ford Thunderbird
1 David Pearson/Chevy Monte Carlo SS
2 Terry Labonte/Chevy Monte Carlo SS
3 Benny Parsons/Chevy Monte Carlo SS
4 Lake Speed/Pontiac Grand Prix
5 Ron Bouchard/Buick Regal
6 Neil Bonnett/Buick Regal
7 Dale Earnhardt/Chevy Monte Carlo SS
8 Ricky Rudd/Ford Thunderbird
9 Buddy Baker/Oldsmobile Cutlass Supreme
10 Geoff Bodine/Chevy Monte Carlo SS
11 Harry Gant/Chevy Monte Carlo SS
12 Darrell Waltrip/Chevy Monte Carlo SS
13 Dave Marcis/Chevy Monte Carlo SS
14 Tim Richmond/Pontiac Grand Prix

CONDITION
(3rd cell — B values)

0 Location on track
1 Tires
2 Brakes
3 Transmission
4 Engine
5 Fuel remaining
6 Current position
7 Pit on next lap?

If you choose the sequential listing, you will be shown the status of all 15 drivers (see Drivers Table), the operational status of their cars (Condition Table), where they are located on the track (Location Table) and their position in the race.

Let's say you are in the sequential listing and you want to give David Pearson more fuel. When the status of Pearson's fuel is shown, you simply add as many gallons as you want, up to a capacity of 25. If you want to change his location on the track, type in any number corresponding to those in the Location Table.

If you want to change his position in the actual standings, type in any number from zero to 15. You probably will have to change a number of other facts in order for Pearson to hold on to the lead because he will probably run low on fuel later or need new tires, for example.

If you want to change a specific cell to give Pearson more fuel, you would enter '1' from the Drivers Table (or the second cell), and '5' from the Condition Table (or third cell) and fill his tank.

Programming Information

A block of one-byte variables begins at memory byte 28000 in the Simulation generator. The 'X' index is always zero; it is not used in this application. The 'Y' index is the driver, using numbers from zero to 14 in their starting positions.

The 'B' index is as follows:

B=0 — track location
B=1 — tire condition, zero to 255 (If too low, an accident may occur.)
B=2 — brake conditon
B=3 — gearbox (or transmission)
B=4 — engine condition

B=5 — fuel level in tenths of gallons (A total of 145 means the driver has 14.5 gallons. The cars average four miles per gallon.)

B=6 — current position in the race (one to 15)

B=7 — pit flag (zero for no; one for yes) This will change automatically if the driver's crew wants him to stop on the next lap.

Variables that cannot be changed while running the program are contained in the A(Y,5) array. The 'Y' index is the driver of the car. The second index indicates: zero for miles traveled, one for current speed, two for current lap, three for top possible speed, four for probable crash lap, and five for time remaining in pits. If A(Y,5) equals 1000, then the car is out of the race.

Want To Drive?

That's easy. Just substitute your name in lines 5102 or 5104. If you want a shorter race, change the lap number variable (LS) in Line 5100. If you want a different track, change the lap length variable (LP) in Line 5000 (the Daytona 500, for example, would be LP=2.5:LS=200).

We would wish you good luck, but since you control all the variables you should do quite well!

Loading Instructions

If you are keying the listings in by hand, key in Listing 1 and then key in either Listing 2 *(WORLDWAR)* or Listing 3 *(STOCKCAR)* with Listing 1 still in the computer. Once you've first typed in and saved one complete program, type DEL 5000 and ENTER to get rid of the scenario, then type in the second scenario.

If you have RAINBOW ON TAPE, just CLOAD either *WORLDWAR* or *STOCKCAR* to load in the entire Simulation.

When you're ready to write your own scenarios, load in one of the complete Simulations, type DEL 5000- and ENTER to delete its scenario and type in your new one.

```
180 .....28
630 .....247
680 .....119
END .....99
```

Listing 1:

```
4 CLEAR 1000,28000
5 DIM X$(15),Y$(15),B$(15),A$(30),A(15,15),B(15)
40 AQ=28000:FLAG=0
50 GOSUB 900 'LOGO
100 CLS(3)
104 GOTO 110
105 PRINT CHR$(23);STRING$(30,CHR$(175));CHR$(23);:RETURN
106 FOR I=1 TO 50:NEXT:RETURN
110 PRINT@0,STRING$(32,CHR$(23));
120 GOSUB 105
130 IF FLAG=0 THEN GOSUB 5100:FLAG=1
132 GOSUB 105:GOSUB 105
135 IF FL=0 THEN FOR I=1 TO 8:GOSUB 105:NEXT:GOTO 180 ELSE 140
140 PRINTCHR$(23);"  G  PRINT GRAPHICS          ";CHR$(23);:GOSUB 105:GOSUB 105
150 PRINTCHR$(23);"  V  VIEW OR CHANGE CONDITIONS";CHR$(23);:GOSUB 105:GOSUB 105
170 PRINTCHR$(23);"  A  ADVANCE SCENARIO        ";CHR$(23);:GOSUB 105
180 IF FLAG=1 THEN PRINT CHR$(23);"  I  ENTER INITIAL DATA <- <- ";CHR$(23);ELSE GOSUB 105
190 GOSUB 105
195 PRINT STRING$(32,CHR$(23));
200 GOSUB 106:PRINT@480,"        - CHOOSE OPTION -        ";:GOSUB 106:PRINT@480,STRING$(31," ");:K$=INKEY$:IF K$="" THEN 200
210 IF K$="A" THEN 500 ELSE IF K$="V" THEN 600 ELSE IF K$="I" THEN 850 ELSE IF K$="G" THEN 950 ELSE 200
220 FOR I=1 TO 8:GOSUB 105:NEXT:GOTO 180
500 PRINT@480,"  ADVANCING SIMULATION";:GOSUB 5300:T=T+1:GOTO 100
600 CLS:PRINT:PRINT"VIEW ALL SEQUENTIALLY (A)          OR ONE SPECIFIC CELL (S)"
601 K$=INKEY$:IF K$="" THEN 601 ELSE IF K$="A" THEN 602 ELSE IF K$="S" THEN 670 ELSE 601
602 FOR X=0 TO IX:FOR Y=0 TO IY:FOR B=0 TO IB:GOSUB 608:NEXT B,Y,X:GOTO 100
608 GOSUB 990:Q=PEEK(AQ+AD)
610 GOSUB 5900 'TRANSLATION
620 CLS:PRINT:PRINT X$(X):PRINT"  ";Y$(Y):PRINT "    ";B$(B):PRINT
630 IF TF=0 THEN PRINT Q ELSE PRINT A$(TF)
631 PRINT
635 PRINT"  K  = KEEP":PRINT"  C  = CHANGE":PRINT"  M  = MENU"
640 K$=INKEY$:IF K$="" THEN 640 ELSE IF K$="M" THEN 655 ELSE IF K$="C" THEN 700 ELSE IF K$="K" THEN 650 ELSE 640
650 RETURN
655 X=IX:Y=IY:B=IB:RETURN
660 GOTO 100
670 CLS:PRINT:PRINT"ENTER FIRST CELL VALUE":FOR X=0 TO IX:PRINT X;"=";X$(X):FOR I=1 TO 150:NEXT:NEXT X
671 INPUT X:IF X<0 OR X>IX THEN PRINT"INVALID ENTRY ... TRY AGAIN":GOTO 670 ELSE 675
675 PRINT:PRINT"ENTER SECOND CELL VALUE":FOR Y=0 TO IY:PRINT Y;"=";Y$(Y):FOR I=1 TO 150:NEXT:NEXT Y
676 INPUT Y:IF Y<0 OR Y>IY THEN PRINT"INVALID ENTRY ... TRY AGAIN":GOTO 675 ELSE 678
678 PRINT:PRINT"ENTER THIRD CELL VALUE":FOR B=0 TO IB:PRINT B;"=";B$(B):FOR I=1 TO 150:NEXT:NEXT B
679 INPUT B:IF B<0 OR B>IB THEN PRINT"INVALID ENTRY ... TRY AGAIN":GOTO 678 ELSE 680
680 GOSUB 608:GOTO 100
700 GOSUB 990:Q=PEEK(AQ+AD)
710 GOSUB 5900 'TRANSLATION
715 CLS:PRINT:PRINT"CURRENT VALUE OF CELL          ";X;",";Y;","
;B:PRINT PEEK(AQ+AD);" WHICH MEANS:"
720 PRINT:PRINT X$(X):PRINT"  ";Y$(Y):PRINT "    ";B$(B):PRINT
730 IF TF=0 THEN PRINT Q ELSE PRINT A$(TF)
731 PRINT
740 PRINT"ENTER NEW VALUE":INPUT Q:IF Q<0 OR Q>255 THEN PRINT "INVALID ENTRY ... SEE MANUAL":GOTO 740 ELSE POKE AQ+AD,Q
750 GOTO 650
850 PRINT@480,"  INITIALIZING SIMULATION  ";:GOSUB 5200:GOTO 1000
900 PMODE 3,1:PCLS(2):SCREEN 1,0:COLOR3,2
910 DRAW"BM42,60;D28E12F12U28;BR12;D28U16R16D16;BR16;H4U8E4R8F4D8G4L8R8E4F4;BR16;U16L6R12L6U8D24R10;BD12;L4R8L4D28L4R8;BR16;U16L4R8L4U8E4F4;BR20;D4U4E4R4F4D4G6D6;BD6;F2L4E2"
920 COLOR 4,2:DRAW"BM16,16;R224D160L224U160;BF16;R192D128L192U128":PAINT(0,0),3,4:PAINT(20,20),4,4
930 FOR K=1 TO 6:GOSUB 991:NEXT:RETURN
950 IF FLAG<>0 THEN 951 ELSE 1000
951 GOSUB 5000:GOTO 1000
990 AD=B+Y*(IB+1)+X*(IB+1)*(IY+1):RETURN
991 FOR I=1 TO 1700:NEXT:RETURN
```

```
5060 .....139
5110 .....31
5230 .....188
5292 .....13
5317 .....227
5414 .....117
5442 .....130
5460 .....174
5478 .....192
5532 .....120
5579 .....224
5627 .....202
5650 .....43
5710 .....109
5820 .....168
5914 .....68
END .....18
```

Listing 2: WORLDWAR

```
5000 PMODE 3,1:PCLS(3):SCREEN 1,0:COLOR1,3
5010 DRAW"BM0,155;E10R5E5U8R3F2R9F6R40F5R10;D3F4G7D5F7R10F20D5R5R7U3L2U10E5R10D5R4D3R10D5R40E10U10R3U10L3U10L8G8L6U3L6D4L6U3L10H6U10L3U3"
5020 DRAW"E2R10U5;L3U2R12U5R10F1 0R10E5R10U5F5R20L20H10L5H5L5H10U2R5E10L10D3L5D3L5D5F5D3L10U10L10"
5021 DRAW"H10L36G16E16U27D27R15E15U20H16F16D20G15R21F10;G5D15L5D10;L15D5R3D12R2D3L8U5H10U3H5U10H7L5"
5030 DRAW"H15L10D5F15R5D3F12L4H5L3D5R3D6G8L5U2E4U8H10L3H5L5H8U4;L5E10U28D28R12D5U5L12G10;G5L5U3L5D4L10G10D5R3D2L6D3;G4L4U2L8G3L5"
5040 DRAW"U8H5L8E8H8U5E10U5R10D3R8F22H7U5R2U5R2U10H8L4U6R5F10U10;R2D4R4D5R10E8R10U8R16E5G5U10L4U5E5R9D5G4D8R7D5R20;U10E5U3R3D7R5U10L2U8"
5050 DRAW"R15E7U2;L10E10U12D12G10L10;H8U7L2D10F5D5G10D10G5L2U2H8U20L3G10L5U10H3E4H5E6H2E5H4E7H8E3"
5060 DRAW"BM40,57;R7D4E4R3F4E4R3F4R7U2L3U2E8U3L5U3R3U10L3U8L2U3R4E8L4U2E4L10H3G3D3G5R5G5R5D5G3D7L5D2G7L5D2F4D2L3D2L9D2"
5070 DRAW"BM31,40;R4E2F2R4E3U6E4U6H3L4G4L8G6D2F3G3D3R3D2;BM108,140;L10F7U4R3U3"
5080 PAINT(56,45),B(2)+1,1:PAINT(36,35),B(2)+1,1:PAINT(70,85),B(3)+1,1
5081 PAINT(100,70),B(4)+1,1:PAINT(100,110),B(5)+1,1:PAINT(250,0),B(6)+1,1
5082 PAINT(140,110),B(7)+1,1:PAINT(120,10),B(8)+1,1:PAINT(104,40),B(8)+1,1
5083 PAINT(140,60),B(9)+1,1:PAINT(20,100),B(10)+1,1:PAINT(0,180),B(11)+1,1
5084 PAINT(250,180),B(12)+1,1:PA
```

```
INT(1Ø5,142),B(5)+1,1
5Ø99 K$=INKEY$:IF K$="" THEN 5Ø9
9 ELSE RETURN
51ØØ REM SET UP SCENARIO RULES
51Ø1 IX=12:IY=6:IB=4
51Ø2 Y$(Ø)="AMERICAN":Y$(1)="BRI
TISH":Y$(2)="SOVIET":Y$(3)="FREN
CH":Y$(4)="GERMAN":Y$(5)="ITALIA
N":Y$(6)="POLISH"
51Ø5 X$(Ø)="ATLANTIC OCEAN":X$(1
)="MEDITERRANEAN":X$(2)="BRITISH
 ISLES":X$(3)="FRANCE":X$(4)="GE
RMANY":X$(5)="ITALY":X$(6)="U.S.
S.R":X$(7)="SLAVIC COUNTRIES":X$
(8)="SCANDINAVIA":X$(9)="POLAND"
:X$(1Ø)="SPAIN":X$(11)="NORTH AF
RICA":X$(12)="PALESTINE"
511Ø B$(Ø)="GROUND TROOPS":B$(1)
="NAVAL FORCES":B$(2)="AIR FORCE
S":B$(3)="TROOP MORALE":B$(4)="S
UPPLIES"
512Ø A$(1)="GOOD MORALE":A$(2)="
POOR MORALE"
5199 RETURN
52ØØ REM SET UP INITIAL SCENARIO
 CONDITIONS
52Ø1 T=Ø:FLAG=2:FOR I=Ø TO 7:FOR
 J=Ø TO 15:A(I,J)=1ØØ:NEXT J,I
52Ø2 A(Ø,1)=999:A(1,1)=1ØØ:A(2,Ø
)=1:A(2,1)=9:A(3,1)=1ØØ:A(4,Ø)=1
:A(4,1)=9
52Ø3 A(5,Ø)=1:A(5,1)=1ØØ:A(6,1)=
1ØØ:A(Ø,Ø)=Ø:A(1,Ø)=Ø:A(3,Ø)=Ø:A
(6,Ø)=Ø'LOYALTY AND GOALS
521Ø FOR I=Ø TO 15:B(I)=Ø:NEXT:B
(4)=1:B(5)=1:B(6)=1:B(11)=1 'REG
ION CONTROL
522Ø FOR I=Ø TO 6:FOR J=11 TO 15
:READ Q:A(I,J)=Q:NEXT J,I
5225 FOR X=Ø TO IX:FOR Y=Ø TO IY
:FOR B=Ø TO IB:GOSUB 99Ø:POKE AQ
+AD,Ø:NEXT B,Y,X
523Ø X=Ø:Y=Ø:GOSUB 525Ø
5231 X=Ø:Y=1:GOSUB 525Ø
5232 X=Ø:Y=4:GOSUB 525Ø
5233 X=1:Y=1:GOSUB 525Ø
5234 X=1:Y=4:GOSUB 525Ø
5235 X=1:Y=5:GOSUB 525Ø
5236 X=2:Y=Ø:GOSUB 525Ø
5237 X=2:Y=1:GOSUB 525Ø
5238 X=3:Y=3:GOSUB 525Ø
5239 X=4:Y=4:GOSUB 525Ø
524Ø X=5:Y=5:GOSUB 525Ø
5241 X=6:Y=2:GOSUB 525Ø
5242 X=9:Y=6:GOSUB 525Ø
5243 X=9:Y=4:GOSUB 525Ø
5244 X=9:Y=2:GOSUB 525Ø
5245 X=11:Y=4:GOSUB 525Ø
5246 X=11:Y=5:GOSUB 525Ø
5247 X=12:Y=1:GOSUB 525Ø
5248 GOTO 5299
525Ø FOR B=Ø TO IB:READ Q:GOSUB
99Ø:POKE AQ+AD,Q:NEXT:RETURN
529Ø DATA 1Ø,1Ø,6,8,6,7,8,7,9,7,
5,6,8,7,5,5,6,8,4,7,1Ø,1Ø,1Ø,5,8
,4,6,7,3,6,2,3,4,1,3
5291 DATA 1ØØ,25,5,6,5,Ø,2Ø,3,8,
5,Ø,35,3,9,7,Ø,17,2,6,4,Ø,2Ø,4,8
,5,Ø,5,1,5,9,2Ø,8,5,6,8,1ØØ,16,2
5,8,1Ø,6Ø,Ø,6,7,1Ø,21Ø,2Ø,25,1Ø,
1Ø
5292 DATA 14Ø,1Ø,7,8,1Ø,18Ø,Ø,1Ø
,7,1Ø,17,Ø,2,9,1Ø,6Ø,Ø,19,8,8,2Ø
,Ø,11,6,8,11,Ø,7,4,3,3,Ø,Ø,5,6,1
9,Ø,7,7,4
5299 RETURN
53ØØ GOSUB 593Ø 'CALC TIME
53Ø1 GOSUB 58ØØ:IF A(4,1)=6 THEN
 A(2,Ø)=Ø 'CHANGE GOALS
53Ø2 FOR C=Ø TO 6:CLS:K=A(C,1):I
F K=999 THEN 5315 ELSE IF K=2ØØ
THEN 5312 ELSE IF K=1ØØ THEN 53Ø
6 ELSE IF (K>=Ø AND K<=11) THEN
532Ø ELSE 534Ø
53Ø6 Y=C:PRINT@128,Y$(C)+" TROOP
S DEFENDING          ALL LINES"
:GOSUB 991:FOR X=Ø TO 12
53Ø7 B=Ø:Z=15:GOSUB 5316:B=1:Z=1
1:GOSUB 5316:B=2:Z=14:GOSUB 5316
:B=4:Z=13:GOSUB 5316
53Ø8 NEXT X:GOTO 534Ø
531Ø PRINT@128,Y$(C)+" SIDE":PRI
NT"REMAINS NEUTRAL":GOSUB 991:GO
TO 534Ø
5312 PRINT@128,Y$(C)+" FORCES NE
UTRALIZED":PRINT"BECAUSE IT IS U
NDER FOREIGN      DOMINATION":GOS
UB 991:GOSUB 991:GOTO 534Ø
5316 GOSUB 99Ø:Q=PEEK(AQ+AD):W=I
NT(((A(Y,Z)*.Ø5)+1)*Q)
5317 IF Q<>Ø AND W<(Q+1) THEN W=
Q+1
5318 IF W>255 THEN W=255
5319 POKE AQ+AD,W:RETURN
532Ø PRINT@128, Y$(C)+" FORCES M
OVING TROOPS          AND SUPPLI
ES INTO ":PRINT X$(K):GOSUB 991
5325 Y=C:X=K:IF X<2 THEN 533Ø
5327 B=Ø:GOSUB 5338:B=2:GOSUB 53
38:B=4:GOSUB 5338:GOTO 5335
533Ø B=1:GOSUB 5338:B=4:GOSUB 53
38:GOTO 5335
5335 B=3:GOSUB 99Ø:POKE AQ+AD,1Ø
:GOTO 534Ø
5338 GOSUB 99Ø:Q=PEEK(AQ+AD)+12:
IF Q>255 THEN Q=255
5339 POKE AQ+AD,Q:RETURN
534Ø NEXT C
54ØØ 'LOOP THRU REGIONS AND FIND
 IF INVASION IS TAKING PLACE
54Ø1 FOR X=Ø TO IX:AK=-1 'INVADE
 FLAG
54Ø2 FOR C=Ø TO IY
54Ø4 IF (A(C,1)<>X OR A(C,Ø)=B(X
))THEN 541Ø
54Ø6 AK=A(C,Ø)  'WHO IS INVADING
  Ø=ALLIED 1=AXIS
54Ø8 C=IY
541Ø NEXT C
5412 IF AK<Ø THEN 5599  'NO INVA
SION IN THIS REGION
5414 FØ=Ø:F1=1:FOR Y=Ø TO IY
5415 'DETERMINE FIGHT FACTORS
5416 IF X=Ø OR X=1 THEN 543Ø
5418 IF A(Y,Ø)=Ø THEN 542Ø ELSE
5425  'GROUND BATTLE
542Ø B=Ø:GOSUB 99Ø:F=PEEK(AQ+AD)
:B=2:GOSUB 99Ø:F=F+PEEK(AQ+AD):B
=4:GOSUB 99Ø:F=F+PEEK(AQ+AD):B=3
:GOSUB 99Ø:F=F*PEEK(AQ+AD):FØ=FØ
+F:GOTO 5439
5425 B=Ø:GOSUB 99Ø:F=PEEK(AQ+AD)
:B=2:GOSUB 99Ø:F=F+PEEK(AQ+AD):B
=4:GOSUB 99Ø:F=F+PEEK(AQ+AD):B=3
:GOSUB 99Ø:F=F*PEEK(AQ+AD):F1=F1
+F:GOTO 5439
543Ø IF A(Y,Ø)=Ø THEN 5432 ELSE

5435 'SEA BATTLE
5432 B=1:GOSUB 99Ø:F=PEEK(AQ+AD)
:B=4:GOSUB 99Ø:F=F+PEEK(AQ+AD):B
=3:GOSUB 99Ø:F=F*PEEK(AQ+AD):FØ=
FØ+F:GOTO 5439
5435 B=1:GOSUB 99Ø:F=PEEK(AQ+AD)
:B=4:GOSUB 99Ø:F=F+PEEK(AQ+AD):B
=3:GOSUB 99Ø:F=F*PEEK(AQ+AD):F1=
F1+F:GOTO 5439
5439 NEXT Y
544Ø IF AK=Ø THEN F1=F1*1.2 ELSE
 FØ=FØ*1.2  ' MULTIPLIER FOR DEF
ENSIVE TROOPS
5442 IF FØ>3*F1 THEN 545Ø ELSE I
F F1>3*FØ THEN 546Ø ELSE IF FØ>1
.5*F1 THEN 547Ø ELSE IF F1>1.5*F
Ø THEN 548Ø
5443 PRINT"** DIFFICULT FIGHTING
 IN          ";X$(X):PRINT"INVASI
ON FAILED, HEAVY LOSSES   ON BOT
H SIDES":GOSUB 991 'CONTROL DOES
 NOT CHANGE
5444 FOR Y=Ø TO IY:FOR B=Ø TO IB
:GOSUB 99Ø:Q=PEEK(AQ+AD):POKE AQ
+AD,INT(.6*Q):NEXT B
5446 A(Y,15)=A(Y,15)*.9:NEXT Y
5448 CT=-1:GOTO 5499
545Ø PRINT"ALLIES HAVE MAJOR VIC
TORY IN    ";X$(X):GOSUB 991
5452 CT=-1:IF B(X)=1 THEN CT=Ø
5453 FOR Y=Ø TO IY
5454 IF A(Y,Ø)=1 THEN 5456
5455 FOR B=Ø TO 3:GOSUB 99Ø:POKE
 AQ+AD,INT(.9*PEEK(AQ+AD)):NEXT
B:B=4:GOSUB 99Ø:POKE AQ+AD,1+PEE
K(AQ+AD):A(Y,15)=A(Y,15)+2:GOTO
5459
5456 FOR B=Ø TO 4:GOSUB 99Ø:POKE
 AQ+AD,Ø:NEXT B:A(Y,13)=A(Y,13)-
2:IF A(Y,13)<Ø THEN A(Y,13)=Ø
5457 A(Y,15)=A(Y,15)-2:IF A(Y,15
)<Ø THEN A(Y,15)=Ø
5459 NEXT Y:GOTO 5499
546Ø PRINT"AXIS HAS MAJOR VICTOR
Y IN          ";X$(X):GOSUB 991
5462 CT=-1:IF B(X)=Ø THEN CT=1
5463 FOR Y=Ø TO IY
5464 IF A(Y,Ø)=1 THEN 5466
5465 FOR B=Ø TO 3:GOSUB 99Ø:POKE
 AQ+AD,INT(.9*PEEK(AQ+AD)):NEXT
B:B=4:GOSUB 99Ø:POKE AQ+AD,1+PEE
K(AQ+AD):A(Y,15)=A(Y,15)+2:GOTO
5469
5466 FOR B=Ø TO 4:GOSUB 99Ø:POKE
 AQ+AD,Ø:NEXT B:A(Y,13)=A(Y,13)-
2:IF A(Y,13)<Ø THEN A(Y,13)=Ø

5467 A(Y,15)=A(Y,15)-2:IF A(Y,15
)<Ø THEN A(Y,15)=Ø
5469 NEXT Y:GOTO 5499
547Ø IF RND(-1)>.8 THEN 5479 ELS
E 549Ø
5471 CT=-1:IF RND(-1)>.8 AND B(X
)=1 THEN CT=Ø
5472 FOR Y=Ø TO IY:FOR B=Ø TO IB
:GOSUB 99Ø
5473 IF A(Y,Ø)=Ø THEN POKE AQ+AD
,INT(.8*PEEK(AQ+AD)) ELSE POKE A
Q+AD,INT(.5*PEEK(AQ+AD))
5474 NEXT B,Y
5475 GOTO 5499
5478 PRINT"ALLIES ARE OUTNUMBERE
D IN          ";X$(X);" BUT STIL
L WINS          MAJOR BATTLE":G
OSUB 991:GOTO 5471
5479 PRINT"AXIS IS OUTNUMBERED I
N          ";X$(X);" BUT STILL W
INS          MAJOR BATTLE":GOSU
B 991:GOTO 5481
548Ø IF RND(-1)>.8 THEN 5478 ELS
E 5491
5481 CT=-1:IF RND(-1)>.8 AND B(X
)=Ø THEN CT=1
5482 FOR Y=Ø TO IY:FOR B=Ø TO IB
:GOSUB 99Ø
5483 IF A(Y,Ø)=1 THEN POKE AQ+AD
,INT(.8*PEEK(AQ+AD)) ELSE POKE A
Q+AD,INT(.5*PEEK(AQ+AD))
5484 NEXT B,Y
5485 GOTO 5499
549Ø PRINT"ALLIES WIN BATTLES IN
          ";X$(X):GOSUB 991:
GOTO 5471
5491 PRINT"AXIS WINS BATTLE IN
          ";X$(X):GOSUB 991:GO
TO 5481
5499 IF CT<Ø THEN 5599
55ØØ 'CONTROL IN X HAS CHANGED T
O CT  RESET GOALS
551Ø B(X)=CT:PRINT X$(X);" FALLS
 UNDER ":IF CT=Ø THEN PRINT"ALLI
ED CONTROL" ELSE IF CT=1 THEN PR
INT"AXIS CONTROL"
552Ø ON X+1 GOTO 557Ø,557Ø,5532,
5533,5534,5535,5536,557Ø,557Ø,55
39,557Ø,557Ø,5542
5532 IF CT=1 THEN Y=1:GOSUB 559Ø
:GOTO 557Ø
5533 IF CT=1 THEN Y=3:GOSUB 559Ø
:GOTO 557Ø
5534 IF CT=Ø THEN Y=4:GOSUB 559Ø
:GOTO 557Ø
5535 IF CT=Ø THEN Y=5:GOSUB 559Ø
:GOTO 557Ø
5536 IF CT=1 THEN Y=2:GOSUB 559Ø
```

```
:GOTO 557Ø
5539 IF CT=1 THEN Y=6:GOSUB 559Ø
:GOTO 557Ø
5542 IF CT=1 THEN Y=4 ELSE Y=1
5543 A(Y,14)=A(Y,14)+1Ø:IF CT=1
THEN PRINT"AXIS CAPTURES ENERGY
RESERVES   IN MIDDLE EAST" ELSE
PRINT"ALLIES CAPTURE ENERGY RESE
RVES  IN MIDDLE EAST"
557Ø GOTO 5599
5571 IF B(4)=1 THEN A(Ø,1)=4
5572 IF B(5)=1 THEN A(Ø,1)=5
5573 IF B(11)=1 THEN A(Ø,1)=11
5574 IF B(1)=1 THEN A(Ø,1)=1
5575 IF B(2)=1 THEN A(Ø,1)=2
5576 IF B(Ø)=1 THEN A(Ø,1)=Ø
5577 IF B(3)=1 THEN A(Ø,1)=3
5578 IF B(2)=1 THEN A(Ø,1)=2
5579 IF B(4)=1 THEN A(1,1)=4
558Ø IF B(1Ø)=1 THEN A(1,1)=1Ø
5581 IF B(5)=1 THEN A(1,1)=5
5582 IF B(11)=1 THEN A(1,1)=11
5583 IF B(8)=1 THEN A(1,1)=8
5584 IF B(1)=1 THEN A(1,1)=1
5585 IF B(Ø)=1 THEN A(1,1)=Ø
5586 IF B(12)=1 THEN A(1,1)=12
5587 IF B(3)=1 THEN A(1,1)=3
5588 IF B(2)=1 THEN A(1,1)=1ØØ
5589 GOTO 57ØØ
559Ø FOR B=11 TO 15:A(Y,B)=INT(.
2*A(Y,B)):NEXT:A(Y,1)=2ØØ:RETURN
5599 NEXT X
56ØØ FOR Y=Ø TO 6:A(Y,13)=A(Y,13
)+INT(.1*(A(Y,11)+A(Y,12)+A(Y,14
))) 'UPDATE COUNTRY CONDITIONS
561Ø IF FL=1Ø THEN GOSUB 5571 EL
SE GOSUB 562Ø
5615 GOTO 5699
562Ø FOR C=Ø TO 6:ON C+1 GOTO 56
25,5626,5627,5628,5629,563Ø,5631
5625 ON T+1 GOTO 565Ø,565Ø,565Ø,
565Ø,565Ø,565Ø,565Ø,565Ø,565Ø,56
5Ø,565Ø,5671,5671,5671,5671,5671
,5665,5665,5651,5651,5663,5663,5
663,5664,5664 'USA GOALS
5626 ON T+1 GOTO 565Ø,5651,5651,
5651,5651,5651,5651,5651,5651,56
51,5651,5671,5671,5671,5671,5671
,5665,5665,5651,5651,5663,5663,5
663,5664,5664 'GB GOALS
5627 ON T+1 GOTO 565Ø,5669,5667,
5667,5668,5668,5668,5651,5651,56
51,5651,5651,5651,5651,5651,5651
,5651,5666,5666,5666,5669,5669,5
669,5664,5664 'USSR GOALS
5628 ON T+1 GOTO 565Ø,5651,5651,
5651,5652,5652,5652,5652,5652,56
52,5652,5652,5652,5652,5652,5652
,5652,5652,5652,5652,5652,5652,5
664,5664,5664 'FRA GOALS
5629 ON T+1 GOTO 565Ø,5669,5667,
5667,5663,5662,5662,5662,5662,56
66,5666,5666,5666,5666,5666,5666
,5666,5651,5651,5651,5651,5651,5
651,5663,5652 'GER GOALS
563Ø ON T+1 GOTO 565Ø,5651,5651,
5651,5651,5651,5651,5651,5651,56
72,5651,5651,5651,5651,5651,5651
,5651,5652,5652,5652,5652,5652,5
652,5652,5652 'ITL GOALS
5631 ON T+1 GOTO 565Ø,5652,5652,
5652,5652,5652,5652,5652,5652,56
52,5652,5652,5652,5652,5652,5652
,5652,5652,5652,5652,5652,5652,5
652,5664,5664 'POL GOALS
565Ø A(C,1)=999:GOTO 568Ø
5651 A(C,1)=1ØØ:GOTO 568Ø
5652 A(C,1)=2ØØ:GOTO 568Ø
566Ø A(C,1)=Ø:GOTO 568Ø
5661 A(C,1)=1:GOTO 568Ø
5662 A(C,1)=2:GOTO 568Ø
5663 A(C,1)=3:GOTO 568Ø
5664 A(C,1)=4:GOTO 568Ø
5665 A(C,1)=5:GOTO 568Ø
5666 A(C,1)=6:GOTO 568Ø
5667 A(C,1)=7:GOTO 568Ø
5668 A(C,1)=8:GOTO 568Ø
5669 A(C,1)=9:GOTO 568Ø
567Ø A(C,1)=1Ø:GOTO 568Ø
5671 A(C,1)=11:GOTO 568Ø
5672 A(C,1)=12:GOTO 568Ø
568Ø NEXT C
5681 RETURN
5699 RETURN
57ØØ IF A(2,Ø)=1 AND B(12)=Ø THE
N A(2,1)=12
57Ø1 IF A(2,Ø)=1 AND B(9)=Ø THEN
 A(2,1)=9
57Ø2 IF A(2,Ø)=Ø AND B(4)=1 THEN
 A(2,1)=4
57Ø3 IF A(2,Ø)=Ø AND B(9)=1 THEN
 A(2,1)=9
57Ø4 IF A(2,Ø)<>B(6) THEN A(2,1)
=1ØØ
57Ø5 IF B(4)=1 THEN A(3,1)=4
57Ø6 IF B(2)=1 THEN A(3,1)=2
57Ø7 IF B(3)=1 THEN A(3,1)=1ØØ
57Ø8 IF B(2)=Ø THEN A(4,1)=2
57Ø9 IF B(3)=Ø THEN A(4,1)=3
571Ø IF B(6)=Ø THEN A(4,1)=6
5711 IF B(Ø)=Ø THEN A(4,1)=Ø
5712 IF B(8)=Ø THEN A(4,1)=8
5713 IF B(7)=Ø THEN A(4,1)=7
5714 IF B(11)=Ø THEN A(4,1)=11
5715 IF B(5)=Ø THEN A(4,1)=5
5716 IF B(4)=Ø THEN 578Ø
5717 IF B(1Ø)=Ø THEN A(5,1)=1Ø
5718 IF B(12)=Ø THEN A(5,1)=12
5719 IF B(3)=Ø THEN A(5,1)=3
572Ø IF B(11)=Ø THEN A(5,1)=11
5721 IF B(4)=Ø THEN A(5,1)=4
5722 IF B(5)=Ø THEN A(5,1)=1ØØ
5723 IF B(9)=1 THEN A(6,1)=1ØØ
573Ø RETURN
578Ø CLS:PRINT@128,"    GERMANY
SURRENDERS !                  T
HE WAR IS OVER !!":GOTO 578Ø
58ØØ CLS:PRINT@128,"IT IS ";K$;"
, ";YR:PRINT:PRINT:PRINT"THE MIL
ITARY GOALS ARE:":GOSUB 991
581Ø FOR C=Ø TO 6:IF A(C,1)=999
THEN K$=" REMAIN NEUTRAL" ELSE I
F A(C,1)=1ØØ THEN K$=" DEFEND PO
SITIONS" ELSE IF A(C,1)=2ØØ THEN
 K$="OCCUPIED" ELSE IF A(C,1)>-1
 AND A(C,1)<13 THEN K$=" INVADE
"+X$(A(C,1))
582Ø PRINT C;" ";Y$(C):PRINTK$:G
OSUB 991:NEXT C
583Ø PRINT"  DO YOU WANT TO CHAN
GE GOALS?   (Y/N)"
5832 K$=INKEY$:IF K$="" THEN 583
2 ELSE IF K$="Y" THEN 584Ø ELSE
589Ø
584Ø PRINT" ENTER THE COUNTRY'S
NUMBER Ø-6":INPUT C:IF C<Ø OR C>
6 THEN 584Ø
5842 PRINT"INPUT NEW GOAL FOR TH
E          ";Y$(C);" TROOPS":PR
INT" -1 = REMAIN NEUTRAL
     -2 = DEFEND CURRENT POSITIO
N    Ø TO 12 = INVADE THAT REGIO
N"
5844 INPUT W:IF W<-2 OR W>12 THE
N 5842 ELSE IF W=-2 THEN A(C,1)=
999 ELSE IF W=-1 THEN A(C,1)=1ØØ
 ELSE A(C,1)=W
585Ø FL=1Ø:GOSUB 593Ø:GOTO 58ØØ
589Ø RETURN
59ØØ TF=Ø
59Ø5 ON B+1 GOTO 591Ø,5911,5912,
5913,5915,5917
591Ø TF=1Ø:A$(1Ø)=STR$(Q)+" DIVI
SIONS":RETURN
5911 TF=1Ø:A$(1Ø)=STR$(Q)+" NAVA
L TASK FORCES":RETURN
5912 TF=1Ø:A$(1Ø)=STR$(Q)+" AIRC
RAFT SQUADRONS":RETURN
5913 IF Q=Ø THEN TF=Ø ELSE IF Q>
=5 THEN TF=1 ELSE TF=2
5914 RETURN
5915 TF=1Ø:A$(1Ø)=STR$(Q)+" UNIT
S OF SUPPLIES":RETURN
5917 TF=1Ø:IF Q=999 THEN 5918 EL
SE IF Q=1ØØ THEN 5919 ELSE IF Q=
2ØØ THEN 5921 ELSE 592Ø
5918 A$(1Ø)=" REMAIN NEUTRAL":RE
TURN
5919 A$(1Ø)=" TAKE DEFENSIVE POS
ITION":RETURN
592Ø A$(1Ø)=X$(Q):RETURN
5921 A$(1Ø)=" ARE UNDER
        FOREIGN DOMINATION":RE
TURN
593Ø IF T=Ø THEN T=2
5931 Y=INT(T/4+.1):B=T-4*Y:IF B=
Ø THEN K$="WINTER" ELSE IF B=1 T
HEN K$="SPRING" ELSE IF B=2 THEN
 K$="SUMMER" ELSE IF B=3 THEN K$
="AUTUMN" ELSE K$=""
5932 YR=1939+Y:IF(T=3 AND FL<>1Ø
) THEN 5935 ELSE RETURNRETURN
5935 X=3:Y=3:B=3:GOSUB 99Ø:POKE
AQ+AD,Ø:RETURN
9ØØØ FOR X=Ø TO 12:FOR Y=Ø TO  6
:FOR B=Ø TO  4:GOSUB 99Ø:PRINT#-
2,X;" ";Y;" ";B;" ";PEEK(29ØØØ+A
D):NEXT B,Y,X
```

```
5066 .....254   5512 .....145
5108 .....11    5551 .....58
5310 .....175   5635 .....228
5330 .....243   5700 .....240
5455 .....42    5877 .....34
                END .....37
```

Listing 3: STOCKCAR

```
5ØØØ 'SCOREBOARD
5ØØ1 CLS:PRINT STRING$(32,CHR$(1
98)):PRINT" TRANSOUTH 5ØØ - DARL
INGTON, SC"
5ØØ2 PRINT USING"###";LS;:PRINT"
 LAPS..TIME  ";:GOSUB 5Ø6Ø:PRINT
 K$
5ØØ3 PRINT"POSIT      DRIVER
      LAP"
5ØØ5 FOR P=1 TO 1Ø:GOSUB 5Ø5Ø:PR
INT USING"#####
   %%####";P," ",Y$(D)," ",A(D,2
)
5ØØ6 NEXT
5ØØ7 PRINT STRING$(31,CHR$(198))
;
5ØØ8 K$=INKEY$:IF K$="" THEN 5ØØ
8 ELSE RETURN
5Ø5Ø X=Ø:B=6:FOR Y=Ø TO IY
5Ø51 GOSUB 99Ø:Q=PEEK(AQ+AD):IF
Q=P THEN D=Y ELSE 5Ø53
5Ø52 Y=15
5Ø53 NEXT Y
5Ø54 RETURN
5Ø6Ø TT=T*4Ø:TØ=INT(TT/36ØØ):T1=
INT((TT-36ØØ*TØ)/6Ø):T2=INT(TT-3
6ØØ*TØ-6Ø*T1)
5Ø66 K$=MID$(STR$(TØ),2,2)+":"+M
ID$(STR$(T1),2,2)+":"+MID$(STR$(
T2),2,2):RETURN
51ØØ Q=RND(-TIMER):X$(Ø)="":LP=1
.36:LS=367:TS=157.7
51Ø2 Y$(Ø)="BILL ELLIOTT - FORD"
:Y$(1)="DAVID PEARSON - CHEVY":Y
$(2)="TERRY LABONTE - CHEVY":Y$(
3)="BENNY PARSONS - CHEVY":Y$(4)
="LAKE SPEED - PONTIAC":Y$(5)="R
ON BOUCHARD - BUICK":Y$(6)="NEIL
 BONNETT - BUICK":Y$(7)="DALE EA
RNHARDT - CHEVY"
51Ø4 Y$(8)="RICKY RUDD - FORD":Y
$(9)="BUDDY BAKER - OLDS":Y$(1Ø)
="GEOFF BODINE - CHEVY":Y$(11)="
HARRY GANT - CHEVY":Y$(12)="DARR
ELL WALTRIP - CHEVY":Y$(13)="DAV
E MARCIS - CHEVY":Y$(14)="TIM RI
CHMOND - PONTIAC"
51Ø6 B$(Ø)="TRACK POSITION":B$(1
)="TIRE CONDITION":B$(2)="BRAKE
CONDITION":B$(3)="GEARBOX CONDIT
ION":B$(4)="ENGINE CONDITION":B$
(5)="FUEL LEVEL":B$(6)="CURRENT
POSITION":B$(7)="PIT ON NEXT LAP
?"
51Ø7 A$(1)="FRONT STRAIGHT":A$(2
)="TURN ONE":A$(3)="TURN TWO":A$
(4)="BACK STRETCH":A$(5)="TURN T
HREE":A$(6)="TURN FOUR":A$(7)="I
N PITS"
```

```
51Ø8 IX=Ø:IY=14:IB=7
5199 RETURN
52ØØ FG=1Ø
52Ø4 B=Ø:FOR Y=Ø TO IY:GOSUB 99Ø
:POKE AQ+AD,(15-Y):NEXT Y
52Ø6 FOR B=1 TO 4:FOR Y=Ø TO IY:
GOSUB 99Ø:POKE AQ+AD,25Ø:NEXT Y,
B
52Ø8 B=5:FOR Y=Ø TO IY:GOSUB 99Ø
:POKE AQ+AD,25Ø:NEXT Y
521Ø B=6:FOR Y=Ø TO IY:GOSUB 99Ø
:POKE AQ+AD,Y+1:NEXT Y
5211 B=7:FOR Y=Ø TO IY:GOSUB 99Ø
:POKE AQ+AD,Ø:NEXT Y
5212 FOR Y=Ø TO IY:A(Y,Ø)=((15-Y
)/255)*LP:A(Y,1)=TS-.3*Y:A(Y,2)=
1:A(Y,3)=TS-.2*Y:A(Y,5)=Ø:NEXT Y
522Ø A(1,4)=7:A(3,4)=11Ø:A(8,4)=
25Ø
5299 FL=2:RETURN
53ØØ CLS(3):FOR Y=Ø TO IY:TL=3Ø:
PRINT:GOSUB 569Ø
53Ø1 IF A(Y,5)=1ØØØ THEN 559Ø EL
SE IF FG=15 THEN 5315 ELSE IF FG
=1Ø THEN 531Ø ELSE IF FG=2Ø THEN
 532Ø ELSE IF FG=3Ø THEN 533Ø
531Ø IF A(Y,5)=999 THEN 55ØØ ELS
E IF A(Y,5)=Ø THEN 54ØØ 'GREEN F
LAG
531Ø IF A(Y,5)=999 THEN 55ØØ ELS
E IF A(Y,5)=Ø THEN 54ØØ 'GREEN F
LAG
5311 TL=TL-A(Y,5):IF TL<Ø THEN 5
312 ELSE 5314  'ALREADY IN PITS
5312 A(Y,5)=A(Y,5)-4Ø:PRINTY$(Y)
:PRINT" IN PITS":GOTO 5599 'STAY
 IN  PITS
5314 A(Y,5)=Ø:PRINTY$(Y):PRINT"
OUT OF PITS":GOTO 54ØØ RUN WITH
TL TIME LEFT
5315 IF A(Y,5)<>Ø THEN 5316 ELSE
 5599
5316 A(Y,Ø)=A(Y,Ø)+A(Y,3)*TL/36Ø
Ø:A(Y,2)=A(Y,2)+1:GOTO 5599
532Ø IF A(Y,5)=999 THEN 55ØØ ELS
E IF A(Y,5)=Ø THEN 5323 ELSE 531
1
5323 PRINTY$(Y):PRINT"RUNNING UN
DER YELLOW":A(Y,1)=124:SQ=A(Y,1)
*TL/36ØØ:A(Y,Ø)=A(Y,Ø)+SQ:B=1:GO
SUB 99Ø:Q=PEEK(AQ+AD):IF Q<1ØØ T
HEN A(Y,5)=999 'RUN UNDER YELLOW
 & CHCK TIRES
5324 B=5:GOSUB 99Ø:Q=INT(PEEK(AQ
+AD)-(SQ*1Ø/(3.5+RND(1ØØ)*.Ø1)))
:IF Q<Ø THEN Q=Ø ELSE IF Q>25Ø T
HEN Q=25Ø
5325 POKE AQ+AD,Q:IF Q<14Ø THEN
A(Y,5)=999
5326 GOTO 549Ø
533Ø X=Ø:FOR B=1 TO 5:GOSUB 99Ø:
POKE AQ+AD,25Ø:NEXT 'RED FLAG
5331 B=Ø:GOSUB 99Ø:POKE AQ+AD,Ø:
A(Y,Ø)=A(Y,2)*LP:GOTO 5599
54ØØ IF A(Y,2)=A(Y,4) AND RND(1Ø
)>3 THEN 5435 ELSE 5431
5431 X=Ø:B=1:GOSUB 99Ø:Q=PEEK(AQ
+AD):B=2:GOSUB 99Ø:Q1=PEEK(AQ+AD
):IF Q<2Ø OR Q1<2Ø OR RND(15ØØØ)
>14998 THEN 5435 ELSE 544Ø
5435 CLS:PRINT@128,Y$(Y):PRINT"
SPINS ON TURN AND HITS WALL ....
.. HE IS OUT OF RACE":GOSUB 991:
FG=2Ø:A(Y,5)=1ØØØ:GOTO 549Ø
544Ø X=Ø:B=3:GOSUB 99Ø:Q=PEEK(AQ
+AD):B=4:GOSUB 99Ø:Q1=PEEK(AQ+AD
):IF Q<6Ø OR Q1<6Ø THEN 545Ø ELS
E 546Ø
545Ø CLS:PRINT@128,Y$(Y)
5452 ON RND(5) GOTO 5453,5454,54
55,5456,5457
5453 PRINT"..BLOWN ENGINE .. OUT
 OF RACE":GOSUB 991:A(Y,5)=1ØØØ:
GOTO 549Ø
5454 PRINT".GEARBOX FAILURE, OUT
 OF RACE":GOSUB 991:A(Y,5)=1ØØØ:
GOTO 549Ø
5455 PRINT".FUEL PUMP FAILURE,
          .. OUT OF RACE":GOSUB
 991:A(Y,5)=1ØØØ:GOTO 549Ø
5456 PRINT".BROKEN CAMSHAFT,
          .. OUT OF RACE":GOSUB
 991:A(Y,5)=1ØØØ:GOTO 549Ø
5457 PRINT".BROKEN PISTON,
          .. OUT OF RACE":GOSUB
 991:A(Y,5)=1ØØØ:GOTO 549Ø
546Ø B=4:GOSUB 99Ø:Q=PEEK(AQ+AD)
*.2/25Ø+.8:SP=A(Y,3)*Q:A(Y,1)=SP
:SQ=SP*TL/36ØØ:A(Y,Ø)=A(Y,Ø)+SQ:
B=1:X=Ø:GOSUB 99Ø:Q=PEEK(AQ+AD):
IF Q<1ØØ THEN A(Y,5)=999 'ADD MI
LES & CHECK TIRES
5461 PRINTY$(Y):PRINT"  RUNNING
AT ";SP;" MPH"
5462 B=5:GOSUB 99Ø:Q=INT(PEEK(AQ
+AD)-(SQ*1Ø/(3.Ø+RND(1ØØ)*.Ø1)))
:IF Q<Ø THEN Q=Ø ELSE IF Q>25Ø T
HEN  Q=25Ø  'USE FUEL
5464 POKE AQ+AD,Q:Q=PEEK(AQ+AD):
IF Q<6Ø THEN A(Y,5)=999 'GO INTO
 PITS?
5465 FOR B=1 TO 4:GOSUB 99Ø:Q=PE
EK(AQ+AD)-RND(3):IF Q<Ø THEN Q=Ø
5466 POKE AQ+AD,Q:NEXT B:GOTO 54
9Ø
549Ø GOSUB 5875:GOTO 5599
55ØØ 'IN PITS
551Ø Q=A(Y,2)*LP:SQ=Q-A(Y,Ø):IF
FG=2Ø THEN TP=SQ*36ØØ/124 ELSE T
P=SQ*36ØØ/A(Y,1)
5512 IF TP<TL THEN 553Ø ELSE 54Ø
Ø
553Ø B=Ø:X=Ø:GOSUB 99Ø:POKE AQ+A
D,Ø:T1=TL-TP  'TIME LEFT FOR PIT
S ON THIS LAP
5531 B=1:GOSUB 99Ø:Q1=PEEK(AQ+AD
):B=2:GOSUB 99Ø:Q2=PEEK(AQ+AD):B
=3:GOSUB 99Ø:Q3=PEEK(AQ+AD):B=4:
GOSUB 99Ø:Q4=PEEK(AQ+AD)
5532 A(Y,Ø)=LP*A(Y,2):PRINT Y$(Y
):PRINT" IS IN PITS AT LAP ";A(Y
,2)+1:GOSUB 991
5534 Q1=25Ø:IF Q2<1ØØ OR Q3<1ØØ
OR Q4<1ØØ THEN 555Ø  'LONG STOP
5535 B=5:GOSUB 99Ø:POKE AQ+AD,25
Ø:B=1:GOSUB 99Ø:POKE(AQ+AD),Q1:B
=2:GOSUB 99Ø:POKE(AQ+AD),Q2:B=3:
GOSUB 99Ø:POKE(AQ+AD),Q3:B=4:GOS
UB 99Ø:POKE(AQ+AD),Q4
5536 Q=RND(1ØØ)*.Ø1:TX=1Ø+2*Q:GO
TO 556Ø
555Ø B=5:GOSUB 99Ø:POKE AQ+AD,25
Ø:B=1:GOSUB 99Ø:POKE(AQ+AD),22Ø:
B=2:GOSUB 99Ø:POKE(AQ+AD),22Ø:B=
3:GOSUB 99Ø:POKE(AQ+AD),22Ø:B=4:
GOSUB 99Ø:POKE(AQ+AD),22Ø
5551 Q=RND(1ØØ)*.Ø1:TX=32+4*Q:GO
TO 556Ø
556Ø PRINT USING"%          %###.
###        %";"STOP TAKES ";TX;
" SECONDS":GOSUB 991:IF TX>T1 TH
EN 5565 ELSE 557Ø
5565 A(Y,5)=TX-T1:GOTO 549Ø
557Ø A(Y,5)=Ø:TL=T1-TX:GOTO 54ØØ
559Ø PRINT Y$(Y):PRINT" IS OUT O
F RACE ON":PRINT"LAP ";A(Y,2):IF
 FG=15 THEN 5315 ELSE 5599
5599 NEXT Y
56ØØ PRINT:PRINT:IF FG=15 THEN 5
7ØØ
56Ø1 FOR Y=Ø TO IY:GOSUB 568Ø:NE
XT
56Ø5 FOR Y=Ø TO IY:IF A(Y,Ø)>LP*
(LS-1) THEN FG=15
56Ø6 NEXT
561Ø FOR P=1 TO IY+1
5611 MN=Ø:IF P=1 THEN MX=1ØØØØ
5612 Q=Ø:FOR Y=Ø TO IY
5613 IF A(Y,Ø)>MN AND A(Y,Ø)<MX
THEN 5615 ELSE 5619
5615 Q=Y:MN=A(Y,Ø)
5619 NEXT Y
562Ø MX=A(Q,Ø):X=Ø:Y=Q:B=6:GOSUB
 99Ø:POKE AQ+AD,P
5621 IF P=1 THEN PRINT Y$(Y):PRI
NT"LEADING THE RACE AFTER":PRINT
A(Y,2)-1;" LAPS"
5622 NEXT P
563Ø Q=RND(1ØØ):IF FG=2Ø THEN 56
35 ELSE IF FG=3Ø THEN 564Ø ELSE
IF FG=1Ø THEN 5645 ELSE IF FG=15
 THEN 565Ø
5635 IF Q>6Ø THEN FG=1Ø:PRINT" G
REEN FLAG !!":GOSUB 991:GOTO 565
Ø ELSE PRINT"YELLOW FLAG IS OUT"
:GOSUB 991:GOTO 565Ø
564Ø IF Q>2Ø THEN FG=1Ø:PRINT" G
REEN FLAG !!":GOSUB 991:GOTO 565
Ø ELSE "RED FLAG IS OUT, THE RAC
E IS    STOPPED.":GOSUB 991:GOTO
 565Ø
5645 IF Q>95 THEN FG=2Ø:PRINT" Y
ELLOW FLAG .. OIL ON TRACK":GOSU
B 991:GOTO 565Ø ELSE 565Ø
565Ø RETURN
568Ø IF A(Y,5)<>Ø THEN 5681 ELSE
 5683
5681 B=7:GOSUB 99Ø:POKE AQ+AD,1:
RETURN
5683 B=7:GOSUB 99Ø:POKE AQ+AD,Ø:
RETURN
569Ø B=7:GOSUB 99Ø:Q=PEEK(AQ+AD)
:IF Q=Ø THEN RETURN
5691 IF A(Y,5)=999 THEN 5692 ELS
E IF A(Y,5)=Ø THEN A(Y,5)=999 EL
SE IF A(Y,5)=1ØØØ THEN 5692
5692 RETURN
57ØØ CLS(7):FOR I=1 TO 48Ø:PRINT
 CHR$(198);:NEXT:PRINT"THE RACE
IS OVER !!":GOSUB 991:GOSUB 991
571Ø FOR P=1 TO IY+1
5711 MN=Ø:IF P=1 THEN MX=1ØØØØ
5712 Q=Ø:FOR Y=Ø TO IY
5713 IF A(Y,Ø)>MN AND A(Y,Ø)<MX
THEN 5715 ELSE 5719
5715 Q=Y:MN=A(Y,Ø)
5719 NEXT Y
572Ø MX=A(Q,Ø):X=Ø:Y=Q:B=6:GOSUB
 99Ø:POKE AQ+AD,P
5722 NEXT P
573Ø FOR P=1 TO 15
5731 FOR Y=Ø TO IY
5732 B=6:GOSUB 99Ø:Q=PEEK(AQ+AD)
:IF P=Q THEN 5733 ELSE 5735
5733 PRINT "#";P;"IS ";Y$(Y):PRI
NT A(Y,2);" LAPS COMPLETE":PRINT
"------------":GOSUB 991
5735 NEXT Y
574Ø NEXT P
575Ø PRINT"SEE RESULTS AGAIN? (Y
/N)"
5752 K$=INKEY$:IF K$="" THEN 575
2 ELSE IF K$="Y" THEN 573Ø ELSE
END
585Ø 'GIVEN Y CONV TO MILES
5852 X=Ø:B=Ø:GOSUB 99Ø:Q=PEEK(AQ
+AD):A(Y,Ø)=(Q/255)*LP+(A(Y,2)-1
)*LP:RETURN
5875 'GIVEN Y CONVERT TO LAPS AN
D POSIT
5877 A(Y,2)=INT(A(Y,Ø)/LP)+1
5879 Q=A(Y,Ø)-(A(Y,2)-1)*LP:X=Ø:
B=Ø:GOSUB 99Ø:POKE AQ+AD,INT(255
*Q/LP):RETURN
59ØØ TF=Ø  'Q INPUT , RETURN TF
5995 ON B+1 GOTO 591Ø,5931,5931,
5931,5931,592Ø,5931,593Ø
591Ø IF Q<=4Ø OR Q>215 THEN TF=1
5911 IF Q<=64 AND Q>4Ø THEN TF=2
5912 IF Q<=88 AND Q>64 THEN TF=3
5913 IF Q<=168 AND Q>88 THEN TF=
4
5914 IF Q<=192 AND Q>168 THEN TF
=5
5915 IF Q<=216 AND Q>192 THEN TF
=6
5917 IF Q=Ø THEN TF=7
5919 RETURN
592Ø TF=1Ø:A$(1Ø)=STR$(Q/1Ø)+" G
ALLONS":RETURN
593Ø TF=1Ø:IF Q=Ø THEN A$(1Ø)="N
O" ELSE A$(1Ø)="YES"
5931 RETURN
```

Rainbow Simulation Generator

Simulations make it possible to experience many exciting situations which normally would not present themselves. The best are those that provide as many realistic options as possible and proceed at a realistic pace.

To make it much easier to create your own special scenarios, THE RAINBOW is providing you with a Simulation generator (Listing 1). It's the same generator which runs the subroutines in the *World War II* and *Stock Car* programs.

To conserve memory, blocks of bytes were used to store variables rather than a full array. The reasoning is simple: If you can scale variables between zero and 255, then only one byte is necessary for each variable. If we used floating point numbers, we would need five bytes for each one. Look at Lines 4 and 40: A block of one-byte variables begins at memory byte 28000.

The variables are addressed in this block by assigning three index values to the variable locations 'X', 'Y' and 'B'. These are similar to using a three-dimensional array, except to find the address of the variable you GOSUB 990 with a particular 'X', 'Y' and 'B' in memory. The subroutine returns a value of AD, which is the offset address you need from the 28000 base. For example, if we want to store the number 76 in the variable indexed by X=2, Y=3, B=0, then type:

```
X=2:Y=3:B=0:GOSUB 990:POKE (AQ+AD),76
```

To retrieve the number and store it in Variable A, just reverse the process:

```
X=2:Y=3:B=0:GOSUB 990:A=PEEK(AQ+AD)
```

To describe what each 'X', 'Y' or 'B' might mean, we provided the three-character arrays X$( ), Y$( ) and B$( ). These are initialized when you load in your scenario. If you really have to have a floating point variable passed between the generator and your scenario, there are two arrays for you to use — the A(15,15) and B(15) dimensioned in Line 5. The variables that fill the 28000 block and the arrays are all initialized to your starting conditions when you choose 'I' on the menu.

To write a scenario, the following conditions must be met:

— The scenario should be between lines 5000 and 5999 for uniformity and interchangeability. Lines higher than 5999 can be used for diagnostics (such as in Listing 2 for *World War II*).

— Lines 5000-5099 contain a graphics subroutine. You can draw a picture, play music or just give a message. It's sort of a free area for fun and games. It assumes the CoCo default PCLEAR4. The *World War II* routine draws the map of Europe; the Stock Car routine draws the leader scoreboard. This subroutine should always end with a RETURN.

— Lines 5100-5199 include a subroutine that loads all the "literals," such as X$( ) and Y$( ) which pertain to the scenario. It also loads the IX, IY and IB parameters, which are the upper limits of 'X', 'Y' and 'B', respectively. You won't exceed memory requirements if IX*IY*IB<4000. If you need more room, you can CLEAR more room in Line 4 and redefine AQ in Line 40. Don't use this subroutine to set initial condition of the scenario.

— Lines 5200-5299 contain the subroutine for loading all initial starting values into the arrays and the 28000 block.

— A subroutine in lines 5300-5899 contains the "laws of propagation." Whenever you want to change the time sequence of your scenario, you jump to Line 5300. This subroutine calculates all the variables as they would change and stores them. It also prints out special messages. When you exit this subroutine, the generator takes over, allowing you to change the scenario or, to just watch the action unfold.

— Lines 5900-5929 contain the translator subroutine. As stated earlier, the variables in the 28000 block are POKEd into memory and are confined to one byte. It is here where you determine what something like Troop Strength=21 really means. You enter this routine from a call in the "View or Change" option on the main menu. The value of 'X', 'Y' and 'B' is given and value is placed in the TF variable. If you exit with just TF=0:RETURN then the geneator will interpret the variable literally (i.e., 21). However, if you exit with TF=2:RETURN, the generator will respond by printing whatever A$(TF) says. A$ is simply a message list and can be used any way you like. You can even define it as you go into this subroutine, like this:

```
5900 IF B=3 THEN TF=21
5902 A$(21)=STR$(PEEK(AQ+AD))+" ARMORED
DIVISIONS":RETURN
```

When you load one of the scenarios you see the opening graphics from the subroutine at Line 900 in the generator. After a few seconds, the main menu appears. You should choose 'I' to initialize the Simulation. This choice causes the program to jump to the subroutine that starts at Line 5200 to begin the scenario. Subsequent showings of the main menu will not show the 'I' option, but you can restart the scenario any time you press the letter.

Let's look at some of the variables we can change. Choose "View" from the main menu and then "As One Specific Cell" from the submenu. You will see all of the variables in order and ready to make changes. If you just want to see one variable, then select 'S' from the submenu. You will see the values of the 'X' index, the 'Y' index and the 'B' index. Simply enter a number, and press the ENTER key. Don't enter a new value less than zero or greater than 255. Remember that you are POKEing numbers directly into memory. At any "Keep or Change" option, you also can press the 'M' key to get back to the menu — a good idea if you aren't familiar with the scenario yet.

Choose the "Graphics" option at the main menu to see the display we have provided for the *World War II* and *Stock Car* scenarios. If it's your own scenario, remember to provide an OUT for your graphics subroutine; the command 5099 GOTO 5099 will not work.

ASSEMBLY FILE

BY KEVIN

Let's talk this month about POSITION INDEPENDENT CODE (PIC). Perhaps you have heard about all of the great qualities of our 6809 Central Processor and not really understood why these things are so great.

The ability to easily write Postion Independant Code is perhaps one of the best features of our CPU. I would venture to say that, unless you have very good reason not to, all of your Assembly Code should be written using PIC. If you have never programmed other processors you may wonder what the big deal is. There is no big deal, its just that PIC is EASY on the 6809.

When writing your assembly code you decide where in memory you wish to locate the final machine code and inevitably that location is chosen to suit your system. It's no good locating your code at $7F00 if the top of your RAM memory is only $3FFF as is found in a 16K CoCo. And yet you may wish to give your program to a mate who has a 32K CoCo and who needs to locate the machine code above $7F00 so as not to interfere with his BASIC program. If the program was written in PIC then all that is required is that you change the ORG statement in your Assembly source code, reassemble the program and lo-and-behold your program will now sit at the new location. Alternatively you may load the original MACHINE CODE program with an offset appropriate to the new location you wish the program to reside in memory and then CSAVEM the program from the new memory location taking care to ensure that the memory addresses you use within the CSAVEM command are correct. Not quite as easy but it will work.

So much for the why. Now for the HOW.

If you examine a copy of the 6809 instruction set you will find a multitude of BRANCH instructions (eg. BRA BEQ BNE etc). Note also the JMP and JSR instructions. These instructions all tell the PROGRAM COUNTER REGISTER to Branch or Jump to a different part of the program. The branch instructions will only branch if the conditions tested by the branch are met (eg. BEQ will only branch if the Z bit of the CC register is set), but that is not relevant to our discussion at present.

What we are interested in is the comment that describes the ADDRESSING MODES of these instructions. You will notice that all of the branch instructions are described as RELATIVE. Relative to what? If the branch is to be taken then the processor will look at the POST-BYTE (the byte following the byte containing the instruction) and add that value to the current value of the Program Counter Register, in effect causing the program counter to contain a new value RELATIVE to its previous value. Remember the program counter points to the address of the next instruction it will execute and so it should now point to an instruction that is the beginning of a subroutine elsewhere in our program.

When the program is assembled the assembler program will examine your instruction and calculate the actual value it will use as the post-byte to pinpoint the relative offset of the subroutine, so again there is no need for you to become intimidated by unecessary calculations. Poor old CoCo must do all the hard work.

But we are not finished with branch instructions yet. An address one byte long can hold any integer value within the range -127 to +127 and since the post-byte of our BRANCH instruction is only one byte long this would seem to imply that we are limited to branches that lie within this range. And this would be true if we were limited to using the simple BRANCH instruction. But we are not. The architects of the 6809 provided us with an alternative group of LONG BRANCH instructions (LBRA LBNE LBLT etc). These instructions consume two bytes for their post-byte to in effect give a range of values from -32K to +32K which covers the entire 64K of memory that is able to be directly addressed by the 6809 CPU. The only penalty is that we consume an extra byte of our limited memory.

How do you know which instruction to use? That's easy. If you are unsure whether or not you must use the LONG BRANCH instruction then use the simple short BRANCH and if the assembler finds the branch outside the range of the single post-byte an error will be generated upon assembly and you will know that you need to change the BRANCH instruction to a LONG BRANCH instruction (all other things being correct).

We have available one other form of instruction that allows relative addressing and that is the PROGRAM COUNTER RELATIVE subset of the INDEXED Adressing Mode.

We have alrady had some experience with Indexed Adressing when we auto-incremented the contents of a register.

eg. STA ,X+

The assembler inspects the operand following the instruction and determines the EFFECTIVE ADDRESS where the instruction will find, store or in some way manipulate its data.

The operand can be used to point to an address relative to the program counter simply by including ,PCR as a part of the operand. This will cause the value or more commonly, the label included in the operand to be evaluated in a similar fashion to the branch instructions and thus act on data contained at an address relative to the program counter.

```
eg.       STA  HERE,PCR
          JMP  $A027
HERE      RMB  $01
```

This simple piece of code will cause the contents of register A to be stored at the address of HERE. RMB is a PSEUDO-OP and asks the assembler to reserve the number of memory bytes specified in the operand for the machine code programs use when the program is EXECuted. But the important point to note is that the postbyte to the STA instruction in the assembled code will contain the value +03 which is the relative offset from the current value of the PC register to the address of HERE. The STA instruction will not change the value of the PC register, but simply use the value contained in the PC register to determine the actual address 3 bytes higher in memory.

By now you should have an understanding of what is meant by the term RELATIVE addressing and it should also be obvious how we are able to use this feature of the 6809 to write our POSTION INDEPENDENT CODE and thus free us from the burden of the need to confine our machine code program to a specific block of memory.

Next month I will present a simple routine to shows how you can put what we have learnt to use and at the same time begin to explore the hidden talents of COLOR BASIC.

DISK UTILITY

64K Disk

Getting On The Right Track

By Colin J. Stearman

With the dramatic price reduction of disk drives and the increased capacity they offer, it is only natural to wish CoCo could make use of them. The CoCo 2 now has 40-track drives as standard, and double-sided and 80-track drives are readily available.

As you know, Disk BASIC only uses 35 tracks on a single-sided disk, giving 68 granules of storage space. Each granule contains nine sectors with 256 bytes of information in each. In this article we will explore how we can make use of all 40 or 80 tracks on our drive and make CoCo use both sides of a double-sided drive.

Double-Sided Drives

The best way to make use of double-sided disk drives is to fool CoCo into thinking that each side of the disk is a separate drive. There are two approaches to this. One is to revise the DOS (Disk Operating System) in the disk controller and the other is to actually modify the disk drive interface. Let's look at the DOS revision first.

Revising the DOS

If you followed my earlier series called "Cooking with CoCo," you should be familiar with how to revise your DOS and move the result to an EPROM. To demonstrate the DOS revision here, we will move the DOS into "all RAM" on our 64K CoCo and modify it there.

Inside the DOS is a table of four bytes which tell CoCo how to access each of the four possible drives. The values in Table 1 are combined with other data and sent to the controller port at address $FF40. If you inspect this table you will find it contains the following:

Table Entry	Value	Drive
1	0000001	0
2	0000010	1
3	0000100	2
4	0100000	3

Table 1

This table starts at 55210 in DOS 1.0 and 55453 in DOS 1.1. I have shown the value in binary because it shows the pattern better. Bits 0, 1, 2 and 6 exactly correspond to four control lines coming out of the disk controller to the drives. These lines are on pins 10, 12, 14 and 32, respectively. These lines are called the drive select lines and Disk BASIC selects which drives see which line by simply removing pins in the plug on the ribbon cable as appropriate. This works, but presents us with a problem as you'll see shortly.

Disk drives are surprisingly standard on their pin assignments. Pins 10, 12 and 14 are the select lines from drives 0, 1 and 2. However, pin 32 is used to select the disk side on double-sided drives.

Most drives employ jumpers on the circuit board inside the drive to determine which drive select line is seen by the drive, so for the drive to be able to be configured for any drive number and employ the side select signal, the ribbon cable must deliver *all* four signals to it. Therefore, it is essential you replace the Radio Shack ribbon cable from the controller to the drives with one which has all the pins in the drive plugs. The supplier who sold you the double-sided drives should be able to furnish this cable.

Now that we can get all four signals to any drive, the table can be modified to deal with double-sided drives. Bit 6 is used to select side 0 or 1, and bits 0 and 1 select which of the two drives to use. If you use double-sided drives, obviously the maximum number of actual drives is two.

Some thought also needs to be given to how the drives are to be numbered. Remember that only disks written on Side 0 of either drive will be compatible with a standard Radio Shack disk system. I have found Table 2 to be the best arrangement.

Drive	Side	Designation
0	0	0
1	0	1
0	1	2
1	1	3

Table 2

To get this arrangement the table will

become:

Table Entry	Value	Drive
1	0000001	0
2	0000010	1
3	0100001	2
4	0100010	3

Table 3

The advantage of this configuration is the DIR0 and DIR1 will work just like a standard two-drive system, but with the added advantage that DIR2 will be on the back of the disk in Drive 0 and DIR3 on the back of Drive 1. Of course, other combinations are possible. Listing 1 will put your 64K CoCo in all RAM mode if necessary and modify as in Table 3. When the program finishes, try doing DIRs to each drive to check for successful patching.

The advantage of this DOS patching technique is that it is fairly simple to implement. However, if you use programs which contain their own DOS, this patch has no effect. The VIP series of programs is an example of this. To access the double-sided capabilities of your drive independent of the software, a hardware approach is the only solution.

Revising the Hardware

To achieve the same effect in hardware, the drive select signals output from the controller must be logically manipulated. We know from Table 1 the pattern which the controller normally puts out to select each drive; Table 3 shows what we would like the drives to see.

A circuit using a quad AND gate chip will perform this logic for us. In order to be completely "FLEX"ible, a switch is included to defeat the circuit. This is necessary because FLEX (and other CoCo DOSs) can deal with double-sided drives without hardware modifications.

Figure 1 shows the circuit. It is best mounted on a small circuit board inside the drive case where both the signals to be intercepted and a five volt supply are available. It is only necessary to intercept signals destined to reach pins 10, 12, 14 and 32; all others pass through to the drive as before. The selector switch can be mounted on the case back panel. The jumpers on each drive should be set to DS0 and DS1, respectively.

Also some brands of drives have head load solenoids — if your drive audibly clicks when accessed you have this feature. If you have two such drives I suggest you make the heads load when the motor is turned on, not when the drive select is activated.

This does mean that both drive heads load whenever either disk drive is accessed, but also saves a lot of wear and tear (not to mention noise) when both drives are sequentially accessed as when doing a COPY. This is achieved by removing the jumper on the disk circuit board from the "HS" position and placing it in the "HM" position.

40-Track Access

Many of you were disappointed when you discovered my DOS modifications in the "Cooking with CoCo" series did not fully incorporate 40-track drives into BASIC. Of course, there was a good reason for it. There are some very definite compromises which must be made before all 40 or 80 tracks can be used fully by BASIC. Let's discuss what they are and why, then you can decide whether or not to go ahead with them.

On Track 17, Sector 2 is a byte map which keeps track (pun intended!) of the used granules on the disk. This map is called the GAT (Granule Allocation Table). On a standard disk there are 35 tracks. One is used for the directory, leaving 34 for data. Each track is split into two granules. Hence, there are 68 granules on a disk, so the GAT has 68 entries.

The data stored in each entry tells the DOS whether the associated granule is available and if not, the data stored in the byte provides a chain of granules which comprise the particular disk file. When the DOS accesses a disk, the GAT is read into memory and written back as necessary when access is finished.

Therefore, there are four areas in RAM set aside for the four granule tables associated with drives 0-3. Because of some additional overhead there are 74 bytes in memory needed for each GAT. Immediately we can see the first problem. If we go to 40, or worse, 80 tracks, the granule count goes to 78 and 158. There's no room in RAM for the larger GATs.

This is where we arrive at the compromises. We can either reduce the number of drives usable and hence the total amount of RAM space needed for the GATs, or we can overlay the GATs for each drive in the same storage area and make sure we never open files on different drives simultaneously. (Multiple files may be open on the same drive, however.)

If you have DOS 1.0 the decision is easy. Because of some nasty bugs in the DOS, it is very unwise to ever have files open simultaneously on different drives. This makes the second choice the obvious one.

If you have 40-track drives it is possible to store three complete GATs in the available 296 bytes in RAM. The 296 is derived from 4*(68+6): the number of drives times the GAT length plus six bytes overhead. For the 40 tracks 3*(78+6) results in 252 bytes being required, so by limiting the maximum number of drives to three we can have files open simultaneously on different drives because we can store three GATs at once in the available 296 bytes (assuming we have DOS 1.1, that is). However, if you have two double-sided 40-track drives, then you will want to use all four drives and must opt for compromise two, whether or not you have DOS 1.1.

80-Track Access

For those of you with the new 80-track drives, the problems are worse. First, one GAT plus overhead requires 164 bytes (79*2+6). Second, the routine which decides how to assign granules to a file only works up to 127 granules. Third, there is only room in the directory for 128 names, even though there are 158 granules on the disk. Let's take each problem in turn.

With each GAT requiring 164 bytes of RAM and only 296 available, we have little choice but to overlay the GATs for each drive and never open files on more than one drive at once if we wish to use all 80 tracks. But if we were to limit ourselves to 72 tracks, then the GAT would need 148 bytes and two could be stored in the 296 available bytes. As a result, we could allow up to two drives and also allow files to be open simultaneously on both. This may be a desirable compromise for some with DOS 1.1.

The granule allocation routine in DOS is quite clever. It allocates granules so they cluster around Track 17, the directory track. Therefore, as a disk fills up, granules are allocated evenly towards the outer and inner tracks centered about Track 17. Unfortunately, this routine cannot handle more than 127 granules; 80 tracks has 158 granules and 72 has 142, so this routine must be changed. I have replaced it in these circumstances with a simple routine which allocates granules starting at Track 0 and working linearly to the highest track.

There is nothing we can do about the

third problem. The original DOS only uses nine sectors for the directory names, with eight names per sector. This patch already expands it to the remaining 16 sectors in the directory track, but this still only allows 128 names (8*16). This is not as bad as it seems. Although theoretically the 80-track drive can store 158 one-granule files, rarely are all files on the disk just one granule long. So, the available storage on the disk is 158 granules or 128 files, whichever comes first.

The Patch Program

Listing 2 is the program to use to patch your DOS for the desired drive type. It puts the DOS in all RAM and patches it according to your instructions. It will properly patch both DOS 1.0 and 1.1, and change *all* BASIC commands to correctly deal with the number of tracks available. The syntax of the BASIC commands is not altered in any way.

It has been further designed to patch a DOS modified by the "Cooking with CoCo" articles. The syntax of the BASIC commands DSKINI and BACKUP were modified by this patch.

Unfortunately, the parameter which checks for the maximum allowable track value cannot be patched here. This is because this data tends to move around in memory depending upon how you configured the patch. Check Line 374 on Page 84 of the September 1984 issue and change it to CMPB #72 or CMPB #80, reassemble and repatch (if you use the 72- or 80-track system). This will allow the necessary number of tracks for both the modified commands. Check that article to see the format of the DSKINI command to select the number of tracks.

Moving the GAT

One solution to the limited space for the RAM storage of the GATs exists: move its storage somewhere else where there is room for all the bytes. Because of the previously mentioned bugs in DOS 1.0, this option only exists for DOS 1.1 owners.

The very first question for DOS 1.1 owners who run Listing 2 concerns the address for the GAT — $800 is the normal place and will be used if no value is entered. The problem is deciding where to put the GAT. In the worst case, if you have four 80-track drives then you would need 656 bytes to store all four GATs, thus allowing multiple files to be opened simultaneously on all four drives. There is no such amount of protected storage in the RAM, but if you intend running in all RAM you could move the GAT storage up high in memory at or above $E000.

In any case, I leave it entirely up to you where you locate the GAT storage. The program makes no tests to see that you have provided sufficient room or that the space is in fact available. It does assume when you enter a value other than $800 that there is sufficient room for all four GATs plus overhead bytes. You can calculate the amount of storage needed from 4*[(# of tracks-1)*2+6].

Compatibility

One of the primary considerations before changing BASIC to handle the 40/80-track drives is that of compatibility. Obviously if you have 80-track drives and modified BASIC, you will not be able to read standard Disk BASIC disks. Because of the closer spacing of the tracks this was probably true even before you modified the DOS.

If you have 40-track drives it is not so bad. You will be able to read and write disks formatted on the standard DOS without any trouble. If you format a 40-track disk which you wish to use on a 35-track system, then run the program in Listing 3 *immediately* after formatting. This will prevent the upper five tracks from being used, therefore maintaining compatibility. Remember, only run this program on a blank disk since it will not work if there is anything stored on the disk.

Wrap Up

I hope this article has given you some insight into how to better use your disk drives. It is my understanding the new white drives for the CoCo 2 have 40-track capability. Try formatting a disk with the DOS patched for 40 tracks. The system will soon tell you if you do not have 40 tracks available!

I have tested the patches on 40-track drives but do not own an 80-track unit. I would like to thank RAINBOW reader Gino Latino of Melbourne, Australia for help in this area.

If you run into problems with these patches I'd be pleased to hear from you. Just write me with a clear definition of the problem and include a stamped, self-addressed envelope (both are essential if you wish a response from me). My address is 143 Ash Street, Hopkinton, MA 01748.

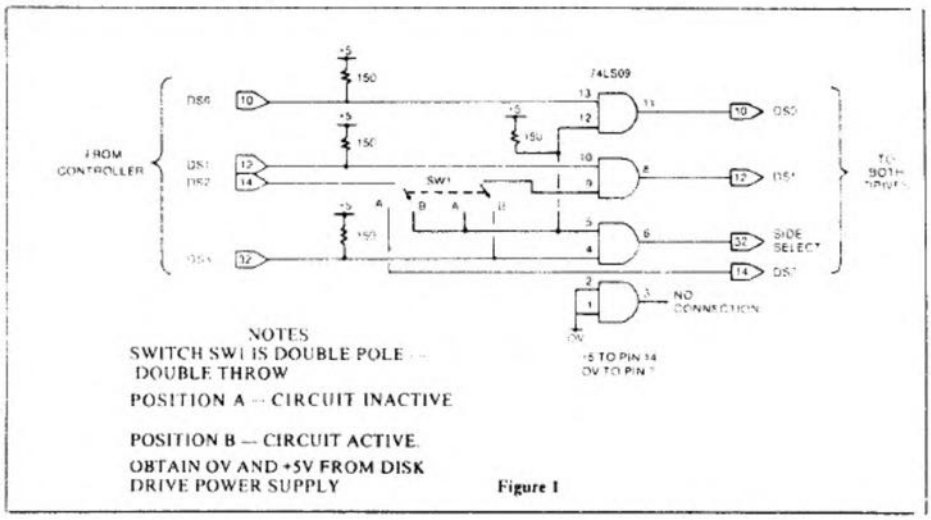
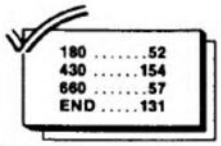


Figure 1

Listing 1: DBL SIDE

```
1 '      ----- LISTING #1 -----
1Ø ' DOUBLE SIDE DISK DRIVE BASI
C REVISIONS
2Ø ' COPYRIGHT 1985 C.J.STEARMAN
3Ø ' This will change the drive
access table
4Ø ' for 2 double sided drives.
5Ø '*****************************
**************
6Ø CLEAR 2ØØ,32511
7Ø 'RAM RELOCATION PROGRAM
8Ø DATA 26,8Ø,142,128,Ø,166,132,
183,255,223,167,128,14Ø,224,Ø,39
,5,183,255,222,32,239,28,175,57
9Ø FOR A=32512 TO 32536
1ØØ READ CODE:POKE A,CODE
11Ø NEXT A
12Ø 'PUT COCO INTO ALL RAM IF NE
CESSARY
13Ø A=PEEK(&HEØØØ):POKE&HEØØØ,55
:IF PEEK(&HEØØØ)<>55 THEN EXEC 3
2512
14Ø POKE&HEØØØ,A
15Ø VER$=CHR$(PEEK(&HC14Ø))+CHR$
(PEEK(&HC141))+CHR$(PEEK(&HC142)
)
16Ø IF VER$="1.Ø" THEN V=Ø:GOTO
19Ø
17Ø VER$=CHR$(PEEK(&HC153))+CHR$
(PEEK(&HC154))+CHR$(PEEK(&HC155)
)
18Ø IF VER$<>"1.1" THEN PRINT"SO
RRY, DON'T RECOGNIZE YOUR DOS":S
TOP ELSE V=1
19Ø POKE 5521Ø+V*243,Ø1
2ØØ POKE 55211+V*243,Ø2
21Ø POKE 55212+V*243,65
22Ø POKE 55213+V*243,66
23Ø PRINT"ALL DONE"
```



Listing 2: 40-80

```
1 '  ----- LISTING #2 -----
1Ø ' 4Ø/8Ø TRACK BASIC REVISIONS
2Ø ' COPYRIGHT 1985 C.J.STEARMAN
3Ø '**************************
**
5Ø CLEAR 2ØØ,32511
6Ø DEF FNUB(X)=INT(X/256)
7Ø DEF FNLB(X)=X-256*FNUB(X)
8Ø 'RAM RELOCATION PROGRAM
9Ø DATA 26,8Ø,142,128,Ø,166,132,
183,255,223,167,128,14Ø,224,Ø,39
,5,183,255,222,32,239,28,175,57
1ØØ FOR A=32512 TO 32536
11Ø READ CODE:POKE A,CODE
12Ø NEXT A
125 'PUT COCO INTO ALL RAM IF NE
CESSARY
13Ø A=PEEK(&HEØØØ):POKE&HEØØØ,55
:IF PEEK(&HEØØØ)<>55 THEN EXEC 3
2512
14Ø POKE&HEØØØ,A
15Ø VER$=CHR$(PEEK(&HC14Ø))+CHR$
(PEEK(&HC141))+CHR$(PEEK(&HC142)
)
16Ø IF VER$="1.Ø" THEN V=Ø:GOTO
19Ø
17Ø VER$=CHR$(PEEK(&HC153))+CHR$
(PEEK(&HC154))+CHR$(PEEK(&HC155)
)
18Ø IF VER$<>"1.1" THEN PRINT"SO
RRY, DON'T RECOGNIZE YOUR DOS":S
TOP ELSE V=1
19Ø CLS
2ØØ DIM PT(22,1)
21Ø FOR I =Ø TO 22
22Ø READ PØ$,P1$
23Ø PT(I,Ø)=VAL("&H"+PØ$):PT(I,1
)=VAL("&H"+P1$):NEXT
24Ø ' DOS1.Ø,1.1 PATCH ADDRESSES
25Ø DATA C7Ø8,C735'GRANULES
26Ø DATA C78B,C7BB'GRANULES
27Ø DATA C7AØ,C7DØ'GRANULES
28Ø DATA C7BF,C7EF'GRANULES
29Ø DATA CC4C,CD26'GRANULES
3ØØ DATA CDD9,CEB5'GRANULES
31Ø DATA D35F,D44D'GRANULES
32Ø DATA D1BØ,D29D'TRACKS
33Ø DATA D572,D65F'TRACKS
34Ø DATA D595,D682'TRACK$
35Ø DATA D446,D534'TRACKS-1
36Ø DATA C72A,C75A'GRANULES (OR
1 IF OVERLAYING GAT)
37Ø DATA C8FC,C9AA'MAXIMUM ALLOW
ABLE DRIVE #
38Ø DATA CDC7,CEA3'MAXIMUM ALLOW
ABLE DRIVE #
39Ø DATA CDED,CEC9'MAXIMUM ALLOW
ABLE DRIVE #
4ØØ DATA D16D,D25A'MAXIMUM ALLOW
ABLE DRIVE #
41Ø DATA D43D,D52B'MAXIMUM ALLOW
ABLE DRIVE #
42Ø DATA C6A5,C6D2'MAXIMUM SECTO
R # FOR DIRECTORY
43Ø DATA C72D,C75D'GAT BASE ADDR
ESS
44Ø DATA CØ73,CØ7D'GAT BASE ADDR
ESS+Ø*(GRANS+6)
45Ø DATA CØ76,CØ8Ø'GAT BASE ADDR
ESS+1*(GRANS+6)
46Ø DATA CØ79,CØ83'GAT BASE ADDR
ESS+2*(GRANS+6)
47Ø DATA CØ7C,CØ86'GAT BASE ADDR
ESS+3*(GRANS+6)
48Ø IF V=Ø THEN  GS$="8ØØ" ELSE
INPUT"GAT STORAGE ADDRESS IN HEX
($8ØØ)";GS$
49Ø GS=VAL("&H"+GS$)
5ØØ IF GS=Ø THEN GS=&H8ØØ
51Ø INPUT"4Ø, 72 OR 8Ø TRACKS";T
RACKS
52Ø IF TRACKS<>4Ø AND TRACKS<>72
AND TRACKS<>8Ø THEN 51Ø
53Ø GRANS=(TRACKS-1)*2
54Ø FOR I=Ø TO 6
55Ø POKE PT(I,V),GRANS
56Ø NEXT
57Ø INPUT"PATCHING 'COOKING WITH
 COCO' DOS(y/N)";X$ :IF LEN(X$)=
Ø THEN X$="N"
58Ø X$=LEFT$(X$,1)
59Ø IF X$="n" THEN X$="N"
6ØØ IF X$<>"N" THEN65Ø
61Ø FOR I =7 TO 9
62Ø POKE PT(I,V),TRACKS
63Ø NEXT
64Ø POKE PT(1Ø,V),TRACKS-1
65Ø IF GS=&H8ØØ THEN 73Ø
66Ø 'CHANGE GAT STORAGE LOCATION
 AND INTIALIZATION
67Ø POKE PT(18,V),FNUB(GS):POKE
PT(18,V)+1,FNLB(GS)
68Ø FOR I=Ø TO 3
69Ø X=GS+I*(GRANS+6)
7ØØ POKE PT(19+I,V),FNUB(X):POKE
 PT(19+I,V)+1,FNLB(X):POKE X,Ø
71Ø NEXT
72Ø POKE PT(11,V),GRANS+6:GOTO81
Ø
73Ø IF TRACKS=8Ø THEN PRINT"8Ø T
RACK SYSTEMS MAY NOT HAVE   FILE
S OPEN SIMULTANEOUSLY ON 2  DIFF
ERENT DRIVES. DOING SO WILL GARB
LE YOUR DISKS. YOU MAY HAVE MULT
IPLE FILES OPEN ON THE SAME DRIV
E HOWEVER":MX=3:GOTO76Ø
74Ø INPUT"DO YOU WANT TO BE ABLE
 TO OPEN  FILES SIMULTANEOUSLY O
N         DIFFERENT DRIVES";A$
75Ø IF A$<>"Y" THEN MX=3 ELSE IF
 TRACKS=4Ø THEN MX=2 ELSE MX=1
76Ø PRINT"MAXIMUM LEGAL DRIVE NU
MBER IS"MX
77Ø FOR I=12 TO 16
78Ø POKE PT(I,V),MX
79Ø NEXT
8ØØ IF MX=3 THEN POKE PT(11,V),1
 ELSE POKE PT(11,V),GRANS+6
81Ø POKE PT(17,V),18
82Ø IF TRACKS<72 THEN 92Ø
83Ø 'REVISE GRANULE ALLOCATION R
OUTINE
84Ø DATA 34,1Ø,63,84,27,9,63,8Ø,
4C,81,8E,24,27,2Ø,F3,35,1Ø,1F,89
,2Ø,28
85Ø OS=V*&H3Ø
86Ø FOR I=&HC794 TO &HC7A8
87Ø READ V$
88Ø VP=VAL("&H"+V$)
89Ø POKE I+OS,VP
9ØØ NEXT
91Ø IF X$="Y" THEN PRINT"DON'T F
ORGET TO PATCH THE        REVISED
 DSKINI/BACKUP MAXIMUM    TRACK V
ALUE PER ARTICLE TEXT.
92Ø PRINT"ALL DONE"
```

Listing 3: 40-35

```
1 '   ----- LISTING #3 -----
1Ø CLS
2Ø CLEAR1ØØØ
3Ø PRINT@6,"4Ø TRACK -> 35 TRACK
"
4Ø PRINT@43,"CONVERSION":PRINT
5Ø INPUT"DRIVE NUMBER";DR
6Ø IF DR<Ø OR DR>3 THEN 1Ø
7Ø DSKI$ DR,17,2,A$,B$
8Ø IF LEFT$(A$,78)<>STRING$(78,2
55) THEN PRINT"DISK NOT EMPTY, C
ANNOT CONVERT":STOP
9Ø MID$(A$,69,1Ø)=STRING$(1Ø,Ø)
1ØØ DSKO$ DR,17,2,A$,B$
11Ø PRINT"DISK NOW COMPATIBLE WI
TH 35     TRACK RSDOS.  DISK NOW
 HAS"
12Ø PRINT FREE(DR);" GRANULES"
```

JOYSTICK UTILITY

The Joystick Fix-It

By John G. Williams

This program is a solution to a problem which plagues users of the standard Radio Shack joysticks. We all know the sticks are inexpensive and work well enough for most applications. However, they seem to be too sensitive near the center so it is easy to over-control screen objects.

Since I am an engineer on the F-16 fighter program, I'm aware of a method used in its flight control computer to cure that same problem in the aircraft. The pilot commands are received by the computer and shaped by a series of equations to provide the desired airplane motion.

The CoCo can do the same thing for us. All it takes is for the stick command to be multiplied by the absolute value of itself and divided by a constant to retain proper scaling. The stick will then have a slower response near center, but will still have quick action near the extremes.

Stik Fix is a demonstration of this capability. RUN the program and slowly move the right stick laterally to see the effect. The screen horizontal axis is the stic motion while the vertical axis is the modified command.

The Listing:

```
 10 PMODE4,1:SCREEN1,1:PCLS(5):CO
 LORØ,5
 15 LINE(128,Ø)-(128,191),PSET:LI
 NE(Ø,96)-(255,96),PSET
 20 X=3.2*(JOYSTK(Ø)-31.5)
 30 Y=(ABS(X)*X)/130
 40 PSET(128+X,96.5-Y)
 50 GOTO2Ø
```

FORTH

FORTH FORUM

by John Poxon

Preamble.

First, the bad news! No doubt you have been following the excellent series of FORTH articles by John Redmond, and look forward to future insightful missives from him. Unfortunately, John is unable to continue presenting articles for the time being, due to pressure of work.

Now for the good news. The FORTH articles will continue, since John has asked me to fill in for him for the time being. I will endeavour to keep up the high standard he has set for the column, and hope that you find the material I present interesting and useful.

For the purpose of simplicity and efficiency in presenting FORTH programmes, I will presume that you own copies of both Starting FORTH and John's A*FORTH (an excellent FORTH, I might add). Starting FORTH was reviewed in the June issue of Australian CoCo, and is the reference text and documentation for A*FORTH. If you wish to buy a copy of A*FORTH, phone John on 02-853751, or write to him at 23 Mirool st, WEST Ryde, NSW. 2114. I understand that A*FORTH sells for $35, which is excellent value when you consider that some "foreign" FORTH programs sell for well in excess of $100! Let me hasten to add that I have no financial interest in A*FORTH, but know value for money when I see it.

Notwithstanding the previous comments, if you own a different FORTH this column is still for you, since the most significant differences between FORTHs lie in editor commands, a difficulty which I'm certain you can live with.

Finally, my contributions to this column will reflect my interests: process control, interfacing, simulation, etc., to which FORTH, in conjunction with interfacing units such as the CoCoConnection, is ideally suited.

Today I will commence with a FORTH version of the initialisation commands for the MK1 CoCoConnection, and then proceed to discuss a simulation subroutine.

CoCoConnection.

Examination of the CoCo memory map reveals that the I/O port is mapped to locations $FF00 - $FFFF (decimal 65280 -65535). The CoCoConnection meets with the CoCo in this region, as follows:

FF60 Port A
FF61 Port A control register
FF62 Port B
FF63 Port B control register

The CoCoConnection is inserted into the cartridge port in the normal manner, including power off first! Following power-up (CoCoConnection first), CLOADM and EXECute A*FORTH; entering the editor by the command EDIT. (Note that if you leave the editor and re-enter using EDIT you will erase the contents of the editor). Use (EDIT) instead to retain your text.

The BASIC initialisation routine supplied with the CoCoConnection will not work if simply translated into FORTH. The FORTH routine (below) will get you going. We will look at this initialisation business more closely in the next article, especially since the MK II CoCoConnection will then be discussed. (The MK II is the one now on sale. G.)

Since this is our first time discussing the use of the A*FORTH, type it in as follows: (note that the menu of editor commands may be examined by typing H). Those of you who are familiar with the editor can of course do their own thing!

a) Press I to enter insert mode.
b) Type in the text.
c) Press (BREAK) to escape insert mode.
d) Type 0 to compile the FORTH. If errors unacceptable to the compiler exist in the text, an error message will show, and pressing any key will place the cursor at the fault.
e) If you wish to save the program to tape: (after readying the recorder);
i) press B to get to the bottom of text,
ii) press 2 to identify the bottom of text to be saved,
iii) press T to get to the top of text,
iv) press 1 to identify the top of text to be saved,
v) press 8, provide a name for the program, then (ENTER),
vi) you will subsequently be asked if you have finished. If so, press 9.

```
HEX
FF60 CONSTANT APT
FF61 CONSTANT ACT
FF62 CONSTANT BPT
FF63 CONSTANT BCT
: BOUT FF BPT C! 00 BCT C! FF BPT C! ;
: BIOUT F BPT C! 0 BCT C! F BPT C! ;
: AIN 00 APT C! 04 ACT C! ;
: SETPORTS BIOUT BOUT AIN ;
: A? APT C@ . ;
```

Note the compiling of addresses as constants. This leads to improved readability of subsequent commands.

Escape the editor using (BREAK), followed by a Y. Execute SETPORTS by typing it in and pressing (ENTER). You should find that all of the output (B) leds are lit,

indicating that each output bit is set. Now execute A?. This will read the value of the input (A) register. With no inputs the value is FF. You can of course redefine A? to operate in base 10 thus:

```
: A? APT C@ DECIMAL . HEX ;
```

If you use the led probe supplied with the equipment you can try placing inputs on various bits of the A port. A? will show that such signals are read into the CoCo.

That's all I'll do using the CoCoConnection for this month. If you own one you can of course investigate setting and resetting individual bits, via ideas in the manual supplied with it, but I will cover that, plus a brief review of the MK II CoCoConnection, (it looks really nice), interfacing to analogue signals and more, next month.

Simulation.

I have two purposes in talking about this: first, in exploring the model we will examine ways of achieving the necessary computations, and second, some of you may be interested in simulation. By the way: if you are, your comments (and critiscisms) are most welcome, either through this column or directly to me.

I have recently become involved in a project to simulate the behaviour of a large boiler supplying steam to a turbine, for the purpose of training apprentices and others. The object of the simulation is to provide a computer based model which can be controlled by a suitable controller external to the computer. By this means a safe, cost effective and compact training aid will be available to assist instruction in the art and science of tuning controllers. If this model is successful, other models of industrial processes will be prepared.

Let's look at one part of the model: the steam boiler. It is envisaged to be a horizontal cylindrical drum with convex hemispherical ends. The boiler has a parallel section 6 m long, with a diameter of 1 m.

A little thought and some mathematics created the equation

```
Q=(11*PI/3)+6(d*SQRT(1-d*d)+ARCSIN(d))-PI*((d*d*d/3)-d)
```

Where Q=volume of liquid in the tank (presumed cold water for the moment).

PI=a constant.

d=deviation from midline (centre-line) of liquid level; i.e. a deviation of 0 means that the tank is half full, a deviation of 1 means that it is full, and a deviation of -1 means that it is empty. (units are metres).

This equation is unsolvable by "ordinary algebra", and numerical analysis must be employed. Specifically, the Newton-Rapson method, an iteritive or recursive technique, will find the deviation given a known value of liquid in the drum, (if the equation is set = to zero). i.e.

```
-Q+(11*PI/3)+6(d*SQRT(1-d*d)+ARCSIN(d))-PI*((d*d*d/3)-d)=0
(EQU 1)
```

Basically, the method relies on putting an initial guess into an equation which will then calculate a new "guess" which is (much) closer to the correct value. This new value is then put into the equation and the process repeated until a value acceptably close to the preceeding value is calculated. Mathematically the process can be represented by

d2=d1-(f(d1))/f'(d1))

Where d2=new value of guess.
d1=old value ofguess.
f=equation 1 (above).
f'=first derivative of equation 1

I should apologise to all the mathematicians for my presentation of equations. Graham has asked me not to use the imbedded control character set for my printer (which would have made the equations look nice) because his printer, being different, wont respond to them.

I have set out the routine in program form so that you can try it out. The BASIC enthusiasts out there may find the BASIC listing interesting, and may even care to create graphics routines to display the water level in the drum. They will find, possibly much to their dismay, that BASIC is very slow in this task, since BASIC re-interpretes each command as it comes to it, no matter how many times before that it has been around the loop. In fact, BASIC is so slow that one would not (or rather, could not) seriously consider using it for simulation, quite apart from its other negative aspects.

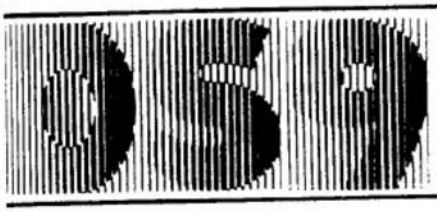
Just below the BASIC listing is a PASCAL listing. This will, of course, run much faster, since it is a compiled language, especially if run in TURBO-PASCAL. Try it, if you have PASCAL. Now consider: how on Earth can one possibly get information from the CoCoConnection using PASCAL? It can be done, but it's not easy.

Note that both BASIC and PASCAL require ARCTAN as a preliminary to calculating ARCSIN. In FORTH one can directly calculate ARCSIN (after a little bit of mucking about). I'll re-phrase that: with certain reservations, after a bit of mucking about!

BASIC listing.

```
10 CLS
20 PRINT@352, " INPUT WATER FRACTION OF BOILER":INPUT X
30 CLS:PRINT:PRINT" WATER FRACTION="X
40 Q=X*23.038
50 GOSUB70
60 PRINT@224,"DEVIATION = "D2" METRES":
   L=((1000+(D2*1000))*100/2000:PRINT@290,"LEVEL % = "L":
   D2=0:GOTO20
70 D1=0
80 '***ROOT***
90 D1=D2
100 PRINT@167,"D1 = "D1" METRES"
110 A1=ATN(D1/SQR(-D1*D1+1))
120 V1=11.519-Q+6*(D1*SQR(1-D1*D1)+A1)-3.142*(((D1*D1*D1/
    3)-D1)
130 V2=3.142*(1-D1*D1)+((-12*D1*D1+12)/(SQR(1-D1*D1)))
140 D2=D1-(V1/V2)
150 IF ABS(D2-D1) <1E-6 THEN RETURN
160 GOTO90
170 END
```

Continued on P 65 ...



MAILLISTFILEMAN

by Tony Ceneviva

Tony continues with a further section of his program for the management of a mailing list.

The Listing:

```
PROCEDURE PRINTTRANSFER
0000        DIM RESP:STRING[1]
000C        DIM DRIVENOTO,DRIVENOFRO:STRING[4]
001C        DIM PRINTERPATH:BYTE
0023        DIM PPATH:STRING
002A        PPATH="/P"
0033        DIM PATH:BYTE
003A        DIM WPATH:BYTE
0041        DIM MENUSELECT:BYTE
0048        DIM PRINTMENUSELECT:BYTE
004F        DIM EOFSW:BOOLEAN
0056        DIM SORTMENUSELECT:BYTE
005D        DIM NEWFILE:STRING
0064        DIM DELETEFILE:STRING
006B        TYPE BLOCKTYPE=SALUT:STRING[9]; NAME:STRING[15]; HOUSENO:STRING
             [5]; STREET:STRING[19]; SUBURB:STRING[16]; TELEPH:STRING
             [10]; FEES:STRING[5]
00BD        DIM BLOCK(150):BLOCKTYPE
00CB        DIM FILEPRINT:STRING
00D2        DIM ONEHUNDRED:REAL
00D9        DIM FIELDTOPRINT:STRING
00E0        DIM TRANSFERFIELD:STRING
00E7        DIM STRINGSIZE:BYTE
00EE        DIM FILEFROM:STRING
00F5        DIM FILETO:STRING
00FC 1
0100        REM ****** PRINT AND DATA MAINIPULATION MENU
012B        PRINT CHR$(12)
0130        PRINT "PRINT & DATA TRANSFER MENU"
014E        PRINT " (1) CREATE NEW FILE"
0166        PRINT " (2) DELETE FILE"
017A        PRINT " (3) LINE PRINTER MENU"
0194        PRINT " (4) TRANSFER RECORDS TO OTHER "
01B7        PRINT "     FILES"
01C5        PRINT " (5) DISPLAY MENU"
01DA        PRINT " (6) DELETE MENU "
01EF        PRINT " (7) FUTURE EXPANSION"
0208        PRINT " (8) END"
0214        MENUSELECT=10
```

```
021B        WHILE MENUSELECT>8 OR MENUSELECT<1 DO
022E          INPUT " MENU SELECTION? ",MENUSELECT
0247        ENDWHILE
024B        ON MENUSELECT GOTO 10,20,30,40,50,60,70,80
0272        GOTO 1
0276 10
027A        PRINT CHR$(12)
027F        PRINT USING "S25^","CREATE NEW FILE"
029A        INPUT "NAME OF NEW FILE? ",NEWFILE
02B4        CREATE #PATH,NEWFILE:UPDATE
02C0        CLOSE #PATH
02C6        GOTO 1
02CA 20
02CE        PRINT CHR$(12)
02D3        PRINT USING "S25^","DELETE OLD FILE"
02EE        INPUT "NAME & PATH OF FILE TO BE DELETED",DELETEFILE
0317        DELETE DELETEFILE
031C        GOTO 1
0320 30
0324        PRINT CHR$(12)
0329        PRINT USING "S25^","PRINT MENU"
033F        PRINT " (1) PRINT COMPLETE FILE"
035B        PRINT " (2) PRINT SELECTED RECORDS"
037A        PRINT " (3) RETURN TO MENU"
0391        PRINTMENUSELECT=20
0398        WHILE PRINTMENUSELECT>4 OR PRINTMENUSELECT<1 DO
03AB          INPUT "SELECTION? ",PRINTMENUSELECT
03BE        ENDWHILE
03C2        IF PRINTMENUSELECT=1 THEN 31
03D1        IF PRINTMENUSELECT=2 THEN 32
03E0        IF PRINTMENUSELECT=3 THEN 1
03EF        GOTO 30
03F3 31
03F7        PRINT CHR$(12)
03FC        PRINT USING "S25^","PRINTING COMPLETE FILE"
041E        OPEN #PRINTERPATH,PPATH:WRITE
042A        INPUT " FILE TO BE PRINTED? ",FILEPRINT
0447        OPEN #PATH,FILEPRINT:READ
0453        SEEK #PATH,0
045C        X=1
0464        REM ** MAY BE CHANGED IF DIM BLOCK CHANGED
048D        ONEHUNDRED=150
0495        LOOP
0497        EXITIF ONEHUNDRED<150 THEN ENDEXIT
04A7          X=1
04AF          LOOP
04B1          EXITIF X>ONEHUNDRED THEN ENDEXIT
04C1          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
04D4          ENDEXIT
04D8            GET #PATH,BLOCK(X)
04E7            X=X+1
04F3          ENDLOOP
04F7          X=1
04FF          LOOP
0501          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
0515            STRINGSIZE=LEN(BLOCK(X).NAME)
0525            BLOCK(X).NAME=LEFT$(BLOCK(X).NAME,STRINGSIZE-1)
0543            STRINGSIZE=LEN(BLOCK(X).STREET)
0553            BLOCK(X).STREET=LEFT$(BLOCK(X).STREET,STRINGSIZE-1)
0571            STRINGSIZE=LEN(BLOCK(X).SUBURB)
0581            BLOCK(X).SUBURB=LEFT$(BLOCK(X).SUBURB,STRINGSIZE-1)
```

```
059F            STRINGSIZE=LEN(BLOCK(X).TELEPH)
05AF            BLOCK(X).TELEPH=LEFT$(BLOCK(X).TELEPH,STRINGSIZE-1)
05CD            PRINT #PRINTERPATH,TAB(4); BLOCK(X).SALUT; BLOCK(X).NAME

                 ;
05EE            PRINT #PRINTERPATH USING "T26,S4>",BLOCK(X).HOUSENO;
060A            PRINT #PRINTERPATH,TAB(31); BLOCK(X).STREET; TAB(50); BLOCK

                 (X).SUBURB; TAB(68); BLOCK(X).TELEPH; TAB(76); BLOCK

                 (X).FEES
064C            X=X+1
0658          ENDLOOP
065C        ENDLOOP
0660        PRINT #PRINTERPATH; "      END OF FILE"
0679        CLOSE #PRINTERPATH
067F        GOTO 30
0683 32
0687        PRINT CHR$(12)
068C        PRINT USING "S25^","PRINTING SELECTED BLOCKS"
06B0        FIELDTOPRINT=""
06B7        INPUT " FILE TO BE PRINTED? ",FILEPRINT
06D4        INPUT " BLOCK TO BE PRINTED -ANY FIELD  ",FIELDTOPRINT
06FD        OPEN #PRINTERPATH,PPATH:WRITE
0709        OPEN #PATH,FILEPRINT:READ
0715        SEEK #PATH,0
071E        X=1
0726        REM ** MAY BE CHANGED IF DIM BLOCK CHANGED
074F        ONEHUNDRED=150
0757        FIELDTOPRINT=FIELDTOPRINT+CHR$($0D)
0764        LOOP
0766        EXITIF ONEHUNDRED<150 THEN ENDEXIT
0776          LOOP
0778          EXITIF X>ONEHUNDRED THEN ENDEXIT
0788          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
079B          ENDEXIT
079F            BLOCK(X).NAME=""
07AE            BLOCK(X).HOUSENO=""
07BD            BLOCK(X).STREET=""
07CC            BLOCK(X).SUBURB=""
07DB            GET #PATH,BLOCK(X)
07EA            IF BLOCK(X).NAME=FIELDTOPRINT THEN X=X+1
0809            ENDIF
080B            IF BLOCK(X).STREET=FIELDTOPRINT THEN X=X+1
082A            ENDIF
082C            IF BLOCK(X).SUBURB=FIELDTOPRINT THEN X=X+1
084B            ENDIF
084D          ENDLOOP
0851          X=1
0859          LOOP
085B          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
086F            STRINGSIZE=LEN(BLOCK(X).NAME)
087F            BLOCK(X).NAME=LEFT$(BLOCK(X).NAME,STRINGSIZE-1)
089D            STRINGSIZE=LEN(BLOCK(X).STREET)
08AD            BLOCK(X).STREET=LEFT$(BLOCK(X).STREET,STRINGSIZE-1)
08CB            STRINGSIZE=LEN(BLOCK(X).SUBURB)
08DB            BLOCK(X).SUBURB=LEFT$(BLOCK(X).SUBURB,STRINGSIZE-1)
08F9            STRINGSIZE=LEN(BLOCK(X).TELEPH)
0909            BLOCK(X).TELEPH=LEFT$(BLOCK(X).TELEPH,STRINGSIZE-1)
0927            PRINT #PRINTERPATH,TAB(4); BLOCK(X).SALUT; BLOCK(X).NAME
```

```
                ;
0948            PRINT #PRINTERPATH USING "T26,S4>",BLOCK(X).HOUSENO;
0964            PRINT #PRINTERPATH,TAB(31); BLOCK(X).STREET; TAB(50); BLOCK
                (X).SUBURB; TAB(68); BLOCK(X).TELEPH; TAB(76); BLOCK
                (X).FEES
09A6            X=X+1
09B2          ENDLOOP
09B6        ENDLOOP
09BA        PRINT #PRINTERPATH; "      END OF FILE"
09D3        CLOSE #PRINTERPATH
09D9        GOTO 30
09DD 40
09E1        PRINT CHR$(12)
09E6        PRINT USING "S25^","TRANSFER RECORDS TO OTHER FILES"
0A11        TRANSFERFIELD=""
0A18        INPUT " FILE TO TRANSFER FROM?  ",FILEFROM
0A39        INPUT " FILE TO TRANSFER TO?  ",FILETO
0A58        PRINT "BLOCKS TO BE TRANSFEREED:-"
0A76        INPUT "ANY FIELD OR <ENTER>FOR ALL REC. ",TRANSFERFIELD
0A9F        OPEN #WPATH,FILETO:UPDATE
0AAB        OPEN #PATH,FILEFROM:READ
0AB7        PATHPT=0
0ABF        WPATHPT=0
0AC7        SEEK #WPATH,0
0AD0        LOOP
0AD2        EXITIF EOF(#WPATH)=TRUE THEN ENDEXIT
0AE1          GET #WPATH,BLOCK(1)
0AEE          WPATHPT=WPATHPT+79
0AFA        ENDLOOP
0AFE        IF WPATHPT>90000. THEN
0B0E          INPUT "NO ROOM ON DISK <ENTER> ",RESP
0B2E          CLOSE #WPATH
0B34          CLOSE #PATH
0B3A          GOTO 1
0B3E        ENDIF
0B40        X=1
0B48        ONEHUNDRED=150
0B50        EOFSW=FALSE
0B56        TRANSFERFIELD=TRANSFERFIELD+CHR$($0D)
0B63        LOOP
0B65        EXITIF EOFSW=TRUE THEN ENDEXIT
0B73          SEEK #PATH,PATHPT
0B7D          X=1
0B85          LOOP
0B87          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
0B9B          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
0BAE            EOFSW=TRUE
0BB4          ENDEXIT
0BB8            BLOCK(X).NAME=""
0BC7            BLOCK(X).HOUSENO=""
0BD6            BLOCK(X).STREET=""
0BE5            BLOCK(X).SUBURB=""
0BF4            GET #PATH,BLOCK(X)
0C03            PATHPT=PATHPT+79
0C0F            IF TRANSFERFIELD<>CHR$($0D) THEN
0C1D              IF BLOCK(X).NAME=TRANSFERFIELD THEN X=X+1
0C3C              ENDIF
0C3E              IF BLOCK(X).STREET=TRANSFERFIELD THEN X=X+1
0C5D              ENDIF
```

```
0C5F              IF BLOCK(X).SUBURB=TRANSFERFIELD THEN X=X+1
0C7E              ENDIF
0C80            ELSE X=X+1
0C8F            ENDIF
0C91          ENDLOOP
0C95          X=1
0C9D          SEEK #WPATH,WPATHPT
0CA7          LOOP
0CA9          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
0CBD            PUT #WPATH,BLOCK(X)
0CCC            WPATHPT=WPATHPT+79
0CD8            PRINT BLOCK(X).NAME
0CE4            X=X+1
0CF0          ENDLOOP
0CF4        ENDLOOP
0CF8        CLOSE #PATH
0CFE        CLOSE #WPATH
0D04        INPUT "TRANSFER COMPLETE <ENTER> ",RESP
0D26        GOTO 1
0D2A 50
0D2E        PRINT CHR$(12)
0D33        PRINT USING "S25^","DISPLAY MENU"
0D4B        PRINT " (1) DISPLAY COMPLETE FILE"
0D69        PRINT " (2) DISPLAY SELECTED RECORDS"
0D8A        PRINT " (3) RETURN TO MAIN MENU"
0DA6        PRINTMENUSELECT=20
0DAD        WHILE PRINTMENUSELECT>4 OR PRINTMENUSELECT<1 DO
0DC0          INPUT "SELECTION? ",PRINTMENUSELECT
0DD3        ENDWHILE
0DD7        IF PRINTMENUSELECT=1 THEN 51
0DE6        IF PRINTMENUSELECT=2 THEN 52
0DF5        IF PRINTMENUSELECT=3 THEN 1
0E04        GOTO 50
0E08 51
0E0C        PRINT CHR$(12)
0E11        PRINT USING "S25^","DISPLAY COMPLETE FILE"
0E32        INPUT " FILE TO BE DISPLAYED? ",FILEPRINT
0E51        OPEN #PATH,FILEPRINT:READ
0E5D        SEEK #PATH,0
0E66        X=1
0E6E        REM ** MAY BE CHANGED IF DIM BLOCK CHANGED
0E97        ONEHUNDRED=150
0E9F        LOOP
0EA1        EXITIF ONEHUNDRED<150 THEN ENDEXIT
0EB1          X=1
0EB9          LOOP
0EBB          EXITIF X>ONEHUNDRED THEN ENDEXIT
0ECB          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
0EDE          ENDEXIT
0EE2            GET #PATH,BLOCK(X)
0EF1            X=X+1
0EFD          ENDLOOP
0F01          X=1
0F09          LOOP
0F0B          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
0F1F            PRINT BLOCK(X).SALUT; "  "; BLOCK(X).NAME
0F3B            PRINT USING "S4,>",BLOCK(X).HOUSENO;
0F50            PRINT TAB(6); BLOCK(X).STREET
0F60            PRINT BLOCK(X).SUBURB
0F6C            X=X+1
0F78            FOR Z=1 TO 1000
```

```
0F8B            NEXT Z
0F96          ENDLOOP
0F9A        ENDLOOP
0F9E        PRINT "      END OF FILE"
0FB2        GOTO 50
0FB6 52
0FBA        PRINT CHR$(12)
0FBF        PRINT USING "S25^","DISPLAYING SELECTED BLOCKS"
0FE5        FIELDTOPRINT=""
0FEC        INPUT " FILE TO BE DISPLAYED? ",FILEPRINT
100B        INPUT " BLOCK TO BE DISPLAYED -ANY FIELD  ",FIELDTOPRINT
1036        OPEN #PATH,FILEPRINT:READ
1042        SEEK #PATH,0
104B        X=1
1053        REM ** MAY BE CHANGED IF DIM BLOCK CHANGED
107C        ONEHUNDRED=150
1084        FIELDTOPRINT=FIELDTOPRINT+CHR$($0D)
1091        LOOP
1093        EXITIF ONEHUNDRED<150 THEN ENDEXIT
10A3          LOOP
10A5          EXITIF X>ONEHUNDRED THEN ENDEXIT
10B5          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
10C8          ENDEXIT
10CC            BLOCK(X).NAME=""
10DB            BLOCK(X).HOUSENO=""
10EA            BLOCK(X).STREET=""
10F9            BLOCK(X).SUBURB=""
1108            GET #PATH,BLOCK(X)
1117            IF BLOCK(X).NAME=FIELDTOPRINT THEN X=X+1
1136            ENDIF
1138            IF BLOCK(X).STREET=FIELDTOPRINT THEN X=X+1
1157            ENDIF
1159            IF BLOCK(X).SUBURB=FIELDTOPRINT THEN X=X+1
1178            ENDIF
117A          ENDLOOP
117E          X=1
1186          LOOP
1188          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
119C            PRINT BLOCK(X).SALUT; "  "; BLOCK(X).NAME
11B8            PRINT USING "S4,>"; BLOCK(X).HOUSENO;
11CD            PRINT TAB(6); BLOCK(X).STREET
11DD            PRINT BLOCK(X).SUBURB
11E9            FOR Z=1 TO 1000
11FC            NEXT Z
1207            X=X+1
1213          ENDLOOP
1217        ENDLOOP
121B        PRINT "      END OF FILE"
122F        GOTO 50
1233 60
1237        PRINT CHR$(12)
123C        PRINT USING "S32^","DELETE RECORDS"
1256        PRINT "FILE TO DELETE ITEMS:-"
1270        INPUT "DRIVE NUMBER - FORMAT /D1/ ",DRIVENOFRO
1293        INPUT "FILE NAME ",FILEFROM
12A5        INPUT "DRIVE N0. FOR TEMPORARY TRANSFER ",DRIVENOTO
12CE        INPUT "BLOCKS TO BE DELETED:- ",TRANSFERFIELD
12ED        FILETO=DRIVENOTO+FILEFROM+"TEMP"
1300        FILEFROM=DRIVENOFRO+FILEFROM
130C        OPEN #PATH,FILEFROM:UPDATE
1318        CREATE #WPATH,FILETO:UPDATE
```

```
1324        PATHPT=0
132C        WPATHPT=0
1334        ONEHUNDRED=150
133C        EOFSW=FALSE
1342        TRANSFERFIELD=TRANSFERFIELD+CHR$($0D)
134F        LOOP
1351        EXITIF EOFSW=TRUE THEN ENDEXIT
135F          SEEK #PATH,PATHPT
1369          X=1
1371          LOOP
1373          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
1387          EXITIF EOF(#PATH)=TRUE THEN ONEHUNDRED=X
139A            EOFSW=TRUE
13A0          ENDEXIT
13A4            GET #PATH,BLOCK(X)
13B3            PATHPT=PATHPT+79
13BF            IF BLOCK(X).NAME=TRANSFERFIELD THEN X=X-1
13DE              PRINT BLOCK(X+1).NAME
13EE              GOTO 62
13F2            ENDIF
13F4            IF BLOCK(X).STREET=TRANSFERFIELD THEN X=X-1
1413              PRINT BLOCK(X+1).NAME
1423              GOTO 62
1427            ENDIF
1429            IF BLOCK(X).SUBURB=TRANSFERFIELD THEN X=X-1
1448              PRINT BLOCK(X+1).NAME
1458              GOTO 62
145C            ENDIF
145E 62
1462            X=X+1
146E          ENDLOOP
1472          X=1
147A          SEEK #WPATH,WPATHPT
1484          LOOP
1486          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
149A            PUT #WPATH,BLOCK(X)
14A9            WPATHPT=WPATHPT+79
14B5            X=X+1
14C1          ENDLOOP
14C5        ENDLOOP
14C9        DELETE FILEFROM
14CE        CLOSE #PATH
14D4        CREATE #PATH,FILEFROM:WRITE
14E0        PATHPT=0
14E8        WPATHPT=0
14F0        ONEHUNDRED=150
14F8        EOFSW=FALSE
14FE        LOOP
1500        EXITIF EOFSW=TRUE THEN ENDEXIT
150E          SEEK #PATH,PATHPT
1518          SEEK #WPATH,WPATHPT
1522          X=1
152A          LOOP
152C          EXITIF X>ONEHUNDRED-1 THEN ENDEXIT
1540            GET #WPATH,BLOCK(X)
154F          EXITIF EOF(#WPATH)=TRUE THEN
155B            ONEHUNDRED=X
1563            EOFSW=TRUE
1569          ENDEXIT
156D            WPATHPT=WPATHPT+79
1579            X=X+1
```

```
1585          ENDLOOP
1589          X=1
1591          SEEK #PATH,PATHPT
159B          LOOP
159D          EXITIF X>ONEHUNDRED THEN ENDEXIT
15AD            PUT #PATH,BLOCK(X)
15BC            PATHPT=PATHPT+79
15C8            X=X+1
15D4          ENDLOOP
15D8        ENDLOOP
15DC        DELETE FILETO
15E1        CLOSE #WPATH
15E7        CLOSE #PATH
15ED        GOTO 1
15F1 70
15F5        GOTO 1
15F9 80
15FD        END
```

FRICKER'S FOLLIES

by Jack Fricker

Welcome to a series of articles in which we will try to clear up some of your questions on the use of OS9 and BASIC09. From the start we will assume that you have at least a working knowledge of Disk BASIC.

The first thing to do is get a working system disk with the necessary files and utilities to get OS9 to work for you.

We will also assume that you have already created a backup system disk and stored the original in a safe place.

OS9 will come up in lower case if you add the following line to your startup file.

```
TMODE .1 -UPC
```

The next thing to do is to fix the errors in the baud rate, the two listings will work for versions 1.0 and 1.1. OS9 will tell you which version you have when you first boot.

When you have typed in the listing you will need to make a new system disk with the files corrected. Type in the following to save the files you have just altered.

```
Save /d0/newrs232 rs232
Verify U </d0/newrs232 >/d0/rs232
```

Use the same procedure to save the updated printer file.

Your module directory (the files in memory now) will probably look very like this.

```
   Module Directory at 10:14:58
Os9      Os9p2    Init
Boot     CCDisk   D0
D1       D2       D3
Term     Ioman    Rbf
Scf      Sys6o    Clock
Shell    P        Pipe
Piper    Pipeman  Printer
Rs232    T1       T2
Aciapak  Mdir
```

The above Mdir is for version 1.1. Unless you are running a terminal or a direct connect modem you do not need t1, t2 or Aciapak, if you only have 2 drives you also don't need D2 and D3.

To create a new system disk use the following as a guide.

```
  Save Temp1 os9 os9p2 init boot ccdisk
  Save Temp2 d0 d1 term ioman rbf scf
  Save temp3 sysgo clock shell p pipe
piper pipeman
```

Do NOT save either rs232 or printer. THen type the following.

```
Build Bootlist
/d0/temp1
/d0/temp2
/d0/temp3
/d0/rs232
/d0/printer
```

Now use format to create a blank disk in drive 1. Then use "os9gen /d1 </d0/Bootlist" to create a new disk that can be booted from Disk Colour Basic. Don't type in the quotes.

You may be wondering where this is all leading to. Well if you have only one SSDD or even one DSDD you may find that all the system files may not fit on your disk or may not leave you any room to do any work on the one drive.

If you have one or more DSDDs then the most popular Solution seems to be a program called SDISK. This excellent program is available from various software vendors(Qld Colour software or Paris Radio) who advertise in this magazine and costs about $60.

The other soloution is to try to write your own disk device descriptors to do the job. I know of at least one person who has done this (hello Ron).

If you have 40 track DSDD & the appropriate device drivers you can more than double the storage capacity or you may choose to go to a second 80 track DSDD and more than quadruple the storage of a 35 track SSDD. If you choose the latter I would suggest that you have at least one 40 track DSDD to maintain compatability with the standard format of 35 track SSDD.

A 40 track DD will read & write to a 35 track disk , but not necessarily the other way around.

If you choose 80 track drives you will find that you should be able to fit all you system files on the disk and use the second drive for Data Storage(yes you do do more than just moving and listing files). All this, in a round about way brings us back to the topic under discussion the deletion of unwanted files.

For example if you are using BACIC09 it is highly unlikely that you will have any use for the following files and directories in your bacic09 system disk. You wouldn't need the defs directory, but you will need that on your assembly language disk. You won't need the sys files except for motd & errmsg. Errmsg is the file that PRINTERRlooks for when an error is generated and prints an expanded message on your screen. Errmsg must reside on drive 0.

If you have a printer you may prefer to list errmsg to the printer and keep it near the terminal. The list is also in the rear of the red commands manual(you did remember to read it at least twice).

Next you use DSAVE to create a file to copy the entire disk.

```
DSAVE /d0 >/d0/copylist
```

Use the editor to remove references in the copylist to programs that you do not wish copied over to the new disks. Then use the entire procedure to create a new disk for Basic09 , C and assembler, pascal or anything else.

Listing 1:

Version 1.1

```
DEBUG
L PRINTER
. .+65
=04
=82
=01
=A2
=00
=CD
=00
=63
=00
=2D
=00
=13
=00
=05
L PRINTER
. .+9C
=12
=C6
=00
=59
=58
L PRINTER
. .+AA
=F2
L PRINTER
. .+11F
=99
=C0
=94
L RS232
. .+B2
=04
=82
=01
=A2
=00
=CD
=00
=63
=00
=2D
=00
=13
=00
=05
L RS232
. .+E1
=12
=C6
=00
=59
=58
L RS232
. .+EF
=42
L RS232
. .+11B
=AC
=FC
=FA
0
```

Listing 2:

Version 1.1

```
DEBUG
L PRINTER
. .+65
=04
=82
=01
=A2
=00
=CD
=00
=63
=00
=2D
=00
=13
=00
=05
L PRINTER
. .+A5
=12
=C6
=00
=59
=58
L PRINTER
. .+BB
=F2
L RS232
. .+72
=04
=82
=01
=A2
=00
=CD
=00
=63
=00
=20
=00
=13
=00
=05
L RS232
. .+B2
=12
=C6
=00
=59
=58
L RS232
. .+BA
=F2
0
```

... From P 41

```
717Ø PRINT:PRINT" PRESS ANY KEY
TO CONTINUE."
718Ø K$=INKEY$
719Ø IFK$=""THEN718ØELSERETURN
1ØØØØ PMODE4,1
1ØØ1Ø COLORØ,1
1ØØ3Ø PCLS1
1ØØ4Ø LINE(4,16)-(252,176),PSET,
B
1ØØ5Ø LINE(128,16)-(128,176),PSE
T
1ØØ6Ø LINE(4,52)-(4Ø,14Ø),PSET,B
1ØØ7Ø LINE(4,76)-(16,116),PSET,B
1ØØ8Ø CIRCLE(128,96),2Ø,Ø
1ØØ9Ø CIRCLE(28,96),2,Ø
1Ø1ØØ CIRCLE(28,96),2Ø,Ø,1,.875,
.125
1Ø11Ø LINE(Ø,84)-(4,88),PSET,BF
1Ø12Ø LINE(Ø,1Ø4)-(4,1Ø8),PSET,B
F
1Ø13Ø LINE(Ø,84)-(Ø,1Ø8),PSET
1Ø14Ø LINE(256,84)-(256,1Ø8),PSE
T
1Ø15Ø LINE(256,1Ø4)-(252,1Ø8),PS
ET,BF
1Ø16Ø LINE(256,84)-(252,88),PSET
,BF
1Ø17Ø CIRCLE(228,96),2Ø,Ø,1,.375
,.625
1Ø18Ø CIRCLE(228,96),2,Ø
1Ø19Ø LINE(252,76)-(24Ø,116),PSE
T,B
1Ø2ØØ LINE(252,52)-(216,14Ø),PSE
T,B
1Ø21Ø CIRCLE(4,16),8,Ø,1,Ø,.25
1Ø22Ø CIRCLE(252,16),8,Ø,1,.25,.
5Ø
1Ø23Ø CIRCLE(252,176),8,Ø,1,.5,.
75
1Ø24Ø CIRCLE(4,176),8,Ø,1,.75,Ø
1Ø245 SCREEN1,Ø
1Ø25Ø FOR T=1 TO 1ØØØØ:NEXT
1Ø26Ø RETURN
```

... From P 56

PASCAL listing.

```
1  PROGRAM TRY1(input,output);
2  VAR
3  d1,d2,Q,X,V1,V2,ARCSIND1,LEVEL:REAL;
4
5  BEGIN
6    WRITE('ENTER THE WATER FRACTION OF THE BOILER: ');
7    READLN(X);
8    d1:=(X-0.5)*2;
9    WRITELN('d1=',d1);
10   Q:=X*23.038;
11   REPEAT
12   d1:=d2;
13   ARCSIND1:=ARCTAN(d1/SQRT(-d1*d1+1));
14   V1:=11.519-Q+6*(d1*SQRT(1-d1*d1)+ARCSIND1)-3.142*((
     d1*d1*d1)/3)-d1);
15   V2:=3.142*(1-d1*d1)+((-12*d1*d1*d1+12)/(SQRT(1-d1*d1
     )));
16   d2:=d1-(V1/V2);
17   UNTIL ABS(d2-d1)<1E-6;
18   WRITELN('d2= ',d2);
19   LEVEL:=((1000+(d2*1000))*100)/2000;
20   WRITELN('LEVEL % := ',ROUND(LEVEL));
21   END
```

Now, having given the oppositon a go, let's consider FORTH. Before we rush into writing a FORTH program, notice that these programs require mathematical functions such as ARCSIN X , SQRT X, etc.. FORTH does'nt have them, so we must create them. Noting also that John has already presented a routine for SQRT, we can ignore it and look at ARCSIN.

Since I am just about out of time for this month, It is expedient for me to pose the question of getting ARCSIN to you. How would you do it? Why not send in your method of getting ARCSIN? Hopefully it will be printed in next months FORTH FORUM!

Please feel free to contact me on (07) 2080893.

John Poxon.

CORRECTIONS

"Animatic: Automatic Animation" (July 1985, P 30) : There has been some confusion about the *Rocket* and *Promnade* programs. These programs (and any other BASIC programs that use the new ANIM command) should always be saved in ASCII; for example, you can use the command CSAVE "ROCKET",A.

If you have saved programs in normal BASIC format (or are using the *Rocket* and *Promnade* files from the June 1985 RAINBOW ON TAPE), load the program, save it in ASCII and use the ASCII copy from now on.

Rita Sabo also tells us that a bug in the main program has been called to her attention. If you typed the *Animatic* program (Listing 4) in by hand, insert a new Line 1551 as follows and reassemble the program.

```
1551          JSR     $A976
```

If you don't have the source code on tape or disk, BASIC programs can be patched by inserting an EXEC &HA976 in a part of the program that comes before any ANIM commands. This is only necessary for programs that use keyboard input to control movement.

In the *Promnade* program (Listing 2), insert this line:

```
510 IF FL=1 THEN EXEC &HA976:GOTO 40
```

"Cardio" (May 1985, P 17) : Rene St. Jacques informs us that he has been getting reports of an FC Error occurring in Line 5010. He passes along the following description to explain how that portion of the program works:

"Once you have determined the string of characters for each letter in a dimensioned array and that you are in graphics mode, you have to place the word to be drawn in A$ and the position, scale and color of the word in OP$. Then the subroutine in Line 5000 will add together the strings of characters for each letter of the word and will draw them. For example, the first three instructions in Line 140 serve to draw the word "CARDIO" at the position X=8, Y=40 at the scale of 14 (see the DRAW instruction). I suggest to your readers to do a TRON to see where the error comes from and I suggest to them to be especially careful about the ';' character."

Rene also tells us there are two errors in Line 1070. First, the word GOOTO should be changed to GOTO, and the last command in the line should be GOSUB 6110 instead of GOSUB 6080. He also says that Line 1220 can be omitted.

... From P 5

PRINT #-2,

Australian Rainbow. $185.00 ea.

* We have back issues of all our magazines and tapes in stock too.

Reference to this magazine, "Network Quarterly", "CoCoBug", our companion magazine "Australian CoCo", and the Tandy Computer catalogue will readily show that support for Tandy products is very strong.

This magazine has made the point before, but it is worth repeating - Tandy's CoCo, unlike so many other computers which appeared five years ago, is still strong because it is very capable.

There is very little a CoCo can't do!

This month I will be in Melbourne for the week beginning 16/9. I will be at Jeff Sheen's User Group meeting on the Wednesday night, and possibly one or two others. You can phone Jeff or myself for further details. I am looking forward to meeting with you folk in Melbourne.

Finally, just a note about the new Rainbow Ware shown a page or two back. These new garments are hand dyed and in mixed pastel colours which run vertically. Its unfortunate that we can't show these in colour, because they really do look the part!

All the previous patterns are still available for another month as well.

Graham.

LETTERS

... From P 6

I've always felt that a column like the one you suggested would be too self-indulgent. We have our problems, but then you probably have yours too, we're trying to entertain you - not bore you!

I could tell you, for example, about the very unsatisfactory Mastercard system. I could tell you that if they don't appologise for getting us in hot water twice in the US and once here, on each occasion because their system couldn't cope, that I'll not only drop them (not that that will worry them in the least), but I will use the magazine to tell the full story!

But I wont tell you 'cos its boring stuff really - and who wants to hear about twits anyway!

Graham.

*

Dear Graham,

After puzzling for some time, I have finally come up with a way to get a PLEAR0.

The assembler for the routine is:

```
LDD   #$601
TFR   D,Y
JMP   $96A5   Part of ROM for PCLEAR
```

For 16K use the following:

```
5 CLEAR 200,16374
10 DATA 204,6,1,31,2,126,150,165
20 forN=16374TO16381:READA:POKEN,A:NEXT:EXEC16374
```

For the 64K:

```
5 CLEAR200,32758
10 DATA 204,6,1,31,2,126,150,165
20
FORN=32758TO32765:READA:POKEN,A:NEXT:EXEC32758
```

These three lines at the benining of a program will act in the same way as a PCLEAR0 command would (if it was available). That is, you get the same amount of memory as a POKE25,6:NEW without loosing your program.

John Carmichael
TAREE, NSW.

SURVEY

Last year we stated in our survey "We know you hate surveys, so we thought we'd do it early in the peace before you find all our other faults".

Now you've had time to figure out what you hate about us, what you love about us, and what you want to see changed anyway, we thought we'd drop another on you!

Upon completion, please pop this page in an envelope and return to:

FREEPOST 5
SOUTHPORT. QLD. 4215. by the 7th October, 1985.

The purpose of the survey is the same - to find out what you like, what you don't like, and to find out where you are up to and who is winning in your relationship with your computer.

There is another contest at the end of the survey - you don't have to enter it if you don't want to. The prize is EITHER a CCR 82 cassette recorder, OR three boxes of disks. If you enter the contest, make sure you let us know which prize you would prefer!

1. Name .....

2. Address .....

3. From where did you obtain this magazine?

. Subscriber? - If so, Number .....

. Newsagent . Meet Group

. Tandy . Other - please specify .....

4. Age Range: . 0 - 10 . 11 - 20

. 21 - 40 . 41 - 65

. 66 - 200

5. What type of computer do you own?

. CoCo 16K CB . Tick here if your

. CoCo 16K ECB CoCo is a model 2

. CoCo 32K ECB (white case).

. CoCo 64K ECB

. MC 10 4K . MC 10 20K

. Tandy 100 . Tandy 200

. Tandy 1000 please specify size .....

. Tandy 2000 please specify size .....

. Other, please specify .....

6. What other items of hardware do you own?

. Cassette Deck. Type .....

. Disk Drive. No .... Type .....

. Printer Type .....

. Modem Type .....

. Monitor Type .....

. Other, please specify .....

7. What items of hardware do you plan on purchasing this coming 12 months?

.....

8. How do you use your computer? Please give % use.

% . Games	% . Communication
% . Business	% . Scientific Appn
% . Software Production (Commercial)	% . Robotics
% . Other Control	% . Education

% . To further own computer knowledge.

9. Would you say you have a 50% or greater proficiency in:

. Basic	. Assembly Language
. Basic 09	. PASCAL
. Forth	. Cobol

. Other, Please specify .....

10. With which of the above languages are you also familiar, as opposed to having a working knowledge of?

.....

11. Magazine buying habits:

I subscribe	Magazine	I purchase separately
.	American Rainbow	.
.	HOT COCO	.
.	Others*	.

* excluding our magazines, please specify: .....

12. What did you like about the last two of our magazines you've received?

.....

13. What did you dislike about the last two of our magazines you've received?

.....

14. Could we include something we are not doing now?

.....

15. Tandy: Which shop do you frequent? .....

16. What is the store manager's name? .....

17. Is the shop tidy and well laid out? .....

18. Does this store usually have what you want? .....

19. Are the people there able to help you with your computing problems?

.....

20. Have you been able to strike up a friendship with any of them?

.....

21. To what other shops do you go to obtain the type of gear you also buy from Tandy? .....

22. CoCoConf '86 might be held in Sydney. Would you plan to come? .....

23. If it were to be on the Gold Coast again, would you come? .....

24. Would you prefer to be charged for CoCoConf '86 on a Daily Basis? .....

25. Meet Groups: Do you regularly attend a Meet Group? .....

26. Are there changes needed to the back of the magazine for your group?

.....

27. This is your opportunity to speak, go:

28. Competition. Complete the following FIVE line program:

```
10 A$="Martha Gritwhistle"
20 B$="
30
40
50 PRINT
```

All entries are the property of this magazine, the worse the entry, the greater likelihood of success (just look at last year's winner!).

NOW! is the time to subscribe to Australian RAINBOW

Copies of back issues can be obtained, subject to the availability of stocks, by using this order form and marking clearly which issues you require to be sent to you.
Each issue costs $4.50 including postage and packing. Please enclose your cheque/money order made payable to: Australian Rainbow Magazine, PO Box 1742, Southport, 4215.

RATES

AUSTRALIAN CoCo/MiCo/softgold		AUSTRALIAN RAINBOW
☐ $3.45	Latest per copy	$4.50 ☐
☐ $19	6 months	$24.75 ☐
☐ $31	12 months	$39.95 ☐

BACK ISSUES

MiCo — First Issue Oct '83 ☐ $3/copy

CoCo/MiCo/softgold First Issue Aug '84 ☐ $3.45/copy

MiCoOz (MiCo on Tape) $8 ea ☐ First Issue Dec '83

AUSTRALIAN RAINBOW

Feb '85-Apr '85 ☐	$3.95
Aug '84-Jan '85 ☐	$3.25
to July '84 ☐	$3.00

DISKS & TAPES

CoCoOz (Aust CoCo on Tape)

☐ Latest $8
☐ 6 months $42
☐ 12 months $75

First Issue Mar '83
or Debit my Credit Card Monthly ☐

Blank tapes 12 for $18 or $1.70 ea ☐
Cassette Cases 10 for $5 ☐
Disks — $3.50 ea ☐ 10 for $29.99 ☐

Rainbow on Tape (program listings) $12 for month of ☐
Please note that RAINBOW on TAPE is issued irregularly
or Annually $144 ☐ or Debit my Credit Card Monthly ☐

BOOKS

Byte
Elementary $5.95 ☐

Help
Medium $9.95 ☐

Facts
Advanced $11.95 ☐

MiCo Help (MC-10 computer).
Medium $9.95 ☐

PLUS

Best of CoCoOz # 1 (Education)
Best of CoCoOz # 2 (Games)
Part 1 for 16K ECB
Part 2 for 32K ECB
(Some 16k CB on P1)
TAPE $ 10.00, DISK $ 21.95

BULLETIN BOARD

Let your computer talk to ours!
CoCoLink — Annual Sub $29 ☐

☐ VISA ☐ BANKCARD
☐ MASTERCARD

_____ Signature

☐ CASH
☐ CHEQUE
☐ MONEY ORDER

Authorised
Amount $ _____

BLOCK CAPITALS PLEASE

If you already subscribe to either Australian Rainbow or Australian CoCo please place Subscription No

Complete the section below with one letter, figure or space per square

FIRST NAME SECOND NAME

Address

PC

Telephone Number STD Code Local Number

New Subscription ☐ Renewal ☐

All the programs listed in this magazine are available on Australian Rainbow on Tape.

Unfortunately due to the irregular arrival of the American source material we can only issue the tape as that source material is received.

Australian Rainbow on tape is available from us monthly or yearly, at $12 an issue . For those of you with FOOTSORE FINGERS.

PARIS RADIO ELECTRONICS

CALL
(02) 344 9111 VOICE
(02) 344 9511 BBS
TO ORDER

CoCo OS-9™ FLEX ™

Free Diskette with Each $50 Purchase

COMPILERS

X-BASIC — A BASIC language to MACHINE language Compiler; includes an Extended Macro Assembler. CCF — $292.00

PL/9 — by Graham Trott. A combination Editor/Compiler/Debugger. Structured Programming at the "almost Assembly Language" Level. CCF — $292.00

INTROL C Compiler — Full Featured C Compiler, Linking Loader includes full Library Manager. CCF — $549.00

•Forth — Forth language for CoCo. CCR — $99.95 CCF — $99.95

MICROWARE BASIC09 — for CoCo OS-9 systems. CCO — $149.95

MICROWARE C Compiler — for CoCo OS-9 Systems. CCO — $149.95

DYNA C — C Compiler from the authors of Dynastar and Dynaforms. CCO — $99.95 CCF — $99.95

DATA BASE'S

XDMS — "Mainframe" User's say: "We don't have anything NEARLY as powerful as XDMS;" pure Assembly Language, F-A-S-T and small enough to operate on a single-sided 5" disk.
XDMS (V) I — CCF — $199.95 XDMS (V) II — CCF — $299.95
XDMS (V) III — CCF — $$399.95

RMS — Machine Language Data Base Manager — Super Fast. CCF — $220.00 CCO — $180.00

OPERATING SYSTEMS

XEX — The latest Flex Operating System from Frank Hogg Laboratory USA. Includes hi res screen drivers, user definable keyboard, supports 128k upgrades, and 35, 40, and 80 track drives. CCF — $149.95

OS-9 for the CoCo. The hotest selling software to hit Australia. CCO — $99.95

DISASSEMBLERS

SUPER SLEUTH — Interactive; extremely POWERFUL!! Disk File Binary/ASCII Examine/Change, Absolute or FULL Disassembly. XREF Generator, Label "Name Changer," and Files of "Standard Label Names" for different Operating Systems included. CCF, Obj. Only $89.95
CCF, w/Source $145.00 CCO, Obj. Only $89.95

DYNAMITE + — Excellent standard "Batch Mode" Disassembler Includes XREF Generator and "Standard Label Names" Files.
CCF, Obj. Only $89.95 CCO, Obj. Only $89.95

WORD PROCESSING

STYLOGRAPH III — A full Screen-oriented WORD PROCESSOR (what you see is what you get); also supports Daisy Wheel proportional printers and WORDPAK. CCF — $165.00 CCO — $165.00

STYLO-SPELL Spelling checker to suit stylo. CCF — $99.95 CCO — $99.95

STYLO-MAIL Mail merge program for Stylograph. CCF — $89.95 CCO — $89.95

DYNASTAR & DYNAFORMS — Another full screen Editor and Word Processor for the CoCo. CCF — $149.95 CCO — $149.95

XWORD — A powerful word processor that can be run under O-PAK, XSCREEN hi res drivers and WORD-PAK 80. CCO — $99.95

SPREAD SHEETS

DYNACALC — CoCo's best and fastest Spread Sheet system. CCR — $149.95

UTILITIES

UTILIX is a unix like utilities package for OS-9. It includes 15 different utilities to aid you in the manipulation of text files. CCO — $75.95

FILTERS KIT No. 1 — Eleven utilities used as filters for OS-9. CCO — $39.95

FILTERS KIT No. 2 — Ten more utilities, APPEND, CONFIRM, FF FORCERROR, MACGEN, NULDEVICE, REP, SIZE, TOUCH, and unload. CCO — $39.95

SDISK and BOOTFIX — Use 40 or 80 track, single or double sided drives with OS-9. Create a bootable double sided system disk. CCO — $59.95

HACKERS KIT No. 1 — This package contains a set of programs useful to anyone working with assembly language, trying to unravel the operating system or other assembly language programmes, or customizing their own. CCO — $59.95

XSCREEN — Hi Res Screen driver for OS-9. CCO — $39.95

O-PAK — Hi Res Screen driver for OS-9. CCO — $56.95

SEARCH AND RESCUE — disk utilities to search and retrieve lost data. CCO — $56.95

XTERM — A communications program for use with XSCREEN. CCO — $82.95

MEMORY MINDER — Disk diagnostics program. The most comprehensive program available to analyse your disk drives.
DOUBLE SIDED CCR — $149.95
SINGLE SIDED CCR — $139.95

XMENU — This is a program that creates a menu driven environment for the coco under OS-9. CCO — $69.95

RAM-DISK — This program simulates a disk drive, utilizing spare RAM memory. Very fast execution of commands. For use with 64K or 128K memory, Flex, XEX, or OS-9. CCF — $49.95 CCO — $49.95

MAGAZINES and BOOKS

AMERICAN RAINBOW — Direct from the USA the same month of publication for the same price as the newsagents. Why wait! $6.80

AMERICAN HOT COCO — Airfreighted direct each month. Subscription available. $5.80

THE OFFICIAL BASIC09 TOUR GUIDE $29.95

HARDWARE

PBJ WORKPAK 80 — an 80 column card to fit in the expansion port of the CoCo. Will run under Flex, or OS-9. $199.95

PBJ WORDPAK II — The ultimate wordpak. Includes smooth scrolling, software controlled video switch, improved character set, hardware inverse video and up to 8K video RAM. $249.95

PBJ CC-BUS — A six slot software selectable expansion bus. Will take your disk controller, WORDPAK 80, GAMES Cards etc. $219.95

PBJ PC-PAK — This is a dual function cartridge which contains a centronics compatible parallel printer port and/or a battery backed real time clock.
w/ real time clock $199.95
w/o real time clock $ 99.95

PBJ 2SP-PAK — A 2 port RS-232 serial interface with programmable baud rates up to 19.2K. Each port can also be interrupt driven. $119.95

DISK CONTROLLERS

J&M Disk Controller with JDOS Disk Basic $219.95

RADIO SHACK Disk Controller Ver 1.1 $199.95

MISC

CC-VID Monochrome video output board $ 35.00
MC6809EP Processor chip $ 24.00
MC6883 SAM chip $ 24.00
4164-150 nSec. 64K chips each $ 7.80

VISA

TM FLEX is a trademark of Technical Consultants
TM OS9 is a trademark of Microware

PRICES SUBJECT TO ALTERATION WITHOUT NOTICE

PARIS RADIO ELECTRONICS
161 BUNNERONG RD, KINGSFORD 2032
P.O. BOX 380, DARLINGHURST 2010
Ph: 344 9111

Availability Legends —
CCF — Color Computer FLEX
CCO — Color Computer OS-9
CCR — Color Computer RS-DOS

PRICES DO NOT INCLUDE POSTAGE AND PACKAGING

user group CONTACTS

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

ADELAIDE	JOHN HAINES	08 278 3560
ADELAIDE NTH STVN	EISENBERG	08 250 6214
ALBURY	RON DUNCAN	060 43 1031
ARMIDALE	TON STUART	067 72 8162
BAIRNSDALE	COLIN LEHMANN	051 57 1545
BALLARAT	MARK BEVELANDER	053 32 6733
BLACKTOWN	KEITH GALLAGHER	02-627-4627
BLACKWATER	ANNIE MEIJER	079.82.6931
BLAXLAND	BRUCE SULLIVAN	047 39 3903
BOWEN	TONY EVANS	077 86 2220
BRASSALL	BOB UNSWORTH	07 201 8659
BRIGHTON	GLENN DAVIES	08 296 7477
BRISBANE EAST	ROB THOMPSON	07 848 5512
BRISBANE SH	PATRICK SIMONIS	07 209 3177
BRISBANE SW	GRAHAM BUTCHER	07 376 3400
BRISBANE WEST	BRIAN DOUGAN	07 30 2072
BROKEN HILL	DEAN PARADICE	080 6701
BUNDABERG	JIM McPHERSON	071 72 8329
BURNIE	ANDREW WYLLIE	002 35 1839
CAMBERWELL	TONY BALDWIN	03 728 3676
CAMDEN	KEVIN WINTERS	046.66.8068
CAMPBELLTOWN	LEO GINLEY	02 605 4572
CANBERRA NTH	JOHN BURGER	062 58 3924
CANBERRA STH	LES THURBON	062 88 9226
CAULFIELD	JEFF SHEEN	03 528 3724
CHATSWOOD	BILL O'DONNELL	02 411 3336
' or	MARK ROTHWELL	02 817 4627
CHURCHILL	GEOFF SPOWART	051 22 1389
COLYTON TEENS	DWAYNE MANSON	02 623 5805
COOMA	ROSS PRATT	0648 23 065
DANDENONG	DAVID HORROCKS	03 793 5157
DARWIN	BRENTON PRIOR	089.81.7766
DENILIQUIN	WAYNE PATTERSON	058 81 3014
DONCASTER	JUSTIN LIPTON	03 857 5149
DUBBO	GRAEME CLARKE	068 89 2095
EMERALD	LEIGH EAMES	059 68 3392
FORBES	JOHANNA VAGG	068 52 2943
FORSTER	GARY BAILEY	065 54 5029
FRANKSTON	BOB HAYTER	03.783.9748
GIPPSLAND STH	PAT KERMODE	056 74 4583
GLADSTONE	ALBERT VAN GORKUM	079 72 2353
GOLD COAST	SHERYL BENTICK	075-39-2003
GOSFORD	PETER SEIFERT	043 32 7874
GOULBURN VALLEY	TONY HILLIS	058 59 2251
GRAFTON	DAVID HULME	066.42.0627
GREENACRES	BETTY LITTLE	08 261 4083
HASTINGS	MICHEAL MONCK	059.86.8288
HERVEY BAY	LESLEY HORWOOD	071 22 4989
HILLS DIST	DENNIS CONROY	02 671 4065
HOBART	BOB DELBOURGO	002 25 3896
HORNSBY	ATHALIE SMART	02 848 8830
IPSWICH	MILTON ROWE	07 281 4059
JUNEE	PAUL MALONEY	069 24 1860
KALGOORLIE	TERRY BURNETT	090.21.5212
KYOGLE	PAUL REIDY	066 32 1028
KENMORE	GRAHAM BUTCHER	07 376 3400
LEETON	CHRIS NAGEL	069 53 2969
LITHGOW	DAVID BERGER	063 52 2282
LIVERPOOL	LEONIE DUGGAN	02-607-3791
MACKAY	LEN MALONEY	079511333x782
MACLEOD	ROBIN ZIUKELIS	03 450211x465
MacQUARIEFIELDS	KIETH ROACH	02 618 2858
MAFFRA	MAX HUCKERBY	051 45 4315
MAITLAND	LYN DAWSON	049 49 8144
MARYBOROUGH	NORM WINN	071 21 6638
MELBOURNE	JEFF SHEEN	03 528 3724
MELTON	MARIO GERADA	03 743 1323
MILDURA	SCOTT HOWISON	050 23 6016
MORPHETTVALE	KEN RICHARDS	08 384 4503
MOREE	ALF BATE	067 52 2465
MORWELL	GEORGE FRANCIS	051 34 5175
MT ISA	PAUL BOUCKLEY-SIMONS	077 43 6280
MUDGEE	BRIAN STONE	063-72-1958
MURGON	PETER ANGEL	071 68 1628
NAMBUCCA HDS	WENDY PETERSON	065 68 6723
NEWCASTLE	LYN DAWSON	049 49 8144
NOWRA	ROY LOPEZ	044 48 7031
ORANGE	STEVE LOVETT	063.62.4025
' or	JIM JAMES	063 62 8625
PARKES	DAVID SMALL	068 62 2682
PENRITH	ALEX SCHOFIELD	047 31 5303
PERTH	IAN MACLEOD	09 448 2136
PORT LINCOLN	JOHN BOARDMAN	086 82 2385
PORT MacQUARIE	RON LALOR	065 83 8223
PORT NOARLUNGA	ROB DALZELL	08 386 1647
PORT PIRIE	KEVIN GOWAN	086 32 1368
RINGWOOD	ANDREW RAWLINGS	03 726 6521
ROCKHAMPTON	KEIRAN SIMPSON	079 28 6162
ROSEVILLE	KEN UZZELL	02 467 1619
SALE	BRYAN McHUGH	051 44 4792
SANDGATE	MARK MIGHELL	07 269 5090
SCARBOROUGH	PETER MAY	07 203 6723
SEACOMBE HTS	GLENN DAVIS	08 296 7477
SMYTHESDALE	TONY PATTERSON	053 42 8815
SPRINGWOOD	DAVID SEAMONS	047 51 2107
STURT	MARY DAVIS	08 296 7477
SUNBURY	JACK SMIT	03.744.1355
SUTHERLAND	IAN ANNABEL	02 528 3391
SWAN HILL	BARRIE GERRAND	050.32.2838
SYDNEY TEENS	ROD HOSKINSON	02 48 5948
SYDNEY EAST	JACKY COCKINOS	02 344 9111
TAMWORTH	ROBERT WEBB	067 65 7256
TAHMOOR	GARY SYLVESTER	046 81 9318
TONGALLA	TONY HILLIS	058 59 2251
TOOWOOMBA	GRAHAM BURGESS	076 30 4259
TOWNSVILLE	JOHN O'CALLAGHAN	077 73 2064
TRARALGON	MORRIS GRADY	051 66 1331
UPPER HUNTER	TERRY GRAVOLIN	065 45 1698
WAGGA WAGGA	CES JENKINSON	069 25 2263
WHYALLA NORRIE	CHRIS HUNTER	086 45 3395
WONTHAGGI	PAT KERMODE	056 74 4583
WYNYARD	ANDREW WYLLIE	002 35 1839
YARRAWONGA	KEN SPONG	057 44 1488

SPECIAL INTEREST GROUPS

BUSINESS

BRIZBIZ BRIAN BERE-STREETER 07 349 4696
TDP - TELEWRITER, DYNACALC PROCOLOR

BRISBANE	GEOFF TOLPUTT	07 44 6084

OS9 GROUPS

BANKSTOWN	CARL STERN	02 646 3619
BRISBANE	JACK FRICKER	07 262 8869
KALGOORLIE	TERRY BURNETT	090.21.5212
MONARO	FRED BISSELING	0648 23263
PENRITH	BOB THOMSON	047 30 2468
SYDNEY EAST	JACKY COCKINOS	02.344.9111
SYDNEY NTH	MARK ROTHWELL	02 817 4627
BLAXLAND 128K	BOB THOMSON	047 30 2468

MiCo GROUPS

CARLISLE	STUART HALL	08 361 1922
LITHGOW	DAVID BERGER	063 52 2282
PORT LINCOLN	JOHN	086 82 2385
ROCKHAMPTON	TIM SHANK	079 28 1846
SYDNEY	RAJA VIJAY	02 519 4106

TANDY 1000 / MS DOS

BRISBANE	BRIAN DOUGAN	07 30 2072
SYDNEY	ROGER RUTHEN	047.39.3903

FORTH

PORT LINCOLN	JOHN BOARDMAN	086 82 2385

AUSTRALIAN RAINBOW MAGAZINE
REGISTERED BY AUSTRALIAN POST — PUBLICATION NO. QBG 4009
AUSTRALIAN COCO/softgold
REGISTERED BY AUSTRALIAN POST — PUBLICATION NO. QBG 4007.
PO BOX 1742,
SOUTHPORT. QLD. 4215.