HOT CoCo®

CoCo Max Reviewed, With Pictures

A CWC/I PUBLICATION
JULY 1985
USA $2.95 CAN $3.95

THE MAGAZINE FOR TRS-80 COLOR COMPUTER® USERS.

Programming Languages
A New Horizon

Fix It Yourself
Easy CoCo Repairs

Five Ways to Sort Data

Free Inside
Appointment Calendar Program

PASCAL
LOGO
C

PLUS:
Columns on Education, Telecomputing, Puzzles, and Assembly-Language Programming

07
0 74470 12067

A FULL LINE OF QUALITY MODEMS

Choose an affordable Radio Shack modem that

Open the Door to an "Electronic Village"

Adding a telephone modem to your computer takes you to the door of a new age in information transfer, to an "electronic village" where the latest news, most recent stock prices, farm futures, scholarly materials and more are always at your fingertips.

And there's never been a better time than now to use your Radio Shack® or Tandy® computer as the key to this new age. With our full line of modems, you can choose the one right for your home or business and get "on line" in a hurry.

With one of our modems and Videotex software for your computer, you can use community bulletin boards to set up an electronic "garage sale" or chat with a neighbor. Send E-Mail or join an information service to research a report, read the news and more. There are even services that let you shop without leaving your house. But whatever you decide, a Radio Shack modem is the first step.

Save Time with Our AC-4 Acoustic Coupler

Our AC-4 Acoustic Coupler (26-1179, $119.95) is perfect for travellers. Just dial the telephone number you want, place the phone handset into the AC-4 and you're ready to go. It's perfect for use in a hotel room, client's office or wherever direct connection to a telephone wall jack is impractical.

The AC-4 operates at 300 baud, with full and half duplex capability and originate and answer modes.

Save on a Sale-Priced Modem II This Month

Our Direct-Connect Modem II is a programmable, auto-answer, auto-dial modem —and it's on

MAKES GETTING "ON LINE" EASY.

fits your needs and never be out of touch again!

sale this month for just $119.95 (26-1173, Reg. $199.95).

Our Basic 300-Baud Modem is Under $60

Our DCM-3 Direct-Connect Modem (26-1178, $59.95) makes data transfer especially affordable for the home computer user. It plugs into a modular jack for full duplex capability.

Our High-Speed Modem Is Value Packed

Finally, a full-featured 1200-baud modem that's value priced. Move up to 1200-baud transmissions and save money on long distance phone bills and information service connect time every time you use the DC-2212.

The DC-2212 (26-1176, $399.95) dials and answers the telephone, receives and transmits information and hangs up when the call is completed.

The DC-2212 is a direct-connect modem for Bell 212A-compatible synchronous or asynchronous operation.

See Them Today

Stop by your nearby Radio Shack Computer Center, participating store or dealer for a first-hand look at our modems.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Send Me a Free Copy of Your 1985 Computer Catalog.

Mail To: Radio Shack, Dept. 86-A-85
300 One Tandy Center, Fort Worth, Texas 76102

Name _____
Address _____
City _____
State _____ Zip _____
Telephone _____

Circle Reader Service card #4

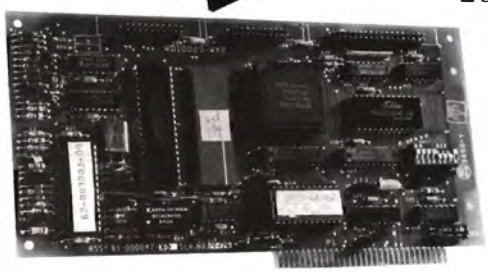
Prices apply at Radio Shack Computer Centers and at participating Radio Shack stores and dealers. All modems FCC registered. Use modems only with Bell-compatible equipment. Direct-connect modems are not for multi-line use without optional-extra controller.

NEW! HARD DISK FOR TANDY® 1000*

introducing **The Rose®** by OSICOM, Inc.

10 MB Internal	20 MB Internal	10 MB External	20 MB External
Controller, Drive Cables, & Manual	*Controller, Drive Cables, & Manual*	*Controller, Drive in enclosure with fan and power supply, Cables, & Manual*	*Controller, Drive in enclosure with fan and power supply, Cables, & Manual*
$699	**$899**	**$975**	**$1199**

Easily user-installed - fully documented instructions.
System "boots" from hard disk without floppy diskette.
Runs Lotus, DBASE II — and Tandy 1000 software.
One year warrantee on all equipment.
Toll-free technical support hot-line.



Orders and Information:
1-800-922-0881 TOLL FREE
201-540-0144 IN NEW JERSEY

18 Bank Street
Morristown, N.J. 07960

Circle Reader Service card #200

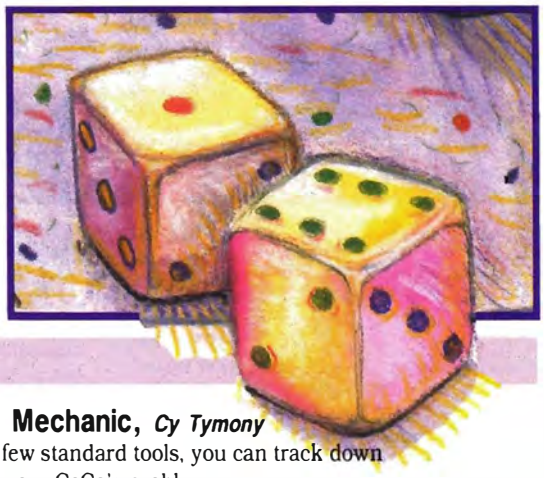
VISA master charge AMERICAN EXPRESS

COD no Surcharge. Visa & MC 3% Amex 5%

*Requires Memory Expansion Board with DMA Controller (25-1004)
Tandy is a registered trademark of Tandy Corporation

HOT CoCo

Vol. 3. No. 2 July 1985



ARTICLES

DEPARTMENTS

HOT CoCo (ISSN 0740-3186) is published monthly by CW Communications, 80 Pine St., Peterborough, NH and additional mailing offices. Subscription rates in U.S. are $24.97 for one year, $38 for two years, and $53 for three years. In Canada and Mexico, $27.97—one year only, U.S. funds. Second class postage paid at Peterborough, NH and additional mailing offices. Canadian 2nd Class mail reg. #9564. Nationally distributed by International Circulation Distributors. Foreign subscriptions (surface mail), $44.97—one year only, U.S. funds drawn on a U.S. bank. Foreign subscriptions (air mail), please inquire. In South Africa contact *HOT CoCo*, P.O. Box 782815, Sandton, South Africa 2146. *All subscription correspondence should be addressed to* HOT CoCo, *Subscription Department, P.O. Box 975, Farmingdale, NY 11737. Please include your address label with any correspondence.* Postmaster: Send address changes to *HOT CoCo*, Subscription Services, P.O. Box 975, Farmingdale, NY 11737. Send Canadian change of address to *HOT CoCo*, P.O. Box 1051, Fort Erie, Ontario Canada, L2A5N8. Return postage guaranteed. For questions concerning your subscription and to place subscription orders, please call us toll free at 1-800-258-5473 between 8 a.m. and 5 p.m. EST or write to *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737.

DIGRESSIONS

Aiming To Please

When we asked for your reactions to our recent redesign (May issue), we expected a good amount of both positive and negative comments, which we received. In fact, many of you sent detailed critiques of that issue. But we did not expect the totally opposite feelings among you concerning specific aspects. The primary case in point: our pull-out program listings.

Our mail is running about three to two in favor of publishing the listings together in the center of the magazine. You are either strongly in favor of the pull-out listings ("Best idea yet," "The best thing to come along in a computer magazine," said two of you) or adamantly opposed to them ("Pull-out programs are a pain," "The most distressing of all the changes," said two others).

Many of you complained about the listings' type size. We have been wrestling with how to make the listings more readable without consuming too much magazine space. The pull-out listings will allow us to print listings slightly larger, because it eliminates the need to run a "box" around them, creating more margin space. Starting with this issue, all Basic listings are 3 percent larger. With the August issue, they will be another 8 percent larger, when we publish them in horizontal rather than vertical columns.

Several of you asked that we put the page number of the accompanying article with each program listing. We will do this beginning this month.

I expect the pull-out listings to stay, because they allow us to print program listings in a more readable size and produce a more attractive, interesting magazine.

Some of you didn't like the larger artwork and open spaces on some of the pages. You said you'd rather see more information printer in the magazine. Those open, or "white," spaces are one way we can increase readability—an important consideration in a magazine where the subject matter can get a little heavy going. Your message was clear, though: You want more articles. And we will do our best to get more material into *HOT CoCo*—I promise.

Everyone who wrote found something to like in the new *HOT CoCo*. Our covers, interior design, and new columns were generally all hits. Next month, we will publish some of the most representative letters we've received on the redesign.

If you haven't written yet, it's not too late. The most valuable tools we have are your comments. If you hate what we are doing, tell us—we editors have thick skins (mostly). If you like what we are doing, tell us why.

To those of you who have already written, thank you.—*Michael E. Nadeau*■

Publisher
JEFF DETRAY

Editor-in-Chief
MICHAEL E. NADEAU

Managing Editor
MARK E. REYNOLDS

Review/New Products Editor
J. SCOT FINNIE

Copy Editor
MARILYN G. MCMASTER

Education Editor
NANCY KIPPERMAN

Technical Editors
PETER PAPLASKAS,
GUIER WRIGHT,
KEITH JOHNSON (INSTANT COCO)

Editorial Administration
KIM McCORMICK

Art Director
DONNA WOHLFARTH

Production Supervisor
ANNE FLEMING

Advertising Director
PETER MONTROSS

Advertising Coordinator
SUSAN WRIGHT
80 Pine Street
Peterborough, NH 03458
1-800-441-4403 or
1-603-924-9471

Manufacturing Manager:
SUSAN GROSS

Typesetting Manager:
DENNIS CHRISTENSEN

Film Preparation Manager:
ROBERT M. VILLENEUVE

President
DEBRA WETHERBEE

Vice President/Finance
ROGER MURPHY

Assistant to President/GM
MATT SMITH

Assistant to Vice President/Finance
DOMINIQUE SMITH

Executive Creative Director
CHRISTINE DESTREMPES

Director of Circulation
WILLIAM P. HOWARD

Assistant Circulation Manager
FRANK S. SMITH

Direct & Newsstand Sales Manager
RAINO WIREIN: 1-800-343-0728

Director of Credit, Sales, and Collection
WILLIAM M. BOYER

Founder
WAYNE GREEN

HOT CoCo **is a member of the CW Communications/inc. group, the world's largest publisher of computer-related information. The group publishes 52 computer publications in 19 major countries. Members of the group include: Argentina's *Computerworld/Argentina*; Australia's *Australia Computerworld*, *Australian Micro Computer Magazine*, *Australian PC World and Directories*; Brazil's *DataNews* and *MicroMundo*; Chins's *China Computerworld*; Denmark's *Computerworld/Danmark* and *MicroVerden*; Finland's *Mikro*; France's *Le Monde Informatique*, *Golden* (Apple) and *OPC* (IBM); Germany's *Computerwoche*, *Microcomputerwelt*, *PC Welt*, *Software Markt*, *CW Edition/Seminar*, *Computer Business* and *Commodore Magazine*; Italy's *Computerworld Italia*; Japan's *Computerworld Japan* and *Perso ComWorld*; Mexico's *Computerworld/Mexico* and *CompuMundo*; Netherland's *CW Benelux* and *Micro/Info*; Norway's *Computerworld Norge* and *MikroData*; Saudi Arabia's *Saudi Computerworld*; Singapore's *The Asian Computerworld*; Spain's *Computerworld/Espana* and *MicroSistemas*; Sweden's *Computer-Sweden*, *MikroDatorn* and *Min Hemdator*; the UK's *Computer Management* and *Computer Business Europe*; United States: *Computerworld*, *HOT CoCo*, *inCider*, *InfoWorld*, *MacWorld*, *Micro MarketWorld*, *PC World*, *PC Jr. World*, *RUN*, *73 Magazine*, and *80 Micro*.**

Article submissions from our readers are welcomed and encouraged. Inquiries should be addressed to: *HOT CoCo* Submissions Editor, 80 Pine Street, Peterborough, NH 03458. Include an SASE for a copy of our writer's guidelines. Payment for accepted articles is made at a rate of approximately $50 per printed page; all rights are purchased. Authors of reviews should contact the *HOT CoCo* Review Editor, 80 Pine Street, Peterborough, NH 03458.

Problems with Subscriptions: Send a description of the problem and your current and/or most recent address to: *HOT CoCo*, Subscription Department, P.O. Box 975, Farmingdale, NY 11737.

Change of Address: Send old label or copy of old address and new address to: *HOT CoCo*, P.O. Box 975, Farmingdale, NY 11737. Please give eight weeks advance notice.

Dealers: Contact Ginnie Boudrieau, Bulk Sales Manager, *HOT CoCo*, 80 Pine Street, Peterborough, NH 03458. (800) 343-0728.

Problems with Advertisers: Send a description of the problem and your current address to: Magazine, Rt. 101 & Elm Street, Peterborough, NH 03458. ATTN.: Rita B. Rivard, Customer Service Manager. If urgent, call 1-800-441-4403.

Telewriter-64™

the Color Computer Word Processor

- **3 display formats: 51/64/85 columns × 24 lines**
- **True lower case characters**
- **User-friendly full-screen editor**
- **Right justification**
- **Easy hyphenation**
- **Drives any printer**
- **Embedded format and control codes**
- **Runs in 16K, 32K, or 64K**
- **Menu-driven disk and cassette I/O**
- **No hardware modifications required**

THE ORIGINAL

Simply stated, Telewriter is the most powerful word processor you can buy for the TRS-80 Color Computer. The original Telewriter has received rave reviews in every major Color Computer and TRS-80 magazine, as well as enthusiastic praise from thousands of satisfied owners. And rightly so.

The standard Color Computer display of 32 characters by 16 lines without lower case is simply inadequate for serious word processing. The checkerboard letters and tiny lines give you no feel for how your writing looks or reads. Telewriter gives the Color Computer a 51 column by 24 line screen display with *true lower case characters*. So a Telewriter screen looks like a printed page, with a good chunk of text on screen at one time. In fact, more on screen text than you'd get with Apple II, Atari, TI, Vic or TRS-80 Model III.

On top of that, the sophisticated Telewriter full-screen editor is so simple to use, it makes writing fun. With single-letter mnemonic commands, and menu-driven I/O and formatting, Telewriter surpasses all others for user friendliness and pure power.

Telewriter's chain printing feature means that the size of your text is never limited by the amount of memory you have, and Telewriter's advanced cassette handler gives you a powerful word processor without the major additional cost of a disk.

...one of the best programs for the Color Computer I have seen...
— Color Computer News, Jan. 1982

TELEWRITER-64

But now we've added more power to Telewriter. Not just bells and whistles, but major features that give you total control over your writing. We call this new supercharged version Telewriter-64. For two reasons.

64K COMPATIBLE

Telewriter-64 runs fully in any Color Computer — 16K, 32K, or 64K, with or without Extended Basic, with disk or cassette or both. It automatically configures itself to take optimum advantage of all available memory. That means that when you upgrade your memory, the Telewriter-64 text buffer grows accordingly. In a 64K cassette based system, for example, you get about 40K of memory to store text. So you don't need disk or FLEX to put all your 64K to work immediately.

64 COLUMNS (AND 85!)

Besides the original 51 column screen, Telewriter-64 now gives you 2 additional high-density displays: 64 × 24 and 85 × 24!! Both high density modes provide all the standard Telewriter editing capabilities, and you can switch instantly to any of the 3 formats with a single control key command.

The 51 × 24 display is clear and crisp on the screen. The two high density modes are more crowded and less easily readable, but they are perfect for showing you the exact layout of your printed page, *all on the screen at one time*. Compare this with cumbersome "windows" that show you only fragments at a time and don't even allow editing.

RIGHT JUSTIFICATION & HYPHENATION

One outstanding advantage of the full-width screen display is that you can now set the screen width to match the width of your printed page, so that "what you see is what you get." This makes exact alignment of columns possible and it makes hyphenation simple.

Since short lines are the reason for the large spaces often found in standard right justified text, and since hyphenation is the most effective way to eliminate short lines, Telewriter-64 can now promise you some of the best looking right justification you can get on the Color Computer.

FEATURES & SPECIFICATIONS:

Printing and formatting: Drives any printer (LPVII/VIII, DMP-100/200, Epson, Okidata, Centronics, NEC, C. Itoh, Smith-Corona, Terminet, etc).

Embedded control codes give full dynamic access to intelligent printer features like: underlining, subscript, superscript, variable font and type size, dot-graphics, etc.

Dynamic (embedded) format controls for: top, bottom, and left margins; line length, lines per page, line spacing, new page, change page numbering, conditional new page, enable/disable justification.

Menu-driven control of these parameters, as well as: pause at page bottom, page numbering, baud rate (so you can run your printer at top speed), and Epson font. "Typewriter" feature sends typed lines directly to your printer, and Direct mode sends control codes right from the keyboard. Special Epson driver simplifies use with MX-80.

Supports single and multi-line headers and automatic centering. Print or save all or any section of the text buffer. Chain print any number of files from cassette or disk.

File and I/O Features: ASCII format files — create and edit BASIC, Assembly, Pascal, and C programs, Smart Terminal files (for uploading or downloading), even text files from other word processors. Compatible with spelling checkers (like Spell 'n Fix).

Cassette verify command for sure saves. Cassette auto-retry means you type a load command only once no matter where you are in the tape.

Read in, save, partial save, and append files with disk and/or cassette. For disk: print directory with free space to screen or printer, kill and rename files, set default drive. Easily customized to the number of drives in the system.

Editing features: Fast, full-screen editor with wordwrap, block copy, block move, block delete, line delete, global search and replace (or delete), wild card search, fast auto-repeat cursor, fast scrolling, cursor up, down, right, left, begin line, end line, top of text, bottom of text; page forward, page backward, align text, tabs, choice of buff or green background, complete error protection, line counter, word counter, space left, current file name, default drive in effect, set line length on screen.

Insert or delete text anywhere on the screen without changing "modes." This fast "free-form" editor provides maximum ease of use. Everything you do appears immediately on the screen in front of you. Commands require only a single key or a single key plus CLEAR.

...truly a state of the art word processor... outstanding in every respect.
— The RAINBOW, Jan. 1982

PROFESSIONAL WORD PROCESSING

You can no longer afford to be without the power and efficiency word processing brings to everything you write. The TRS-80 Color Computer is the lowest priced micro with the capability for serious word processing. And only Telewriter-64 fully unleashes that capability.

Telewriter-64 costs $49.95 on cassette, $59.95 on disk, and comes complete with over 70 pages of well-written documentation. (The step-by-step tutorial will have your writing with Telewriter-64 in a matter of minutes.)

To order, send check or money order to:

Cognitec
704 N. Nob St.
Del Mar, CA 92014

Or check your local software store. If you have questions, or would like to order by Visa or Mastercard, call us at (619) 755-1258 (weekdays, 8AM-4PM PST). Add $2.00 for shipping & handling. CA residents add 6% state tax.

Now available at Radio Shack stores via express order.

Instant CoCo is a cassette tape containing the major programs from this issue of *HOT CoCo*. Its purpose is to save you the time and effort of typing long program listings into your Color Computer. You simply load the programs from the Instant CoCo tape using your cassette recorder. The instructions for operating each program are found in the corresponding *HOT CoCo* article. Both Basic and Assembly-language programs are included on the tape.

The Instant CoCo symbol appears in *HOT CoCo*'s table of contents and on the program listing for each article with a listing used on the Instant CoCo tape. As an added extra, each tape also contains a never-before-published Bonus Program, complete with instructions.

The directory below lists all programs included on this month's Instant CoCo cassette. Shown first are the name of the article with a descriptive blurb and its author, followed by the page number in this issue where the article appears. Next comes the file name of the program on cassette. Finally, there is a brief description of the Color Computer system needed to run the program.

This month's Instant CoCo cassette is available for just $11.47, including postage and handling, from **Instant CoCo, 80 Pine St., Peterborough, NH 03458.** See our ad on p. 64 for more details.

Instant CoCo Directory July 1985

Side A

Article Name/Author/Description	Page #	File Name	System
Copyright Statement	---	TITLE	16K CB
Mindbusters/Ramella	16	CRYPT	16K ECB
Solve these puzzles based on the alphabet, if you can.		ALPHA	16K CB
Getting It Together/Tipps	23	MAKEDATA	16K ECB or DECB
Combine Basic with Assembly-language code.			
All Sorts Of Sorts/Meredith	33	LIST1	16K ECB
Next time you need a sort routine, grab one of these programs.		LIST2	16K ECB
		LIST3	16K ECB
		LIST4	16K ECB
		LIST5	16K ECB
		LIST6	16K ECB
		LIST7	16K ECB
		LIST8	16K ECB

Side B

The John-B System, Part I/Barbarello	38	PHONE	16K ECB
Test your Phone I/O unit with auto-dialer.			
Tricky Dice/Meinhardt	72	TRICKY	16K ECB
Roll the dice, rack up points.			
Date Minder/Huckabey	74	DATEMIND	32K ECB
Let your CoCo remind you of those important appointments			

***** BONUS PROGRAM *****

Star War 27/Lewis	---	STRWAR 27(m)	16K CB or DECB
Fight off attackers from the front and back.			
Adding And Subtracting Feet, Inches, And Fractions/Gill	---	FEET-IN	4K CB
It all measures up to the nearest 1/64 of an inch.			

CB = Color Basic, DECB = Disk Extended Color Basic, ECB = Extended Color Basic
(m) = machine-language program (use CLOADM)

Back Issues

Yes, back issues of *HOT CoCo* are available for all months. This list shows the features in each issue:

June 1983—The CoCo word processor; a serial-to-parallel interface project; and the adventure, Cavehunt.

July 1983—How to upgrade your CoCo to 64K; cure video RFI.

August 1983—Speech synthesis via software; get more colors; build a color monitor driver.

September 1983—Disk utilities; hi-res character generator.

October 1983—Animation techniques; ROM disassembly, part I.

November 1983—Nuclear submarine simulation; ROM-pack primer; banner printer.

December 1983—World capitals quiz program; talking spelling tutor; vocabulary-building program.

January 1984—Programs for the businessman and investor; ins and outs of database management.

February 1984—CoCo-aided circuit design; simulate Extended Basic in Color Basic; change your CoCo's vocabulary.

March 1984—How a disk stores information; create your own wordsearch puzzles; dental/medical bill balancer.

April 1984—Peripherals buyer's guide; how to shop for a disk drive; disk-fix utility; Lisp interpreter.

May 1984—OS-9 review; financial transactions tracker; homebrew spelling checker; CoCo Reversi game.

June 1984—Horse-racing and stock-market simulators.

July 1984—Do-it-yourself lowercase mod; variable cross-referencer; the game, Python.

August 1984—Basic-09 review; database manager program; graphics tutorials; hurricane tracker.

September 1984—Educational software buyer's guide; typing-teacher program; the CoCo as a marketing aid.

October 1984—A collection of sounds for your CoCo; how to make programs auto-execute; printer spooler.

November 1984—Personal money manager program; disk-file protection utility.

December 1984—Disk-drive timer; disk drive maintenance tips; full-featured text-editing program.

January 1985—Spreadsheet program; stock-charting program; make fancy graphics with your printer.

February 1985—Drawing program; user's group list; Space Hawks game.

March 1985—Universal screen-dump program; POKE list; utilities.

April 1985—Telewriter-64 mods; modem comparison; satellite-tracking program.

You'll also find in each issue our regular features, reviews of popular software and hardware, and dozens of useful programs that are yours for the typing in.

Each back issue costs $3.50 plus $1 shipping and handling. On orders of 10 or more back issues, there is a flat $10 shipping fee. Send your orders to *HOT CoCo*, Attn. Back Issue Orders, 80 Pine St., Peterborough, NH 03458. ■

DYNACALC®

CoCo's Best & Fastest Spreadsheet System

ACCLAIMED BY THE EXPERTS

"DYNACALC is my choice for a CoCo spreadsheet."
Dan Downard, RAINBOW, September, 1984.

"Eat your heart out, Lotus 1-2-3!"
Scott Norman, HOT CoCo, October, 1984.

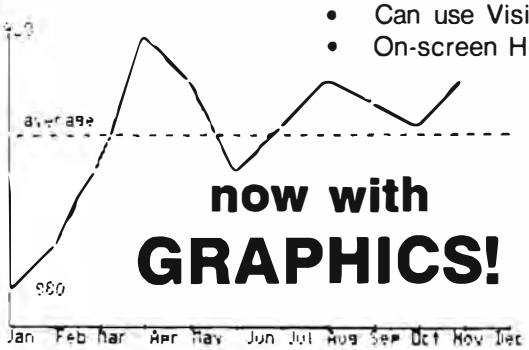
NOW ONLY $99.95

Built-in Features:

- 51 x 24 Display with Lower Case
- Super-fast Smart Screen Refresh
- Auto-Repeat Keyboard Driver
- Keysaver (TM) repeats last command x times
- Disk Operating System (works just like ROM DOS)
- Easy communication with BASIC/DOS programs
- Two-way communications with PRO-COLOR-FILE ★Enhanced★
- Outputs to ASCII Word Processors like Telewriter-64
- Fast 16-Digit Arithmetic with Scientific Functions
- Summation, Mean, and Standard Deviation Functions
- Logical Functions with String & Numeric Comparison
- String locate command to navigate large worksheets
- Sort full or partial worksheet by columns or rows
- Line, Bar, Hi/Lo/Close, Circle Graphs
- Full Graphics captioning and overlay facility
- Graphics Drivers for all popular Printers
- Joystick/Mouse Driver for Cursor Movement
- Works with any ROM versions, even JDOS
- 33k Available Worksheet Space
- Up to 256 Columns or 256 Rows
- Can use VisiCalc worksheets & training materials
- On-screen Help Messages

RAINBOW CERTIFICATION SEAL

OS-9 version available from Tandy/Radio Shack #26-3275



now with GRAPHICS!

FOR 64K DISK SYSTEMS
available from

COMPUTER SYSTEMS CENTER
42 Four Seasons Center #122
Chesterfield, MO 63017 USA
(314) 576-5020
10 to 6 Mon.-Fri.

NEW ADDRESS

CSC

or your local DYNACALC dealer

Price US$99.95
Outside North America add $10 postage

CANADA-CDN $129.95
DATAMAN INTERNATIONAL
420 FERGUSON AVE. N.
HAMILTON, ONT. L8L 4Y9
(416) 529-1319

AUSTRALIA
PARIS RADIO
161 BUNNERONG RD.
KINGSFORD 2032 NSW
(612) 344-9111

MasterCard VISA

How to Use HOT CoCo

Each month *HOT CoCo* provides program listings for you to type into your Color Computer and use. If you are new to computing, read this page for advice that will help you avoid problems often encountered when entering programs manually.

Know the Basics

Before you begin, you should be familiar with the basic operation of your Color Computer. Read the manual and make sure you understand how to enter a program line, save a program to cassette or disk, and make corrections to a program line. The Color Computer manuals are well written, and you will enjoy your CoCo much more if you've read them.

Check the Requirements

The first thing you should do is make sure that the program you want to enter will run on your version of the Color Computer. You need to know the memory requirements, the type of Basic used (Color, Micro Color, Extended Color, or Disk Extended Color Basic), what peripherals might be needed, and in some cases whether a particular ROM version is needed. (See below for an explanation of the different ROMs.)

All this information is provided in the System Requirements box included with each article that has a program listing. This box gives the minimum requirements to use the program. If, for instance, the box reads "16K RAM, Color Basic," the program should also work on 32K or higher, Extended or Disk Extended Color Basic CoCos.

Once you've established that the program will work on your CoCo, read the article thoroughly. Sometimes it will include information vital to typing in the listing.

What You See is What You Get

We print all Basic program listings 32 characters across—just as they appear on your video screen. Type in the listings exactly as it appears in the magazine, being particularly careful with spaces and punctuation. If you do this, the 32-character format will aid in proofreading what you have typed in by letting you match beginning and ending characters on corresponding lines. If you have a line that ends on a character other than what appears in the magazine, go back and check for a typo.

Common Errors

Some characters are easier to confuse than others when you are typing in program listings. And since your Color Computer interprets everything literally, the smallest error can crash a program. Below is a list of characters commonly confused with one another:

zero and the letter O
colon and semicolon
lowercase l and the numeral one
uppercase B and the numeral eight

Weird Characters

The up arrow indicates exponentiation on the Color Computer. Unfortunately, most printers do not have an up arrow. Our printer prints a caret (^) instead. Be sure to type an up arrow in place of all carets in Basic program listings.

Assembly-Language Listings

HOT CoCo often publishes programs written in Assembly language rather than Basic. Assembly listings "talk" to your computer on a much more direct level; Basic requires some translation before your CoCo can execute it. Therefore, Assembly works much faster than Basic. Unfortunately, it is more difficult to learn Assembly-language programming than Basic programming.

But you do not need to know how to program in Assembly to use these programs. You do need, however, something called an editor/assembler. An editor/assembler allows you to manually enter an Assembly listing, and then it "assembles" it into a form that your CoCo can execute. Since editor/assemblers can cost as much as $80, you probably don't need one unless you want to learn Assembly-language programming.

It is possible to hand assemble an Assembly listing, but this is a tedious process that is best left to someone with a little experience with Assembly programming. It also requires a short Basic routine that prepares your CoCo for hand assembly.

We convert some Assembly programs to Basic DATA statements and include a short Basic routine to load and execute the DATA statements. This gives you a program that you can type in just like a Basic listing, yet it operates much like one written in Assembly.

If you want to run one of *HOT CoCo*'s Assembly listings, but it hasn't been converted to DATA statements and you do not own an editor/assembler, check to see if the program is included on our Instant CoCo cassette. All Assembly programs on Instant CoCo are in assembled form, meaning you can load and execute them immediately.

Speaking of DATA Statements

Since DATA statements often consist of numbers only, it is easy to make a mistake typing them in. One wrong number can crash the program or lock up your machine. When this happens, the only way to recover is often to turn off the computer for a few seconds and then turn it back on. Of course, this wipes out your program in memory.

To avoid this, always save what you have typed in before running it. That way, if you did make a mistake, you can load the program from tape or disk to look for the error, rather than retyping the entire listing.

One last thing about DATA statements: Error messages that occur due to a mistyped DATA statement line will refer to the corresponding READ statement line earlier in the program. Yet it is the DATA statement that in incorrect.

If All Else Fails

If you cannot get your typed-in listing to run after checking and double-checking for typos, you can ask us for help. Send a detailed description of your problem along with any error messages given. Ideally we'd like a printout of what you typed. Send a self-addressed, stamped envelope for the fastest reply. Sorry, but we cannot help you if you have modified the orignal program in any way. Write to *HOT CoCo, attn. Technical Editor, 80 Pine St., Peterborough, NH 03458.*

Different ROMS

Radio Shack has updated the Basic ROMs in the Color Computer several times since it was introduced. Below is a list of the ROMs and the problems and benefits you might encounter with each one:

- Color Basic 1.0—Cannot fully use the 64K upgrade and has only a 7-bit serial printer routine, which inhibits sending graphics data to a printer.
- Color Basic 1.1—Fully supports 64K and has an 8-bit serial printer routine for graphics.
- Color Basic 1.2—Executes code faster than previous versions, but changed the way the ROM reads the keyboard. This makes some software written for the older ROMs incompatible with the 1.2 ROM. There is a simple fix, which *HOT CoCo* incorporates into every program in which this problem is encountered.

If you don't know what Color Basic ROM version you have, type EXEC 41175 after you first turn on your computer. The ROM version will be printed on the screen.

- Extended Basic 1.0—Has bugs in the PCLEAR, PRINT USING, and DLOAD statements.
- Extended Basic 1.1—Fixes the abovementioned bugs.
- Disk Basic 1.0—This is in the disk controller cartridge used with the grey CoCos and grey disk drives. The 1.0 Disk ROM is incompatible with CoCo 2's.
- Disk Basic 1.1—Works faster than 1.0, but you can use the 1.1 Disk Basic controller with the older, grey CoCos. Also, many routines have been moved, making some programs written using the 1.0 Disk ROM incompatible with the 1.1 ROM. (See "A Quick Fix for Your Disk ROM," by Mike Meehan, *HOT CoCo*, February 1985, p. 44, for a utility that overcomes this incompatibilty in most cases.)■

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

PRICE BREAKTHROUGH

DEALER INQUIRIES INVITED

Prices Change Every Day. Please Call 1-800-343-8841 For Lower Prices.

MEGADISK™ HARD DISK DRIVE SYSTEMS — **TOLL FREE ORDERING 1-800-343-8841**

For the { **IBM/PC, Tandy 1000, TRS/80 Models I/III/IV/4P,** Compaq, Eagle, Sanyo, Tava, PC Workalikes, **Color Computers,** Heath/Zenith, Max/80
Complete with Hardware, Cables, Software and Quikfit Installation

Wow! New Low Prices

5 Megabytes Internal Mount	starting at	$ 299.95
10 Megabytes Internal Mount	starting at	544.95
20 Megabytes Internal Mount	starting at	799.95
5 Megabytes External System	starting at	499.95
10 Megabytes External System	starting at	699.95
20 Megabytes External System	starting at	999.95
Streaming Tape Backup System — Internal Or External	starting at	499.95

DOS Systems Available: { IBM/Heath — DOS, 1.0, 2.0, 2.1, 3.0, or later
TRS/80-LDOS, TRSDOS 6.x, Newdos/80, Dosplus 3.5, CP/M, COCO DOS, Max/80 LDOS, OS9

FULLY WARRANTIED — PARTS AND LABOR — 24 HOUR SERVICE — CALL TOLL FREE — 1-800-343-8841

FLOPPY DISK DRIVES, POWER SUPPLIES AND CABINETS

Our Disk Drives are UL approved — Our Floppy Drive Cabinets and Power Supplies are Underwriters Laboratory Listed and have passed the required Federal Communications Part 15 Section B-EMI/RFI tests.
Warranty on all disk drives is one full year parts and labor. Warranty on floppy disk drive power supplies is five (5) years. In warranty or out of warranty service is 24 hour turn-a-round on all disk drives and power supplies.

Tandon — Full Height Drives		
100-1	Single Sided 40 tk Bare	$125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	326.95
100-2	Dual Sided 40 tk Bare	162.00
	In Case with Power Supply	206.95
	Dual Drives in One Cabinet	381.95
	Half High Drives	
65-1	Single Sided 40 tk Bare	122.00
	In Case with Power Supply	156.95
	Dual Drives in One Cabinet	306.95
65-2	**Dual Sided 40 tk Bare**	**142.00**
	In Case with Power Supply	**176.95**
	Dual Drives in One Cabinet	**346.95**
65-4	Dual Sided	162.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
TEAC Half High Drives		
55A	Single Sided 40 tk Bare	125.00
	In Case with Power Supply	166.95
	Dual Drives in One Cabinet	316.95
55B	Dual Sided 40 tk Bare	157.00
	In Case with Power Supply	196.95
	Dual Drives in One Cabinet	356.95
55F	Dual Sided 80 tk Bare	177.00
	In Case with Power Supply	216.95
	Dual Drives in One Cabinet	386.95
Apple/Franklin Disk Drives		
	35/40 Track in Case with Cable and Software	159.95
	Controller Card for Two Disk Drives	46.95
	Combination Price for Disk Drive and Controller	196.95
Commodore Disk Drives		236.95
Power Supplies and Cabinets 5¼ and Hard Drive Systems		starting at 42.00

COLOR COMPUTER DISK DRIVE SYSTEMS AND ADD IN PRODUCTS

40 Track Single Head Drive with Case, Power Supply, Cable Controller, Instruction Booklet, Diskettes	$239.95
Above with Dual Drives in One Cabinet	374.95
40 Track Dual Head with Case, Power Supply, Cable, Controller, Instruction Booklet, Diskettes	**259.95**
Above with Dual Drives in One Cabinet	**399.95**
128 Memory Upgrade Kit	129.95
Dual DOS Switch	19.95
With Second DOS System — JDOS, RSDOS, Micro DOS, and Booklet	69.95

MODEMS

Volksmodem 300 Baud	$ 69.95
Signalman Mark X Autodial	123.95
Mark XII 1200/300 Baud Autodi	284.95

ALL IN-STOCK ITEMS SHIPPED WITHIN 24 HOURS. SAME DAY SHIPPING PROVIDED BY REQUEST WITHOUT ANY EXTRA HANDLING CHARGES.

********** MEGADISK SPECIAL **********

With a 10 megabyte system, you can purchase a complete IBM or PC compatible host, software and cable for only $15.00!!
P.S. Ask about the MEGAPLEXER.

IBM - PC/XT WORKALIKE

All of our computers have: 8 slot motherboard, 640K, monochrome adapter, parallel printer port, 130 watt power supply, free software — all with our full warranty.

/PC- 2 Floppy Drives - Monochrome Monitor, MS-DOS	$1,595.00
Color system resolution - 720h x 240v, MS-DOS	$1,995.00
5meg/XTtra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	$1,745.00
Color system resolution - 720h x 240v, MS-DOS	2,145.00
10meg/XTtra - 1 Floppy Drive - Monochrome Monitor, MS-DOS	2,095.00
XTtra - Color system resolution - 720h x 240v, MS-DOS	2,495.00
20meg/XTtra - 1 Floppy Drive Monochrome Monitor, MS-DOS	2395.00
XTtra - Color system resolution - 720h x 240v, MS-DOS	2,795.00
Internal Tape Backup For Any Of Above systems Add	$449.95

PRINTERS

Dot Matrix		
Citizen		$ Call
Star Micronics		
Gemini 120 CPS Near letter quality	starting at	$259.95
Delta 10/15 160 CPS	starting at	424.95
Radix 10/15 200 CPS	starting at	649.95
Panasonic 1090		249.95
Daisy Wheel		
Silver Reed 440 80 Column 12 CPS		315.95
550 132 Column 19 CPS		439.95
770 132 Column 36 CPS		895.00
Olympia 132 Column 14 CPS with Form and Tractor		399.95
Apple/Franklin Printer Interface w/Graphics and Cable		84.95
Printer Cables	starting at	19.95
Printer Paper — Microperf Edge 1000 Sheets		16.95

ELECTRICAL

Surge Protectors — Line Filters — SL Waber — 6 Outlets with Switch	$ 39.95
Uninterruptable Power Supplies	399.95

MISCELLANEOUS

Diskettes in 10 Pack	**from**	**$ 10.95**
Twoprint Switches	from	99.95
Disk Drive Cables	from	16.00
Maintenance Cleaning Kits		12.00
Parallel Printer Buffers 8K		149.95
Floppy Disk Drive Cables		
1 Drive		16.00
2 Drives		18.95
Heath/Zenith 2 Drive Cables — Shielded		24.95

Letters to the Editor

Send your Letters to the Editor, HOT CoCo, 80 Pine St., Peterborough, NH 03458.

Express Order Works

After reading the note in Digressions on Radio Shack's Express Order (*HOT CoCo*, March 1985, p. 5), I ordered the VIP Integrated Library through my local dealer, Electronics Etc. At first, my dealer was told that this program wasn't available through Express Order. A few days later, however, the dealer called me to say that the program was in the warehouse. He said they would ship it if I still wanted it.

I had the program within 10 days of seeing the notice in *HOT CoCo*. That is what I call express service. I credit my dealer with making the service work and am glad Radio Shack is selling third-party software.

I also like VIP's Integrated Library because I don't have to learn new commands for each program. If you haven't used this program, you're missing something.

Thanks for a good publication. I learn a lot from your magazine and advertising.

Gary R. Hawkins
Ladysmith, WI

April Issue

My very first *HOT CoCo*, your April 1985 issue, was great. The cover article was interesting, and I could tell that the spacecraft shown on the cover is a Russian-made Salyut. The picture was taken from an Apollo spacecraft during a Russian-American hookup in 1975.

The article on modems was informative and convinced me to buy the Radio Shack Modem IB. All in all, the April *HOT CoCo* was an excellent piece of journalism. Your magazine stands out as the best of all computer magazines.

James Sharer
Dallas, TX

Modem II

I wasn't very pleased with "Your First Modem" (*HOT CoCo*, April 1985, p.24). You made Modem II sound worthless, but I find it answers many of my needs. Your review said to turn the force DTR switch off when it should be on. With the switch on, the modem can autodial and autoanswer. Modem II allows slow and fast rotary dialing and touch-tone dialing, too.

I use Mikeyterm, Mike Ward's excellent free program downloadable through CompuServe's Color SIG, for autodialing and it works fine. This program has advanced features, such as the XMODEM error-checking protocol. My friends and I who use Mikeyterm thank Mike.

Jason Johnson
Milan, MI

The documentation that comes with the Modem II states the following. "The Color Computer does not have a DTR (data terminal ready) as an RS-232 control line. The Modem II uses the DTR line as its master-control line. In all operation modes, DTR must be asserted to cause the internal microprocessor to execute its program."

While your software might make up for this loss of features, HOT CoCo *cannot review a product outside of its stated capabilities. Although the Modem II can answer the phone line, its internal microprocessor cannot be used with the Color Computer. The built-in microprocessor adds to the price and yet doesn't allow Modem II to autodial. For this reason, I feel the purchase price is too high for autoanswer alone.*

Bobby Ballard

Nesting Habits

I wish to thank Stuart Hawkinson for a fine review of our product, SBasic 1.0 (*HOT CoCo*, April 1985, p. 72). However, there were two errors. First, SBasic does allow you to nest Basic commands within a CASEOF statement. Only CASEOF statements may not be nested within CASEOF. Second, the correct price of SBasic 1.0 is $24.95 in the U.S. and $30.95 in Canada, including shipping.

R. Lainevool
Tandar Software
Agincourt, Ontario

On Line Down Under

Does anyone want to communicate with a CoCo user from down under? I'd like to hear from someone prepared to exchange information via the CoCo. I promise to reply.

I'd also like to know of any CoCo programs resembling the arcade version of Battle Zone.

Joel Mielle
3 Rylston Court
Mt. Eliza 3930
Victoria, Australia

Help Wanted

Thank you for an excellent magazine. I've read every article since I got my first copy.

Can anyone give me the address of a company called Fenwal? Or can you provide clues to solving Madness and the Minotaur? I've been working on it for a month and a half.

Steve Warrick
1721 Chicago St.
Peru, IL 61354

Subscriber Complaint

The CoCo 2 is an excellent computer; *HOT CoCo* is a well-written, interesting magazine. However, no matter how good a magazine is, it isn't worth a cent unless there are customers to purchase and make use of the information it provides.

The management of *HOT CoCo* doesn't seem to know that the real test of a magazine's value comes when it is time for the customer to renew a subscription. Last June, I inadvertently paid twice for one renewal. I realized my error immediately and notified the subscription department. They told me I would receive a refund within four weeks. To make a long story short, I finally got my money back in January of this year.

You need to be reminded that, however good your product, you must satisfy your customers to stay in business.

C.W. Waldron
Acton, MA

You have our apologies for the poor service. Subscription service is the bane of all magazines. It creates such a logistical nightmare that most magazines, including HOT CoCo, *use an outside fulfillment service. Unfortunately, this sometimes results in slow (in your case, extremely slow) responses to complaints. We have recently streamlined our refund process because of problems such as yours. Anyone with subscription problems can call 1-800-645-9559, or in New York, 1-800-732-9119. Complaints take four to six weeks to process.*

By the way, of the four computer magazines in our publications group, HOT CoCo *has the highest renewal rate.—eds.*

Misquoted

R.F. Miller's program to remove spaces in a Basic program (*HOT CoCo*, March 1985, p. 56) causes a problem if you leave out the end quotation mark at the end of a string in a program line. When there is no end quotation mark, Remove continues until it finds the next quotation mark, which would occur at the beginning of the next string. It then deletes the spaces in that string.

There is a simple fix. Add:

```
425 BEQ NXTOKE
```

This line tests for a zero as well as a quotation mark.

Please keep up the good work with your magazine.

Jeff Larsen
Queensland, Australia

Handicapped Resources

I'm writing in response to Dave Meredith's request for computer information for the handicapped (*HOT CoCo*, April 1985, p. 10). I recommend the following resources:

Closing the Gap
Budd Hagan, Editor
Route 2, Box 39
Henderson, MN 56004
(published quarterly)

Communication Outlook
Artificial Language Lab
Computer Science Dept.
Michigan State University
East Lansing, MI 48824
(quarterly newsletter)

COPH-2
2030 W. Irving Park Road
Chicago, IL 60618
(computer group)

Trace Center
314 Waisman Center
University of Wisconsin
Madison, WI 53706
(has listings of programs written or adapted for the handicapped)

Marian G. Hall, System Manager
National Rehabilitation Information Center
4407 Eighth St.
Washington, DC 20017
(the Abledata data bank on rehabilitation products)

Personal Computers for the Physically Disabled
Apple Computer Co.
10260 Bandley Drive
Cupertino, CA 95014
(resource guide)

John Dalhaus
Highland, IL

Comparative, Superlative

I have recently finished my newest copy of *HOT CoCo*. I appreciate the excellent coverage given to terminal programs. Your reviews last year made me decide to buy Autoterm, and I have been quite satisfied with it.

I am interested in comparisons of other programs, like spreadsheets, word processors, and disk zappers. The presentation in your September issue was the most useful I have seen in any of the magazines for the CoCo. If you continue your objective comparisons, you will be doing your readers a service.

Thanks for *HOT CoCo!*

Stephen A. Haughey
Milwaukee, WI

Telewriter POKE

I was surprised at the number of readers who let me know they liked my modifications to Telewriter-64 (*HOT CoCo*, April 1985, p. 40). So many people wanted to change the default value of the Eps/Oki/LF menu item that it became impossible to answer everyone separately. The correct POKE address is: POKE 90,n.

Mick McGuire
Orange Park, FL

NEW DISK DRIVES

STARTING AT

$ 89.95

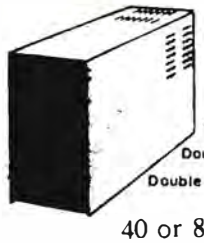
WITH CASE & POWER SUPPLY

$129.95

TANDON MPI TEAC

Speed 6 ms tk to tk and up
Capacity 250k unformatted
Tracks 40
Warranty **now 1 YEAR**

New Low Price!



40 or 80 Tracks
1/2 Hght. Teac/Panasonic

CALL FOR SALE PRICES

We carry only the finest quality disk drives • no seconds • no surplus

SATISFACTION GUARANTEED!! ALL DRIVES FULLY TESTED&WARRANTEED

Complete Disk Drive with Power Supply&Case ..... Teac $129.95

Two Drives in Dual Case & Power Supply ..... Teac ..... ~~$279.95~~ CALL

1/2 ht double sided double density Disk Drives (Panasonic/Teac) $ **$119.95**

1/2 ht double sided double density Disk Drive with ps&case ..... ~~$199.95~~ CALL

free How to use your new drive system on audio cassette

Single ps&case $44.95 Dual 1/2 ht ps&case .....$54.95 Dual ps&case ... **Call**

Color Computer Controller (J&M) **$129.95**

DRIVE Ø FOR RADIO SHACK COLOR COMPUTER

TANDON, MPI OR TEAC DRIVE (SINGLE SIDED 40 TRACKS SPEED 5 MS TRK TO TRK & UP)
POWER SUPPLY and CASE, TWO DRIVE CABLE WITH ALL GOLD CONNECTORS
J&M CONTROLLER, MANUAL and DOCUMENTATION ..... ~~$249.95~~ ..... **$ SALE!**

BUY THE BEST FOR LESS! CALL!

DRIVE Ø FOR RADIO SHACK COLOR COMPUTER

PANASONIC 1/2 HEIGHT DOUBLE SIDED DOUBLE DENSITY DRIVE 500K unformatted
POWER SUPPLY and CASE, 2 DRIVE CABLE WITH ALL GOLD CONNECTORS
J&M CONTROLLER, MANUAL and DOCUMENTATION ..... ~~$279.95~~ ..... Super!! **SALE!**

TAKE ADDED SAVINGS ON TWO DRIVE SYSTEMS

10 Diskettes
CONTROL DATA **$17.95**

DISKETTES **with free library case** .....

Unadvertised Specials ..... **$Call**

Drives cleaned, aligned & tested **$29.95**

TECHNICAL STAFF ON DUTY, PLEASE CALL FOR ASSISTANCE.

CALL US TODAY!! (617)234-7047
ORDER TOLL FREE **1-800-635-0300**

* DEALER INQUIRIES INVITED.
(617) 234-7047

TRUE DATA PRODUCTS

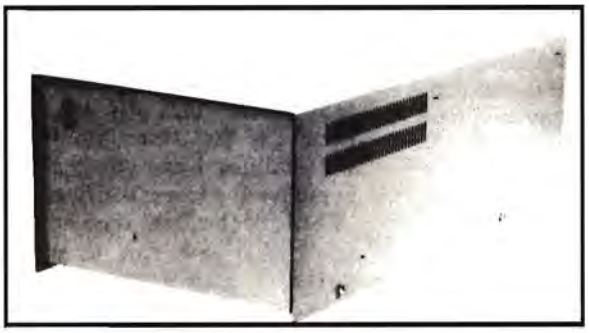
195 Linwood Street, P.O. Box 546
Linwood, Massachusetts 01525
(617) 234-7047
HOURS MON-SAT 9-6 (EST)

We welcome
• Visa / Master Charge
• Checks (allow 2 weeks for clearing)
• C.O.D. Add $2.00

New Hard Drives

---------- COMPLETE SYSTEM ---------- JUST PLUG IN ----------

Call For
BEST PRICE



Warranty - One Full Year

5 to 20 Megabyte, ready to run on the TRS 80 Model I/III/IV/4P, color computer,

64K UPGRADES $39.95

NEW **QUALITY VIDEO MONITORS**

Starting at **$79.95**

MONOCHROME

COLOR MONITORS

VIDEO DRIVER
ENABLES YOUR COCO TO OPERATE WITH A VIDEO MONITOR INSTEAD OF A TELEVISION!

SCREEN DUMP PROGRAM

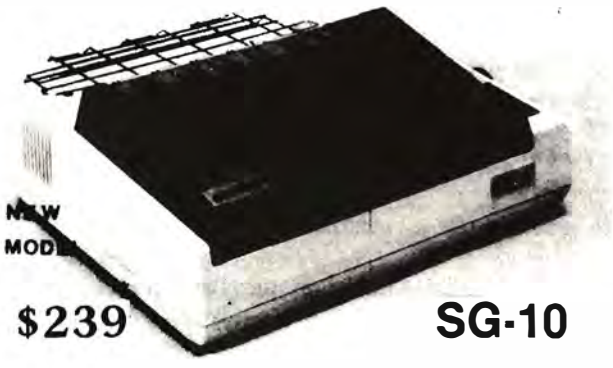
The best screen dump program for the Epson & Gemini printers ever!! Have the option of standard images or reverse w/regular or double sized pictures $19.95

PRINTER CABLES AND INTERFACES AVAILABLE
Call for current pricing

star PRINTERS

- 100 - 120 - 160 CPS
- Bidirectional Logic Seeking
- Friction and Tractor
- 9X9 Dot Matrix
- True Decenders
- High Res-Bit Image Block Graphics
- Super Script-Subscript
- Underlining
- Backspacing Doublestrike
- 5, 6, 8 1/2, 10, 12 and 17 Pitch
- Programmable Line Spacing
- SIX (6) MONTH WARRANTY

GEMINI 10X (9 Inch Carriage, 120cps) Friction and Tractor . . . . $CALL
GEMINI 15X (15 Inch Carriage, 120cps) Friction and Tractor $CALL
DELTA 10 (10 Inch Carriage, 160cps) Friction and Tractor . . . $CALL
POWER TYPE Letter quality



NEW MODEL

$239 SG-10

SP-3 INTERFACE for Color Computer

- 300-19,200 BAUD rates
- External to printer — No AC Plugs
- Built in modem/printer switch—no need for Y-cables or plugging/unplugging cables

Only:

~~$59.95~~

COMPLETE SYSTEM

299.95 NEW LOW PRICE

now with screen dump

Nothing more to buy!

Dealer inquiries invited

TRUE DATA PRODUCTS
195 Linwood Street, P.O. Box 546
Linwood, Massachusetts 01525

CALL US TODAY!! (617)234-7047
ORDER TOLL FREE 1-800-635-0300

Doctor ASCII

by Richard E. Esposito

Having technical difficulties? Consult the Doctor for an answer. Due to the volume of mail Doctor ASCII receives, we cannot guarantee that your query will be published. Please send a self-addressed, stamped envelope with all letters to Doctor ASCII, c/o HOT CoCo, 80 Pine St., Peterborough, NH 03458.

Q. In your January 1985 column, you said Radio Shack part no. 26-3018 will give my CoCo Extended Basic. My local Radio Shack store referred me to a local service office that would not sell it to me without the "required" installation. This is a mess! I recently upgraded my unit to 64K myself, and after that I think I can plug in a chip.—***Bill Brewer, Austin, TX***

A. Try ordering it using the replacement part no. AXX-7072. If they still give you a hassle, you can order the Extended Basic chip through one of several advertisers in this magazine.

Q. I would like to put two disk units in the space that Tandy uses for one on my work table. Is this possible?—***Daniel J. Marcella, Watervliet, NY***

A. Tandy is now selling single-sided, half-height drives for the CoCo mounted horizontally. But personally, I would go with double-sided TEAC drives.

Q. There are times when I want to inspect a Basic program without sending it to a printer. I'd like to recall code that has scrolled off the screen and have the ability to modify it. Is a 9- or 10-inch color TV adequate for a software-generated 51-character display?—***Spence Blakely, Portsmouth, RI***

A. CSAVE your programs with the A option (e.g., CSAVE "file".A). Then use a word processor that accepts ASCII files to do your editing. I prefer a 12- or 13-inch screen. I suggest you try one before buying.

Q. How can I disable the LIST command? When running a program that has columns of text, how do I get the cursor to go from column to column and wait for input?—***David Hutchinson, Alamogordo, NM***

A. POKE 383,158 disables LIST. The cursor address is stored in memory at addresses 136 and 137. By POKEing values into these addresses, you can make the cursor jump around the screen. For example, the following program lets you enter data alternately into two columns:

```
10 CLS
20 PRINT "COLUMN 1","COLUMN 2"
30 INPUT A
40 POKE 136,4: POKE137,48
50 INPUT B
60 INPUT C
70 POKE 136,4: POKE 137,80
80 INPUT D
90 PRINT"INPUT VALUES WERE:"; A; B; C; D
```

Q. Using Tandy's X-Pad, I create pictures that are stored in machine-language form. We use Network II system at our school. I would like to combine the X-Pad pictures with the Basic programs that use them so that a fifth- or sixth-grader need only download one file.—***Stephen M. Demko, Principal, Central Middle School, Vermilion, OH***

A. You can save a high-resolution picture and a Basic program together by fooling the CoCo into thinking that the whole thing is one machine-language program. Create your picture and write your Basic program. Add the following lines to the very beginning of your Basic program (renumber it if necessary):

```
10 POKE 27,000:POKE 28,000
20 POKE 29,000:POKE 30,000
30 POKE 31,000:POKE 32,000
40 the rest of your program
```

Now type this direct command: PRINT PEEK(27),PEEK(28) which will print two values. Edit lines 10, 20, and 30 so that those values are included in the POKEs instead of the 000s. You must have three-digit POKE values; use leading zeros if necessary. For example, if you receive the values 38 and 119, your edited lines will look like:

```
10 POKE 27,038:POKE 28,119
20 POKE 29,038:POKE 30,119
30 POKE 31,038:POKE 32,119
```

Finally, you can save the program and picture with:

```
SAVEM"name",&HE00,PEEK(27)*256 + PEEK(28),44661
```

for disk or:

```
CSAVEM"name",&H600,PEEK(27)*256 + PEEK(28),44661
```

for tape.

If all you need is to save the Basic program as a machine-language program without a picture, use a starting address of PEEK(25) *256 + PEEK(26) instead of &HE00 or &H600. Use either EXEC or RUN to execute the program after LOADMing or CLOADMing.

Q. Is there a way to write a Basic program that will, when finished, CLOAD and RUN another program without user intervention?—***Virgil W. Davis, Mountain Lakes, NJ***

A. Yes, but the auto-CLOADed and auto-RUN program must be CSAVEMed as a machine-language routine, as described in my answer to Stephen Demko (above). The first program must contain a CLOADM"second":EXEC ("second" is the name of the second program in the sequence). Any auto-run program must end with a STOP, END, or CLOADM"NEXT":EXEC to prevent the program from crashing into remnants of the previous program. You could fill a C90 cassette with 20 or 30 programs that run in sequence using this technique. You could even have PMODE 4 pictures on the screen during tape-loading sequences.

Q. The POKE 359,16 to inhibit a return to SCREEN 0,0 does not work on my 16K Extended Basic CoCo 2.—***Blaine Tempest, Lyons, CO***

A. The problem is due to the newer ROM version you have. POKE 359,57 will work on your machine.

Q. I can't load some machine-language cassette programs on my CoCo 2 with the disk controller plugged in. Is this because the disk memory is in the same place as the program?—***Doug Everhardt, Dansville, NY***

A. Disk Basic uses an additional 2K of memory starting at $600, which is at the start of graphics memory for Extended Color Basic. Because this 2K is used for disk memory, Disk Basic starts graphics memory at $E00. This incompatibility causes problems for many machine-language programs written for tape-based CoCos. The Tapefix program in "Disk Utilities" (*HOT CoCo*, September, 1983, p. 134) adds a loader that lets the program load from disk into high memory to 16K tape programs with this problem. Then after disabling Disk Basic, it moves the program into its normal position where it will execute.

Q. What does the error message, "?Extra Ignored," mean?—***Andrew Athan, Pelham Manor, NY***

A. Suppose you have the statement, 100 INPUT X,Y,Z, in your program. If, after running the program, you respond to that statement with 1.3, 4.5, 7.2, 99, you will get the "?Extra Ignored" message because the computer expects three values for X, Y, and Z, respectively. The fourth value is extra and, therefore, ignored.

Q. How can I dump a CoCo 2 directory to my DMP-120? Is there a program or command to scroll just a screenful of directory listings and then stop until a key is pressed?—***E.M. Goodsell, Hudson, OH***

A. The simplest way to print your directory is with a POKE 111, 254:DIR. The following program answers your second question. It will print 30 entries from your disk directory on a screen at a time in two columns:

```
10 CLEAR 1000
20 DIM I$(58),F$(2)
30 N=0
40 INPUT"DRIVE";DR
50 FOR I=1 TO 9
60 DSKI$DR,17,2+I,F$(1),F$(2)
70 FOR F=1 TO 2
80 FOR J=1 TO 97 STEP 32
90 IF MID$(F$(F),J,1)=CHR$(255)
THEN GOTO170 ELSE IF MID$(F$(F),
J,1)=CHR$(32) THEN GOTO 140
100 Q$=MID$(F$(F),J+8,3)
110 C$=MID$(F$(F),J,8)+"/"+Q$
120 N=N+1
130 I$(N)=C$
140 NEXTJ
150 NEXTF
160 NEXTI
170 CLS:PA=0
180 FOR I=1 TO N
190 IF I>30 AND PA=0 THEN PRINT@
480,"press enter for more";:PA=-
480:EXEC 44539:CLS
200 PRINT@PA+2+(I-1)*16,I$(I);
210 NEXTI
220 PRINT@480,"enter to repeat -
 break to stop";:EXEC44539
230 A$=INKEY$:IF A$=""THEN 230
250 GOTO170
```

Q. I want to find the memory locations of Telewriter-64's character sets to redefine some characters, including those for user-definable codes.—***Anton K.C. Fernhout, Chemin du Moulin, Switzerland***

A. Wizard, a software-modification package for Telewriter-64, is sold by Nexus, 2604 Bridalwood Drive, Knoxville, TN 37919, 615-522-1977. It includes modified character sets and the means to create your own. See the review in the February 1985 *HOT CoCo*, p. 84.

Q. Is there a way to merge two CSAVEd programs on tape with my 16K Extended Basic CoCo 2?—***Clarence Neece III, Madera, CA***

A. This simple method works on disk systems, too (if you don't want to save programs in ASCII format):

- CLOAD (LOAD for disk) the first of the two programs.
- Type PRINT HEX$(PEEK(25)*256 + PEEK(26)) and note the values. E.g., you might get 2601.
- Type PRINT HEX$(PEEK(25)*256 + PEEK(28) – 2) and again note the values. You might get 27FC.
- POKE the values from the previous step into locations 25 and 26—e.g., POKE 25,&H27:POKE 26,&HFC.
- CLOAD (or LOAD) the second program and renumber it if necessary to avoid conflicting line numbers.
- POKE the values from the second step into locations 25 and 26—e.g., POKE 25,&H26:POKE 26,&H01.

Repeat the third through sixth steps as additional programs are merged.

Q. How do you use the JOYIN [$A00A] mentioned in the back of the Color Basic manual?—***Kevin Holbrook, Binghampton, NY***

A. Type EXEC PEEK(&HA00A)*256 + PEEK(&HA00B). You can then find your joystick values by PEEKing addresses &H15A, &H15B, &H15C, and &H15D.

Q. I use Telewriter and a DMP-200 printer. I would give my left arm to justify text in my printer's proportional character mode.

Is there a way to get the Directory Print program from "Disk Utilities", *HOT CoCo*, September 1983, to shift its print to the right so that I can punch holes for a three-ring binder?

Can I use my disk drives, printer, and modem at the same time using the Deluxe RS-232 Program Pak?

Is there a way to get more memory out of the disk version of Telewriter?—***Bill Ripley, Lethbridge, Alberta***

A. Great Plains Computer Co. (P.O. Box 916, Idaho Falls, ID 83403, 208-529-3210) sells Stylograph III for $99.95 in both Flex and OS-9 versions. Even after adding the cost of Flex or OS-9, it's a lot cheaper than a left arm. Both versions support true justified proportional spacing. The OS-9 version has its own hi-res, 51 by 24 display. The Flex version has a larger text buffer. Both are compatible with PBJ's Word-Pak.

You can modify the program I wrote for E.M. Goodsell above to indent your directories. Delete lines 190, 220, 230, and 250 and replace line 200 with:

```
200 PRINT# -2,STRING$(10," "); I$(I)
```

Tandy has added a driver for a second RS-232 port via the Deluxe RS-232 Program Pak to its latest version, 01.01.00, of OS-9. By the time this article reaches print, someone should have an OS-9 terminal program that meets your needs.

The disk-based Telewriter has less buffer space because it cannot delete Extended Basic, as the cassette version does. Stylograph lets you scroll text off the disk so that you can edit files larger than the text buffer.■

Mindbusters

by Richard Ramella

There are only 26 of the little dancers; yet they open worlds to all who understand their myriad forms.

I refer to the alphabet.

Alphabet-based games and puzzles abound. Scrabble and Boggle are two copyrighted examples, the former drawing on crosswords, the latter on the pleasures of adjacency.

Anagrams use all the letters of one word or phrase to form another, so that *angered* might become *enraged* and *detour* might become *routed*. Palindromes are words or phrases that are spelled the same forward and backward: *A man, a plan, a canal: Panama*. Stephen King's novel *The Shining* uses the palindrome for *red rum* to form a plot device. Another example is the fruit peddler's lament: *No lemons, no melon!* Lewis Carroll created a word-puzzle format he called Doublets in which you transform a word into other words one letter at a time: *rum, ram, ham, hat.*

This month's puzzles require some ability with the alphabet. But before we get to them, here is the answer to last month's puzzle called Bull's Eye. To see the answer, run the program and type in the following sequence of movement commands, waiting after each for the computer to complete the move.

A, Z, A..Z,, A.Z, Z.AA, Z.Z,

Cryptonice

Listing 1, Cryptonice, requires Extended Color Basic. When you run the program, the computer presents a phrase consisting of two lines of words in reversed letters. It is encrypted so that none of the letters appears in the right place. But for every instance of a given letter, the same code letter has been substituted by Cryptonice. Move the orange cursor that ranges above the encrypted lines to a position over a letter about whose identity you would like to speculate. Then press the letter key that seems likely to you. The letter you choose appears over all instances of the encrypted letter. The cursor moves to the right only and wraps around from the end of the phrase to the beginning. When you have correctly deciphered the cryptogram, a tone sounds and the word "winner" wiggles at the bottom of the screen, from which you can escape by pressing the break key.

You can add to the phrases that Cryptonice holds by inserting new data material in lines 100 to 220. Note the total number of phrases in the DATA lines. If there are 100, for example, change the DIM expressions in lines 220 and 240 to read DIM W$(100) and DIM J$(100), respectively. In line 270, you would change FOR X = 1 TO 10 to FOR X = 1 TO 100 and K$ = J$(RND(10)) to K$ = J$(RND(100)). If you add a very long list of phrases to Cryptonice, you have to increase the CLEAR 2000 on line 220.

Puzzle Contest III

Now for this month's contest. Mindbusters puzzle contests run on the Color Computer in Color Basic or the MC-10. Program Listing 2 is called Alphasequence. The object is to come up with 15 words whose letters read in alphabetical order. Some examples of this are bin, cot, and tux.

When you run Alphasequence, it prints the alphabet at the top of the screen for reference. The prompt WORD 1 appears on the left side of the screen. Type a word and press the enter key. A legal word scores its number of letters squared. A three-letter word scores 9 points; a four-letter word scores 16. The program blocks words that do not read in alphabetical order and tells you to try again. Words with more than one of the same letter are not allowed. When you have entered 15 legal words, Cryptonice prompts you to press the enter key to see your words and final score. Next month: grid games. ■

Program listing on page 45.

Ed. note—To enter the contest, write your name, address, highest score, and the list of 15 legal words that make up that score on a piece of paper. Legal words read from left to right alphabetically and are listed in Webster's Ninth New Collegiate Dictionary. *Place the piece of paper in an envelope addressed to Richard Ramella, 1493 Mt. View Ave., Chico, CA 95926. Be sure to write your final score on the outside of the envelope flap. Entries must be postmarked by July 31, 1985. The winner will be the entrant with the highest score. In case of a tie, the winner will be decided in a random drawing. The winner will receive a free, one-year subscription to* HOT CoCo. *The winner's name and solution will appear in a future issue.*

System Requirements

Color Computer or MC-10
16K RAM
Micro Color or Color Basic

DIGISECTOR™ DS-69 VIDEO DIGITIZER FOR THE COCO

Give your COCO the gift of sight!
The Micro Works is happy to introduce the newest member of our Digisector™ family — the DS-69 Video Digitizer for your COCO. It has all the standard features of its big brothers but comes with a price tag that's right for you.

- **High Resolution** 256 by 256 spatial resolution.
- **Precision** 64 levels of grey scale.
- **SPEED!** ½ second for a full screen of video.
- **Compactness** Self contained in a plug in Rompack.
- **Ease of Use** Software on disk will get you up and running fast!

The DS-69 Digisector opens up a whole new world for you and your COCO. Your computer can be a security system, take portraits, analyze signatures, inspect assembly work . . . the DS-69 is your COCO's eyes. Use the DS-69 and a TV camera to get fast, precise conversion of video signals into digital data.

Powerful C-SEE™ software.
C-See is a menu-driven software package included with your DS-69. It provides high speed 5 level digitizing to the screen, high precision 16 level digitizing for superb hard copy printout, and simple software control of brightness and contrast. Or call our driver routines from your own Basic program for easy 64 level random access digitizing. Pictures taken by the DS-69 may be saved on disk by C-See and then edited by our optional MAGIGRAPH package for enhancements and special effects.

The DS-69 comes with a one year warranty. C-See supports both cassette and disk operation with the Multi-Pak adaptor and requires 64K. Cameras and other accessories are available from The Micro Works. Let your COCO see the World!

- DS-69 Digisector & C-See Software $149.95
- MAGIGRAPH Graphics Package on disk $ 39.95

Terms: Visa, Mastercard, Check or C.O.D.

Purveyors of Fine Video Digitizers Since 1977.

The MICRO WORKS Established 1977

P.O. Box 1110 Del Mar, CA 92014 (619) 942-2400

Circle Reader Service card #196

The Computer Room

by Scott L. Norman

Handi-Writer, Telewriter, And My Pal Edwin

Elsewhere in this issue, Jim Barbarello begins his multi-part series on a personal-assistance system for the physically handicapped. That gives me an excuse to discuss a related item. My subject is called Handi-Writer, and it's a low-cost cassette program designed to turn the Color Computer into a communications tool for anyone lacking the ability to speak. It works by displaying letters, numbers, and some common words and phrases on the TV screen, along with a moving cursor that lets you designate the next item you want to include in a message. All that's required is the ability to manipulate at least one switch of some sort.

I first heard of Handi-Writer through the *Bulletin on Science and Technology for the Handicapped*, a publication of the Project on the Handicapped in Science (run by the prestigious American Association for the Advancement of Science). *Bulletin* editor Susan Forman put me in touch with program author and vendor Howard Batie, who responded with a long letter describing his approach to designing systems for the real world, copies of both current versions of the program for the CoCo, and loads of documentation. He even sent the source code.

Handi-Writer was originally written for an Athetoid cerebal palsy victim who was unable to control a pointer or joystick, the selection tools most commonly used in computerized message-composition systems. The original Model I version was described in the *IEEE Computer Society Proceedings* for October 1981, and in *Byte* in December of the same year (p. 474). The first Handi-Writer has been in continuous operation ever since.

Both CoCo versions of the program generate a video display of message elements (the alphabet, numbers, common punctuation marks, and words or phrases), along with a few commands. In version 4.10, a blinking cursor scans through the material in a fashion designed to minimize the time required to construct messages. You compose messages by activating a switch as the cursor passes over the desired message element. You can use many types of hardware familiar to those who work with the handicapped community: pushbuttons, treadles, eyebrow "wrinkle" microswitches, tongue-actuated levers, breath-actuated "Sip-n-Puff" straws, and so on.

Version 4.51 is intended for those with the ability to use a joystick or a five-button keypad to control the direction of the cursor's motion, as well as to make a selection. It should make for faster message composition.

Messages are built up in a special area on the video screen, but you can route them to a printer by stopping the cursor as it scans over a PRINT command. There is also a provision for hooking an alarm bell or buzzer to the system to let you summon assistance or alert a companion or attendant to the fact that a message is waiting to be read.

Handi-Writer can display messages of up to six lines of 31 characters each, with word-wrap. If you enter a longer message, the topmost line is automatically sent to the printer as it scrolls off the screen.

Current versions of Handi-Writer include an elementary math-drill package. This was added at the request of the principal of a local school for the handicapped who was interested in using the computer to teach mathematical concepts. It could, perhaps, be deleted in favor of other features, such as driving the Radio Shack Appliance Controller.

Cursor scanning is based on a two-directional pattern: The cursor starts at a home position and moves horizontally until you press the control switch; then it shifts to a vertical sweep. That lets you pick the column and row in which the next message element lies.

The motion isn't continuous; the cursor stops and blinks at each position to give you time to activate the switch. Two switch closures are necessary to make a selection: one to freeze the cursor, and another to verify the choice. This desensitizes the system to movements you might make accidentally. The number of blinks, and thus the speed of the whole system are under your control.

At the fastest speed (one blink per position), the cursor takes just over eight seconds to scan across a row. It reaches the 30-second-per-scan mark at a speed setting of 4; the system can actually go down to 10, so it can accommodate a wide range of user abilities.

This method generally gives faster access than a TV-like raster. With the latter, on every pass you would have to wait for the cursor to scan through *everything* lying above the element you wished to select. Handi-Writer's alphabet is arranged so as to minimize the time required to get the most common letters.

The startup display includes 10 system commands (cursor speed, an erasure option, and so on) and one of two built-in lists of 21 words apiece. There is actually storage space for five such word lists in a 16K computer. You can order custom programs, although it would be simple enough for anyone with a smattering of Basic to create special word lists. There can also be two lists of seven phrases each, and your name and address can be programmed into the system and added to a message as a single selection.

Communicating through a program like Handi-Writer is not an unblemished joy. It is terribly slow compared to speech, and I'm sure that controlling the cursor can be frustrating to the new user. Still, it is probably unfair for anyone with verbal abilities to criticize the program too much. It certainly opens up the possibility of communication for some nonverbal individuals, and it seems to have been put together by someone who is both sensitive to his intended market and well aware of the appropriate design considerations.

Another Small Step For Telewriter-64

I've been a Telewriter user since my early days with a 16K CoCo. Although I've processed a lot of words with various versions of Howard Cohen's creation, I don't think my objectivity has been totally subverted; I will cheerfully admit that the program could use some touching up. You have to pump in too many routine commands at every working session, for example.

A number of patches have been published. One of the most ambitious, and very best, is Mick McGuire's in the April 1985 issue of this magazine (pages 40–43). He showed how to modify the loader routines and the two I/O (input/output) drivers, of all which are written in Basic, so as to customize Telewriter-64. I incorporated most of his suggestions, and now my system comes up with my standard print-formatting choices already in place. It also does its own switching between upper- (for specifying file names on the I/O menus) and lowercase (for the actual writing).

I wanted to make one additional improvement, however. Telewriter's standard text screen is buff, while the I/O and formatting menus appear on a green background. The buff and green differ greatly in apparent brightness on my monochrome monitor. If I set the controls to give me a pleasing text screen, the menus would be very murky and hard to read; if I turned up the brightness to give good-looking menus, the text screen would knock me out of my chair.

An obvious solution was to patch the program so as to produce a green text screen. The manual gives a POKE that is supposed to accomplish this, but you must enter it by hand after working your way through Telewriter to the disk I/O menu and selecting the Return-to-Basic option. It won't work if you type it in before you load the program—

clumsy, to say the least.

Filled with confidence after seeing how well McGuire's modifications worked, I started nosing about in U/BAS (the loader routine) for a likely place to make some changes. It turned out to be a bit tricky, but eventually I found a spot that worked with the POKE from the Telewriter manual. It is in the program line that loads and executes TW64, the machine-language editor itself. In the 64K disk version of the program, at least, that's line 210. Just insert

POKE 61262, 240:

immediately before the EXEC instruction, and save this as the new version of U/BAS. (Use the POKE address appropriate to your own RAM size and storage medium). Now sit back and enjoy; you never again ride the brightness control on your monitor or TV set.

Now if someone would only come up with a true cut-and-paste method of moving blocks of text, instead of that copy/insert/delete routine.

Edwin, The Writer's Friend

Last month, I reviewed Dynamic Electronics' ME-128-64 memory expansion kit, which gives a 64K CoCo a second 64K of RAM. The new memory makes the machine act like two independent computers; at this point you can't swap variables between the two halves of memory. Worse yet, if you are to switch successfully between programs stored in the two banks, both must be in the same display mode. Even that isn't always a guarantee of success; lots of program combinations cause the machine to hang up.

The product is really meant for the CoCo programmer, experimenter, or other enthusiast, and not for those of you who just use commercial applications software. The one exception is Telewriter-64; you can load it into both banks and put each copy to work on a separate text file. You can jump back and forth all day, using the toggle switch that comes with the kit.

But why would anyone want to write two things at once? Suppose you're preparing a paper that will have a bibliography at the end: You can work on the main text in one bank of RAM, then switch to the other bank whenever you want to add a reference. Type your footnote or bibliography entry into the second file, then return to your principal work and continue. You can print the two files separately when you're finished.

This method avoids having to push around a large chunk of text, intended for the end of a piece of writing, while you're working on earlier portions. That's a chore that really slows Telewriter down. I often use the two-file method to prepare the "Products Mentioned" box for these columns.

I have another application. Since I'm cursed with a porous memory where the fine points of grammar and spelling are concerned, I often use the second bank of RAM to hold a computerized crib sheet: a Telewriter text file containing notes on things that I tend to forget. I use the word processor's Find command to locate items on my reference list, and of course I can add to it whenever I work my way through a sticky point that I think will come up again. It's like having a very condensed version of some of my reference books on line.

My reference file is called Edwin, in honor of newscaster and linguistic consciousness-raiser Edwin Newman. Edwin resides on the disk that contains my working copy of Telewriter-64, my personal letterhead, and the setups for product reviews and these columns. He's slowly growing. Right now he has sections on rules of punctuation, compound words, and British/American spelling differences. (I write for a British magazine whose editor insists that I use what he thinks is "real English".)

Edwin will never replace a full-fledged spelling checker—a Telewriter text file is limited to about 4,100 words—but he can be awfully convenient for rapid access to notes on a wide range of topics. ■

Scott Norman is the manager of solid-state science at GTE Laboratories in Waltham, MA. Write to him at 8 Doris Road, Framingham, MA 01701.

Products Mentioned In The Computer Room

Bulletin on Science and Technology for the Handicapped
Susam B. Forman, Editor
c/o American Association for the Advancement of Science
1515 Mass Ave. NW
Washington, DC 20005

Handi-Writer versions 4.10 and 4.51
Howard F. Batie
400 Maple Court
Herndon, VA 22070
16K CoCo or CoCo 2
Cassette, $14.95

Telewriter-64
Cognitec
704 N. Nob St.
Del Mar, CA 92014
16K-64K
$49.95, cassette
$59.95, disk

ME-128-64
Dynamic Electronics, Inc.
P.O. Box 896
Hartselle, AL 35640
64K
$129 to $189, depending on computer model

REVIEW

by Scott L. Norman

The Facts On CoCo Max

Take your CoCo to the artistic max With this Macintosh-like graphics editor.

Have you been looking for an outstanding artist's kit for your Color Computer? CoCo Max from Colorware is a combined hardware and software system with a distinctive set of features. This well-conceived graphics editor deserves the attention of all those people interested in doing classy illustrations with their CoCos—assuming they aren't averse to having a great deal of fun in the process!

Icons And Menus

CoCo Max is unashamedly based on the Apple Macintosh's MacPaint program. That was enough to get my attention because I use a Macintosh at my office. MacPaint might be the best thing about the Macintosh. It is a powerful and innovative example of what a fresh look at how human beings interact with computers can accomplish. MacPaint is based on pioneering human-factors research carried out at Xerox's Palo Alto Research Center. Two of the more important concepts that emerged from this research—icons and pull-down menus—are used throughout CoCo Max.

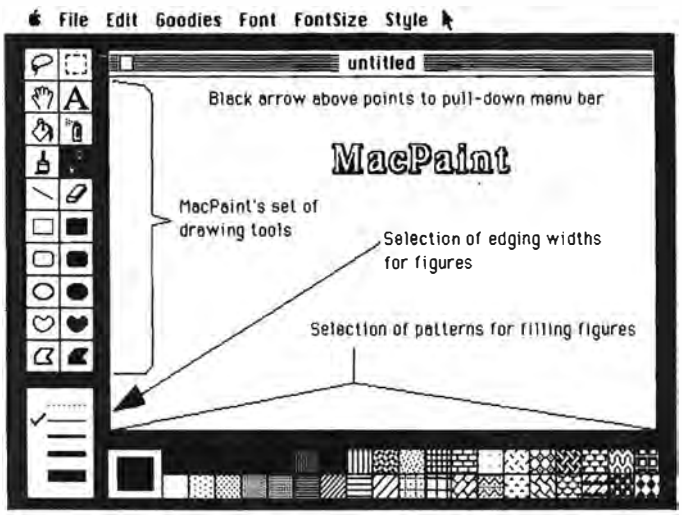


Fig. 1. The MacPaint Screen on an Apple Macintosh

The icons in CoCo Max and MacPaint are little pictures that represent program options or "tools" and are available at any time throughout these programs. A pencil-shaped icon, for example, represents freehand drawing. A shaded rectangular icon represents a tool that lets you construct rectangles containing predefined patterns. To select an icon and use its function, you point to it with the arrow-shaped cursor. CoCo Max works best with a mouse, but a joystick or graphics tablet works, too. Once you position the cursor on an icon, you press down the button on your cursor-control device. Then you move the cursor to the point on the screen where you would like to begin that operation.

Pull-down menus provide program options that do not lend themselves to graphic representation. Point the cursor to the name of a menu, click the button, and a collection of related options appears on the screen in menu format. The illusion is that you have pulled down a window shade on which the menu is written. (See Fig. 3.) You select options by sliding a cursor down the menu to the appropriate spot and clicking your interface-device button.

Icons and pull-down menus are attractive alternatives to the use of keyboard commands for making choices within a computer program. Icons have begun to appear in a variety of Color Computer applications software. There is some question about their role in other areas, but they work extremely well for CoCo Max's application.

The Hardware

All Color Computer graphics programs must reconcile the 64- by 64-element resolution of the computer's joystick port with the 256- by 192-element resolution of its high-resolution display. In other words, all Color Computer graphics packages face the problem of how to address individual pixels under the constraints of a limited I/O (input/ouput) capability. Several programs on the market employ a dual-range scheme in which the drawing instrument moves a cursor on a pixel-by-pixel basis within a small inset box. When the cursor reaches the border of the closeup box, it pushes the box across the screen to allow you to draw in a new location. This kind of system works, but requires you to become accustomed to the differences in speed between cursor and box movement.

CoCo Max takes a different approach, ignoring the joystick port. Your input device connects to CoCo Max's Hi-Res Input Unit, a small

circuit cartridge that plugs into the computer's ROM-pack port. The CoCo Max interface unit provides unrestricted access to every point on a PMODE 4 screen, giving the program a special feel.

CoCo Max works with Radio Shack's disk controller. It is not compatible with J&M's JDOS controller. Because both the disk controller and interface unit must connect to the ROM port, you need some sort of multiple ROM-pack interface. I use a simple Y-branching cable, but one of the interface units sold by Radio Shack or third-party vendors does the job as well. Colorware sells a 5-inch Y cable for $27.95.

Although I used both a joystick and Radio Shack's Color Mouse in my early trials, I quickly settled on the mouse for serious work. That was due partly to my previous experience with the Macintosh, but it is still a logical choice. The mouse is not a perfect instrument—the standard joke among Macintosh owners is that using one is like drawing with a bar of soap—but it works very well for this application.

The Software

The quickest way to gain an appreciation for CoCo Max's power is to head for the nearest Apple computer dealer and request a demonstration of MacPaint. The programs are that similar. As the accompanying figures show, the working screen of Colorware's product is outfitted with an amazingly MacPaint-like collection of drawing tools, fill-in patterns, and border widths. The contents of the pull-down menus are similar, although MacPaint has an additional Apple option that controls features peculiar to the Macintosh operating system and a font-size menu that reflects its greater diversity of text features.

There are a few differences in the way the pull-down menus work, dictated by Apple's patent position on menus. There are also considerable differences in screen resolution and RAM size between the CoCo and the Macintosh. Nevertheless, comparing the two programs is the only sensible way to describe CoCo Max. That is a tribute to the durability of the CoCo's design and the skill of CoCo Max author, Tim Jenison.

The first tasks to complete when setting up CoCo Max are making a working copy of the unprotected system disk and running a printer configuration program called Config to set the baud rate (from 600 to 9,600) and printer type. CoCo Max supports all the most common printers. These include most models from Radio Shack, Gemini, Epson, Star, Okidata, and C.Itoh. The program also has printer routines for the Apple Imagewriter and the Hewlett-Packard Thinkjet. And Colorware promises that more driver routines are on the way.

I found a bug in the driver for my Epson FX-series printer. It does not handle the double-strike option properly. This is not a fatal flaw, and Colorware claims to have fixed the problem just before this review went to print.

The CoCo Max system disk includes four demonstration images that you will probably want to delete after a short time. Unused printer drivers should go, too, because CoCo Max is strictly a one-drive program. It does not respond to attempts to store images on a second drive, and at six granules per picture, you can eat up storage space in a hurry. Prolific aritists will want to make several copies of their properly configured CoCo Max disk before going to work.

Working with CoCo Max is a delight. The program loads with a Macintosh-like chime and presents you with an option-laden screen topped by the titles of its pull-down menus. The CoCo Max tools let you draw freehand or piece together images from straight-line segments, rectangles, or ellipses. Rectangles can have square or rounded corners and, along with the ellipses, can be filled with a pattern selected from the palette at the bottom of the screen. There are also paintbrush, paint-bucket, and spray-can icons, whose functions help you fill in your drawing with various degrees of selectivity.

CoCo Max offers a total of 60 painting patterns in two different arrays; MacPaint users must get by with only 38 designs. Many of the extra CoCo Max patterns are intended primarily for generating artifact colors on a TV screen and might be of little interest to monitor owners. However, CoCo Max also lets you customize patterns with an option from the Goodies menu.

CoCo Max's set of drawing tools is filled out by options for erasing, adding text, and selecting portions of a picture for special treatment. Many of the tools can be accessed in more than one way or made to do double duty. Consider the eraser icon, for example. Moving the cursor to the eraser and clicking the mouse button selects the erase option. The cursor actually takes on the appearance of the eraser. When you pull the eraser away from the option palette, you can undo any part of a picture by wiping the icon over it as though it were a real eraser. If you double click the mouse button while the cursor is resting on the eraser icon, everything in the window portion of your image is erased.

A complete CoCo Max drawing, or page, can be considerably larger than one window. It can cover nearly all of an 8½- by 11-inch sheet of paper when you print it with the program's double-size option. It's a simple matter to scroll from one part of a page to another by using the hand icon. There is a separate menu command for clearing an entire page so that you can start a fresh drawing. There is also a method for printing the whole page or just as much as will fit on the video screen at one time.

Many other CoCo Max operations function in two or more ways. You can select Fat Bits, the closeup, pixel-by-pixel editing feature you use to touch up the details of a picture, from the Goodies menu or invoke it by double clicking the mouse on the pencil icon. Another Goodies option lets you choose from among 32 shapes and sizes for the paintbrush. You can also get to this menu by double clicking the brush icon.

The list of CoCo Max's special features is a long one. You can make "rubber stamp" copies of part of an image for use elsewhere on a drawing or store any portion of a drawing in a "clipboard" disk file for later use. You can also draw a continuous string of any shape by pressing down the shift key while you point the cursor to an object and pull it across your drawing. The shift key performs other tricks, too. It creates "transparent paint", which lets underlying patterns

Fig. 2. The CoCo Max Screen on the Color Computer

show through, changes the edges of solid objects from black to palette patterns, and so on.

A Prescription For Perfection

Despite my unflagging enthusiasm for CoCo Max, I would be remiss if I failed to point out a few missing features. Although its text handling features have a lot to offer, it would be nice to have a text-editing capability.

The five fonts that come with the system (Fig. 4) include a decorative style called Paris and a font called Fort Worth, which is large enough to use on overhead-transparency masters. One aspect of the program's text capabilities that might seem to be a shortcoming probably isn't. MacPaint has more fonts than CoCo Max, but I don't miss them. I use only a few of MacPaint's many type styles in the illustration work I do. Colorware has also mentioned that they intend to add fonts in the near future.

CoCo Max's style menu lets you make your text italic, boldface, shadowed, or outlined. These are nice features. One drawback is that it is a little difficult to place text on a drawing with precision. Some of the fancier options make for slow going, too. Putting shadowed, boldface, Fort Worth lettering on the screen, for example, calls for a great deal of computation between letters.

CoCo Max's Fat Bits feature can be a little tricky. The problem is not in using the pixel-at-a-time editor, but in locating the region on which you want to work. Once you have selected the Fat Bits mode, you must use the hand icon to move the drawing around until you locate the area you want. That takes practice. Unlike MacPaint, CoCo Max has no provision for predefining a region for magnification.

Some features you'll find in other graphics editors that are missing in CoCo Max concern sizing and positioning of images. Although you can select an isolated portion of a CoCo Max drawing for certain operations, there is no provision for stretching or shrinking it. The program also does not have an option for rotating a region by 90 degrees. These are useful features that the graphics editor should provide. CoCo Max does, however, give you the option to flip image segments through 180 degrees horizontally or vertically.Colorware has dropped the inertia function, mentioned in the manuals of an early version of CoCo Max.

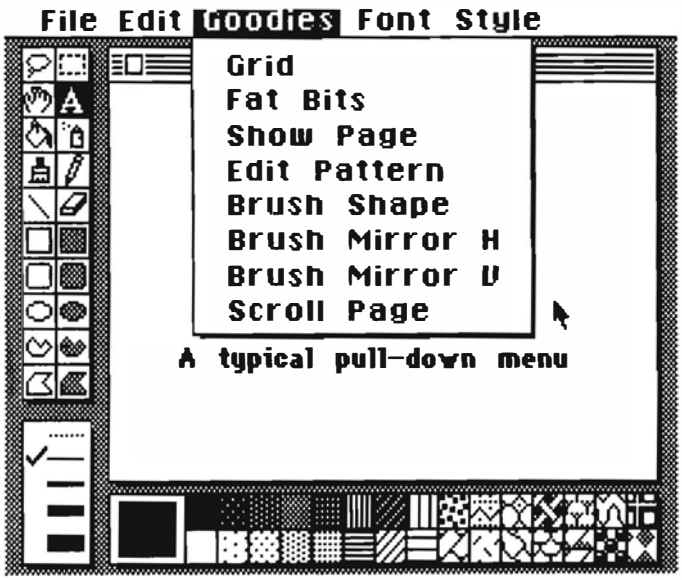


Fig. 3. An Example of CoCo Max's Pull-Down Menus

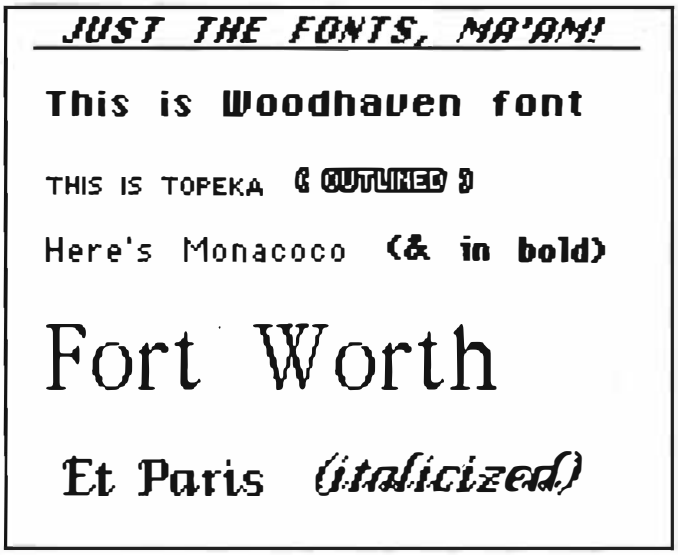


Fig. 4. CoCo Max's Fonts

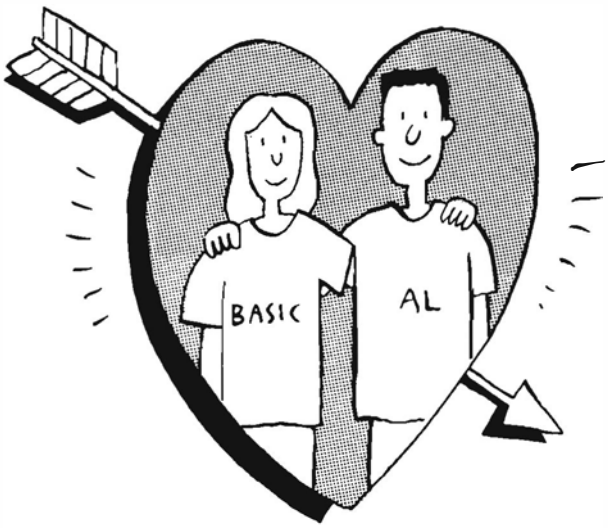
Something to note before you buy CoCo Max: It has no telecommunications capability. CoCo Max is meant to be used as a stand-alone drawing program or a generator of title screens and game boards for Basic programs. If you are interested in exchanging graphics with your friends on-line or transferring them to and from a BBS, you should check into other programs.

CoCo Max is, however, able to receive imagery directly from another computer or other video source (not tested here). Its manual describes how the Digisector DS-69 video digitizer, an additional hardware/software system from The Micro Works, can be used to take electronic snapshots (sometimes called "frame grabbing") from a variety of standard video sources, such as cameras, VCRs, or computers. You can store these kinds of images on disk for manipulation by CoCo Max.

Summary

CoCo Max puts the fun back into computing. Despite some minor drawbacks, this graphics package does an excellent job of offering the kind of state-of-the-art environment you find on a much more expensive machine. This is a program with which you can create. Colorware has invested the kind of time and research in this product that virtually secures its success—and that shows up on your screen.■

CoCo Max is a hardware/software device manufactured by Colorware Inc., 78-03B Jamaica Ave., Woodhaven, NY 11421, 718-647-2864. It requires 64K, a disk drive, and a mouse, joystick, or graphics tablet. It sells for $69.95.



Getting It Together

Unite Basic and machine code without DATA statements.

Do you ever want to combine machine-language code and a Basic program? The routine accompanying this article makes that combination painless. You will no longer have to load the Basic and the machine language separately.

Furthermore, using Program Listing 1 instead of DATA statements saves memory. With DATA statements, you separate elements with commas. Each comma is a byte. If you have 300–400 bytes of machine code, you use an additional 200 bytes for commas. With several DATA statements, you will use more memory for line numbers and DATA tokens.

Program Explanation

With your machine code in memory and protected from Basic by a CLEAR command, load and run Listing 1. The routine creates a tape or disk file of Basic strings, containing an ASCII representation of your machine code and the restore sequence.

First, the program asks for the starting and ending addresses of the machine code in memory. If the starting address is greater than the ending address, you must resubmit the addresses.

Respond with a T or D when the program asks whether you are using a tape or disk. You must then specify a file name. For tape systems, enter a legal file name of up to eight characters. At the prompt, place a tape in your cassette player, press the record and play buttons, and then press your CoCo's enter key. For disk systems, add a slash or dot and a three-character file extension to the file name. Be sure to add a colon and the drive number if you are not using drive 0.

There are 64 or fewer characters per string. Since the strings are in ASCII, you can merge them with a Basic program. The listing as written safely generates Basic strings for 1,664 bytes (64 by 26) of code. To allow 2,600 bytes, change the byte value from 64 to 100 in lines 160–180 and from 63 to 99 in line 190.

You are now ready to merge the Basic strings with your program. For tape systems, turn off and then turn on your computer to clear memory. If you are creating a new Basic program, CLOAD the Basic strings from tape and type the rest of your Basic program.

If your Basic program already exists, you must use another method to combine the Basic strings and the Basic program. CLOAD the Basic strings and then execute the following line in direct mode (without line numbers) to find the starting address for Basic:

```
PRINT PEEK(25);PEEK(26)
```

Copy the values returned by this line—you will need them later.

Next, execute another line in direct mode:

```
POKE25,PEEK(27):POKE26,PEEK(28):NEW
```

Now CLOAD your Basic program. Using the RENUM function, renumber your program to start at line 200 so that the line numbers of the Basic strings and the restore sequence do not conflict with your Basic program.

Execute two POKEs to addresses 25 and 26 with the numbers you copied above. If, for example, the values returned were 38 and 1, you would type the following in direct mode:

```
POKE25,38:POKE26,1
```

System Requirements
16K RAM
Disk, or Extended Color Basic

Illustration by Michael Bartalos

For disk systems, turn your system off and on to clear memory. LOAD your Basic program and renumber it to start at line 200. To use MERGE for loading in the Basic strings, you must add a line such as:

```
1 CLEAR 1000,xxxx:S = xxx + 1
```

where xxxx is the last address to be used by Basic, and xxx + 1 is the starting address of machine code. If your program needs more than 1,000 bytes of string space, increase the space cleared accordingly.

Renumber the combined program if you wish. SAVE or CSAVE your merged program before running it.

Applications

If you are comfortable with machine code, you can use the technique I used in Listing 1 to change Assembly language to Basic without an assembler. Create a Basic string from the Assembly program's object code and then POKE the string into memory.

For example, the routine below issues a form feed to a printer:

```
86FE      LDA  #$FE   PRINTER DEV NO. - 2
976F      STA  $6F    DEV NO.
860C      LDA  #$0C   FORM FEED CODE
AD9FA002  JSR  [$A002] ISSUE FORM FEED
0F6F      CLR  $6F    CLEAR DEV NO.
39        RTS         RETURN TO BASIC
```

The Basic program below shows the object code from the first column of the Assembly routine converted to a Basic string in line 20. The FOR. . .NEXT loop POKEs the code into memory.

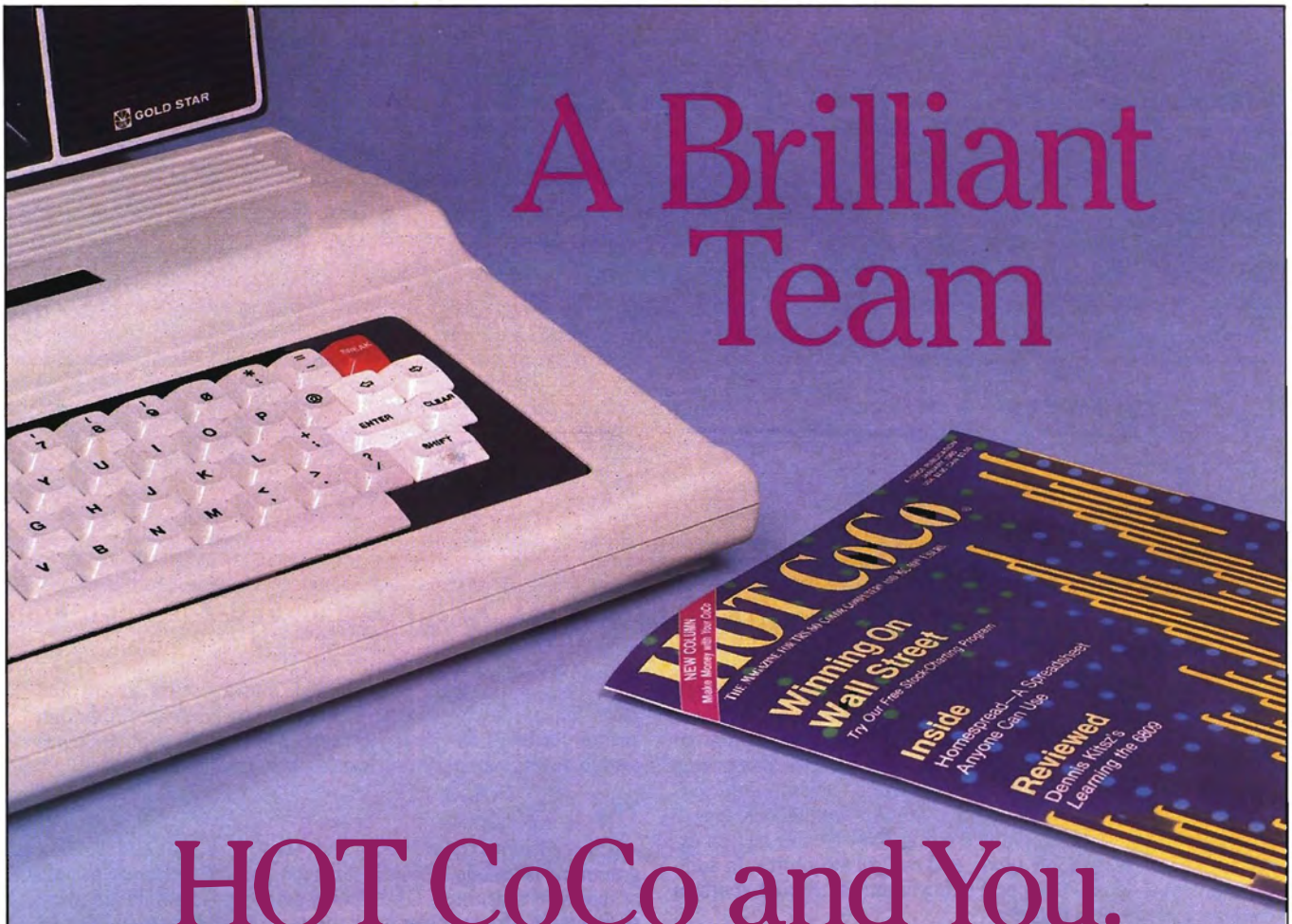
```
10 CLEAR 200,&H7FF0
20 DA$ = "86FE976F860CAD9FA0020
   F6F39"
30 I = 1
40 FOR AD = &H7FF0 TO (&H7FF0 +
   (LEN(DA$)/2) - 1)
50 D = VAL("&H" + MID$(DA$,I,2))
60 POKE AD,D
70 I = I + 2
80 NEXT AD
90 EXEC &H7FF0
```

(If your printer uses hexadecimal &H1E, decimal 30, instead of &H0C, decimal 12, to perform a form feed, change line 20 accordingly.)

There are two ASCII characters for each byte of machine code. A leading zero is significant. Bytes of values less than 16, decimal, must have a leading zero. So a byte value of five is written as 05.■

Program listing on p. 52

Address correspondence to Frank Tipps, 1837 Cartlen Drive, Placentia, CA 92670.



A Brilliant Team

HOT CoCo and You.

Those with the latest and most thorough information run their computers with confidence. Those without it are run ragged with frustration.

Color Computerists who've gained the upper hand read **HOT CoCo**, the monthly magazine that unlocks the full potential of TRS-80* and MC-10* Color Computers.

HOT CoCo's inside stories can make you and your Color Computer a brilliant team. By subscribing to **HOT CoCo** for the low one-year price of $24.97, you'll get information-packed issues at 30% off the cover price!

- **HOT CoCo** specializes in presenting time-efficient, money-saving programs for business, home, and school. Easy to use, and practical!
- Novice and expert users alike will benefit from **HOT CoCo**'s tips and tutorials. Sharpen your skill. Watch your CoCo become more versatile!
- Graphics? Discover how easy it is to create your own!
- Games? **HOT CoCo**'s wide assortment provides hours of entertainment and challenge for the entire family.
- Looking to gain an advantage by adding more hardware or software? Read our reviews first—decide for yourself what's worth its weight in gold, and what isn't worth a hill of beans.

Everything on the pages of **HOT CoCo** can bring out the best in you and your Color Computer. A wealth of knowledge—for an entire year—for **just $24.97!**

Order your subscription today by filling in the coupon, or by calling TOLL FREE 1-800-258-5473. In New Hampshire, call 1-924-9471.

☐ **YES!** **I want to be part of a brilliant team...**

Send me 12 issues of HOT CoCo for the low price of $24.97. I'll save 30% off the newsstand price.

☐ Payment enclosed ☐ Bill me

Name _____

Address _____

City _____ State _____ Zip _____

Canada & Mexico $27.97, one year only, US funds drawn on US bank. Foreign surface $44.97, one year only, US funds drawn on US bank. Foreign airmail please inquire.
Allow 6 to 8 weeks for delivery.

HOT CoCo • PO Box 975
Farmingdale, NY 11737

357B4

* TRS-80 and MC-10 Color Computers are registered trademarks of the Radio Shack Tandy Division of Tandy Corp.

by Mark D. Goodwin

C'ing is Believing

The best of Assembly in a high-level language.

The high-level structured programming language, C, is used to develop systems software, such as operating systems, high-level programming languages, word processors, program generators, and assemblers. Because C is so effective for systems software development, many programmers mistakenly believe C is either a low-level or mid-level programming language. In fact, C supports character, integer, and floating-point data as well as extensive operators, variable types, and control structures. In addition, many versions of C support string functions through routines contained in run-time libraries.

A Brief History

To better understand C, it is helpful to look at its roots. C derives from Algol 60, the grandfather of structured programming languages designed by an international committee in 1960. The committee's objective was to create a language with uniform syntax and a modular program structure. In 1963, Cambridge University and the University of London introduced a spin-off of Algol, CPL (Combined Programming Language).

CPL wasn't as general as Algol and was difficult to learn and cumbersome to implement. So in 1967, Martin Richards of Cambridge wrote a simpler version of CPL called BCPL (Basic Combined Programming Language). To further simplify CPL, Ken Thompson of Bell Laboratories wrote B in 1970. Because of simplification, both BCPL and B were limited.

Dennis Ritchie, also of Bell Laboratories, wrote C in 1972, retaining the simplicity of B and BCPL and incorporating CPL's generality. Because systems programmers found C easy to use and powerful, it quickly became their preferred language. Although written for Bell Lab's Unix operating system, C's popularity has lead to its implementation on non-Unix computer systems.

```
/*
** Factorial Function
** Returns n!
*/
factorial (n)
int i,j,n;
  {
  i = 1;
  for (j = 1; j <= n; ++j)
    i = i*j;
  return(i);
  }
```

Program Listing 1. Factorial Function

The C Language

Unlike interpreted languages such as Basic, C is a compiled programming language. In contrast to compilers that produce pseudo or machine code, most C compilers generate Assembly-language source code. An assembler must then compile this source code into machine code, making compilation slow. Nevertheless, assembling compiler-generated source code allows experienced Assembly-language programmers to modify their programs before they are assembled into machine code. This enhances C's usefulness as a systems software development tool.

The building blocks of all C programs are functions. They are either predefined by the C language, contained in the run-time library, or defined by the programmer. The function in Listing 1, which returns the factorial of an integer passed to it, is typical of a programmer-defined function.

Although the C language supports character, integer, and floating-point data, some compilers support only characters and integers. A compiler without the floating-point feature might seem limited, but systems software rarely requires it. So a version of C with only the character and integer subsets is still powerful. Most C compilers support many variable types: static, local, global, array, pointer variables, and programmer-defined structures.

Like other structured languages, C supports many program-control structures: IF. . .ELSE, WHILE, DO. . .WHILE, FOR and SWITCH. . .CASE. . .DEFAULT. If wisely used, C's flexible control structures promote efficient programming.

Most implementations of C provide a library of common functions, such as terminal input/output, file input/output, and string functions. These are not part of C; their syntax and semantics differ from version to version.

I hope this overview encourages readers to take a detailed look at the C programming language. Anybody who writes Assembly-language can benefit from C. An accessible, high-level, structured language with the power of Assembly language, C is unequaled as a tool for developing systems software. ■

Address correspondence to Mark D. Goodwin, HCR 79, Box 103, Orland, ME 04472.

Another Price Breakthrough from Radio Shack

A VERSATILE HOME PRINTER FOR ONLY $200

Now you can get the versatility and convenience of a full-size printer at home—without a full-size price tag.

For just $199.95, you can take home our new DMP-105 dot-matrix printer for high-speed printouts on full-size fanfold and sheet paper.

High-Speed Performance and Bit-Image Graphics

The DMP-105's bi-directional print head prints 80 characters per second and makes the DMP-105 ideal for data processing, program listings and general-purpose use.

And the DMP-105's 30 special block graphics characters or its bit-image graphics mode make it easy to insert charts, diagrams and other graphics into your text.

Three Print Modes for Personalized Text

The DMP-105 even lets you choose condensed, normal and elongated printing modes. An adjustable tractor feed lets you use 4″ to 9½″ fanfold paper, and a friction platen lets you use single letter-sized sheets easily.

The DMP-105 includes parallel and Color Computer-compatible serial interfaces, and prints an original, plus one copy.

See It Today

Just visit your nearby Radio Shack Computer Center or participating store or dealer for a demonstration of our affordable new DMP-105.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION
Circle Reader Service card #4

Price applies at Radio Shack Computer Centers and at participating stores and dealers.

Send Me a New 1985 Computer Catalog!
Mail To: Radio Shack, Dept. 86-A-173, 300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

LANGUAGE

by Delmar E. Searls

A Short Course In PASCAL

Will Pascal replace Basic as your programming language choice?

Many computer-science educators consider Pascal to be the best beginner's language available. It is a modern language incorporating many desirable features. Pascal encourages the development of well-written, properly structured programs. This short course will explore some of Pascal's features.

Clarify The Problem

The first step in developing a program is to clarify the task to be performed—a step that is often overlooked. The description of the problem should include the input, the processing needed, and the output.

My example is a program that gives a short addition or subtraction quiz. The program determines the type of quiz desired, generates 10 problems, and displays a report stating how many problems were solved correctly. The program prompts for a response and tells you whether or not the response for each problem is correct. If you give an incorrect response, the correct answer appears. All the problems use numbers from 1 to 99 and none have a negative answer.

Program Structure

Once the problem statement is clear, the programmer can sit down and design a solution. Stepwise refinement is a problem-solving technique that takes a large problem and breaks it down into its component parts. If necessary, each of these parts is broken into smaller parts and so on. This approach is also called "top-down" and is a fundamental part of top-down programming.

In my example, the problem can be broken down into three major components: determine the type of quiz desired, give the quiz, and print the results. This breakdown is shown graphically in Fig. 1. This chart also shows the program structure. The top rectangle represents the main driver, which consists of calls to the procedures or functions that perform the tasks identified in the rectangles on the second level.

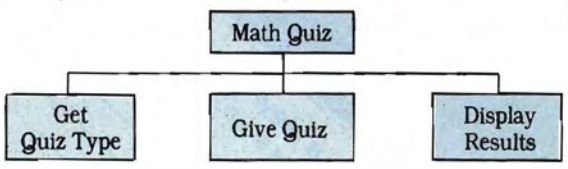


Fig. 1. A Preliminary Structure Chart for MathQuiz

Consider the tasks on the second level. Determining the quiz type has two components: display a menu and get the user's choice. Giving the quiz involves four subtasks: generate two random numbers, display the problem, get the user's response, and process the response. Each subtask represents a specific, well-defined module. You have completed the refinement process.

Before you begin writing the program, you must add one more task to the second level. The random-number generator needs a seed number that determines the sequence of numbers generated. Consequently, the main driver must call a module that lets you input this seed number.

The complete structure chart appears in Fig. 2. Rectangles on a given level (except the topmost) represent subroutines (Pascal procedures or functions) that are called by modules on the next level up. Thus the topmost module (the main driver) calls the modules on the second level. The Get Quiz Type module calls the Display Menu module, and the Give Quiz module calls the Get Random Number module.

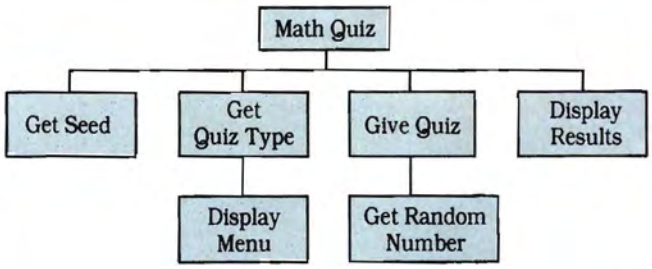


Fig. 2. The Complete Structure Chart for MathQuiz

Although giving the quiz consists of four subtasks, I included the last three within the module itself. Only the random-number generator will be written as a separate function. Other programmers might write all four subtasks as procedures or functions with the Give Quiz module, which calls each in turn.

Program Logic

Note that you have planned your program structure without worrying about how you will implement each module. Don't be

intimidated by an apparently difficult module. Plan the program structure first and then worry about the logic. Breaking the problem down into a number of relatively simple subtasks greatly reduces the amount of time spent on developing the logic. Each module represents a separate logic unit, and you can work on the modules in any order you desire.

The Heading

The first section of every Pascal program is called the "heading" and has the following syntax:

```
PROGRAM program name (INPUT, OUTPUT);
```

Words in uppercase letters are Pascal keywords and must appear as shown. Lowercase represents words that are determined by the programmer. "Program name" is just the name the programmer gives the program. The semicolon at the end of the heading is required because Pascal uses it to mark the end of program statements.

INPUT and OUTPUT refer to standard input and output files, which would normally be the keyboard and the screen, respectively. Some Pascal versions let you omit the (INPUT, OUTPUT) phrase. For our program the heading is:

```
PROGRAM Mathquiz;
```

The program name, procedure and function names, and constant and variable names are referred to as "identifiers." Identifiers consist of any number of letters and digits but must start with a letter. Usually both upper- and lowercase letters are allowed. Identifiers can be of any length, but the compiler recognizes only the first few characters. For example, Computerware's Pascal recognizes only the first four letters.

Declarations

Following the heading is the declarations section. Unlike Basic, Pascal requires that you declare all variables. In Basic you distinguish between floating-point numbers and string variables by appending a dollar sign to the end of the string variable. In Pascal you declare each variable by name and type.

There are several standard types: INTEGER, REAL, CHAR, and BOOLEAN. INTEGER and REAL correspond to integers and floating-point numbers, respectively. CHAR refers to a single character (letter, numeral, or punctuation). BOOLEAN variables have one of two values: TRUE or FALSE.

In addition to variables, you can declare constants in the declaration section. Constants are declared much like variable assignments in Basic. For example:

```
CONST  Pi = 3.14156;
       BestGrade = 'A';
       PayRate = 8.75;
```

The CONST marks the beginning of the constant declaration section. You could put all declarations on one line, but using a separate line for each declaration makes the program easier to read. A semicolon is required after each declaration. Constants (if any) are declared immediately after the heading. In the example above, the second constant is a character constant. Single quotation marks rather than Basic's double quotation marks denote character data.

You define new variable types following the declaration of constants. Declared types are one of Pascal's more powerful features, but space does not allow a detailed explanation here. I'll give you an example, though, to illustrate the concept.

One of the most useful declared types is a string:

```
TYPE String = ARRAY [1..20] OF CHAR;
```

In this example, I have defined a string to be an array of 20 characters. Once this has been done, you can declare a variable as a string. A string variable prints out one character at a time. The characters are accessed using subscripts much as one-dimensional arrays are accessed in Basic. (Some Pascal versions include language extensions that greatly simplify the use of strings.)

Variables are declared after any constants or types. The variable declarations begin with the Pascal keyword VAR:

```
VAR  HighScore,
     LowScore,
     Median: INTEGER;
     Mean: REAL;
     Grade: CHAR;
     DataValid: BOOLEAN;
     TestName: String;
```

```
PROGRAM MathQuiz;

(***********************************************
     This program generates a short math
     quiz in addition or subtraction.  The
     number of right and wrong answers is
     displayed at the end of the test.
***********************************************)

CONST M = 10000;     (* Used in Random Number *)
      B = 3821;      (* Generator function    *)

VAR Seed,  (* Seed for Random Number Generator *)
    NumRight,            (* Number right *)
    NumWrong: INTEGER;   (* Number wrong *)
    QuizType: CHAR;      (* Type of quiz desired *)
```

Program Listing 1. The Declarations

Note that you can write several variables of the same type with commas between them. You can make the last declaration (TestName) only if the TYPE string was previously declared as above.

Only global variables are declared in the main declaration section. Global variables are used in a number of modules within the program or in the main driver itself. Local variables, on the other hand, are those that are used by a procedure or function and are not needed elsewhere.

My program will use two constants: M and B. Both of these are used in the random-number generator. I also need four global variables: Seed, NumRight, NumWrong, and QuizType. Listing 1 gives the declarations. Notice that comments explain the purpose of the various identifiers. A comment always begins with the characters (* and ends with the characters *), as in this example: (*Generator function*).

The Main Driver

The remainder of the program consists of the statements that perform the desired task. I'll go through the program in the same order it was written. I followed the structure chart working from top to bottom, and from left to right.

The top box represents the main driver. (See Listing 2.) Notice that it is written between the keywords BEGIN and END. The period after the keyword END is required and signals the end of the program to the compiler.

The main driver consists of three procedures and one function call (the second line). Notice that a semicolon separates statements. (The semicolon is usually not required immediately before the END keyword.)

The main driver comes last in the listing, although I wrote it first. The procedures and functions must be inserted into the program in front of the main driver.

Getting The Seed Number

The first procedure (Listing 3) called by the main driver illustrates Pascal's input and output statements. Every procedure

```
(*=============================================

     THE MAIN DRIVER STARTS HERE

=============================================*)

BEGIN
  GetSeed;
  QuizType := UserInput;
  GiveQuiz;
  ShowResults
END. (* PROGRAM MathQuiz *)
```

Program Listing 2. The Main Driver

starts with the keyword PROCEDURE followed by the procedure name. Local variables can be declared but you don't need any here. The statements in a procedure (or function) are written between the BEGIN and END keywords as was done in the main driver. Notice that a semicolon follows the END in a procedure or a function. Only the main driver has a period after the END keyword.

Think of the WRITELN statement as a built-in procedure. It is much like Basic's PRINT statement. When you use WRITELN, the next output prints on the next line down. If you use the WRITE statement, the next output starts in the next print position of the same line. List items to be printed within parentheses. If two or more items are listed, separate them by commas. Enclose literal strings within single quotation marks.

```
(*-----------------------------------------------
     The procedure GetSeed allows the user
     to input the seed value for the random
     number
generator. ----------------------------------------------*)

PROCEDURE GetSeed;

BEGIN
  WRITE('ENTER A THREE DIGIT NUMBER: ');
  READ(Seed);
  WRITELN
END;
```

Program Listing 3. The First Procedure

The user prompt is printed using the WRITE statement. As a result, the number you enter appears on the same screen line as the prompt. The WRITELN statement at the end of the procedure generates a line feed and a carriage return. Consequently, whatever is next printed to the screen will appear at the beginning of a new line.

READ and READLN are the built-in input procedures. Think of the input as being on computer punch cards. READLN means to read the data list and go to a new card. READ means to read the data list and stay on the same card waiting for the next READ statement. Similar meanings apply if the data is being read from a text file of some sort. When using the keyboard as the input device, the READ statement is sufficient.

Getting The Quiz Type

The statement QuizType := UserInput; illustrates two points. First, the assignment operator in Pascal is a colon followed by the equals sign. If you are used to Basic programming, you'll frequently forget this point. On the right side of the assignment statement the compiler expects to find one of four things: an arithmetic expression, a variable (already assigned a value), a constant, or a function name. In this case the name on the right refers to a function. The function name is used as though it were a variable.

```
(*-----------------------------------------------
     The procedure DisplayMenu is called by
     the UserInput function and displays a
     list of the available types of
quizzes. ----------------------------------------------*)

PROCEDURE DisplayMenu;

BEGIN
  WRITELN('    MATH QUIZ MENU');
  WRITELN;
  WRITELN('    1. ADDITION');
  WRITELN('    2. SUBTRACTION');
  WRITELN;
END; (* DisplayMenu *)

(*-----------------------------------------------
     The function UserInput is called in the
     main driver and returns the type of
     quiz that the user desires to
take. ----------------------------------------------*)

FUNCTION UserInput: CHAR;

VAR Option: CHAR; (* User's choice from menu *)
    Ok: BOOLEAN;  (* Valid option flag *)

BEGIN
  Ok := FALSE;
  DisplayMenu;

       (* Get a valid user response *)

  REPEAT
    WRITE('ENTER NUMBER OF DESIRED QUIZ: ');
    READ(Option);
    WRITELN;
    IF (Option = '1') OR (Option = '2')
      THEN Ok := TRUE;
  UNTIL Ok;
  UserInput := Option
END; (* UserInput*)
```

Program Listing 4. Function and Procedure Code

A function is written much like a procedure. A function, however, always returns a single value. In this case the value returned by the function will be a CHAR type and assigned to the variable

QuizType. (I could have done the same task with a procedure but chose to use a function in order to illustrate the technique.) From Fig. 2 you see that the FUNCTION User Input calls the PROCEDURE DisplayMenu. Listing 4 is the code for the function and the procedure. In the program it would immediately follow the PROCEDURE GetSeed.

The first line of code in a function has the syntax:

FUNCTION function name: type;

You must declare what type of variable the function is to return. In the example the function will return a CHAR value (that is, a single character). There is one basic rule that applies to functions. Somewhere within the body of the function a value must be assigned to the function name. (Remember that the function name is, in effect, a variable.) Usually this assignment is made just prior to the END statement, as it is in this example.

Notice the declaration of local variables within this function. These variables are recognized only within the function and also within procedures or functions called by the function. The variable Ok is a Boolean variable and will have a value of either TRUE or FALSE. It indicates a valid user response (a 1 or a 2). The value of Ok is initially set to FALSE, and the menu appears on the screen. The PROCEDURE DisplayMenu consists entirely of WRITELN statements.

The next part of the function illustrates one of Pascal's iteration or looping structures. The REPEAT. . .UNTIL loop repeatedly executes the same statement (or group of statements) until a given condition is met. If a group of statements is to be executed, they are included between BEGIN and END as shown. In this case the body of the loop repeats until the variable Ok has a value of TRUE. This occurs only if the user selects a valid option. The first line in the body of the loop prints a prompt for the user. The second line gets a response, and a third line sends a line feed and a carriage return.

The fourth statement is an example of a selection structure. If the user responds with either a 1 or a 2, then the variable Ok is assigned a value of TRUE and the program exits the loop. Otherwise the value of Ok remains FALSE and the body of the loop executes again. Unlike Basic, when two or more conditions appear in the IF clause each is enclosed within parentheses.

The last line of the function assigns the value of the valid response (Option) to the variable UserInput, which is the name of the function. This variable (UserInput) should not be used in any other way in the function. If it is, the program will think you are calling the function a second time, and then a third, and then a fourth, and so on. It is similar to infinite loop and generates an error when you try to run the program. Back in the main driver. the value of UserInput will be assigned to the variable QuizType.

Giving The Quiz

The PROCEDURE GiveQuiz is the main portion of the program and it consists primarily of a large loop that executes 10 times. The body of the loop generates an arithmetic problem, gets a response from the user, and checks the user's response. It also updates the number right or the number wrong as needed. This procedure calls the FUNCTION RandomInteger, which returns a random value in the range 1 to 99. (Some Pascal versions have a built-in random-number generator, but the Computerware version does not.) The GiveQuiz module (Listing 5) immediately follows the code in Listing 4.

Continued on page 68

```
(*-------------------------------------------
     The function RandomInteger is called
     by the GiveQuiz procedure and returns
     a random integer
value. -------------------------------------------*)

FUNCTION RandomInteger: INTEGER;

BEGIN
  REPEAT
    BEGIN
      Seed := (Seed * B + 1) MOD M;
      IF Seed < 0
        THEN Seed := -Seed
    END
  UNTIL Seed > 99;
  RandomInteger := Seed DIV 100
END; (* RandomInteger *)

(*-------------------------------------------
     The procedure GiveQuiz is called by
     the main driver and generates 10
     questions.  The number of correct and
     incorrect responses is
maintained. -------------------------------------------*)

PROCEDURE GiveQuiz;

VAR First,      (* First number in problem *)
    Second,     (* Second number in problem *)
    Answer,     (* The correct solution *)
    Response,   (* The user's solution *)
    Temp,       (* A temporary storage location *)
    Count: INTEGER;  (* A counter *)

BEGIN
  NumRight := 0;
  NumWrong := 0;
  FOR Count := 1 TO 10 DO
    BEGIN
      First := RandomInteger;
      Second := RandomInteger;

      (* Display the problem *)

      CASE QuizType OF
        '1': BEGIN  (* Addition *)
               Answer := First + Second;
               WRITE(First:2,' + ',Second:2,' = ')
             END;
        '2': BEGIN  (* Subtraction *)
               IF First<Second
                 THEN BEGIN
                        Temp := First;
                        First := Second;
                        Second := Temp
                      END;
               Answer := First - Second;
               WRITE(First:2,' - ',Second:2,' = ')
             END
      END; (* of CASE *)

      (* Get the user's response *)

      READ(Response);
      WRITELN;

      (* Check user's response *)

      IF Response = Answer
        THEN BEGIN
               NumRight := NumRight + 1;
               WRITELN('THAT IS RIGHT!')
             END
        ELSE BEGIN
               NumWrong := NumWrong + 1;
               WRITELN('SORRY, THAT IS WRONG');
               WRITELN('THE ANSWER IS ',Answer)
             END;
      WRITELN
    END (* DO Loop *)
END; (* GiveQuiz *)
```

Program Listing 5. GiveQuiz Module

UTILITY

by David Meredith

All Sorts Of Sorts

These programs are made to order.

Sorting is perhaps the most important, and the most studied, computer operation. It occupies an estimated ¼ to ½ of the processing time on mainframe computers. Consequently, the invention of fast-sorting algorithms has been a priority for computer scientists. Their discoveries form the basis of this article.

Efficient sorting algorithms are also crucial for many CoCo applications, especially those involving data storage and retrieval. As the number of items to be sorted approaches 1,000, the difference between an efficient and inefficient sort varies from a few minutes to an hour.

I will show you five standard sorting algorithms. The number and complexity of items to be sorted determines which sort you should use. The first four are in-memory sorts for data stored in an array. The last is for data in disk files too long to fit in memory.

First, let me explain sorting. Suppose you have entered data into an array using numeric or string data. Often it is convenient to keep such data in numerical or alphabetical order within the array, but it might not have been entered in the correct order. Sorting is the rearrangement of data in an array into ascending (or descending) numeric or alphabetic order. Table 1 shows an array before and after sorting.

		1	2	3	4
Before Sorting	A(I)	10	6	11	3
After Sorting	A(I)	3	6	10	11

Table 1. An Array Before And After Sorting

If you have never written a sorting program, it would be instructive for you to stop reading, go to your CoCo, and try writing a sorting program by filling in the missing subroutine in Program Listing 1.

System Requirements

16K RAM
Disk Basic

Selection Sort

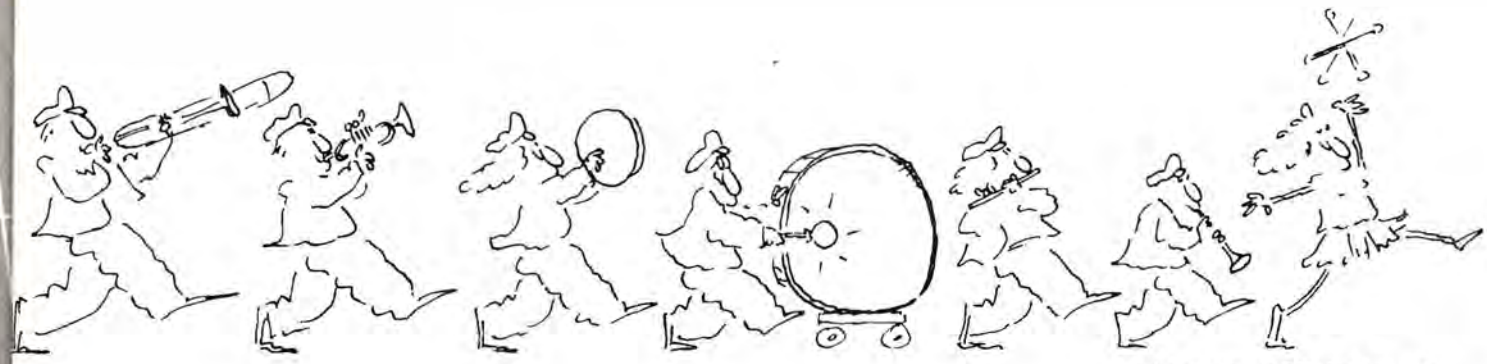
The simplest sort is the selection sort (Program Listing 2). You can test this program by merging it with Listing 1. This method first puts the smallest element of the array into the first position of the array. Then it puts the smallest remaining element into the second position of the array and continues this process until the array is sorted.

Listing 2 accomplishes this task by locating the smallest element in the array and exchanging it for the first element of the array.

10	3	3	3
6	6	6	6
11	11	11	10
3	10	10	11
Initial Array	After First Swap	After Second Swap (no change)	After Third Swap

Table 2. Selection Sort

Illustration by Chris Demarest



It then finds the smallest remaining element and exchanges it for the second member of the array. This process continues until the array is sorted. Table 2 shows the steps in the selection sort of an array.

The selection sort is slow on large arrays, but it is compact. (See Program Listing 3.) It works best on arrays of fewer than 30 items.

Bubble Sort

The bubble sort (Program Listing 4), a component of the fast Shell sort described below, is more complex than the selection sort but slightly faster. It is useful for arrays with 20 to 50 elements.

The bubble sort sorts the initial segments of an array A(1..N). Suppose A(1..I – 1) is already sorted. To extend the ordering to segment A(1..I), examine elements A(I – 1), A(I – 2), . . .until you find A(J) < = A(I) or J = 0. Then move A(J + 1). . .A(I – 1) to A(J + 2). . .A(I), and place the original value of A(I) into A(J + 1). Now A(1..I) is sorted.

A(I) is said to "bubble up" through A(I – 1), A(I – 2). . .until it finds its proper level. Table 3 shows the steps using a bubble sort on an array.

10	6	6	3
6	10	10	6
11	11	11	10
3	3	3	11
Initial Array	Two Elements Sorted	Three Elements Sorted (no change)	Four Elements Sorted

Table 3. Bubble Sort

Although the bubble sort is perhaps 20 percent faster than the selection sort on random data, it is still slow compared to the advanced sorting methods discussed next. The bubble sort is superior for ordering partially sorted data.

The selection sort cannot take advantage of partial ordering in the data to be sorted. It always performs the same number of comparisons on the data (line 140 of Listing 2). But the bubble sort rushes through previously sorted data, executing line 130 (Listing 4) only once for each element of the array. Use the bubble sort when a small percentage of your data is out of order.

Shell Sort

The exceptional efficiency of the bubble sort applied to partially sorted data led the Dutch computer scientist D.L. Shell to propose a fast sorting method based on repeated applications of the bubble sort in 1959.

Shell discovered that he could increase the speed of the bubble sort by sorting subsets of an array before sorting the entire array. Table 4 shows an array A(1. .N) subdivided by Shell into subarrays. The subarrays are not the same length, but their length is either INT(N/M) or INT(N/M) + 1.

To sort the array A(1. .N), bubble-sort each of the subarrays constructed in Table 4; then

A(1), A(1 + M), A(1 + 2M), . . . ;
A(2), A(2 + M), A(2 + 2M), . . . ;
. . .
A(M), A(M + M), A(M + 2M), ...

Table 4. The Array A(1. .N) As Divided Into M Disjointed Subarrays For The Shell Sort

divide M by 2, rounding off your result to an integer, and repeat the process.

If you start with M = INT(N/2), the first subarrays sorted contain only two or three elements. On each successive pass there are approximately half as many subarrays, but each is about twice as long as those from the previous pass. Eventually the entire data set is one subarray.

After the first pass, the subarrays are partially sorted, so bubble-sorting them proceeds rapidly. That is why the Shell sort is so much faster than the bubble sort.Table 4 shows the Shell sort at work.

The Shell sort is surprisingly effective. Despite its moderately complex code, it runs faster than the bubble sort on arrays longer than 20 items. On arrays as short as 100 items, it runs more than twice as fast. Although it is theoretically not as fast as the quick sort presented next, it is so much simpler than the quick sort that it actually runs faster on arrays shorter than 1,000 items.

10	10	10	3
6	6	3	6
11	11	11	10
3	3	6	11
	M = 2		M = 1
Initial Array	Bubble Sort A(1) A(3) (No Change)	Bubble Sort A(2) A(4)	Bubble Sort A(1. .4)

Table 5. Shell Sort. Initially M = 2

Quick Sort

The quick sort, invented by the British computer scientist, C.A.R. Hoare in 1962, is the fastest general-purpose sort. (See works by Aho, *et al*, and Sedgwick under Further Reading for tricks on improving its performance.)

To quick-sort an array, first rearrange, or partition, its elements into two segments, with the elements in the left segment smaller than those in the right segment. To sort the two segments, partition each of them into two smaller segments, and then partition each of the four segments into two parts. Continue until all segments are length 1. Sorting is then complete.

Writing a program to keep track of partitioning is difficult and requires a stack. In Program Listing 6, the stack is the array ST(.), and the variable SP (stack pointer) is its index.

Save information by placing it in the lowest unused position of the stack (always the position with index SP + 1), and recover information from the highest filled stack position (always the position with index SP). The information recovered is always the information most recently placed on the stack and not yet removed. The stack is a LIFO (last in, first out) stack.

Once you have a stack, you can write a quick sort. Table 6 outlines the code for the quick sort contained in Listing 6.

1. Begin with an empty stack, L = 1, R = N. The segment of A(1. .N) being worked on is always the segment A(L. .R).

2. If the segment has length 1 (L = R), then go to step 6.

3. Find the first two different values in the segment A(L. .R) and set V equal to the larger value. If all the values are the same, then go to step 6 (the segment is sorted).

4. Partition the segment. Working from both ends of the array, move array elements smaller than V to the left end of the array, and move elements larger than V to the right end. When you are done, you will have an index D between L and R such that A(L. .D – 1) <V< = A(D. .R). (See Listing 6 for more details of this operation.)

5. You now have partitioned the segment A(L. .R) into two parts: A(L. .D – 1) and A(D. .R). Save the left and right boundaries of the longer partition on the stack, and make L and R equal to the left and right boundaries of the shorter partition. Go to step 2.

6. If the stack is empty, you are done sorting. Otherwise get the left and right boundaries (new values for L and R) for a new segment from the stack and repeat step 2.

Table 6. Outline of Quick Sort

The stack keeps track of work to be performed. First, partition the array into two segments. Remember this means permuting the elements so that those in the lower seg-

ment are less than those in the upper segment. Lines 150–220 accomplish this task. Then place the boundaries of the longer partition on the stack (line 240), and return to line 140 to begin partitioning the shorter segment.

Partitioning, saving the longer segment and repartitioning the shorter segment, stops when the shorter segment reaches length 1 because it cannot be further partitioned. When a segment of length 1 is presented to line 140 for partitioning, the program gets the boundaries for the next segment to partition from the stack (line 260). If the stack is empty, the sort is complete.

The quick sort is not easy to understand, but you can see it work if you add a printing trace to a line in Program Listing 6:

```
130 PRINT "PARTITIONING A FROM"; L;
    "TO"; R: IF R>L THEN 160
```

Each time this line prints, it shows you the left and right boundaries of the segment about to be partitioned.

The quick sort is most efficient when applied to large sets of randomly arranged data. It has too much overhead for short arrays, and it is not very efficient on partially sorted data sets. If you need to sort partially sorted data, it is best to use the Shell sort or the bubble sort.

Performance Report

I tested each of my sorting programs on long and short arrays of random decimals between 0 and 1. See Tables 8 through 10 for results. The tabulated values are averages. Individual program runs vary from 10 percent for short arrays to 5 percent for the longest arrays.

Clearly the bubble sort is best for short arrays; the Shell sort is fastest for arrays of intermediate length; and the quick sort asserts its power on very long arrays. Let's see why this is true.

Both the selection sort and the bubble sort are $O(n^2)$ sorts; that is their speed is proportional to the square of the number of items sorted. On each pass through the main loop (lines 100–180), the selection sort does N – I comparisons (lines 130–150) and three data moves (line 160) for a total of $(N-1) + (N-2) + \ldots + 1 = (1/2)N^2 - (1/2)N$ comparisons and $3(N-1)$ data moves. In contrast, each pass through the main loop of the bubble sort performs on the average $(I-1)/2$ comparisons and $(I+1)/2$ data moves (lines 120–140) for and average total of $(I+2+\ldots+(N-1))/2 = (1/4)N^2 - (1/4)N$ comparisons and $(3+4+\ldots+(N+1))/2 = (1/4)N^2 + (3/4)N - 1$ data moves.

Since data moves are less time consuming than comparisons, the bubble sort is generally faster than the selection sort. In empirical tests (Table 7), the bubble sort always ran faster than the selection sort.

The Shell sort and the quick sort are, in principal, much faster than the selection sort or the bubble sort. An analysis of the Shell sort (see Aho, *et al*, under Further Reading) shows that it is an $O(n^{3\backslash 2})$ sort, where n is the number of items to sort. For reasons I don't understand, my Shell sort increases its running time by slightly less than this.

The quick sort is an O(nlog(n)) sort. That is,

L 10	L 3	L 3	3
6	6	R=D 6	6
11	D 11	11	L 10
R 3	R 10	10	R=D 11
Initial Array	After First Partition	After Second Partition (No Change)	After Third Partition

Table 7. Quick Sort. Partitions Of Length 1 Are Skipped

Number Of Items	Selection Sort	Bubble Sort	Shell Sort	Quick Sort	Best Sort
10	1.0	0.9	1.3	2.6	1.7
20	3.0	2.7	2.3	4.0	2.4
50	16.9	12.4	8.6	11.5	5.8
100	64.9	54.4	21.1	25.0	15.1
200	253.4	192.9	52.1	54.4	31.5
500	1,573.7	1,103.2	150.4	153.2	101.3
1,000	> 1 hr.	> 1 hr.	341.9	344.3	223.1
5,000	est. > 1 day		2,689.7	2,088.3	1,445.6

Table 8. Sorting Times In Seconds

Number Of Items	Selection Sort	Bubble Sort	Shell Sort	Quick Sort
10	19	38	29	24
20	51	146	95	67
50	135	688	319	196
100	288	2,762	834	507
200	585	9,851	2,225	1,149
500	1,482	60,189	6,387	3,345
1,000	2,982	249,788	16,793	7,356

Table 9. Numbers Of Data Moves While Sorting (Experimental Data)

Number Of Items	Selection Sort	Bubble Sort	Shell Sort	Quick Sort
10	45	31	29	24
20	190	126	92	99
50	1,225	646	315	331
100	4,950	2,666	837	818
200	19,900	9,654	2,204	1,867
500	124,750	59,701	6,225	5,572
1,000	499,500	248,794	16,220	12,548

Table 10. Numbers Of Data Comparisons While Sorting (Experimental Data)

the time required to sort n items is approximately proportional to nlog(n). My quick sort obeys this rule closely. The time (in seconds) to sort n items is nearly 11.5nlog(n).

Calculations show that, as n grows large, n^2 is larger than $n^{3\backslash 2}$, which in turn is larger than nlog(n). Moreover, as n grows, the differences between the three functions increase dramatically.

I don't mean that if n^2 is larger than nlog(n), then the bubble sort will be slower than the quick sort. The running time for the bubble sort is not exactly n^2. Rather it is proportional to n^2. In my experiments, the time required to bubble-sort n items is $.005n^2$, and the time to quick sort n items is 0.115nlog(n). Thus the bubble sort is faster than the quick sort when $n < 38$.

An $O(n^2)$ sort can be faster than an O(nlogn) sort when n is small. But for large arrays the O(nlogn) sort is faster, and it is much faster for very large arrays.

Similarly, as the array grows, the Shell sort increases more rapidly than does the time for the quick sort. Yet the constants of proportionality are such that the quick sort actually runs more slowly than the Shell sort on all but the longest arrays.

Sorting uses three kinds of computer operations: data comparisons, data moves, and control structures (e.g., FOR. . .NEXT loops, GOTOs, and index manipulation). Clever programming reduces the time spent controlling program flow, but nothing can reduce the number of data comparisons and moves required by each sorting algorithm.

Tables 9 and 10 show results of experimental comparisons and data moves required by each sorting algorithm on variously sized data sets. Use these tables to select the sorting method best suited to your situation.

If data comparisons are time consuming, then use the bubble sort (for short arrays). If your data comparisons are fast but data moves are slow, then the selection sort might run faster than the bubble sort. Note that although the selection sort requires far more data comparisons and correspondingly fewer data moves than the bubble sort, the sum of both operations for the two sorts is approximately equal.

For longer arrays use the Shell sort or the quick sort. The quick sort requires fewer data comparisons and data moves than the Shell sort. However, the complexity of the quick sort's control structure makes it slower than the Shell sort on all but the largest data sets.

In Assembly language, the percentage of running time devoted to managing array indices, FOR. . .NEXT loops, and GOTOs can be made smaller than is possible in a Basic program. With medium-sized data sets, an Assembly-language quick sort is faster than an Assembly-language Shell sort.

Performance Improvements

You can speed up the bubble sort by about 20 percent by adding one line and changing another:

```
95   A(0) = -1E30
130  IF A(J)>T THEN A(J+1)=A(J):
     J=J-1: GOTO 130
```

The large negative value placed in A(0) is called a guard. Its purpose is to terminate the loop in line 130 if J = 0. The new line 130 performs one comparison instead of two each time it is executed; this accounts for the speedup.

To improve the Shell sort by 5 to 10 percent, initialize M to a value one less than a power of two. Replace line 100 with:

```
100  M = 1
105  IF M+M<=N THEN M=M+M:
     GOTO 105 ELSE M=M/2-1
```

The most dramatic speedup results from improving the quick sort. I call this improvement the "best sort". It performs 30 percent faster than the quick sort (Table 7).

You'll recall that the quick sort is slow on short arrays, while the bubble sort is fast on partially sorted arrays. The best sort takes advantage of these facts, using the quick-sort algorithm to partition an array down to unsorted segments of length less than 14. It then uses the bubble sort to finish sorting the array.

The best sort (Program Listing 7) begins like the quick sort, but segments shorter than length 14 are not partitioned. (Compare lines 140 in Program Listings 6 and 7.)

When the best sort completes its quick sort (lines 120–260), the array A(1. . .N) is a series of segments of length 13 or less. The contents of each segment are smaller than those in the next segments, but within segments the data is unsorted. The best sort then applies the bubble sort to this partially sorted array.

Merge Sort

Sometimes you must sort a disk file that is too large to fit into memory. In such cases, use the merge sort (Program Listing 8), which requires that only a small part of the data being sorted reside in memory.

The concept of the merge sort is simple. First, read blocks of data from a disk file into an array of length NN. Make NN as large as possible to reduce the running time of the sort. Sort these blocks in memory one at a time, and write them alternately to two files called the output scratch files.

Now the data is divided into sorted blocks in the two output files, with approximately half the data in each file. There are no separators between the blocks of sorted data. One block ends and another begins when a smaller datum follows a larger one in the file. The comparisons with W in line 310, 320, 340, and 370 of Listing 8 test for this condition.

Next change the two output scratch files into input files and open two new output scratch files. Merge the first blocks from the two input files into one double-length sorted block on the first output file. Merge the second blocks from the input files and send the result to the second output file. Send the next merged blocks to the first output file, the next to the second output file, and so forth.

When you finish merging all the blocks on the two input files, the data resides in sorted blocks in the two output files, just as it did at the end of the first step. But now the blocks are twice as long as before, and there are only half as many.

Repeat the merge process as often as necessary, reversing the roles of the two input and two output files each time. Passes double the length of the sorted data blocks. Eventually the data is merged into a single long block and sorted.

As the blocks grow longer, they might become too large to reside in memory. However, it is not necessary to read an entire block into memory to accomplish the merge. Memory must contain only the next datum from each of the blocks being merged.

In my merge sort, up to three copies of the data reside on the disk while the sort is running. One of these copies is the original data file, which is retained for security purposes until the sort is complete. The other two copies are in the input and output files. Thus the sort is limited to files of less than 50,000 bytes.

If you kill the original data file immediately after reading it (line 190), you can use the entire disk for scratch files, increasing the maximum file size to 75,000 bytes. Don't forget to eliminate the reference to the original data file in line 400. If you use two disk drives and remove the original data disk after reading it, you can save the original data file and increase the maximum file size to 150,000 bytes.

The merge sort is slower than the other sorts in this article. If the first step produces B sorted blocks, then the number of passes required to merge these into a single sorted sequence is roughly $\log_2(B)$. Including the first step, during which the initial data blocks are sorted, the merge sort reads and writes the entire data file $\log_2(B) + 1$ times. The time required for the merge sort is roughly double the time required to read and write the file that many times.

You can improve the running time of the merge sort by increasing the number of scratch files used. You can merge as many as seven input files to seven output files using the CoCo DOS. Aho, *et al*, describe this more complex merge sort (see Further Reading below).

The merge sort uses only sequential files. In fact, it was originally used with magnetic tape files. You might think that direct-access disk files would allow faster sorting methods than sequential tape files, but that isn't so. Sequential files can be read and written so much faster than direct-access files that, even for a disk-based sort, sequential files are preferable to direct-access files.

Further Reading

The bible of sorting is Knuth's *The Art of Computer Programming—Vol. III: Sorting and Searching* (Addison-Wesley, Reading, MA 1973). However, you have to read volumes I and II before volume III makes sense. So, only serious students of computer science study Knuth.

A more accessible text is Aho, Hopcroft, and Ullman: *Data Structures and Algorithms* (Addison-Wesley, Reading, MA 1983). All the sorting algorithms presented here are analyzed in this text.

Sedgwick's article, "Implementing Quicksort Programs," *Communications of the Association for Computing Machinery*, (vol. 21, no. 10, Oct. 1978, pp. 847–857), discusses the quick sort.

For the last word (as of this writing) about sorting, read J.L. Bently's "Programming Pearls," *Comm. ACM*, (vol. 27, no. 4, April 1984, pp. 287–291). Reading Dr. Bently's fascinating column, which appears monthly in the *Comm. ACM*, requires no advanced knowledge. It is filled with important ideas for experienced and inexperienced programmers.

These reference materials should be available at any college library, many of which are open to the public. Take advantage of the one near you. ■

See program listing on p. 46

David Meredith is a professor of mathematics at San Francisco State University. Address correspondence to him at the Dept. of Mathematics, 1600 Holloway Ave., San Francisco, CA 94132.

by James J. Barbarello

The CoCo provides the brains for this person-assistance/ home-control project.

Photo by André Moreaux

The John-B System, Part One

In May 1984, I was involved in a serious automobile accident. One of the results was an incapacitated right arm (I am right-handed). This seemingly minor alteration created a major change in my lifestyle. I couldn't perform simple functions like putting my clothes on, tying my shoes, or writing a letter! I felt helpless, but I reminded myself that it wasn't permanent, and I could still walk and use my left arm fully. It did, however, make clear the importance of the work being done by a friend from Johnstown, PA.

Joe Sobieski first contacted me in August 1983 to obtain permission to use my Typewriter for the Handicapped program ("PC Printer—Part II", *HOT CoCo*, August 1983). Joe has been making CoCo-based assistance systems for severely handicapped people for some time, and wanted to include the type-

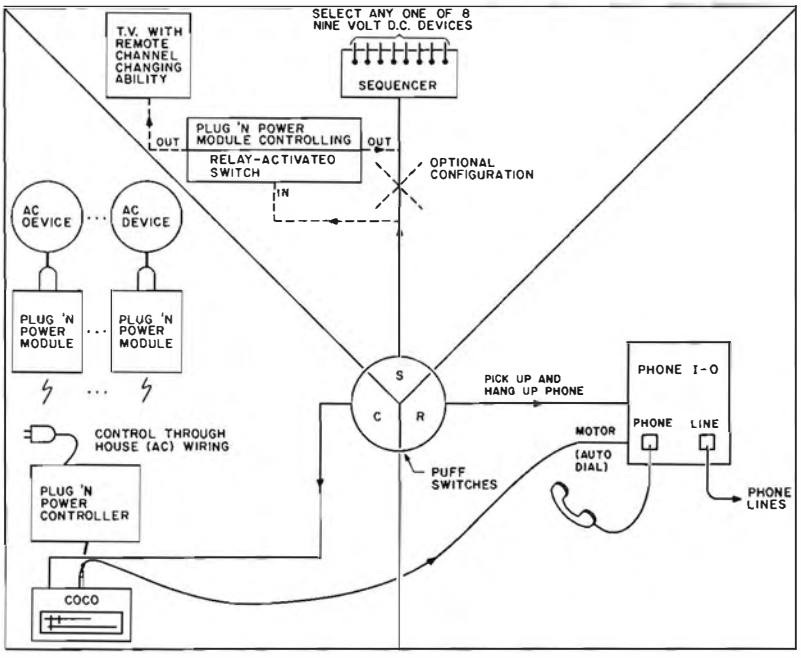


Fig. 1. The John-B System

System Requirements

16K RAM
Extended Color Basic

writer function. I wondered why Joe would bother; with medical science leaping forward every day there couldn't be that many potential users for such a system, could there? To my surprise, I learned that thousands of young girls and boys are injured each year in auto accidents, diving and swimming mishaps, and other tragedies that result in upper-vertebra injuries and paralysis. Unfortunately, they are not alone in their disabilities. Many veterans and elderly people are also disabled.

The major burden in keeping a severely handicapped person at home is the constant attention he requires. The proper application of Joe's John-B System can provide family members relief in caring for the handicapped person. But most of all, it can allow that person to take control of a great portion of his life.

Joe hoped the details of his John-B System could be made available to the CoCo public. That is the purpose of this series. You can build a John-B system for a relative or friend (or for sale), or you can use the system in other ways. Although intended for the handicapped, it can (with some simple programming changes) provide automated environmental control or home security. In fact, you can easily adapt anything controlled by a switch or relay for control by the system.

Concept

The system has three hardware-related and four software functions, three of which are computer controlled (Fig. 1). You can configure a system using any combination of functions, since they can all work independently. The first function is remote control. It allows the CoCo to control Plug 'n Power remote-control modules through the house's ac power lines. You can connect up to 16 separate ac devices anywhere in your house to their own Plug 'n Power remote-control module and turn them on or off. This includes appliances, lights, and even door locks (using electric door latches).

The second function is automatic phone dialing. The CoCo will "pick up" the handset (place it off hook), allow you to select a phone number from a stored list, and then dial it for you. When you're done calling, the CoCo will let you "hang up" the phone (place it on hook).

The third function is not controlled by the computer. It is a manual sequencing control that allows you to select one of eight devices (or none of them) by closing a single switch. In the John-B system, the user blows at a "puff flag". The flag, like a sail, catches the puff and closes a switch, selecting the next device in sequence. A typical example of its use is selecting one of a number of transistor radios (each tuned to a different station) to listen to. Each sequencer output can control up to 9 volts at 500 milliamps. If ac or higher dc current levels are required to be controlled, you can use the sequencer to control an appropriate sized relay. As an alternative, a Plug 'n Power module can control a relay-

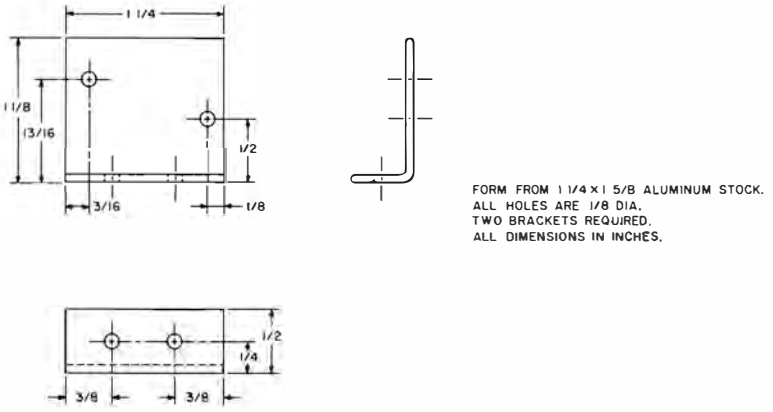


Fig. 2a. Switch Bracket

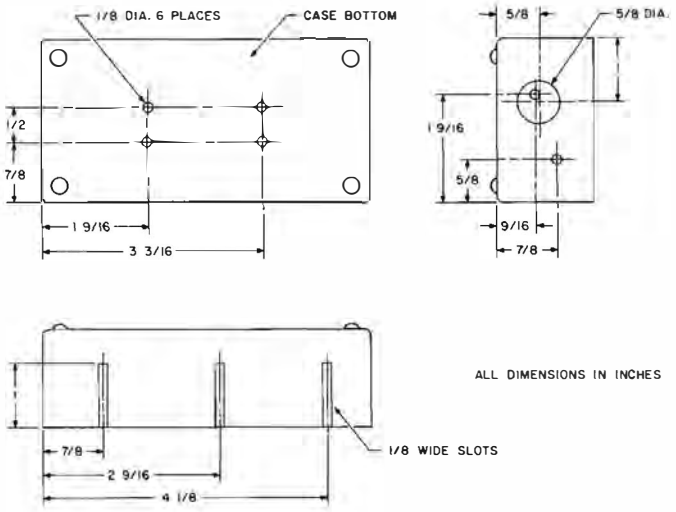


Fig. 2b. Case Machining

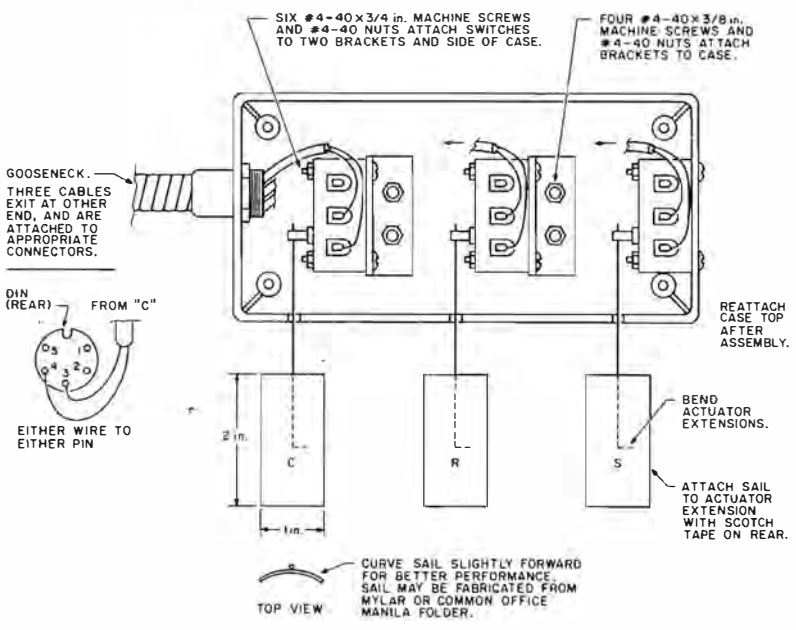


Fig. 2c. Puff-Switch Assembly Pictorial

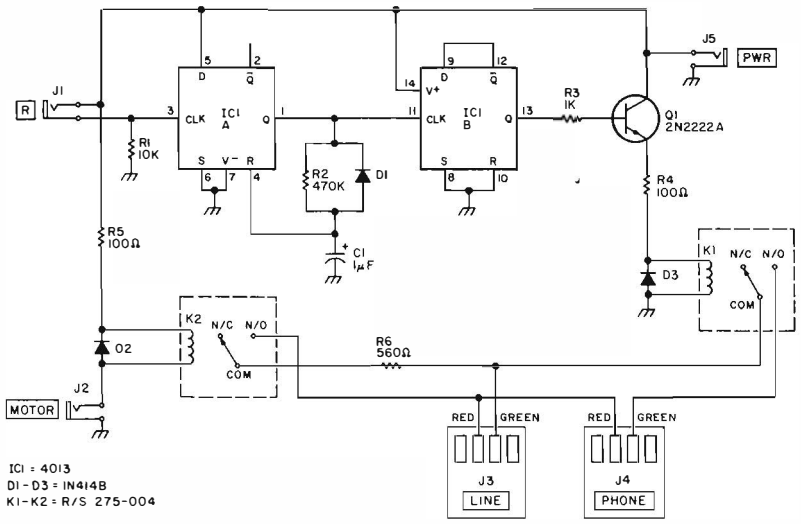


Fig. 3. John-B Phone I/O

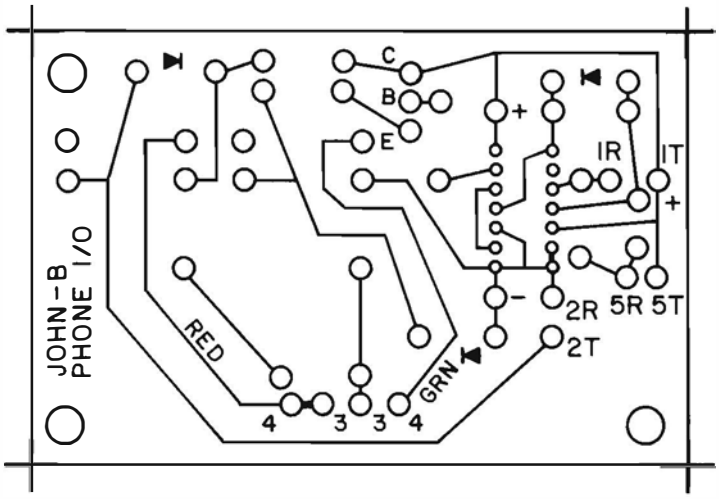


Fig. 4. Phone I/O PC Layout

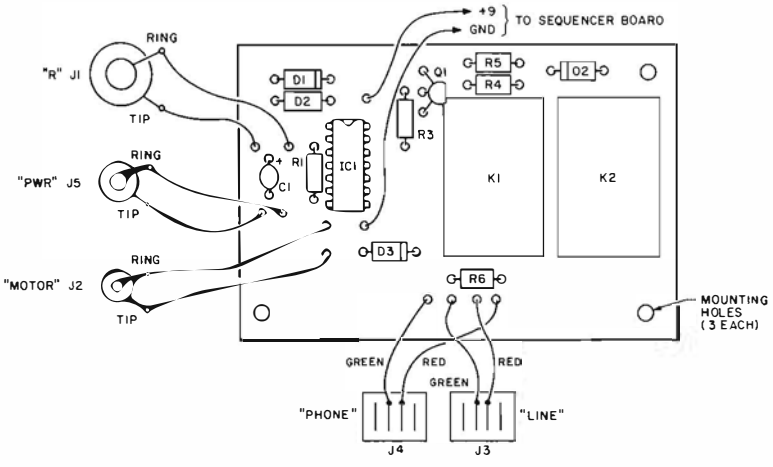


Fig. 5. Phone I/O Component Placement Final Wiring

activated switch to alternate between the sequencer and an entertainment TV (as opposed to the CoCo TV "monitor"). In this way, the user can change channels with the same (S) puff switch used for the sequencer.

The fourth function (specifically designed for handicapped application) is the Typewriter. Using a single puff switch, the user can compose text and transmit it to a printer.

Approach

I'll complete this project in three parts. This first part covers the puff switches and the automatic phone dialer. Next month, the second part describes the construction of the sequence and power controllers. In September, I'll wrap up the series with the software and installation information.

Puff-Switch Assembly

The puff-switch assembly is required only for the John-B System. (For other applications, the software will request keyboard input.) The John-B System presumes the user has minimal (if any) mobility, so computer input is obtained with a switch that can be activated by blowing (or puffing) at it. The switch (labeled C for computer) is connected to the right joystick port, as would be the fire button on a standard joystick.

The assembly has two other puff switches. The second (labeled R) is used to pick up and hang up the telephone receiver. The third (labeled S) is used to activate the sequence controller. The puff-switch assembly is positioned in front of the user so minimum movement is required to operate it, and there is no obstruction of the user's view of the monitor. This is accomplished by mounting the box containing the three switches on a flexible (gooseneck) microphone stand. The stand is then mounted on the chair or bed that the user occupies and adjusted as necessary.

Figure 2 shows construction details of the puff-switch assembly. Special low-torque switches are necessary for satisfactory operation. (See Table 1 for part number and supply source.) The switch box is attached to the microphone stand with an aluminum or wood bracket. You can also attach the stand directly to the box. Another bracket attaches to the base of the microphone stand to secure it to a chair or bedrail. As a substitute, secure a C clamp to the end of the microphone stand.

Automatic Phone Dialer

The phone dialer has two functions: It performs the electrical equivalent of picking up and replacing the receiver. It also, of course, performs the actual dialing.

Most phone service is provided on two wires (red and green). Picking up the receiver connects these lines to the phone. The phone resistance (about 600 ohms) lowers the voltage coming in on these lines from about 50 volts to about 5 volts. This tells the central office that you want a dial tone. You then begin dialing, which is no more than opening and closing the green line.

Our phone dialer performs these functions with two relays (Fig. 3). Relay K2 is driven by the cassette motor relay in the CoCo. Both K1 and K2 draw very little current (12 milliamps is typical), so there is no appreciable load on the CoCo's relay. The software energizes relay K2 when you are ready to dial. This closes its contacts, placing a 560-ohm resistor across the lines. The software now waits a specified time while the central office recognizes the resistance and provides a dial tone. Then the CoCo begins pulsing K2 to perform the actual dialing. After the software has completed the dialing, it informs the user to puff on the R switch.

The R switch provides +9 volts to the input (clock) of IC1A. As wired, IC1A debounces (eliminates multiple closures of) the R switch, producing a single positive pulse of about ½ second. The output from IC1A triggers IC1B, which is wired as a flip-flop. When IC1B is triggered, it reverses state (high to low or low to high). The output of IC1B controls transistor Q1, which provides the necessary current to operate relay K1. Thus, the momentary closure of the R switch is transformed into a latched change in the relay's status (on or off).

Presuming relay K1 was previously off, puffing on the R switch energizes K1 and connects the receiver (so you can talk and listen). At the same time, the software is holding K2 to maintain the phone connection until the receiver is lifted. After a predetermined time, the software deenergizes K2 to remove the 560-ohm resistor (which would otherwise load the line and lower the volume).

With this setup, you can permanently position the receiver next to the John-B user who is unable to manipulate it. It also lets the user answer an incoming phone call by simply puffing at the R switch. The approach presumes that there is another phone on the same line close enough so the user can hear it ring, since the "on hook" mode also disconnects the ringer. If there isn't, add one of the new $10 phones to the line before relay K2. This approach is cheaper than buying a separate ringer, which can cost as much as $20!

For the non-John-B user, simplify the system eliminating relay K1 and picking up the receiver when instructed by the software.

Construction

Fabricate a PC board from the pattern shown in Fig. 4. Mount all components on the board, being sure to observe the orientation shown in Fig. 5 for C1, IC1, and Q1. Since IC1 is static sensitive, handle it as little as possible (and preferrably at its ends). Before touching IC1, remove any static charge you might have by touching a ground point (such as the screw holding an electric outlet cover).

Perform final wiring per Fig. 5. Mount the completed unit in any suitable case, making sure J1 through J5 are readily available. To use the Phone I/O, connect the phone line to J4 (line) and a phone to J3 (phone). Connect the small grey cassette-deck plug from your CoCo to J2 (motor). Attach a 9-volt dc power cube to J5 (pwr). Finally, connect the R puff switch (or any SPST switch) to J1 (R) and the C puff switch to the CoCo's right joystick port. Load the Program Listing and run it. When the desired phone number appears, momentarily close the C switch. You will now see the number you selected appear as it is being dialed. When dialing is completed, you will be asked to pick up the receiver. Momentarily cose the R switch and note that the receiver is now connected. After the phone call has been completed, momentarily close the R switch once again to hang up the phone. ■

Program listing on page 48

Jim Barbarello has had numerous hardware projects published. Write him at R.D. #1, Box 241H, Tennent Road, Englishtown, NJ 07726.

Puff-Switch Assembly

S1–S3	Low-torque cherry microswitch #E51, available from Herback and Rademan (their P/N TM22K376 at $1.25 each)
P1	6- or 5-pin (240-degree model) DIN plug
P2, P3	¼" phone plug (such as RS P/N 274-1536)

Miscellaneous: Case (RS P/N 270-222 or similar), 13-inch Microphone Gooseneck (RS P/N 33-330 or similar), six-conductor (or three pairs of two-conductor) cable, mounting hardware (see note 6 below), mylar sheet stock or stiff paper (see Fig. 2 for details)

Automatic Phone Dialer

C1	1.0 uf,10 v or greater capacitor (such as RS P/N 272-1434)
D1, D2, D3	1N4148 silicon signal diode (such as RS P/N 276-1620)
IC1	4013 CMOS dual "D" flip-flop IC (such as RS P/N 276-2413)
J1	¼-inch phone jack (RS P/N 274-252 or similar)
J2	⅛-inch phone jack (RS P/N 274-251 or similar)
J3, J4	Duplex modular telephone-line-cord wall jack or two single modular telephone-line-cord jacks (duplex jack is Gemini P/N TA74 or similar, available at Rickel's Home Center or other stores carrying telephone accessories).
J5	3/32-inch phone jack (RS P/N 274-292 or similar)
K1, K2	SPDT sensitive mini relay (RS P/N 275-004)
Q1	2222A silicon NPN transistor (such as RS P/N 276-2009)
R1	10-Kohm, ¼-watt resistor (such as RS P/N 271-1335)
R2	470-Kohm, ¼-watt resistor (such as RS P/N 271-1354)
R3	1-Kohm, ¼-watt resistor (such as RS P/N 271-1321)
R4, R5	100-ohm, ¼-watt resistor (such as RS P/N 271-1311)
R6	560-ohm, ¼-watt fixed resistor (RS P/N 271-020 (½-watt version can be used as an alternative).

Table 1. Parts List

Notes

- RS = Radio Shack
- Herback and Rademan, 401 E. Erie Ave., Philedelphia, PA 19134, 215-426-1708. $10 minimum order. Visa and M/C accepted.
- Jameco Electronics, 1355 Shoreway Road, Belmont, CA 94002, 415-592-8097. $10 minimum order. Visa and M/C accepted.
- All phone I/O and sequencer parts (except jacks and relays) are also available from Jameco Electronics.
- Both the Phone I/O and sequencer can be housed in a single case or powered from the same 9-volt power cube. Additional power (+9 and Gnd) takeoff points are provided on the Phone I/O board, which can be connected to the power input points on the Sequencer PC Board to provide the required 9 volts. Then only one power cube is required.
- In the absence of standard aluminum stock, form the Puff-Switch assembly (and other mounting) brackets from a suitable size aluminum chassis box (such as RS P/N 270-239).

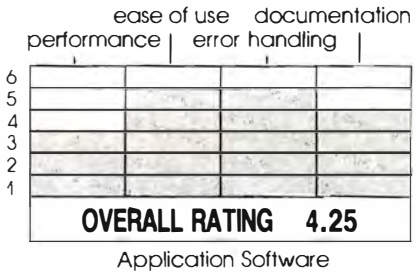
Reviews

edited by J. Scot Finnie

Contents

CoCo Cookbook: A Culinary Adventure

by Scott and Sharon Norman



Application Software

Norman Manchevshy's CoCo Cookbook from Computerware is an attempt to apply computer technology to an aspect of home life that can be either an enjoyable or tedious undertaking—cooking. The program is actually a data-file manager of the free-form, "electronic Rolodex" variety, specialized for the storage of recipes. It comes with a 50-recipe sample file (of delicious-looking selections) that provides useful practice material for learning the program. CoCo Cookbook is easy to use. But you cannot merely shove information into it and expect the computer to spit out new insights.

CoCo Cookbook requires a 32K computer and one disk drive. It cannot capitalize on the additional memory of 64K machines or the additional storage space of more than one disk drive. The program stores recipes on its own disk only. However, it can manage up to 270 recipes, each of which can be up to 3,040 characters long. That's well in excess of the 156K storage capacity of a CoCo disk, not to mention that the CoCo Cookbook and its auxiliary-indexing routines must fit on there, too. The trick is accomplished by some clever data-compression techniques. The program compresses 126 words commonly used in recipes to 1- or 2-byte codes, much as "reserved words" are encoded in a Basic program.

CoCo Cookbook is written in Color Basic. You can easily customize it for your printer baud rate and theoretically change the length of a printed line and number of lines per page. To change the baud-rate, POKE the key number for your printer hookup into RAM address 159 (program line 70 is the designated spot). The other two modifications do not always work well. We recommend leaving the original code alone. Recipes print across the full width of the page, with adequate top and bottom margins.

Major Functions

The program's major functions include adding new recipes to the disk, changing existing recipes or their titles (similar, but not identical, to editing a Basic program) deleting recipes, finding specified recipes, and listing the titles of all recipes in the file. You can arrange the list alphabetically by title or numerically according to the order in which you entered the recipes.

Titles and recipe numbers are important in CoCo Cookbook. It is important to keep an up-to-date printout of your list handy. You must specify a recipe's number when deleting or changing it. However, the program's find-recipe option lets you specify a recipe's number or a "keyword" in its title. CoCo Cookbook's search routine examines a recipe's title—not its list of ingredients or instructions. If you want the program to find everything in your file that has to do with chocolate, make sure that word is included in all appropriate titles.

Because titles play a crucial role in the operation of CoCo Cookbook, users should consider expanding their concept of titles to ease the job of searching through recipes. One helpful method of organization, taken from an example in the 50-recipe initial file, is to list major ingredients along with the name of the dish: Fettucine al Burro (pasta, cream, & cheese). This method would make Fettucine al Burro pop up if you were searching with a keyword, for example, for cheese dishes.

You can use the same method to highlight recipes for special dietary requirements. You might want to set up your own codes for Kosher, low-sodium, or vegetarian diets, for example. An alternative is to make up a separate disk for every kind of diet. It is easy to make as many additional copies of the program disk as you want.

To avoid a confusion over dietary concerns, you might use a code in the title set off by a special character, such as an asterisk: Filet of Sole *LC *K. This identifies Filet of Sole as an acceptable low-calorie and Kosher meal. However you set up your recipe titles, you must be concise because titles can be only 44 characters long.

CoCo Cookbook cannot perform logical AND or OR operations when searching through recipe titles. There is no way to specify recipes that meet arbitrary combinations of criteria. However, the program does have a bypass option that lets you scan everything it finds in response to a search request so that you can discard any recipe that doesn't suit your fancy. It also has an auto option that spews out every record containing a specified keyword in its title.

Summary

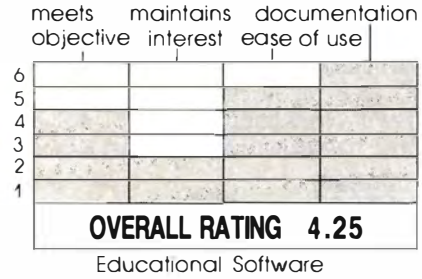
CoCo Cookbook is no speed demon. It takes about 20 seconds to locate the last entry in a 50-recipe file by keyword (but only two seconds to find it by record number). In addition to speed, other features have been omitted in the interest of fitting everything into 32K. There is no provision for scaling the number of meal portions up or down and no method of menu-planning assistance. We would like to see these features added to an upgraded 64K version of CoCo Cookbook.

Does CoCo Cookbook perform any useful functions? We think so. Here are a couple of helpful ideas for CoCo Cookbook owners with printers. You might print out a fresh copy of a favorite recipe every time you want to prepare it. Why risk spoiling a bound cookbook in the kitchen? If you are involved with a cooking class, you might use CoCo Cookbook to provide class members with their own copies of recipes—eliminating the need for frantic scribbling during a demonstration. You are apt to come up with more ideas. And don't overlook the recipes that come with CoCo Cookbook. If you are a culinary adventurer, this program could be for you. ■

CoCo Cookbook is manufactured by Computerware, 4403 Manchester Ave., Box 668, Encinitas, CA 92024, 619-436-3512. It requires 32K and one disk drive. It sells for $27.95.

Recounting Memory Castle

by Richard Ramella



Educational Software

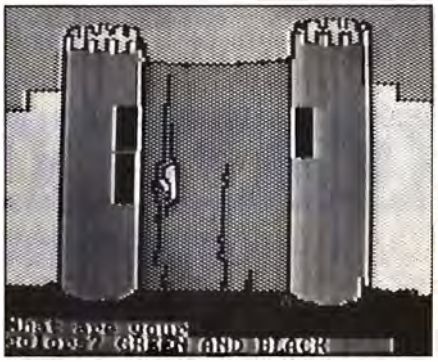
Memory Castle is a learning game from Sunburst Communications. The object of the game is to make a trip through a castle and its environs while remembering as many of the places to go and things to do as possible. The easiest level has six things to remember; the medium level has 12; the most difficult level has 20.

As the game begins, a knight appears on horseback in two colors that vary with every program run. Pressing the enter key brings up the level-section screen. Then players are shown the gate of the castle where they must type the colors of the knight. A wrong answer elicits the response: "Intruder! Ye claim false colors!" The game ends with an invitation to try again.

Once the knight's colors are properly identified, the game begins in earnest. The easy level might instruct a player to go to the fireplace and watch the fire burn, to the music room and play the flute, and to the shield room and get a green shield. These instructions stay on the screen for as long as the player wishes. The rest of the game is a matter of correctly answering a series of questions about destination and action. Memory Castle has 10 castle locations drawn in full-screen graphics, three reward scenes, the knight, and the castle gate.

I tested the game on three children, ages 9 to 12. None had any interest beyond three runs of the program. Their opinions boiled down to this: A finite number of events can occur in the game. Remembering them is little more than a mechanical act. Once a memory trick is learned, it is not much fun to repeat the game's variations. I asked the 9-year-old what his trick was. "Just remember the main words," he adivsed: "fire, burn, music, flute, shield, and green."

This child formed a strategy for solution that is the actual aim of this program—memorization. Much education is only memorizing facts, tables, abbreviations, and numbers. Some are useful, and some just give otherwise loafing brain cells something to do until they are called into play by *Trivial Pursuit* later in life.


The Gate To Memory Castle

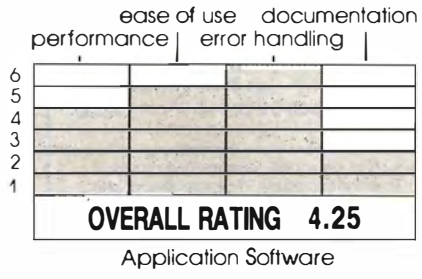
The Memory Castle package contains 69 pages of material in a loose-leaf binder. Included are excellent instructions for play, a section on working with the Color Computer, and an article on problem solving. The Memory Castle authors are Dr. Carol Chomsky and Dr. Judah L. Schwartz, part of a 29-member team working on Sunburst's series of educational games designed to provide students with methods for problem solving. They have adopted the ideas of a program called Computer Assisted Problem Solving Skills Practice, which presents a "problem solving matrix" that comprises memory, cognitive skills and control, and creativity. The teaching material that comes with Memory Castle suggests using mnemonics to help children learn facts.

Memory Castle is well packaged, well documented, and offers nice graphics. As an educational program, it also achieves what it sets out to—make children use their memories. Its only drawback is that it does not maintain interest well. It would probably make a good supplementary activity to a memory exercise in a classroom setting.■

Memory Castle is produced by Sunburst Communications, 39 Washington Ave., Pleasantville, NY 10570, 800-431-1934. It requires 16K, Extended Color Basic, and a disk drive. It sells for $55.

Over Easy-Edit

by Steve Brown



Application Software

The longer I bang away at my keyboard, the more tiresome programming seems. Let's face it, typing and retyping GOSUBs with the same or similar lines is tedious work. But Mark Data Products has introduced a tool to help alleviate the ennui associated with typing Basic or Assembly-language code. Easy-Edit is a text editor that offers a great deal of flexibility for handling text on a line-by-line basis.

Easy-Edit is not merely a line editor. It provides automatic key repeat, its own mini-DOS (disk-operating system), automatic adaptation to either 32 or 64K, and Mark Data Products' Superscreen, a 51-character by 24-line text-screen generator that reroutes all normal text input to a graphics page and includes an ON ERROR GOTO error-trapping routine. (For more information on Superscreen, see the review on p. 40 of the January, 1984, issue of *HOT CoCo*.) Easy-Edit is also compatible with popular assembler programs.

Performance

When you load Easy-Edit, its machine-language driver program tests the memory capacity of your computer and automatically adjusts to your configuration. Easy-Edit requires a 32 or 64K machine. There is quite a difference in the line capacities of computers with these two amounts of memory. A 32K machine holds about 1,000 lines of text and a 64K CoCo holds about 3,000 lines. I haven't

gotten near capacity, but if your text files become too large for available memory, Easy-Edit has a provision for dumping what is in memory to disk.

At the same time that Easy-Edit tests the memory capacity of your CoCo, Superscreen takes control of the text screen. Easy-Edit's functions are controlled by approximately 30 one- or two-keystroke commands. One of these commands changes the screen colors. The program defaults to a heavily artifacted black-on-buff screen that is hard to read. I found that black letters on a green background were easy to read.

Because Easy-Edit is a line-oriented editor, it numbers each line of text you type. In order to perform a function on a line, you must reference its line number in your command. The line numbers are for reference and don't function like line numbers in a Basic program. If you type a Basic program with Easy-Edit, you must also type Basic line numbers.

Easy-Edit has many time-saving features. Among other functions, you can list to the screen or printer what you have typed, renumber it, move and insert lines, find and change strings of text inside lines, copy lines to other parts of your program, and copy parts of lines. The last command is very helpful. How many times have you retyped part of a line, such as =INT((XM)*(VAL(X$)): GOSUB 2000, on several lines of a Basic program and then had to proofread each entry? Easy-Edit lets you copy just a portion of a repeated line.

Easy-Edit also has its own mini-DOS with single-keystroke commands for handling disk files. This means that you don't have to leave the editor and return to Basic in order to load, save, kill, or display a directory of files.

Easy-Edit can only manipulate files saved in ASCII format: Basic programs saved with the A command or Assembly-language programs saved by Radio Shack's EDTASM + editor/assembler as .TXT files. Easy-Edit can read, save, kill, and merge files. The merge capability is particularly useful when you are building a large program from a number of subroutines. I tend to write subroutines for my programs in small segments and save them as single files. Easy-Edit makes it easy to tack these bits together to make one large program.

Easy-Edit is easy to use. It has logical commands and flexibility. You can set printer parameters, such as baud rate, from the command mode. The editor lets you append text to a line, change all or some occurrences of a string of text, and overlay new text on existing lines. This program does what it sets out to do—make editing your programs easy.

Documentation

Easy-Edit's documentation is less complete than what I am accustomed to from Mark Data Products. A demonstration file and a set of step-by-step instructions showing how each command works should come with this program. I could not figure out how to get the columns of an Assembly-language program to line up so that they were acceptable to Radio Shack's EDTASM +. Despite these problems, the program's text-manipulation commands are logical and easy to execute.

Summary

Easy-Edit is a capable text-file editor. It can handle large text files easily. Its command structure offers the programmer flexibility. Its screen environment and efficient time-saving features, such as short command executions, automatic key repeat, and mini-DOS make text editing easier. Although its manual could cover more material, it is written in a clear style. Overall, Easy-Edit will make a useful addition to your programmer's toolkit of utility programs. ■

Easy-Edit is manufactured by Mark Data Products, 24001 Alicia Parkway, No. 207, Mission Viejo, CA 92691, 714-768-1551. It requires 32K and a disk drive. It sells for $34.95.

Reviews continue on page 54

HOT CoCo's Pull-Out Program Listings

All program listings are available on our Instant CoCo cassette.

Mindbusters

See page 16

Program Listing 1. Cryptonice

```
10Ø REM * CRYPTONICE * TRS-8Ø EX
TENDED COLOR BASIC 16K
11Ø DATA LIFE WOULD BE TOLERABLE
 WERE IT NOT FOR ITS AMUSEMENTS
12Ø DATA MANY A GENIUS HAS BEEN
SLOW IN GROWTH. OAKS THAT FLOURI
SH FOR A THOUSAND YEARS DO NOT S
PRING UP INTO BEAUTY LIKE A REED
.
13Ø DATA HABITS ARE FIRST COBWEB
S THEN CABLES
14Ø DATA BE WISELY WORLDLY NOT W
ORLDLY WISE
15Ø DATA ACCURST BE HE THAT FIRS
T INVENTED WAR
16Ø DATA NO PEOPLE DO SO MUCH HA
RM AS THOSE THAT GO ABOUT DOING
GOOD
17Ø DATA A PRECEDENT EMBALMS A P
RINCIPLE
18Ø DATA EVERY ONE IS MORE OR LE
SS MAD ON ONE POINT
19Ø DATA GLORY'S NO COMPENSATION
 FOR A BELLY-ACHE
2ØØ DATA PENSIVE POETS PAINFUL V
IGILS KEEP - SLEEPLESS THEMSELVE
S TO GIVE THEIR READERS SLEEP
21Ø CLS: PRINT "CRYPTONICE SETTI
NG UP. PATIENCE"
22Ø T=TIMER: CLEAR 2ØØØ: DIM W$(
1Ø): Q$=CHR$(32): PL$="T64;O4;CA
BBAGE"
23Ø IF T>1ØØØ THEN T=T-1ØØØ: GOT
O 23Ø ELSE FOR X=1 TO T: D=RND(Ø
): NEXT
24Ø FOR X=33 TO 9Ø: F$=F$+CHR$(X
): NEXT: F$=F$+CHR$(9): DIM J$(1
Ø)
25Ø FOR X=65 TO 9Ø: V$=V$+CHR$(X
):NEXT: A$=V$: FOR X=1 TO 26
26Ø B=RND(26): IF MID$(V$,B,1)=Q
$ THEN 26Ø ELSE B$=B$+CHR$(64+B)
: MID$(V$,B,1)=Q$: NEXT
27Ø FOR X=1 TO 1Ø: READ J$(X): N
EXT: K$=J$(RND(1Ø)): FOR X=1 TO
LEN(K$)
28Ø R$=MID$(K$,X,1): J=ASC(R$):
IF J<65 OR J>9Ø THEN SAY$=SAY$+M
ID$(K$,X,1): GOTO 3ØØ
29Ø SAY$=SAY$+CHR$(64+INSTR(B$,R
$))
3ØØ NEXT X: CLS: PUZ$=STRING$(LE
N(K$),32)
31Ø Z=32: FOR X=1 TO LEN(SAY$)
32Ø H$=CHR$(ASC(MID$(SAY$,X,1))+
32)
33Ø IF INSTR(A$,CHR$(ASC(H$)-32)
)=Ø THEN H$=CHR$(ASC(H$)-32)
34Ø PRINT @ L+Z,H$;: L=L+1: IF L
=32 THEN Z=Z+64: L=Ø
35Ø NEXT: PK=1Ø24: L=Ø
36Ø Z$=INKEY$: ST=PEEK(PK+L): PO
KE PK+L,255: FOR T=1 TO 1Ø: NEXT
 T
37Ø POKE PK+L,ST: IF INSTR(F$,Z$
)<2 THEN 36Ø
38Ø IF Z$=CHR$(9) THEN L=L+1: M=
M+1: IF L=32 THEN L=L+32
39Ø IF M=LEN(SAY$) THEN L=Ø: M=Ø
4ØØ IF Z$=CHR$(9) THEN 36Ø
41Ø POKE PK+L,ASC(Z$): GOSUB 42Ø
```

```
: GOTO 36Ø
42Ø MID$(PUZ$,M+1,1)=Z$: P1=PEEK
(PK+L+32): AS=ASC(Z$)
43Ø FOR J=1 TO LEN(K$): IF CHR$(
P1)=CHR$(PEEK(PK+LL+32)) THEN PO
KE PK+LL,AS: MID$(PUZ$,J,1)=Z$
44Ø LL=LL+1: IF LL=32 THEN LL=LL
+32
45Ø NEXT: LL=Ø: IF PUZ$<>K$ THEN
 RETURN
46Ø FOR P=1 TO 2: PRINT @ 48Ø+P,
Q$+"W I N N E R"+Q$;: PLAY PL$:
NEXT P: GOTO 46Ø
47Ø END
```

Program Listing 2. Alphasequence

```
1ØØ REM * ALPHASEQUENCE *
11Ø REM * A MINDBUSTERS CONTEST
- WORKS ON ANY COCO *
12Ø CLEAR 75Ø: DIM W$(25): FOR X
=1 TO 15
13Ø CLS: FOR L=65 TO 9Ø: PRINT C
HR$(L);: NEXT L: PRINT: PRINT
14Ø PRINT "WORD"X: INPUT Z$: PRI
NT
15Ø FOR A=2 TO LEN(Z$)
16Ø IF MID$(Z$,A-1,1)=>MID$(Z$,A
,1) THEN PRINT "ILLEGAL": SOUND
RND(255),1Ø: GOTO 13Ø
17Ø NEXT A: PRINT Z$" IS A GOOD
WORD."
18Ø W$=W$+CHR$(32)+Z$: Z=LEN(Z$)
: SC=SC+(Z*Z): PRINT "SCORE";SC
19Ø PRINT: INPUT "PRESS A KEY TO
 CONTINUE";V: NEXT X
2ØØ PRINT "RUN COMPLETE. PRESS E
NTER"
21Ø INPUT "TO SEE WORDS AND SCOR
E";V
22Ø CLS: FOR X=1 TO LEN(W$): J$=
MID$(W$,X,1): P=P+1: PRINT J$;
23Ø IF P>2Ø AND J$=CHR$(32) THEN
 PRINT: P=Ø
24Ø NEXT X: PRINT: PRINT "SCORE:
";SC: END
```

All Sorts of Sorts

See page 33

Program Listing 1. The shell of a sorting program— add the sorting algorithm.

```
5 REM LISTING #1
1Ø REM SHELL FOR TESTING SORTING
 PROGRAMS
2Ø INPUT"NUMBER OF ITEMS TO SORT
";N: DIM A(N)
3Ø FOR I=1 TO N: A(I)=RND(Ø): NE
XT: REM FILL THE ARRAY A(.) WITH
 RANDOM DECIMALS
4Ø GOSUB 1ØØ: REM SORT THE ARRAY
 A(.)
5Ø FOR I=1 TO N-1: IF A(I)>A(I+1
) THEN PRINT"SORT FAILS": STOP E
LSE NEXT I: PRINT"SORT CHECKED":
 STOP
1ØØ REM THE SORTING SUBROUTINE S
TARTS HERE.
```

Program Listing 2. Selection Sort

```
5 REM LISTING #2
1ØØ REM SELECTION SORT OF A(1..N
) INTO ASCENDING ORDER
11Ø FOR I=1 TO N-1: REM EXCHANGE
 A(I) AND THE SMALLEST OF A(I..N
)
12Ø M=I: REM M WILL BE THE INDEX
 OF THE SMALLEST ELEMENT IN A(I.
.N)
13Ø FOR J=I+1 TO N: REM SEARCH F
OR THE SMALLEST ELEMENT
14Ø IF A(J) < A(M) THEN M=J: REM
 MAINTAIN M=INDEX OF SMALLEST EL
EMENT FOUND SO FAR
15Ø NEXT J: REM NOW M HAS THE CO
RRECT VALUE
16Ø T=A(I): A(I)=A(M): A(M)=T: R
EM SWAP A(I) AND A(M).
17Ø REM AT THIS POINT A(I) <= A(
I+1..N)
18Ø NEXT
19Ø RETURN
```

Program Listing 3. Selection Sort Compacted

```
5 REM LISTING #3
1ØØ REM COMPACT VERSION OF THE S
ELECTION SORT
11Ø FORI=1TON-1:M=I:FORL=I+1TON:
IFA(J)<A(M)THENM=J
12Ø NEXT: T=A(M): A(M)=A(I): A(I
)=T: NEXT
13Ø RETURN
```

Program Listing 4. Bubble Sort

```
5 REM LISTING #4
1ØØ REM BUBBLE SORT A(1..N) INTO
 ASCENDING ORDER
11Ø FOR I=2 TO N
12Ø REM BUBBLE A(I) UP INTO ALRE
ADY SORTED SEGMENT A(1..I-1)--FO
R J=I-1, I-2, ... MOVE A(J) TO A
(J+1) UNTIL A(J)>=T (THE ORIGINA
L VALUE OF A(I)) THEN MOVE T INT
O THE LAST VACATED SPACE A(J+1)
13Ø J=I-1: T=A(I)
14Ø IF J>Ø THEN IF A(J)>T THEN A
(J+1)=A(J): J=J-1: GOTO 14Ø
15Ø A(J+1)=T
16Ø NEXT I
17Ø RETURN
```

Program Listing 5. Shell Sort

```
5 LISTING #5
1ØØ REM SHELL SORT A(1..N) IN AS
CENDING ORDER
11Ø M = INT(N/2)
12Ø REM WHILE M>Ø, BUBBLE SORT T
HE SUBARRAYS A(K), A(K+M), A(K+2
M)..., K=1..M
13Ø FOR K=1 TO M
14Ø REM BUBBLE SORT A(K), A(+M),
 A(K+2M),...
15Ø FOR I=K+M TO N STEP M
16Ø REM BUBBLE A(I) UP INTO PRIO
R SEGMENT
17Ø J=I-M: T=A(I)
18Ø IF J>Ø THEN IFA(J)>T THEN A(
J+M)=A(J): J=J-M: GOTO 18Ø
19Ø A(J+M)=T
2ØØ NEXT I
21Ø NEXT K
22Ø M=INT(M/2)> IF M>Ø THEN 13Ø
23Ø RETURN
```

Program Listing 6. Quick Sort

```
5 REM LISTING #6
1ØØ REM QUICK SORT OF A(1..N) IN
TO ASCENDING ORDER. VARIABLES US
ED: R,L,D,E,M,SP.  ALSO USES AN
ARRAY ST(*) WHICH MUST BE DIMENS
IONED AT THE BEGINNING OF THE CA
LLING PROGRAM WITH LENGTH AT LEA
ST 2*SQR(N)
11Ø REM INITIALIZE EMPTY STACK.
 FIRST SEGMENT TO PARTITION IS E
NTIRE ARRAY
12Ø SP=-1: L=1: R=N
13Ø REM CURRENT SEGMENT TO SORT
IS A(L..R).  IF IT IS LENGTH 1,
THEN POP LIMITS FOR NEXT SEGMENT
 OFF STACK. IF STACK EMPTY, THEN
 SORT COMPLETED
14Ø IF R=L THEN 26Ø: REM IF SEGM
ENT LENGTH 1 GET NEXT SEGMENT OF
F STACK
15Ø REM FIND A VALUE V: MIN(A(L.
.R)) < V <= MAX(A(L..R))
16Ø I=L
17Ø IF I=R THEN GOTO 26Ø ELSE V=
A(I): REM IF I=R THEN ALL A(L..R
) THE SAME.  GET NEXT SEGMENT TO
 SORT OFF STACK.
18Ø IF V=A(I+1) THEN I=I+1: GOTO
 17Ø ELSE IF V<A(I+1) THEN V=A(I
+1)
19Ø REM PARTITION A(L..R) INTO A
(L..E)<V; A(D..R)>=V.  NOTE E=D-
1 WHEN LINE 24Ø ENTERED
2ØØ D=L: E=R
21Ø IF A(D)<V THEN D=D+1:GOTO 21
Ø
22Ø IF A(E)>=V THEN E=E-1: GOTO
22Ø ELSE IF D<E THEN X=A(D):A(D)
=A(E):A(E)=X:GOTO 21Ø
23Ø REM A(L..R) IS NOW PARTITION
ED.  PUSH LIMITS OF LONGER PARTI
TION ON STACK AND REPARTITION SH
ORTER PARTITION
24Ø SP=SP+2: IF E-L>R-D THEN ST(
SP-1)=L: ST(SP)=E: L=D ELSE ST(S
P-1)=D: ST(SP)=R: R=E
25Ø GOTO 14Ø
26Ø IF SP>Ø THEN R=ST(SP): L=ST(
SP-1): SP=SP-2: GOTO 14Ø ELSE RE
TURN
```

Program Listing 7. Best Sort

```
5 REM LISTING #7
1ØØ REM BEST SORT OF A(1..N) INT
O ASCENDING ORDER. VARIABLES USE
```

```
D: R,L,D,E,M,SP.  ALSO USES AN A
RRAY ST(*) WHICH MUST BE DIMENSI
ONED AT THE BEGINNING OF THE CAL
LING PROGRAM WITH LENGTH AT LEAS
T 2*SQR(N)
11Ø REM INITIALIZE EMPTY STACK.
 FIRST SEGMENT TO PARTITION IS E
NTIRE ARRAY. SEGMENTS OF LENGTH<
13 ARE NOT SORTED BY PARTITIONIN
G, BUT RATHER AT END THE ENTIRE,
 PARTIALLY SORTED, ARRAY IS COMP
LETELY SORTED BY BUBBLE SORT
12Ø SP=-1: L=1: R=N: M=13
13Ø REM CURRENT SEGMENT TO SORT
IS A(L..R).  IF IT IS LENGTH<14,
 THEN POP LIMITS FOR NEXT SEGMEN
T OFF STACK. IF STACK EMPTY, THE
N A(I) IS DIVIDED INTO UNSORTED
CLUMPS OF ELEMENTS OF INCREASING
 SIZE. FINISH BY BUBBLE SORTING
ENTIRE ARRAY.
14Ø IF R-L<M THEN 26Ø
15Ø REM FIND V: MIN(A(L..R)) < V
 <= MAX(A(L..R))
16Ø I=L
17Ø IF I=R THEN 26Ø ELSE V=A(I):
 REM IF I=R THEN ALL A(L..R) THE
 SAME.  GET NEXT SEGMENT TO SORT
.
18Ø IF V=A(I+) THEN I=I+1: GOTO
17Ø ELSE IF V<A(I+1) THEN V=A(I+
1)
19Ø REM PARTITION A(L..R) INTO A
(L..E)<V; A(D..R)>=V.
2ØØ D=L: E=R
21Ø IF A(D)<V THEN D=D+1:GOTO 21
Ø
22Ø IF A(E)>=V THEN E=E-1: GOTO
22Ø ELSE IF D<E THEN X=A(D): A(D
)=A(E): A(E)=X: GOTO 21Ø
23Ø REM A(L..R) IS NOW PARTITION
ED.  PUSH LIMITS OF LONGER PARTI
TION ON STACK AND REPARTITION SH
ORTER PARTITION
24Ø SP=SP+2: IF E-L>R-D THEN ST(
SP-1)=L: ST(SP)=E: L=D ELSE ST(S
P-1)=D: ST(SP)=R: R=E
25Ø GOTO 14Ø
26Ø IF SP>Ø THEN R=ST(SP): L=ST(
SP-1): SP=SP-2: GOTO 14Ø
27Ø REM BUBBLE SORT A(1..N) TO C
OMPLETE SORTING ARRAY
28Ø A(Ø)=-1E3Ø: REM INSERT GUARD
 VALUE AT BEGINNING OF ARRAY
29Ø FOR D=2 TO N
3ØØ V=A(D): E=D-1
31Ø IF A(E)>V THEN A(E+1)=A(E
E=E-1: GOTO 31Ø: REM GUARD VALUE
 WILL TERMINATE THIS LOOP IF E G
ETS DOWN TO Ø
32Ø A(E+1)=V: NEXT D
33Ø RETURN
```

Program Listing 8. Merge Sort

```
5 REM LISTING #8
9Ø REM MERGE SORT OF FILE F$.  U
SES FOUR SCRATCH FILES FI$(1..4)
.  FILE F$ IS LEFT UNALTERED UNT
IL END, WHEN IT IS RENAMED 'XXXX
/DAT', WHICHEVER OF FI$(.) HOLDS
 SORTED DATA IS RENAMED FI$, AND
 ONLY THEN IS THE ORIGINAL FILE
KILLED.
1ØØ REM BASIC OUTLINE OF MERGE S
ORT:  WRITE OUT ORIGINAL DATA IN
 SORTED SEQUENCES OF LENGTH NN T
O FILES FI$(1) AND FI$(2).  THEN
 MERGE SEQUENCES TO FILES 3,4; M
ERGE BACK TO FILES 1,2; ... UNTI
L DATA MERGED INTO SINGLE SEQUEN
CE.
11Ø REM user must specify origin
al data file F$ (INCLUDING EXTEN
SION) and the size NN of the ori
ginal sequences.  make NN as lar
ge as possible to speed up the s
ort, subject to limitation that
sequences of size NN are sorted
in memory.
12Ø REM user must also provide a
n array A(NN) and a stack ST(2*S
QR(NN)) for sorting initial sequ
ences.  user must also set aside
 at least 4 files with command F
ILES 4.
13Ø FOR I=1 TO 4: READ FI$(I): N
EXT I: OU=1:DATA FF1/DAT,FF2/DAT
,FF3/DAT,FF4/DAT: REM ESTABLISH
NAMES OF SCRATCH FILES
14Ø OPEN"I",#3,F$:FORI=1 TO 2:OP
EN"O",#I,FI$(I): NEXT I
15Ø N=1
16Ø INPUT#3,A(N):IF N<NN AND NOT
EOF(3) THEN N=N+1:GOTO 16Ø ELSE
GOSUB 42Ø: REM INPUT A SEQUENCE
AND SORT IT
17Ø FOR I=1 TO N: WRITE#OU,A(I):
 NEXT I
18Ø IF NOT EOF(3) THEN OU = 3-OU
: GOTO 15Ø: REM OU FLOPS BETWEEN
 1 AND 2.  IF NOT END OF INPUT,
GET AND SORT NEXT SEQUENCE OF IN
PUT.
19Ø CLOSE#1: CLOSE#2: CLOSE#3
2ØØ REM INPUT NOW IN SORTED BLOC
KS ON FILES 1 AND 2.  START MERG
ING.
21Ø IN=1: REM FIRST INPUT FILE
22Ø REM MERGE FILES IN, IN+1 TO
OUTPUT FILES
23Ø OU=4-IN: REM FIRST OUTPUT FI
LE
24Ø FOR I=Ø TO 1:OPEN"I",#IN+I,F
I$(IN+I):OPEN"O",#OU+I,FI$(OU+I)
:NEXT I
25Ø IF EOF(IN+1) THEN 39Ø: REM S
ORT COMPLETE
26Ø INPUT#IN,X:INPUT#IN+1,Y
27Ø IF NOT EOF(IN) AND NOT EOF(I
N+1) THEN GOSUB 3ØØ ELSE IF NOT
 EOF(IN) THEN GOSUB 33Ø ELSE IF N
OT EOF(IN+1) THEN GOSUB 36Ø ELSE
 29Ø: REM MERGE BLOCKS FROM BOTH
 INPUT FILES IF POSSIBLE ELSE SE
ND NEXT BLOCK FROM REMAINING FIL
E TO OUTPUT ELSE MERGE CO
28Ø OU=9-IN-IN-OU: GOTO 27Ø: REM
 SWITCH OUTPUT FILES.  IF IN=1 TH
EN OU FLOPS BETWEEN 3 AND 4; IF
IN=3 THEN OU FLOPS BETWEEN 1 AND
 2
29Ø FOR I=1 TO 4:´CLOSE#I: NEXT
I: IN=4-IN: GOTO 22Ø: REM END OF
 ONE PASS OF MERGE
3ØØ REM MERGE NEXT BLOCKS ON FIL
ES IN AND IN+1.  FIRST DATA IN X
 AND Y.  AT END FIRST DATA FROM
NEXT BLOCKS OF FILES (IF ANY) IN
 X AND Y
31Ø IF X<Y THEN WRITE#OU,X: IF N
OT EOF(IN) THEN INPUT#IN,W: IF W
>=X THEN X=W: GOTO 31Ø ELSE X=W:
 GOSUB 36Ø: RETURN ELSE GOSUB 36
Ø: RETURN
32Ø WRITE#OU,Y: IF NOT EOF(IN+1)
 THEN INPUT#IN+1,W: IF W>=Y THEN
 Y=W: GOTO 31Ø ELSE Y=W: GOSUB 3
3Ø: RETURN ELSE GOSUB 33Ø: RETUR
N
33Ø REM FLUSH REMAINING DATA FRO
M SEQUENCE IN FILE IN.  FIRST DA
TUM IN X.  AT END FIRST DATUM OF
 NEXT BLOCK (IF ANY) IN X.
34Ø WRITE#OU,X: IF NOT EOF(IN) T
HEN INPUT#IN,W: IF W>=X THEN X=W
: GOTO 34Ø ELSE X=W
35Ø RETURN
36Ø REM FLUSH REMAINING DATA FRO
M SEQUENCE IN FILE IN+1.  FIRST
DATUM IN Y.  AT END FIRST DATUM
OF NEXT BLOCK (IF ANY) IN Y.
37Ø WRITE#OU,Y: IF NOT EOF(IN+1)
 THEN INPUT#IN+1,W: IF W>=Y THEN
 Y=W: GOTO 37Ø ELSE Y=W
38Ø RETURN
39Ø REM MERGE SORT COMPLETE.  OU
TPUT ON FILE IN.  TEMPORARILY RE
NAME INITIAL DATA FILE 'XXXX', T
HEN RENAME SORTED OUTPUT F$, THE
 NAME OF THE ORIGINAL DATA FILE.
  THUS ORIGINAL DATA FILE IS SOR
TED. KILL ALL OTHER FILES CREATE
D.
4ØØ FOR I=1 TO 4: CLOSE#I: NEXT
I: RENAME F$ TO "XXXX/DAT": RENA
ME FI$(IN) TO F$: KILL "XXXX/DAT
": FOR I=1 TO 4: IF I><IN THEN K
ILL FI$(I): NEXT I ELSE NEXT I
41Ø RETURN
42Ø REM QUICK SORT ARRAY A(1..N)
. INITIALIZE STAC TO EMPTY.  FIR
ST SEGMENT TO PARTITION IS ENTIR
E ARRAY.  SEGMENTS OF LENGTH<13
ARE NOT SORTED BY PARTITIONING,
BUT RATHER AT END THE ENTIRE ARR
AY IS SORTED BY BUBBLE SORT.
43Ø SP=-1: L=1: R=N: M=13
44Ø REM CURRENT SEGMENT TO SORT
IS A(L..R).  IF IT IS LENGTH<14,
 THEN POP LIMITS FOR NEXT SEGMEN
T OFF STACK. IF STACK EMPTY, THE
N A(I) IS DIVIDED INTO UNSORTED
CLUMPS OF ELEMENTS OF INCREASING
 SIZE. FINISH BY BUBBLE SORTING
ENTIRE ARRAY.
45Ø IF R-L<M THEN IF SP>Ø THEN R
=ST(SP): L=ST(SP-1): SP=SP-2: GO
TO 45Ø ELSE 56Ø
46Ø REM FIND V: MIN(A(L..R)) < V
 <= MAX(A(L..R))
47Ø I=L
48Ø IF I=R THEN GOTO 46Ø ELSE V=
A(I): IF V=A(I+1) THEN I=I+1: GO
TO 48Ø ELSE IF V<A(I+1) THEN V=A
(I+1): REM IF I=R THEN ALL A(L..
R) THE SAME.  GET NEXT SEGMENT T
O SORT.
49Ø REM PARTITION A(L..R) INTO A
(L..E)<V; A(D..R)>=V.
5ØØ D=L: E=R
51Ø IF A(D)<V THEN D=D+1:GOTO 51
Ø
52Ø IF A(E)>=V THEN E=E-1: GOTO
52Ø ELSE IF D<E THEN X=A(D): A(D
)=A(E): A(E)=X: GOTO 51Ø
53Ø REM A(L..R) IS NOW PARTITION
ED.  PUSH LIMITS OF LONGER PARTI
TION ON STACK AND REPARTITION SH
ORTER PARTITION
54Ø SP=SP+2: IF E-L>R-D THEN ST(
SP-1)=L: ST(SP)=E: L=D ELSE ST(S
```

```
P-1)=D: ST(SP)=R: R=E
55Ø GOTO 45Ø
56Ø REM BUBBLE SORT A(1..N) TO C
OMPLETE SORTING ARRAY
57Ø A(Ø)=-1E3Ø: REM INSERT GUARD
 VALUE AT BEGINNING OF ARRAY
58Ø FOR D=2 TO N
59Ø V=A(D): E=D-1
6ØØ IF A(E)>V THEN A(E+1)=A(E):
E=E-1: GOTO 6ØØ: REM GUARD VALUE
 WILL TERMINATE THIS LOOP IF E G
ETS DOWN TO Ø
61Ø A(E+1)=V: NEXT D
62Ø RETURN
```

The John-B System

See page 38

```
1 REM** JOHN-B SYSTEM
2 REM** PHONE I/O MODULE
3 REM** UNRESTRICTED ACCESS
        TO USE, COPY OR MODIFY
1Ø DIM Z(21)
2Ø PK=6528Ø
3Ø CLS 3
49Ø REM****** TELEPHONE ******
5ØØ CLS 3:FOR DLY=1 TO 5Ø:NEXT
51Ø PRINT"START-WHEN READY TO DI
AL"
52Ø S=PEEK(PK) AND 1:IF S<>Ø THE
N 52Ø
53Ø CLSØ:RESTORE:K=1
54Ø PRINT"CHOOSE PHONE NUMBER":F
OR I=1 TO 8:PRINT:NEXT I
55Ø FOR I=1 TO 5ØØ:NEXT I
56Ø PRINT@96,K"- ";:READ A$
57Ø IF A$="X" THEN 83Ø ELSE READ
 N$:PRINTA$,N$
58Ø FOR G=1 TO 1ØØ
59Ø S=PEEK(PK) AND 1
6ØØ IF S=Ø THEN 63Ø
61Ø NEXT G
62Ø K=K+1:GOTO 56Ø
63Ø PRINT@97,"DIALING "A$" NOW..
.":PRINT:PRINT
64Ø POKE 65313,4:J=LEN(N$)
65Ø FOR G=1 TO 1ØØØ:NEXT G
66Ø FOR I=1 TO J
67Ø Z(I)=ASC(MID$(N$,I,1))-48
68Ø IF Z(I)=Ø THEN Z(I)=1Ø
69Ø NEXT I
7ØØ FOR I=1 TO J:IF Z(I)=1Ø THEN
 PRINT Ø; ELSE PRINT Z(I);
71Ø FOR V=1 TO Z(I)
72Ø POKE 65313,52
73Ø FOR G=1 TO 18:NEXT G
74Ø POKE 65313,4
75Ø FOR G=1 TO 18:NEXT G
76Ø NEXT V
77Ø FOR G=1 TO 5ØØ:NEXT G
78Ø NEXT I
79Ø CLS2:PRINT"LIFT THE RECEIVER
-SAY HELLO"
8ØØ FOR G=1 TO 2ØØØ:NEXT G
81Ø POKE 65313,52
82Ø RESTORE:GOTO 3Ø
83Ø PRINT@97,"end of list...PLEA
SE TRY AGAIN":FOR G=1 TO 2ØØØ:NE
XT G:GOTO 82Ø
84Ø REM*PHONE NUMBERS (END LIST
WITH "X")
85Ø DATA OPERATOR,Ø
86Ø DATA JIM B., 12Ø15365499
87Ø DATA JOE,2556469
88Ø DATA TOM,2552913
89Ø DATA TEST.RING,9544
9ØØ DATA EDDIE,17544Ø67
91Ø DATA SHOP,4794Ø23
92Ø DATA JOHN,2456872
93Ø DATA MATT.B,3234Ø22
94Ø DATA BECKY,2881926
95Ø DATA INEZ,2Ø64832
96Ø DATA EMORY,2786543
97Ø DATA SKIP,2554Ø26
98Ø DATA BLACKIE,2Ø58933
99Ø DATA IDA,5356654
1ØØØ DATA KENA,2556743
1Ø1Ø DATA GLEN K.,3221124
1Ø2Ø DATA PENITO,17544923
1Ø3Ø DATA DON W.,2546467
1Ø4Ø DATA X
```

Date Minder

See page 74

```
2Ø P=PEEK(25):FOR X=2Ø TO 62 STE
P 6:IF P=X THEN PO=1 ELSE NEXTX:
FOR X=18 TO 54 STEP 6:IF P=X THE
N PO=Ø ELSE NEXTX:GOTO4Ø
3Ø CLS:PRINT@193,"COMPUTER HAS B
EEN SET TO PCLEAR2ERO. RELOAD AN
D RERUN PROGRAM.":IFPO=1THENPOKE
25,14:POKE3584,Ø:NEW ELSE IFPO=Ø
THENPMODEØ:PCLEAR1
4Ø 'load variables
5Ø CLEAR67ØØ:PL$="T4L2ØO2GO3G":P
1$=STRING$(32,61):P2$=STRING$(31
,61):P3$=P1$+P2$:P4$=P1$+P1$:P6$
=STRING$(12,62)+" ":P8$="NO DATA
 THIS DATE":SP$="*SPECIAL DAY*
*SPECIAL DAY*"
6Ø GOSUB17Ø:YL=365:DIM CE$(366,6
),MO$(12),DM(12),SD$(6),SD(6),TS
(6):FOR X=1 TO 12:READ MO$,DM:MO
$(X)=MO$:DM(X)=DM:NEXTX:GOTO2ØØ
7Ø DATA JANUARY,31,FEBRUARY,28,M
ARCH,31,APRIL,3Ø,MAY,31,JUNE,3Ø,
JULY,31,AUGUST,31,SEPTEMBER,3Ø,O
CTOBER,31,NOVEMBER,3Ø,DECEMBER,3
1
8Ø 'sub routines
9Ø PLAY"T8O3L2GL4CDEFL2GCP16CL2A
L4FGABO4L2CO3CP16CFL4GFEDL2EL4FE
DCL2O2BO3L4CDECL2EL1D":RETURN
1ØØ FL=Ø:FOR Z=1 TO 6:IF X+YD=TS
(Z)THENPRINT@2,SP$;:PRINT@482,SP
$;:SZ$="***"+SD$(Z)+"***":PRINT@
143-INT(LEN(SZ$)/2),SZ$:PRINT@36
7-INT(LEN(SZ$)/2),SZ$;:FL=1:GOSU
B9Ø:RETURN ELSE NEXTZ:RETURN
11Ø FORZ1=1TO6:SD=Ø:S$=STR$(SD(Z
1)):T1$=LEFT$(S$,2):T1=VAL(T1$):
T2$=RIGHT$(S$,2):FORZ2=1TOT1:SD=
SD+DM(Z2-1):TS(Z1)=SD+VAL(T2$):N
EXTZ2,Z1:RETURN
12Ø FL=Ø:FORZ=1TO6:IFX+YD=TS(Z)T
HENPT=INT(LEN(SD$(Z))/2-1):PRIN
T#-2:PRINT#-2,TAB(PT)SP$:PRINT#-
2:PRINT#-2,STRING$(12,62)" "SD$(
Z)" "STRING$(12,6Ø):PRINT#-2:PRI
NT#-2,TAB(PT)SP$:PRINT#-2:FL=1:R
ETURN ELSE NEXTZ:RETURN
13Ø IF MO=12 AND DA+14>31 THEN T
M=Ø:TY=Ø:RETURN ELSE TM=MO:TY=YD
+DM(TM):RETURN
14Ø MO=VAL(LEFT$(TD$,2)):DA=VAL(
RIGHT$(TD$,2)):FOR X=1 TO MO:YD=
YD+DM(X-1):NEXTX
15Ø D2=Ø:IF DA+14>DM(MO) THEN D1
=DM(MO):D2=DA+14-D1:RETURN ELSE
D1=DA+14:RETURN
16Ø PRINT@389,"PRESS KEY FOR NEX
T DAY";:EXEC 44539
17Ø CLS:PRINT@32,P4$;:PRINT@416,
P4$;:PLAYPL$:RETURN
18Ø FORT=1TO46Ø:NEXTT:RETURN
19Ø IF NE<1 THEN NE=Ø:RETURN ELS
E RETURN
2ØØ 'start program
21Ø PRINT@162,"CALENDAR PROGRAM
VERSION 5.3":PRINT@265,"COPYRIGH
T 1984":PRINT@3Ø3,"BY":PRINT@329
,"JAMES HUCKABEY":GOSUB9Ø:GOSUB1
8Ø
22Ø GOSUB17Ø:PRINT@227,"IS THIS
A LEAP YEAR (Y/N)"
23Ø I$=INKEY$:IFI$="Y"THEN DM(2)
=29:YL=366 ELSE IFI$<>"N"THEN23Ø
24Ø PRINT@224:PRINT@256:PLAYPL$:
PRINT@196,"ENTER TODAYS DATE (MM
DD)":PRINT@267,"";:INPUT TD$:IFL
EN(TD$)<>4ORVAL(TD$)<Ø1Ø1ORVAL(T
D$)>1231THEN24Ø
25Ø 'main menu
26Ø II$=INKEY$:GOSUB17Ø:PRINT@11
,"MAIN MENU":PRINT@134,"1=VIEW D
ATA":PRINT@166,"2=ENTER DATA":PR
INT@198,"3=SAVE TO TAPE"
27Ø PRINT@23Ø,"4=SAVE TO DISK":P
RINT@262,"5=LOAD FROM TAPE":PRIN
T@294,"6=LOAD FROM DISK":PRINT@3
26,"7=PRINT CURRENT DATA":PRINT@
358,"8=EXIT PROGRAM"
28Ø I$=INKEY$:ON INSTR("@1234567
8",I$)+1 GOTO 28Ø,28Ø,63Ø,29Ø,75
Ø,75Ø,8ØØ,8ØØ,88Ø,99Ø
29Ø 'data entry routines
3ØØ GOSUB17Ø:PRINT@1Ø4,"DATA ENT
RY MENU";:PRINT@163,"A=JANUARY":
PRINT@195,"B=FEBRUARY":PRINT@227
,"C=MARCH":PRINT@259,"D=APRIL":P
RINT@291,"E=MAY":PRINT@323,"F=JU
NE":PRINT@355,"G=JULY"
31Ø PRINT@178,"H=AUGUST":PRINT@2
1Ø,"I=SEPTEMBER":PRINT@242,"J=OC
TOBER":PRINT@274,"K=NOVEMBER":PR
INT@3Ø6,"L=DECEMBER":PRINT@338,"
M=SPECIALS":PRINT@37Ø,"N=MAIN ME
NU"
32Ø I$=INKEY$:IFI$="N"THEN25Ø EL
SE IF I$=""THEN32Ø ELSE I=ASC(I$
)-64:IFI<1ORI>13THENPLAYP7$:GOTO
32Ø ELSE IFI<13THEN49Ø
33Ø 'special day entries
34Ø GOSUB17Ø:C=1:FORX=1TO6:IFSD$
(X)=""THENC=X:TC=X-1 ELSE NEXTX:
GOTO38Ø
35Ø TC$=STR$(TC):TC$=RIGHT$(TC$,
LEN(TC$)-1):PRINT@1Ø4,"SIX SPECI
AL DAYS":PRINT@137,"USED SO FAR:
"TC$:PRINT@384,"TYPE '#' FOR EDI
TOR '@' FOR MENU";:PA$="TOTAL EN
TRIES="+STR$(NE)
36Ø PRINT@495-INT(LEN(PA$)/2),PA
$;:PRINT@224:PRINT@228,"";:INPUT
"ENTER DATE(MMDD)";S1$:IFS1$="#"
THEN38Ø ELSE IFS1$="@"THEN29Ø EL
SE SD=VAL(S1$):IFSD<Ø1Ø1ORSD>123
1THEN33Ø ELSE SD(C)=SD:EF=1:NE=N
```

```
E+1
37Ø PA$="ENTER DATA FOR "+S1$:PR
INT@2Ø7-INT(LEN(PA$)/2),PA$:PRIN
T@224:PRINT@284,"<24":PLAYPL$:PR
INT@257,"";:INPUT S2$:IFLEN(S2$)
>24 OR S2$=""THENPRINT@256:GOTO3
7Ø ELSE SD$(C)=S2$:TC=C:IF C<6 T
HEN C=C+1:GOSUB17Ø:GOTO35Ø
38Ø GOSUB19Ø:CLS:PRINT@3,"NUMBER
 OF SPECIAL DAYS =";TC:PRINT@32,
P1$:PRINT@48Ø,P2$;:POKE1535,125:
PRINT@452,"WANT TO CHANGE DATA(Y
/N)";:PLAYPL$
39Ø PA=64:Y=1:FORX=1TO6:IF SD(X)
=Ø THEN PRINT@PA,Y:PA=PA+32:Y=Y+
1:PRINT@PA,Y:PA=PA+32:Y=Y+1:NEXT
X:GOTO42Ø
4ØØ SD$=STR$(SD(X)):SD$=RIGHT$(S
D$,LEN(SD$)-1):IF LEN(SD$)<4 THE
N S$="Ø"+SD$ ELSE S$=SD$
41Ø PRINT@PA,Y;S$:PA=PA+32:Y=Y+1
:PRINT@PA,Y;SD$(X):PA=PA+32:Y=Y+
1:NEXTX
42Ø I$=INKEY$:IFI$="N"THEN29ØELS
E IFI$<>"Y"THEN42Ø
43Ø PRINT@448:PLAYPL$:PRINT@45Ø,
"";:INPUT"ENTER NUMBER(1-12) #=A
LL";I$:PRINT@448:FORX=1TO6:IFSD$
(X)>""THENTE=X:NEXTX ELSE NEXTX
44Ø IFI$="#"THENC=1:FORX=1TO6:SD
$(X)="":SD(X)=Ø:NEXT:NE=NE-TE:GO
TO34Ø ELSE IFI$="@"THEN 29Ø ELSE
 I=VAL(I$):IFI<1ORI>12ORINT((I+1
)/2)>TE THENPRINT@448:GOTO43Ø
45Ø PA=32+(I*32):PRINT@PA:PRINT@
PA+28,"<24";:PRINT@PA,I$;:PLAYPL
$:INPUTSD$:IFLEN(SD$)>24THENPRIN
T@PA:GOTO45Ø
46Ø TI=INT((I+1)/2):IF SD$=""THE
N 48Ø ELSE IFI/2=INT(I/2)THENSD$
(I/2)=SD$:GOTO38Ø
47Ø IFSD$=""THEN 48Ø ELSE SD=VAL
(SD$):IFSD<Ø1Ø1ORSD>1231THENPRIN
T@PA:GOTO45Ø ELSE SD(TI)=SD:GOTO
38Ø
48Ø TC=TC-1:NE=NE-1:IF TI=6 THEN
 SD$(TI)="":SD(TI)=Ø:GOTO38Ø ELS
E FOR SW=TI TO 5:SD$(SW)=SD$(SW+
1):SD(SW)=SD(SW+1):NEXT SW:SD$(S
W)="":SD(SW)=Ø:GOTO 38Ø
49Ø 'daily entries
5ØØ GOSUB17Ø:DA=Ø:YD=Ø:FOR X=1 T
O I:YD=YD+DM(X-1):NEXT:PT$=LEFT$
(MO$(I),3)
51Ø GOSUB19Ø:P$=MO$(I)+STR$(DM(I
))+" DAYS":PRINT@143-INT(LEN(P$)
/2),P$:PRINT@354,"TO ACCESS EDIT
OR ENTER DATE":PRINT@391,"ENTER
'@' TO EXIT"
52Ø PRINT@229,"";:INPUT"ENTER DA
Y OF MONTH";DA$:DA=VAL(DA$):IFDA
$="@"THEN29Ø ELSE IF DA<1 OR DA>
DM(I) THENPRINT@224:GOTO52Ø
53Ø FORX=1TO6:IF CE$(DA+YD,X)=""
THEN C=X:TC=C-1:PRINT@29Ø,"NUMBE
R OF ENTRIES ON"DA"="TC:PRINT@35
2 ELSE NEXTX:GOTO57Ø
54Ø PRINT@224:PA$="NUMBER OF ENT
RIES ="+STR$(NE):PRINT@495-INT(L
EN(PA$)/2),PA$;:PLAYPL$
55Ø PRINT@353,"ENTER '#' TO EDIT
 '*' FOR MENU":PRINT@252,"<24";:
PRINT@225,"";:INPUT DE$:IF LEN(D
E$)>24ORDE$=""THEN55Ø ELSE IFDE$
="#"THEN57Ø ELSE IFDE$="@"THENDE
$="":GOSUB17Ø:GOTO51Ø ELSE IF DE
$="*"THEN29Ø
56Ø NE=NE+1:EF=1:CE$(DA+YD,C)=DE
$:PRINT@31Ø,DA"="C:C=C+1:IFC<7TH
EN54Ø
57Ø GOSUB17Ø:GOSUB19Ø:PRINT@12,"
EDITOR":PN$="ENTRIES FOR "+PT$+S
TR$(DA):PRINT@111-INT(LEN(PN$)/2
),PN$:PA=16Ø:FORX=1TO6:PRINT@PA,
X;CE$(DA+YD,X):PA=PA+32:NEXTX:PR
INT@388,"WANT TO CHANGE DATA(Y/N
)";
58Ø I$=INKEY$:IFI$="N"THENGOSUB1
7Ø:GOTO51ØELSEIFI$<>"Y"THEN58Ø
59Ø TE=1:PRINT@385,"ENTER ENTRY
NUMBER (1-6) #=ALL";:PLAYPL$:FOR
X=1TO6:IFCE$(DA+YD,X)>""THEN TE=
X:NEXTX ELSE NEXTX
6ØØ I$=INKEY$:TI=VAL(I$):IFI$="#
"THENFORX=1TO6:CE$(DA+YD,X)="":N
EXT:NE=NE-TE:GOSUB17Ø:GOTO51Ø EL
SE IFI$="@"THEN5ØØ ELSE IFTI<1OR
TI>TE THEN6ØØ
61Ø PRINT@384:PRINT@393,"ENTER N
EW DATA":PA=128+(TI*32):PRINT@PA
:PRINT@PA+28,"<24";:PRINT@PA,I$;
:PLAYPL$:INPUT DE$:IFLEN(DE$)>24
THENPRINT@PA:GOTO61Ø ELSE IF DE$
="@"THENGOSUB17Ø:GOTO51Ø
62Ø IFDE$>""THENCE$(DA+YD,TI)=DE
$:GOTO57Ø ELSE NE=NE-1:IF TI=6 T
HEN CE$(DA+YD,TI)="":GOTO57Ø ELS
E FOR SW=TI TO 5:CE$(DA+YD,SW)=C
E$(DA+YD,SW+1):NEXT SW:CE$(DA+YD
,SW)="":GOTO57Ø
63Ø 'scan routine
64Ø YD=Ø:GOSUB17Ø:PRINT@1Ø7,"VIE
W MENU":PRINT@196,"1=SCAN FROM T
ODAYS DATE":PRINT@26Ø,"2=CHANGE
SCAN DATE":PRINT@324,"3=RETURN T
O MAIN MENU"
65Ø I$=INKEY$:IFI$="1"THENGOSUB1
4Ø:GOTO68Ø ELSE IFI$="3"THEN26Ø
ELSE IFI$<>"2"THEN65Ø
66Ø GOSUB17Ø:PRINT@163,"ENTER NE
W SCAN DATE (MMDD)":PRINT@233,"O
R '@' TO EXIT":PRINT@299,"";:INP
UT ND$:IFND$="@"THEN63Ø ELSE IFL
EN(ND$)<>4ORVAL(ND$)<Ø1Ø1ORVAL(N
D$)>1231THEN66Ø
67Ø MO=VAL(LEFT$(ND$,2)):DA=VAL(
RIGHT$(ND$,2)):FORX=1TOMO:YD=YD+
DM(X-1):NEXT:GOSUB15Ø
68Ø GOSUB17Ø:PRINT@234,"PLEASE W
AIT":PA=162:GOSUB11Ø:PRINT@224:P
LAYPL$
69Ø FOR X=DA TO D1:PR$=MO$(MO)+S
TR$(X):PRINT@111-INT(LEN(PR$)/2)
,PR$:PRINT@224:FOR Y=1 TO 6:IFCE
$(X+YD,1)=""THEN71Ø ELSE PRINT@P
A,Y;CE$(X+YD,Y):PA=PA+32:NEXTY:P
A=162:GOSUB1ØØ
7ØØ GOSUB16Ø:NEXTX:IFD2=ØTHEN63Ø
 ELSE 72Ø
71Ø PRINT@224:GOSUB1ØØ:IFFL=1THE
N7ØØ ELSE PRINT@231,P8$:PLAYPL$:
GOSUB18Ø:NEXTX:IFD2=ØTHEN63Ø
72Ø GOSUB13Ø:PRINT@96:PA=162:FOR
 X=1 TO D2:PR$=MO$(TM+1)+STR$(X)
:PRINT@111-INT(LEN(PR$)/2),PR$:P
RINT@224:FOR Y=1 TO 6:IF CE$(X+T
Y,1)=""THEN 74Ø ELSE PRINT@PA,Y;
CE$(X+TY,Y):PA=PA+32:NEXTY:PA=16
2:YD=TY:GOSUB1ØØ
73Ø GOSUB16Ø:NEXTX:GOTO63Ø
74Ø YD=TY:GOSUB1ØØ:IFFL=1THEN73Ø
 ELSE PRINT@231,P8$:PLAYPL$:GOSU
B18Ø:NEXTX:GOTO63Ø
75Ø 'save routines
76Ø IFI$="3"THEN GOSUB17Ø:DN$="T
APE":DN=-1:PRINT@225,"PREPARE RE
CORDER,PRESS ANY KEY":EXEC44539
ELSE DN$="DISK":DN=1
77Ø GOSUB17Ø:PRINT@226,"SAVING T
O "DN$", PLEASE WAIT."
78Ø OPEN"O",#DN,"CALENDAR":PRINT
#DN,NE:FORX=1TO366:FORY=1TO6:PRI
NT#DN,CE$(X,Y):NEXTY,X
79Ø FORX=1TO6:PRINT#DN,SD$(X):PR
INT#DN,SD(X):NEXT:CLOSE#DN:PRINT
@224:PRINT@231,"DATA SAVED TO "D
N$:PLAYPL$:GOSUB18Ø:GOTO26Ø
8ØØ 'load routines
81Ø IFEF=ØTHEN84Ø ELSE II$=INKEY
$:GOSUB17Ø:PRINT@13Ø,"TYPE IN 'Y
' TO CONFIRM LOAD":PRINT@196,"IF
 YOU LOAD A NEW FILE":PRINT@258,
"THE EXISTING FILE IS DELETED":P
RINT@322,"ANY OTHER KEY FOR MAIN
 MENU"
82Ø II$=INKEY$:IFII$=""THEN82Ø E
LSE IF II$<>"Y"THEN25Ø
83Ø GOSUB17Ø:PRINT@226,"EXISTING
 FILE BEING DELETED":NE=Ø:FORX=1
TO366:FORY=1TO6:CE$(X,Y)="":NEXT
Y,X:FORX=1TO6:SD$(X)="":SD(X)=Ø:
TS(X)=Ø:NEXTX:EF=Ø
84Ø IFI$="5"THEN GOSUB17Ø:DN$="T
APE":DN=-1:PRINT@225,"PREPARE RE
CORDER,PRESS ANY KEY":EXEC44539
ELSE DN$="DISK":DN=1
85Ø GOSUB17Ø:PRINT@225,"LOADING
FROM "DN$", PLEASE WAIT"
86Ø OPEN"I",#DN,"CALENDAR":INPUT
#DN,NE:FORX=1TO366:FORY=1TO6:INP
UT#DN,CE$(X,Y):NEXTY,X:FORX=1TO6
:INPUT#DN,SD$(X):INPUT#DN,SD(X):
IF EOF(DN)=-1 THEN 87Ø ELSE NEXT
X
87Ø CLOSE#DN:PRINT@224:PRINT@229
,"DATA LOADED FROM "DN$:EF=1:PLA
YPL$:GOSUB18Ø:GOTO25Ø
88Ø 'printer routine
89Ø GOSUB17Ø:YD=Ø:PRINT@232,"DAT
A TO PRINTER":GOSUB14Ø:FOR X=1 T
O 4:PRINT#-2:NEXTX:PP$="CALENDAR
 PRINTER DUMP FOR "+MO$(MO)+STR$
(DA):PT=4Ø-INT(LEN(PP$)/2):PRINT
#-2,TAB(PT)PP$:PRINT#-2
9ØØ GOSUB11Ø:FOR X=DA TO D1:PRIN
T#-2:PRINT#-2,P6$;MO$(MO);X:PRIN
T#-2:SD=Ø:GOSUB12Ø
91Ø IF FL=1 AND CE$(X+YD,1)=""TH
EN94Ø ELSE IF FL=Ø AND CE$(X+YD,
1)=""THENPRINT#-2,TAB(13)P8$:GOT
O94Ø ELSE FOR Y=1 TO 6 STEP3:PRI
NT#-2,Y;CE$(X+YD,Y);
92Ø IFCE$(X+YD,Y+1)=""THEN PRINT
#-2:GOTO94Ø ELSE PRINT#-2,TAB(28
)Y+1;CE$(X+YD,Y+1);
93Ø IFCE$(X+YD,Y+2)=""THEN PRINT
#-2:GOTO94Ø ELSE PRINT#-2,TAB(56
)Y+2;CE$(X+YD,Y+2):NEXTY
94Ø NEXTX:IF D2=ØTHEN25Ø ELSE GO
SUB13Ø
95Ø FORX=1TOD2:PRINT#-2:PRINT#-2
,P6$;MO$(MO+1);X:PRINT#-2:SD=Ø:Y
D=TY:GOSUB12Ø:FOR Y=1 TO 6 STEP3
:IFCE$(X+TY,Y)=""THEN PRINT#-2,T
AB(13)P8$:GOTO98Ø ELSE PRINT#-2,
Y;CE$(X+TY,Y);
96Ø IF CE$(X+TY,Y+1)=""THEN PRIN
T#-2:GOTO98Ø ELSE PRINT#-2,TAB(2
8)Y+1;CE$(X+TY,Y+1);
97Ø IF CE$(X+TY,Y+2)=""THEN PRIN
T#-2:GOTO98Ø ELSE PRINT#-2,TAB(5
6)Y+2;CE$(X+TY,Y+2):NEXTY
98Ø NEXTX:GOTO25Ø
99Ø 'end routine
1ØØØ GOSUB17Ø:PRINT@233,"END OF
PROGRAM":GOSUB9Ø:END:GOTO25Ø
1Ø1Ø 'CALENDAR VERSION 5.3
```

Tricky Dice

See page 72

```
10 '***** tricky-dice *****
20 '*** BY VOLKER MEINHARDT ***
30 '**** game preparation ****
40 B$="T4;O3;L1;C;O2;L4;B;O3;L4;
C;O2;L2;B;L2;G;L4;E;L3;A;L2;E;L4
;D;L2.;C;D;L1;O3;C;L4;D;E;L2;G;L
2;E;L4;D;L1;C;L4;D;E;L1;D;L4;P2;
T5;C;C;D;O2;L4.;B;O3;L8;C;L4.;D;
L4;E;E;F;L4.;E;L8;D;L4.;C;;D;C;O
2;B;O3;L2;C;"
50 '*** player ? ***
60 FORI=1TO8:CLS(I):PRINT@234,"T
RICKY-DICE";:SOUND20*I,3:NEXT:CL
S
70 PRINT@160,"PLEASE, TYPE IN TH
E NUMBER OF   PLAYERS";:INPUT A
80 DIM NAME$(A,10):DIM Z(6)
90 FORI=1TOA:PRINT"NAME, PLAYER"
;I;:INPUT NAME$(I,10):NEXT
100 '*** squares ***
110 DIM WERT(A,15):DIM W(5):CLS
120 '**** game starts ****
130 '***loop-passages***
140 FORD=1TO13
150 '***loop-player***
160 FORS=1TOA:CLS
170 W1=1:W2=1:W3=1:W4=1:W5=1
180 FORI=1TO5:W(I)=0:NEXT
190 GOSUB1970
200 PRINT@54,NAME$(S,10);
210 GOSUB2410
220 '***loop-throw***
230 FORW=1TO3
240 FORI=363TO383:PRINT@I,CHR$(1
75);:NEXTI:PRINT@299,CHR$(191);
250 '**joystick horizontal**
260 HA=11
270 H=JOYSTK(0)/2
280 P=PEEK(65280)
290 IFH<11THEN H=11
300 IFHA<>H THENPRINT@HA+352,CHR
$(175);
310 PRINT@352+H,CHR$(94);
320 IFPOINT(22,22)<>3AND(P=126OR
 P=254)THEN520
330 IFPOINT(22,22)<>3THENSOUND12
0,1
340 IF(POINT(26,22)<>3OR POINT(2
8,22)<>3OR POINT(30,22)<>3)AND(P
=126OR P=254)THEN420
350 IF(POINT(34,22)<>3OR POINT(3
6,22)<>3OR POINT(38,22)<>3)AND(P
=126OR P=254)THEN440
360 IF(POINT(42,22)<>3OR POINT(4
4,22)<>3OR POINT(46,22)<>3)AND(P
=126OR P=254)THEN460
370 IF(POINT(50,22)<>3OR POINT(5
2,22)<>3OR POINT(54,22)<>3)AND(P
=126OR P=254)THEN480
380 IF(POINT(58,22)<>3OR POINT(6
0,22)<>3OR POINT(62,22)<>3)AND(P
=126OR P=254)THEN500
390 HA=H
400 GOTO270
410 '**dice-make stand**
420 IFW1=1THENW1=-1:PRINT@269,CH
R$(255);CHR$(255);CHR$(255);:GOS
UB2400:GOTO270
430 IFW1=-1THENW1=1:PRINT@269,"
  ";:GOSUB2400:GOTO270
440 IFW2=1THEN W2=-1:PRINT@273,C
HR$(255);CHR$(255);CHR$(255);:GO
SUB2400:GOTO270
450 IFW2=-1THEN W2=1:PRINT@273,"
  ";:GOSUB2400:GOTO270
460 IFW3=1THEN W3=-1:PRINT@277,C
HR$(255);CHR$(255);CHR$(255);:GO
SUB2400:GOTO270
470 IFW3=-1THEN W3=1:PRINT@277,"
  ";:GOSUB2400:GOTO270
480 IFW4=1THEN W4=-1:PRINT@281,C
HR$(255);CHR$(255);CHR$(255);:GO
SUB2400:GOTO270
490 IFW4=-1THEN W4=1:PRINT@281,"
  ";:GOSUB2400:GOTO270
500 IFW5=1THEN W5=-1:PRINT@285,C
HR$(255);CHR$(255);CHR$(255);:GO
SUB2400:GOTO270
510 IFW5=-1THEN W5=1:PRINT@285,"
  ";:GOSUB2400:GOTO270
520 '**throw ????  **
530 PRINT@120,W;
540 IFW1=1THEN O=26:GOSUB640
550 IFW2=1THEN O=34:GOSUB640
560 IFW3=1THEN O=42:GOSUB640
570 IFW4=1THEN O=50:GOSUB640
580 IFW5=1THEN O=58:GOSUB640
590 IFW1=-1ANDW2=-1ANDW3=-1ANDW4
=-1ANDW5=-1THEN FORI=1TO300:NEXT
I:GOTO680
600 '**tricky ????  **
610 IFW(1)=W(2)ANDW(1)=W(3)ANDW(
1)=W(4)ANDW(1)=W(5)THEN ZAHL=0:F
ORI=1TO5:ZAHL=ZAHL+W(I):NEXTI:GO
TO1720
620 NEXTW
630 GOTO680
640 '**throw !!  **
650 X=RND(6):J=(O-18)/8:W(J)=X
660 ON X GOSUB2030,2070,2110,215
0,2190,2230
670 RETURN
680 ZAHL=0:FORI=1TO5:ZAHL=ZAHL+W
(I):NEXTI
690 '*** notations ***
700 PRINT@363,"":PRINT@299," ";
710 FORI=12TO428STEP32:PRINT@I,C
HR$(175);:NEXTI
720 '**joystick-vertical**
730 VA=428
740 H=JOYSTK(0)
750 V=INT(JOYSTK(1)/4)*32+12
760 P=PEEK(65280)
770 IFV>428THEN V=428
780 IFVA<>V THENPRINT@VA,CHR$(17
5);
790 PRINT@V,CHR$(60);
800 IFV=12AND(P=126OR P=254)THEN
950
810 IFV=44AND(P=126OR P=254)THEN
1000
820 IFV=76AND(P=126OR P=254)THEN
1050
830 IFV=108AND(P=126OR P=254)THE
N1100
840 IFV=140AND(P=126OR P=254)THE
N1150
850 IFV=172AND(P=126OR P=254)THE
N1200
860 IFV=236AND(P=126OR P=254)THE
N1320
870 IFV=268AND(P=126OR P=254)THE
N1380
880 IFV=300AND(P=126OR P=254)THE
N1440
890 IFV=332AND(P=126OR P=254)THE
N1520
900 IFV=364AND(P=126OR P=254)THE
N1590
910 IFV=396AND(P=126OR P=254)THE
N1660
920 IFV=428AND(P=126OR P=254)THE
N1690
930 VA=V
940 GOTO740
950 '**1-SPOTS ???**
960 IFWERT(S,1)=-1OR WERT(S,1)>0
THEN 740
970 GOSUB2300
980 IFZ(1)=0THEN WERT(S,1)=-1:GO
SUB2410:GOSUB2380:GOTO1810
990 IFZ(1)>0THEN WERT(S,1)=Z(1):
GOTO1250
1000 '**2-SPOTS ???**
1010 IFWERT(S,2)=-1OR WERT(S,2)>
0THEN 740
1020 GOSUB2300
1030 IFZ(2)=0THEN WERT(S,2)=-1:G
OSUB2410:GOSUB2380:GOTO1810
1040 IFZ(2)>0THEN WERT(S,2)=Z(2)
*2:GOTO1250
1050 '**3-SPOTS ???**
1060 IFWERT(S,3)=-1OR WERT(S,3)>
0THEN 740
1070 GOSUB2300
1080 IFZ(3)=0THEN WERT(S,3)=-1:G
OSUB2410:GOSUB2380:GOTO1810
1090 IFZ(3)>0THEN WERT(S,3)=Z(3)
*3:GOTO1250
1100 '**4-SPOTS ???**
1110 IF WERT(S,4)=-1OR WERT(S,4)
>0THEN 740
1120 GOSUB2300
1130 IFZ(4)=0THEN WERT(S,4)=-1:G
OSUB2410:GOSUB2380:GOTO1810
1140 IFZ(4)>0THEN WERT(S,4)=Z(4)
*4:GOTO1250
1150 '**5-SPOTS ???**
1160 IF WERT(S,5)=-1OR WERT(S,5)
>0THEN 740
1170 GOSUB2300
1180 IFZ(5)=0THEN WERT(S,5)=-1:G
OSUB2410:GOSUB2380:GOTO1810
1190 IFZ(5)>0THEN WERT(S,5)=Z(5)
*5:GOTO1250
1200 '**6-SPOTS ???**
1210 IFWERT(S,6)=-1OR WERT(S,6)>
0THEN 740
1220 GOSUB2300
1230 IFZ(6)=0THEN WERT(S,6)=-1:G
OSUB2410:GOSUB2380:GOTO1810
1240 IFZ(6)>0THEN WERT(S,6)=Z(6)
*6
1250 SUM=0:FORI=1TO6
1260 IFWERT(S,I)>0THEN SUM=SUM+W
ERT(S,I)
1270 NEXTI
1280 IF SUM>62 THEN SUM=SUM+35
1290 IFWERT(S,7)<63AND SUM>96THE
N FORI=10 TO 250STEP10:SOUNDI,1:
NEXTI
1300 WERT(S,7)=SUM
1310 GOTO1760
1320 '**3-DOUBLETS ???**
1330 IFWERT(S,8)=-1OR WERT(S,8)>
0THEN 740
1340 GOSUB2300
1350 FORI=1TO6:IF Z(I)>=3THEN WE
RT(S,8)=ZAHL:GOTO1760
1360 NEXTI
1370 WERT(S,8)=-1:GOSUB2410:GOSU
B2380:GOTO1810
1380 '**4-DOUBLETS ???**
1390 IFWERT(S,9)=-1OR WERT(S,9)>
0THEN 740
1400 GOSUB2300
1410 FORI=1TO6:IF Z(I)>=4THEN WE
RT(S,9)=ZAHL:GOTO1760
```

```
1420 NEXTI
1430 WERT(S,9)=-1:GOSUB2410:GOSU
B2380:GOTO1810
1440 '**FULL HOUSE ???**
1450 IFWERT(S,10)=-1OR WERT(S,10
)>0THEN 740
1460 GOSUB2300
1470 FORI=1TO6:IF Z(I)=3 THEN 14
90
1480 NEXTI:GOTO1510
1490 FORJ=1TO6:IF Z(J)=2 THEN WE
RT(S,10)=25:GOTO1760
1500 NEXTJ
1510 WERT(S,10)=-1:GOSUB2410:GOS
UB2380:GOTO1810
1520 '**SMALL STREET ???**
1530 IFWERT(S,11)=-1OR WERT(S,11
)>0THEN 740
1540 GOSUB2300
1550 J=0:FORI=1TO6:IF Z(I)>0THEN
 J=J+1:IF J=4THEN WERT(S,11)=30:
GOTO1760
1560 IF Z(I)=0THEN J=0
1570 NEXTI
1580 WERT(S,11)=-1:GOSUB2410:GOS
UB2380:GOTO1810
1590 '**GEAT STREET ???**
1600 IFWERT(S,12)=-1OR WERT(S,12
)>0THEN 740
1610 GOSUB2300
1620 J=0:FORI=1TO6:IF Z(I)>0THEN
 J=J+1:IF J=5THEN WERT(S,12)=40:
GOTO1760
1630 IF Z(I)=0THEN J=0
1640 NEXTI
1650 WERT(S,12)=-1:GOSUB2410:GOS
UB2380:GOTO1810
1660 '**delete tricky !! **
1670 IFWERT(S,13)>0 OR WERT(S,13
)=-1THEN 740
1680 WERT(S,13)=-1:GOSUB2410:GOS
UB2380:GOTO1810
1690 '**CHANCE !!!**
1700 IFWERT(S,14)>0THEN 740
1710 WERT(S,14)=ZAHL:GOTO1760
1720 '**note down tricky ?? **
1730 IFWERT(S,13)=-1THENPRINT@17
7,"POOR GAMBLER !";:FORI=200TO1S
TEP-10:SOUNDI,2:NEXTI:GOTO700
1740 IFWERT(S,13)=0THEN WERT(S,1
3)=50:FORI=1TO3:FORJ=I*5TOI*75ST
EP3:SOUNDJ,1:NEXTJ,I:GOTO1760
1750 IFWERT(S,13)>0THEN WERT(S,1
3)=WERT(S,13)+100:FORI=1TO3:FORJ
=I*10TOI*75STEP2:SOUNDJ,1:NEXTJ,
I:GOTO700
1760 '** calculate sum **
1770 J=0:FORI=7TO14:IFWERT(S,I)>
0THEN J=J+WERT(S,I)
1780 NEXTI
1790 WERT(S,15)=J:GOSUB2410
1800 FORI=1TO400:NEXTI
1810 NEXTS
1820 NEXTD
1830 '*** final result ***
1840 X=0:FORI=1TOA:IFWERT(I,15)>
X THEN X=WERT(I,15):J=I
1850 NEXTI
1860 CLS(4):PRINT@44,"WINNER";:P
RINT@199,"BY ";X;" POINTS";:PRIN
T@330,NAME$(J,10);
1870 PLAY B$:CLS(3)
1880 PRINTTAB(9);"final result";
:PRINT
1890 FORK=1TOA:J=0:X=0
1900 FORI=1TOA:IFWERT(I,15)>X TH
EN X=WERT(I,15):J=I
1910 NEXTI:PRINTK;". ";NAME$(J,1
0);"  - ";X;" POINTS":WERT(J,15)
=0:NEXTK
1920 IFPEEK(65280)=126 OR PEEK(6
5280)=254 THEN CLS:PRINT@228,"ON
E MORE GAME ? (J/N)";:GOTO1940
1930 GOTO1920
1940 A$=INKEY$
1950 IFA$=""THEN 1940
1960 IFA$="J"THEN RUN ELSE END
1970 '*sp-field construction*
1980 PRINT@0,"1-SPOTS:  ";CHR$(1
59);:PRINT@32,"2-SPOTS: ";CHR$(1
59);CHR$(159);:PRINT@46,"PLAYER:
";:PRINT@64,"3-SPOTS: ";CHR$(159
);CHR$(159);
1990 PRINT@96,"4-SPOTS: ";CHR$(1
59);CHR$(159);:PRINT@110,"THROW-
NR.:";:PRINT@128,"5-SPOTS: ";CHR
$(159);CHR$(159);:PRINT@160,"6-S
POTS: ";CHR$(159);CHR$(159);:PRI
NT@192,"TOP SUM:";
2000 PRINT@225,"3 SAME: ";CHR$(1
59);CHR$(159);:PRINT@257,"4 SAME
: ";CHR$(159);CHR$(159);:PRINT@2
88,"F.HOUSE: ";CHR$(159);CHR$(15
9);:PRINT@320,"LT.STR.: ";CHR$(1
59);CHR$(159);
2010 PRINT@352,"GR.STR.: ";CHR$(
159);CHR$(159);:PRINT@385,"TRICK
Y:";CHR$(159);CHR$(159);CHR$(159
);:PRINT@417,"CHANCE: ";CHR$(159
);CHR$(159);:PRINT@482,"TOTAL: "
;
2020 RETURN
2030 '*sp-die 1*
2040 GOSUB2270
2050 SET(O+2,19,5)
2060 RETURN
2070 '*sp-die 2*
2080 GOSUB2270
2090 SET(O+1,20,5):SET(O+3,18,5)
2100 RETURN
2110 '*sp-die 3*
2120 GOSUB2270
2130 SET(O,18,5):SET(O+2,19,5):S
ET(O+4,20,5)
2140 RETURN
2150 '*sp-die 4*
2160 GOSUB2270
2170 SET(O,18,5):SET(O,20,5):SET
(O+4,18,5):SET(O+4,20,5)
2180 RETURN
2190 '*sp-die 5*
2200 GOSUB2270
2210 SET(O,18,5):SET(O,20,5):SET
(O+2,19,5):SET(O+4,18,5):SET(O+4
,20,5)
2220 RETURN
2230 '*sp-die 6*
2240 GOSUB2270
2250 SET(O,18,5):SET(O,20,5):SET
(O+2,18,5):SET(O+2,20,5):SET(O+4
,18,5):SET(O+4,20,5)
2260 RETURN
2270 '*sp-reset dice*
2280 FORI=O TO O+5:RESET(I,18):R
ESET(I,19):RESET(I,20):RESET(I,2
1):NEXTI
2290 RETURN
2300 '*sp-part check routine*
2310 FORI=1TO6
2320 Z(I)=0
2330 FORJ=1TO5
2340 IFW(J)=I THEN Z(I)=Z(I)+1
2350 NEXTJ,I
2360 RETURN
2370 '*sp-deletions*
2380 FORI=1TO4:PRINT@177,"delete
d";:SOUNDI*50,2:PRINT@177,"
  ";:SOUNDI*10,2:NEXTI:RETURN
2390 '*sp-time delay*
2400 FORI=1TO400:NEXTI:RETURN
2410 '*sp-write notation-list*
2420 POKE65495,0
2430 IFWERT(S,1)>0THENPRINT@9,WE
RT(S,1); ELSE IF WERT(S,1)=-1THE
NPRINT@10,CHR$(239);
2440 IFWERT(S,2)>0THEN IFWERT(S,
2)<10THENPRINT@41,WERT(S,2);ELSE
PRINT@40,WERT(S,2);ELSE IFWERT(S
,2)=-1THENPRINT@41,CHR$(239);CHR
$(239);
2450 IFWERT(S,3)>0THEN IFWERT(S,
3)<10THENPRINT@73,WERT(S,3);ELSE
PRINT@72,WERT(S,3);ELSE IFWERT(S
,3)=-1THENPRINT@73,CHR$(239);CHR
$(239);
2460 IF WERT(S,4)>0THEN IFWERT(S
,4)<10THENPRINT@105,WERT(S,4);EL
SEPRINT@104,WERT(S,4);ELSE IF WE
RT(S,4)=-1THENPRINT@105,CHR$(239
);CHR$(239);
2470 IFWERT(S,5)>0THEN IFWERT(S,
5)<10THENPRINT@137,WERT(S,5);ELS
EPRINT@136,WERT(S,5);ELSE IFWERT
(S,5)=-1THENPRINT@137,CHR$(239);
CHR$239);
2480 IFWERT(S,6)>0THEN IFWERT(S,
6)<10THENPRINT@169,WERT(S,6);ELS
EPRINT@168,WERT(S,6);ELSE IFWERT
(S,6)=-1THENPRINT@169,CHR$(239);
CHR$(239);
2490 IFWERT(S,7)>0ANDWERT(S,7)<1
0THENPRINT@201,WERT(S,7);
2500 IFWERT(S,7)>9THEN IF WERT(S
,7)<100THENPRINT@200,WERT(S,7);E
LSEPRINT@199,WERT(S,7);
2510 IFWERT(S,8)>0THEN IFWERT(S,
8)<10THENPRINT@233,WERT(S,8);ELS
EPRINT@232,WERT(S,8);ELSE IFWERT
(S,8)=-1THENPRINT@233,CHR$(239);
CHR$(239);
2520 IFWERT(S,9)>0THEN IFWERT(S,
9)<10THENPRINT@265,WERT(S,9);ELS
EPRINT@264,WERT(S,9);ELSE IFWERT
(S,9)=-1THENPRINT@265,CHR$(239);
CHR$(239);
2530 IFWERT(S,10)>0THENPRINT@296
,WERT(S,10);ELSE IFWERT(S,10)=-1
THENPRINT@297,CHR$(239);CHR$(239
);
2540 IFWERT(S,11)>0THENPRINT@328
,WERT(S,11);ELSE IFWERT(S,11)=-1
THENPRINT@329,CHR$(239);CHR$(239
);
2550 IFWERT(S,12)>0THENPRINT@360
,WERT(S,12);ELSE IFWERT(S,12)=-1
THENPRINT@361,CHR$(239);CHR$(239
);
2560 IFWERT(S,13)>0ANDWERT(S,13)
<100THENPRINT@392,WERT(S,13);
2570 IFWERT(S,13)>50THENPRINT@39
1,WERT(S,13);ELSE IF WERT(S,13)=
-1THENPRINT@392,CHR$(239);CHR$(2
39);CHR$(239);
2580 IFWERT(S,14)>0THEN IFWERT(S
,14)<10THENPRINT@425,WERT(S,14);
ELSEPRINT@424,WERT(S,14);
2590 IFWERT(S,15)>0ANDWERT(S,15)
<10THENPRINT@489,WERT(S,15);
2600 IFWERT(S,15)>9ANDWERT(S,15)
<100THENPRINT@488,WERT(S,15);
2610 IFWERT(S,15)>99THENPRINT@48
7,WERT(S,15);
2620 POKE65494,0
2630 RETURN
```

Getting It Together

See page 23

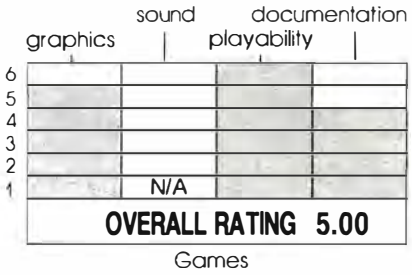
Program Listing. Makedata

```
1Ø CLEAR1ØØØ
2Ø INPUT "ENTER START ADR";S
3Ø INPUT "ENTER END ADR";E
4Ø IF E<=S THEN PRINT"BAD ADDRES
SES":GOTO2Ø
5Ø I=1Ø
6Ø INPUT"TAPE OR DISK (T OR D)";
DV$
7Ø IF DV$<>"T" AND DV$<>"D" THEN
 6Ø
8Ø IF DV$="T"THEN DV=-1 ELSE DV=
1
9Ø IF DV=1 THEN INPUT"FILE NAME
+ EXT";F$ ELSE INPUT"FILE NAME (
8 CHAR MAX)";F$
1ØØ IF LEN(F$)>8 AND DV=-1 THEN
PRINT"BAD FILE NAME":GOTO 9Ø
11Ø IF DV=1 GOTO 14Ø
12Ø PRINT"READY TAPE"
13Ø A$=INKEY$:IF A$<>CHR$(13)GOT
O13Ø
14Ø DA$="":K=65
15Ø OPEN"O",#DV,F$
16Ø IF (E-S+1)<=64THEN L=S:GOTO2
8Ø
17Ø L=64*(INT((E-S+1)/64))+S
18Ø FOR J=S TO (L-1) STEP 64
19Ø FOR A=J TO J+63
2ØØ D=PEEK(A)
21Ø IF D>15 THEN DA$=DA$+HEX$(D)
 ELSE DA$=DA$+"Ø"+HEX$(D)
22Ø NEXTA
23Ø A$=STR$(I)+" D"+CHR$(K)+"$="
+CHR$(34)+DA$+CHR$(34)
24Ø PRINT#DV,A$
25Ø DA$="":I=I+1:K=K+1
26Ø NEXTJ
27Ø IF L>E THEN 35Ø
28Ø FOR A=L TO E
29Ø D=PEEK(A)
3ØØ IF D>15 THEN DA$=DA$+HEX$(D)
 ELSE DA$=DA$+"Ø"+HEX$(D)
31Ø NEXT A
32Ø A$=STR$(I)+" D"+CHR$(K)+"$="
+CHR$(34)+DA$+CHR$(34)
33Ø PRINT#DV,A$
34Ø I=I+1
35Ø PRINT#DV,STR$(I)+"GOSUB1ØØ"
36Ø I=I+1
37Ø FOR J=66 TO (K-1)
38Ø A$=STR$(I)+"DA$=D"+CHR$(J)+"
$:GOSUB1ØØ"
39Ø PRINT#DV,A$
4ØØ I=I+1
41Ø NEXT J
42Ø PRINT#DV,STR$(I)+"GOTO1Ø8"
43Ø I=1ØØ
44Ø PRINT#DV,STR$(I)+"I=1"
45Ø PRINT#DV,STR$(I+1)+"FOR AD=S
 TO (S+(LEN(DA$)/2)-1)"
46Ø PRINT#DV,STR$(I+2)+"D=VAL("+
CHR$(34)+"&H"+CHR$(34)+CHR$(43)+
"MID$(DA$,I,2))"
47Ø PRINT#DV,STR$(I+3)+"POKE AD,
D"
48Ø PRINT#DV,STR$(I+4)+"I=I+2"
49Ø PRINT#DV,STR$(I+5) + "NEXT A
D"
5ØØ PRINT#DV,STR$(I+6)+"S=AD"
51Ø PRINT#DV,STR$(I+7)+"RETURN"
52Ø PRINT#DV,STR$(I+8)+"'CONTINU
E"
53Ø CLOSE#DV
54Ø END
```

Continued from p. 44

Fashion Design With The Designer

by Beth Norman



The Designer is an easy-to-use, clothing-and-fabric design program specifically billed by Cognitive Development as a way to interest young women in computers. The program follows in a tradition of pencil-and-paper dress-design games that have absorbed young people of all ages for generations.

The Designer lets you create clothing for a blank-faced female model who poses in a fashion-model stance. You use the cursor to make selections from pictorial menus of necklines, sleeves, skirts, and pants. You make choices of fabric patterns and colors for each section in turn, designing your own pattern if you like. Once you have created a pleasing outfit, you store it in the model's "closet".

It would take a long time to exhaust The Designers mix-and-match possibilities. Choices of neckline include square and round scoops, turtleneck, lace collar, oxford collar, halter, and two strapless designs. Arms can be sleeveless, puffed, short, clinging, leg-of-mutton style, long, dolman, or cap sleeves. Skirts are available in several lengths and fullnesses. Pants range in style from bell bottoms to jodhpurs to capris. Shorts and culottes of different lengths, bathing suits, and leotard bottoms complete the selection.

When you choose a style or pattern with The Designer, you are not locked into it. The computer displays only the last selection of any piece of the outfits you work on. You can always go back and change your mind about an earlier part of your design.

The Designer offers 26 video pages of "swatches" from which you choose fabrics. You can select solids, stripes, checks, plaids, polka dots, and houndstooth patterns in a variety of sizes—a total of 208 designs. The program uses PMODE 4 graphics, complete with artifact color effects, to give you black, white (buff), red, and green. If the fabric pattern you want is not part of The Designer menu, you can create it. To do this, you change the code that represents any of the standard samples and watch the effects of your changes on the TV screen.

Think of each pattern as being made of five rows of four columns each. The Designer's pattern code consists of five groups of four letters. The letters represent the colors you select for each position. A design of solid black, white, red, and green rows followed by a row of alternating green and white elements looks like this: BBBB WWWW RRRR GGGG GWGW. Patterns repeat over the entire section of the outfit on which you use them.

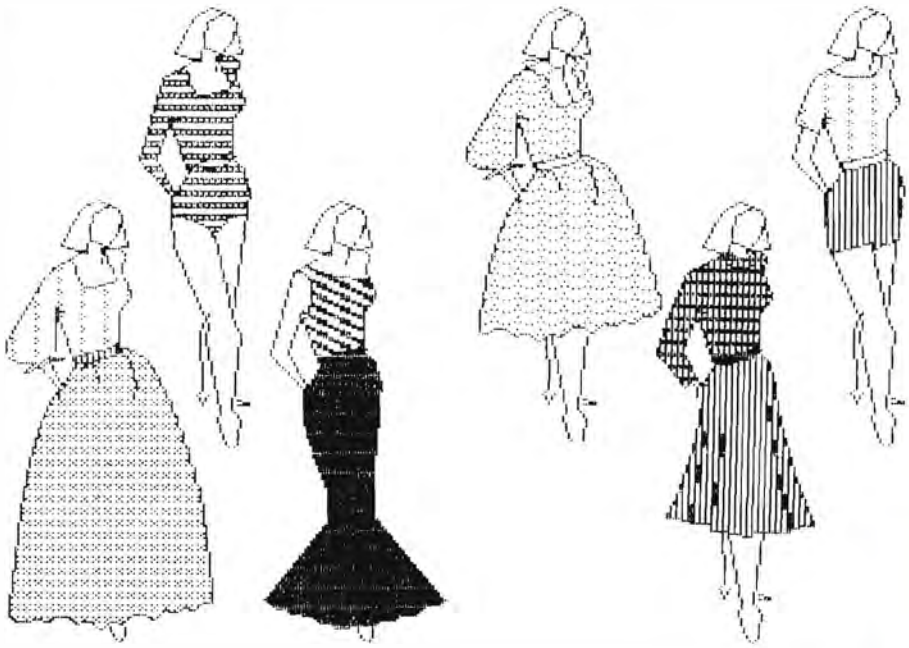
The Designer offers a great deal of room for experimentation. Its instruction leaflet points out that the "technical aspects of the Color Computer and color television" you use, could lead to surprises when you place some color elements beside one another. It is possible to create new colors in this way.

The Designer lets you store 36 of your favorite outfits in 12 "racks", or disk files, of three outfits each. (The program stores the complete image, including the model.) Three racks of completed example designs come with the program. The Designer's "check closet" option on the main menu lets you scan through the files whenever you want.

Some helpful additions to the program would be: a provision for accessories, such as hats, shoes, and belts; a way to choose hairstyles for the model; and an option for modifying or designing necklines, sleeves, and so on. But these are minor points. What I really miss is a way to make printed copies of The Designer's outfits. Although The Designer strives to promote computer literacy among young women, I think that at least part of its audience might enjoy coloring or cutting out creations once they have been computer designed.

Although the program does not have a print routine, there is a way to get its creation off the TV screen and onto the page. The Designer's picture files are recorded in a conventional binary format that at least one of the CoCo's powerful graphic's utilities can read. (I used Four Star Software's CoCo Paint.) Other graphics processors will probably have this capability, too. Along with providing a way to get a printout, using one of these programs lets you touch up any part of a drawing you like. You might even give the model a face.

Except for a few minor points, I think The Designer is an excellent program. It has an original idea and should provide a creative outlet for the young people who use it. The Designer is well worth its purchase price. ■

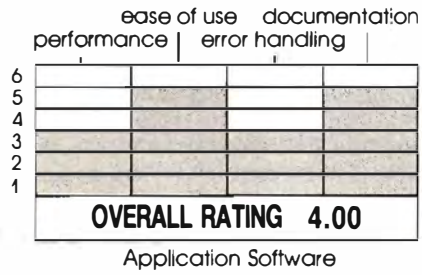


The three outfits on the left are included in The Designer's sample wardrobe. The three on the right are the reviewer's originals.

The Designer is produced by the Cognitive Development Co., Suite 141A, 12345 Lake City Way, N.E., Seattle, WA 98125, 206-382-6661. It requires 32K and a disk drive. It sells for $24.95.

Making The Color Connection II For OS-9

by Jeffrey S. Parker



The Color Connection II for OS-9 from Computerware (for OS-9 version 1.0) provides a versatile smart-terminal program that takes advantage of the features of the Unix-based operating system. The Color Connection II is a menu- and submenu-oriented system that does not have built-in help menus. However, it operates quickly and easily once you learn its logical commands. This program is one of the fastest terminal programs I have tested for the Color Computer.

The Color Connection II has a well-organized and easy-to-understand manual. It takes you through the menu options in a step-by-step manner, covering how to configure the communications settings, how to access the OS-9 shell, how to implement direct-shell commands from within the program, how to do uploads and downloads, and so on. It also contains several helpful appendices, including a summary of the ASCII characters, a troubleshooting chart (which gives probable causes and solutions for common difficulties), a quick-reference summary of the program's operations, and a glossary of telecomputing terms.

Drawbacks

The Color Connection II has several drawbacks, one of which is the size of its buffer. The terminal is configured to use only a 5K default buffer, from which several hundred bytes are deducted for the overhead required by the program. The manual claims a maximum buffer size of about 30K; I was able to install a 28K buffer. Another drawback is that The Color Connection II is preset and limited to 300-baud communications. If you do not use 1,200-baud communications frequently, this program might be a good choice.

Computerware notes in the manual that the program has a bug that causes a partial loss of transmitted characters if the keyboard is used while information is being received from a host computer. This is an annoying problem when you want to transmit an escape character to tell a host computer to stop transmission. In my tests, host computers could not understand the transmission of escape characters because they were garbled. Computerware is working on correcting this problem for the next release of The Color Connection II.

Another shortcoming of The Color Connection II is its inability to download machine-language programs. Although the terminal can read a machine-language file into memory, it cannot download a program with its memory addresses and have it remain intact.

The last drawback to the smart terminal is a command in its main menu that exits the program to OS-9 immediately, without an error prompt. Any information you might have stored in the buffer is lost at this point. You must be careful not to select that option unless you are ready to clear the buffer and leave the program. An error option, giving the user the chance to abort the exit to OS-9 before it takes place, would be a help here.

Advantages And Conclusions

The Color Connection II runs very rapidly. It is easy to progress through its menu options and manipulate the data it contains. The Color Connection II is easy to learn and use. It lets you type directly into the buffer, which means a savings of on-line time and money. You can type in a message before getting on line and then execute a buffer dump with two keystrokes once on line.

It is possible to upload and download to a variety of machines with The Color Connection II. It lets you set parameters for uploading or downloading to a given host computer on a separate menu. A particularly nice feature of the terminal is its support of XON/XOFF protocol.

Because of its simplicity, speed, and efficiency, The Color Connection II for OS-9 should be a serious consideration for anyone who spends a great deal of time on BBSes. It provides a way to store, upload, or download data in OS-9 without the added requirement of printing them out while on or off line from within the program. The Color Connection II has several advantages and disadvantages. You'll need to weigh them carefully. But if you can live with its sometimes annoying drawbacks, the program has a lot to offer. ■

The Color Connection II for OS-9 is manufactured by Computerware, Box 668, Encinitas, CA 92024, 619-436-3512. It requires 64K, OS-9, and a disk drive. It sells for $39.95.

Get Your *Hands-On Basic*

by Jeffrey S. Parker

Hands-On Basic takes an approach to learning Basic that is different from most introductory textbooks. This book is designed for people with no computer experience and no mathematical aptitude beyond the ability to perform addition, subtraction, division, and multiplication. According to Herbert Peckham, you can use the book either with minimal supervision or as a self-study guide. *Hands-On Basic* is targeted at the junior-high-school level and above.

The authors have organized the chapters of this book in unique six-section units. The first sections or units state the objectives of each chapter. The second sections contain "discovery exercises." These let the student use the computer to experiment with the concepts the book discusses. They are designed to build on each other, encouraging the student to experiment further. The third units of each chapter are discussion sections, which describe the workings of various functions and commands in detail. The fourth units go directly into applications with programming examples. The fifth sections contain sample problems and practice tests.

Chapters 1 to 6, which teach the rudiments of Basic, incorporate the programming examples and practice problems with the discovery exercises until the reader has enough background to continue to a more advanced level. The book supplies answers to odd-numbered questions and to the practice tests so that students can check their own progress. A well-defined glossary of terms and an index provide good cross-referencing.

On the whole, this format is far more student-computer interactive than most texts that attempt to teach Basic. Because students actually input data and come up with results, they are stimulated to explore further. The text is very clear, straightforward, and refreshingly easy to read—not technical or stuffy. The authors incorporate a little dry humor, particularly in difficult spots, and this helps readers to progress through the text: "The only difficulty with exponentiation is the intimidating word used to describe the process. . . ."

Hands-On Basic does have a few drawbacks. The first is that the thrust of the book lies in teaching the student what to do with Basic and how to do it, but it doesn't always explain why. For example, the book contains a lesson on combining sound and graphics, but does not mention that you could apply this knowledge to education or game programs. For all of its many good applications, this book is more of a how-to book than a textbook.

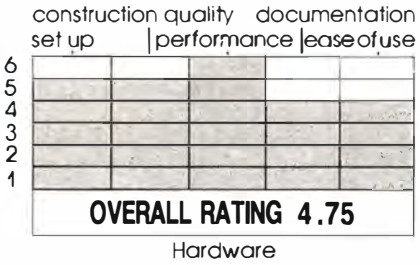
A second drawback involves hardware. The book pays much less attention to 16K machines and cassette drives than it should, given the large number of people—particularly those just getting started—who learn on less powerful, cassette-based systems. This orientation toward larger memory, disk-based computers seems to be aimed at the established education market, despite Peckham's claim that the book is ideal for self-educating students.

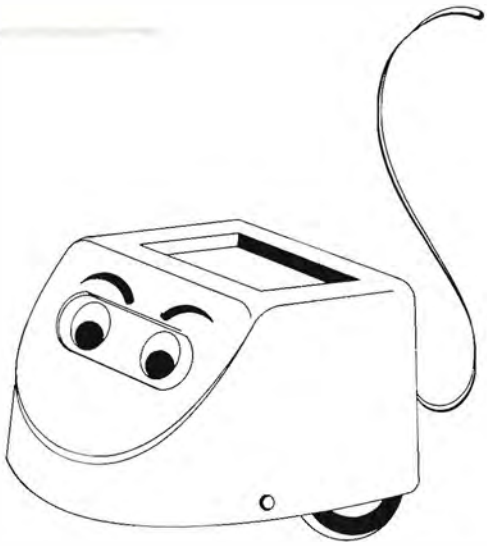
Hands-On Basic is well organized, well written, and easy to read. Students should pace themselves with the book, not moving onto the next chapter until they firmly grasp the present lesson. It is a good idea to secure the advice and guidance of a teacher or a more experienced user in conjunction with this book. ■

Hands-On Basic was written by Herbert Peckham, Wade Ellis Jr., and Ed Lodi. It is published by McGraw-Hill Books, New York, NY 10016, 212-512-4100. The book has 343 spiralbound pp. and sells for $22.

Nomad's Land

by James J. Barbarello





Nomad is a roving robot from Frank Hogg Laboratories that uses an ultrasonic range finder with a resolution of .2 of an inch to determine where it is in relation to other objects. Nomad connects to your Color Computer through a ROM-pack case containing a custom PIA (peripheral-interface adapter) board. The ROM-pack is connected to the robot by a 25-foot cord. Nomad gets power from a 24-volt dc power pack that plugs into the ac line. It measures 8-inches long by 7-inches wide by 5-inches high. Nomad also has a cargo area that is 4¼ by 3 by ¼ inches. Nomad can move in precision steps as small as .04 of an inch. It can also travel as fast as 17 inches per second and climb a slope as steep as 40 degrees.

Documentation and Setup

Nomad comes in a well-packed box with markings that reveal its manufacturer, the Genesis Computer Corp. It contains unpacking instructions and the documentation. The 22-page manual is well organized. It has a table of contents, a "getting started" section, explanations of Nomad's commands, a schematic diagram of the ROM interface, tips on programming Nomad in Basic, and information for programming Nomad in machine language and changing its hardware address in case it conflicts with other hardware you are using on your CoCo. The manual also provides listings of the Basic example programs that come with Nomad. One feature that would be helpful is a complete list of technical specifications (for example: Nomad's schematic diagram, parts placement descriptions, and parts list).

Nomad's setup is simple to complete and is described fully in the documentation. You plug the ROM cartridge into the CoCo's ROM port, turn on the computer, and plug the power pack into an ac outlet. Place the Nomad cassette in your deck and type CLOADM"NOMAD". Once the program loads, you'll have several new commands and functions that you can use from Basic.

Nomad's documentation advises you to hang the cord connecting the robot to its ROM pack from the ceiling, leaving slack for manuevering, to prevent it from tangling. This is a good idea. If you don't do this the cable could get caught under Nomad during circular movements.

Hardware

Nomad and its ROM pack are well-constructed pieces of equipment that use double-sided, glass-epoxy, printed-circuit boards. The edge-connector pads on the ROM pack are gold plated. My preference runs toward thicker printed-circuit lands. I also found two "fixes," or components tack soldered to the underside of the board, in one of the two units I examined. Although this is undesirable, it won't affect Nomad's operation, and I suspect that Frank Hogg Labs has stopped using fixes on their later units.

Nomad's drive mechanism consists of two 7.5-degree stepping motors controlled by electronics. Stepping motors are high-precision devices that contribute a significant amount to the cost of Nomad's manufacture. Nomad's stepping motors use gears to drive two hard rubber wheels. A small, hard-plastic ball mounted in the front of Nomad helps keep it level. The robot's $250 selling price is not unreasonable in light of the high quality of the components that go into its construction.

Nomad's ultrasonic ranging system is a significant plus. The robot contains two ultrasonic transducers (a transmitter and a receiver) driven by on-board electronics. The system begins with a free-running oscillator that produces a short pulse approximately 10 times per second. One transducer transmits the pulses and the other takes in the sonic rebounds that echo back from nearby objects. The computer determines the distance between Nomad and the objects it detects by measuring the time between pulse transmission and reception.

This is a sophisticated system, but as the manual states, it cannot always be accurate. There is considerable spread of the transmitted beam. This is not a problem in the case of a flat wall, for example, but if there is an object off to one side, Nomad could focus on it, throwing off its reading of the wall. Nomad can also misjudge its distance from a slanted wall. Sounds bouncing off such a wall do not return to the robot in as direct a path as do rebounds from a perpendicular wall. If the angle is great enough, Nomad's receiving transducer might not pick up the echoes at all. Nomad receives echoes from the walls that form a corner, not from the corner, itself. This causes Nomad to anticipate shorter-than-realistic distances from corners. Despite these shortcomings, Nomad's ranging system is a useful feature that gives it an advantage over similar machines.

Software

Nomad comes with two kinds of software. One is a machine-language program that uses about 1K of RAM and adds several keywords to Basic for controlling the robot. Once loaded and executed, they become part of Basic's vocabulary. The second is a set of example programs in Basic that demonstrate the robot's features.

Nomad's machine-language program adds four new commands to Basic: NOMAD, SPEED, ACCEL, and NCONV. You use the NOMAD command to control direction. Typing NOMAD 10,45, for instance, instructs Nomad to move 10 inches forward while rotating 45 degrees to the right. Negative values produce left turns and reverse movements. The SPEED command tells Nomad what speed to maintain in inches per second. SPEED lets you vary Nomad's speed between 0 and 17 inches per second. The ACCEL command controls Nomad's acceleration. The manual suggests typing ACCEL 10000 to make Nomad accelerate as fast as it can. The NCONV command lets you change conversion constants. You can use this command to instruct Nomad to interpret the distance parameters you specify in the other commands as centimeters instead of inches.

The machine-language code that comes with Nomad also provides some helpful functions. The RANGE function tells you Nomad's distance from objects and lets you direct Nomad's movement to a specified distance from an object. The CALRANGE function lets you calibrate the range function so that you can compensate for differences that result from wheel slippage, for example. The SAVNCONV function lets you save your modified NCONV values. You do not have to use the machine-language program to control Nomad. The manual has a section that describes how to use your own Basic programs to control Nomad.

Considering the price and experimental nature of this device, I think that Frank Hogg Labs' documentation should have a source listing of the machine-language program that adds Nomad's commands to Basic.

Nomad comes with several Basic example programs on cassette, most of which require Extended Color Basic's graphic commands. Descriptions of some of these programs follow. RADAR/BAS uses Nomad's ranging capability to display a continuously updated "radar" scan of the robot's surroundings on your computer's screen. WANDER/BAS causes Nomad to wander aimlessly and avoid objects that it encounters. TRACKRNG/BAS directs Nomad to attempt to maintain a specified distance from an object in front of it, such as your hand. This is an entertaining program that makes good use of Nomad's ranging capability. GOTAXY/BAS lets you move Nomad to any x,y position within a coordinate range that you specify.

REMOTE/BAS simulates the use of a robot in a hazardous or remote environment. Nomad draws a picture of its surroundings while you watch it on the screen and direct it with the arrow keys. I commanded Nomad to back up a long way with this program and received an FC (illegal function) error. Although this is a minor problem, the program should have better error trapping.

Performance

Nomad performs to its specifications. The manual makes you aware that many factors can affect performance. For instance, the command NOMAD 0,360 does not produce the same results as FOR I = 1 to 10:NOMAD 0,36:NEXT 1. Small errors due to slippage and inertia accrue with each of Nomad's stops and starts causing the FOR. . .NEXT loop to produce a range of movement that is different from the shorter command. The smoothness of the surface over which Nomad rolls also causes varying degrees of wheel slippage. And any obstacle, such as a ½-inch step up, can stop movement, leaving its wheels spinning in a futile manner. For proper operation, you need a uniformly smooth, wide open space.

One problem with Nomad that shows up when the pivot ball that balances the robot's wheels begins to wear is an annoying squealing sound on tile floors. But this is a minor point. Nomad performs consistently well.

The Acid Test

Frank Hogg Labs claims that Nomad places you "on the leading edge with advanced experiments in robotics, artificial intelligence, and real-world applications." This is an overstatement. While the TRACKRNG demonstration does simulate a low level of artificial intelligence and the REMOTE program simulates a possible real-world application, they are far from advanced experiments in robotics. Nomad's best asset is that it can be useful in specific applications, such as education. Objects that move in the real world are always interesting to children. I decided to find out just how interesting Nomad is to kids.

To do this, I sent Nomad to school. It debuted in my daughter's second-grade class. The test consisted of a 45-minute demonstration followed by a question-and-answer period. Nomad was extremely well received by the kids. Their interest and participation showed that Nomad can fare well in the school environment. But second-graders are not the most critical of audiences. Later in the day I tried Nomad in my older daughter's sixth-grade class.

If you're not already aware of it, sixth graders can be absolutely merciless when it comes to critiquing something. But Nomad passed with flying colors—they were entranced. The questions began as soon as Nomad started moving. Without a dissenting voice, all (including the teacher) indicated that Nomad could be a long-term friend in their classroom. This is a class that has used a "turtle tot" with drawing and Logo capabilities in conjunction with an Apple computer. I finally had to end the demonstration because we ran out of time.

While most of the students would have preferred that Nomad have a Logo-like user environment and an attachment for drawing in the tradition of other turtles (Nomad resembles the turtles produced by other companies for this purpose), they were fascinated by Nomad's ability to "see".

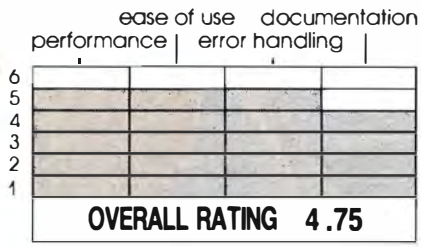
Summary

Nomad is a good buy if you have a specific application in mind. I recommend it highly for educators with programming experience. Although you can use it right away. more software is needed to make it viable for classroom use, especially because it does not use Logo. Frank Hogg Labs has taken an important first step in introducing Nomad. I suggest that the company develop more software and find a way to give Nomad the capability to hold a drawing pen. Despite these shortcomings, the ultrasonic ranging capability is a unique feature among turtles in the Color Computer market that is sure to intrigue most anyone who has the chance to use Nomad. ■

Nomad is marketed by Frank Hogg Labs, The Regency Tower, Suite 215, 770 James St., Syracuse, NY 13203, 315-474-7856. It requires 16K and runs on cassette or disk (a multiple ROM-pack interface or a Y cable is required for disk-drive operation) in Color Basic. Some of the demonstration programs require Extended Color Basic. Nomad sells for $250 plus $3.50 for shipping.

Cool Off With Cbreeze

by Stuart Hawkinson



Application Software

Cbreeze is a full-screen editor from PBJ for the Color Computer programmer who uses OS-9. The program matches the capabilities of PBJ's Word-Pak 80-character by 24-line video-display hardware. It can also be used with several software video drivers for OS-9. Cbreeze provides most of the editing features that programmers require. Once you use this well-designed screen editor, you won't want to go back to OS-9's line editor.

Full-Screen Editing

Cbreeze is perfect for programmers and OS-9 buffs. It lets you drive effortlessly around the screen with the arrow keys. Single-key (and clear key) commands offer insert, delete, truncate, and duplicate-line capabilities. Shifted arrows move the cursor to the start or end of a line and scroll pages up or down.

The top line of Cbreeze's screen gives file and error information. It is separated from the text area by a broken line. The bottom line is used for longer command-mode input. To switch from screen editing to command mode, you press the break key.

Cbreeze has 21 lines for text editing. The display is designed for an 80-column screen, but the program works with some 24-line by 51-column video drivers. You can actually edit lines of more than 500 characters by scrolling the window left or right.

A unique feature of Cbreeze "remembers" indentation from the previous line. Typing structured programs that have indented loops and case statements with the program is a breeze. When you are typing Assembly programs, it keeps op-code mnemonics in a straight line

I used the autoindent feature to outline this review. It was more convenient than my standard word-processing software. But I missed one important feature. Cbreeze does not switch between insert and overstrike modes; it is always in the overstrike mode. To insert text, you must open up a line by pressing the clear/up-arrow key combination several times—once for each extra space you need. This is frustrating.

Split-Screen Editing

Editing multiple files is a hacker's delight. It is a feature that Cbreeze supports, limited only by the CoCo's screen capabilities. You do not get variable-size windows, such as those you find on Apple's Macintosh, but you can split the screen horizontally among as many as four edit buffers. You move from one window to another via the command line by typing the window number.

You can also move or copy lines from one window to another. This is a natural for programmers who need to replicate code segments in various modules. Moving text between files makes writing documentation from program headers a simple process.

The program is limited to four files for a practical reason. Each edit window has its own information, divider, and command lines, leaving only two lines per window when four files are open.

Command Mode

The command mode offers a wide range of editing functions. You move through the text with line- and page-scroll commands. You can also locate and change strings (not multiple occurrences with a single command). Another option is to "jump" to a string in the text. This is a powerful way to move directly into screen editing at the point you require.

For multiple-command execution, Cbreeze provides several new function keys. You can type CTRL/A in OS-9 to execute a command repeatedly. Cbreeze has keys for the last command and the next to last command. It also lets you exchange the two. These features allow you to execute multiple locate/replace sequences by using only two keystrokes per substitution.

It is a simple matter to manipulate files from Cbreeze's command line. You load a new file (append to current text), change default names, save a file, and exit from the command line. Cbreeze lets you take advantage of some of the power OS-9 offers. You can enter the command shell of OS-9 by typing a dollar sign before the command. This gives you access to all OS-9's utilities from your text file. You can list a directory, copy a file, or format a new disk from Cbreeze.

The Cbreeze Package

Cbreeze comes on an OS-9 formatted disk. You copy the program to OS-9's CMDS (commands) directory. It occupies approximately 12K when loaded into memory. That leaves 18 to 28K (depending on the video driver and other custom modules you use) for editing. The program warns you when the edit buffer has been filled so that you can split files before you have a problem.

The documentation for Cbreeze is a 20-page leaflet. All the program's commands are detailed in the first section. The second section contains two tutorial lessons that lead you through each command's use. Sample text files are provided for these exercises. A summary table of commands makes a handy reference. I noticed that the load command and the block-move markers were missing from this table.

An Experiment

Cbreeze was designed to work with PBJ's Word-Pak II 80-column card, and it works well with that product. I decided to experiment with some software-generated, high-resolution screens: Frank Hogg Laboratories' O-Pak and Bee Color Computer & Software's SKIO. Cbreeze does fine in the 51-column by 24-line environment of O-Pak. But there were problems concerning the find command.

Cbreeze highlights strings that it finds in reverse video. But when you use it with O-Pak, the rest of the display turns completely black. The find command works fine with SKIO but does not invert located strings. The shell commands do not execute from SKIO's high-resolution screen, and the delete-line command does not function correctly. Aside from these incompatibilities, Cbreeze performs surprisingly well with these two software-generated high-resolution screens.

Summary

Cbreeze is an excellent editor for the OS-9 programmer. It can also perform light writing chores as a sideline. The program makes full use of PBJ's Word-Pak II, 24-line by 80-column video-display hardware. It is also somewhat compatible with software-generated, 24-line graphics displays. Cbreeze is a winner for simplicity and ease of use. It combines many of the features you need most in a program editor with multiple-source files. If you get a little hot about OS-9's line editor, Cbreeze is a refreshing remedy. ■

Cbreeze, version 2.0, is available from PBJ Inc., P.O. Box 813, N. Bergen, NJ 07047, 201-330-1898. It requires 64K, OS-9, one disk drive, and Word-Pak or Word-Pak II. It sells for $39.93 plus $2.50 for shipping.

by Cy Tymony

CoCo Mechanic

Simple CoCo repairs are no trouble at all.

Repairing your CoCo is not as complex as it seems. In fact, if you know the proper voltage for a particular circuit, it is like testing any other electrical device. And the benefits of peeking into your machine to familiarize yourself with its components are enormous.

One obvious advantage is the savings on labor charges. You can also get bargains on damaged CoCos and peripherals. A friend of mine purchased a 32K Extended Color Basic Computer for only $8. The problem was two bent keyboard-connector pins on the printed circuit board. With a few minutes work, he increased his investment 30 fold without spending money on additional parts.

Not every repair is that rewarding, but it does pay to learn how your CoCo operates. With the right knowledge, you can make simple repairs to your printer, modem, and disk drive, too. You might even repair friends' machines for a profit.

Tools

The most important tools are knowledge of the principles of electricity and familiarity with electronic components. Such information is beyond the scope of this article, but there are many books available in libraries and bookstores. My favorite book on introductory electronics is *Getting Started in Electronics*, by Forest Mims III. (Radio Shack no. 276-5003).

Necessary repair equipment includes a digital logic probe, a volt-ohm multitester, Phillips and standard screwdrivers, and needlenose pliers. An oscilloscope is helpful. Opening your CoCo's case voids the warranty, so let your Radio Shack Service Center make repairs if your machine is less than 90 days old.

Prevention

The best maintenance is preventative maintenance. Keep your CoCo cool by placing it in a well-ventilated area. Remember to shut it off. If you use it for hours at a time, invest in a quiet fan that attaches to the computer's case.

Keep articles and liquids away from the computer's vent slots. Carefully install and remove ROM (read-only memory) packs with the machine off. If you add hardware, do not exceed the rated output of available current.

Replacement Parts

You can find capacitors, resistors, diodes, transistors, and common digital ICs (integrated circuits) for the CoCo at local electronic stores. There are five, LSI (large-scale integration) ICs: the above 6809E CPU (central processing unit), 6847 VDG (video display generator), two 6821 PIAs (peripheral interface adapters), the Color Basic ROM 68A364, and Extended Basic ROM Expansion IC. Order them through Radio Shack or through companies that advertise in electronics or computer magazines.

How The CoCo Works

Understanding the CoCo helps focus your attention on the appropriate subsystem in the machine during troubleshooting. The CoCo and the CoCo 2 use a 6809E CPU as the controlling IC—the brain. (See Fig. 1.) The E stands for external clock. The five other 40-pin LSI ICs produce the system time signals; direct orderly execution of instructions; process input; store and output data; and service the computer's cassette, RS-232, and joystick ports.

Basic's ROM contains a preprogrammed body of permanent instructions, which direct the 6809 CPU in processing data from memory or a cartridge. (See Fig. 2.) ROM also supplies a set of functions for Basic's interpreter; they translate English commands into instructions the CPU can understand. If your CoCo has Extended Basic, an Extended Basic ROM allows you to generate high-resolution graphics and take advantage of other convenient features.

The 6883 SAM (system address multiplexer, Fig. 3) IC provides the timing signals for the CoCo's components. (See Fig. 4.) It generates a 14.31818-MHz signal—a voltage level that rises and falls 14.31818 million times per second—to which other components synchronize. This signal is divided by four to produce the standard 3.579545-MHz TV clock signal needed to produce a picture on your TV monitor. The SAM IC also divides

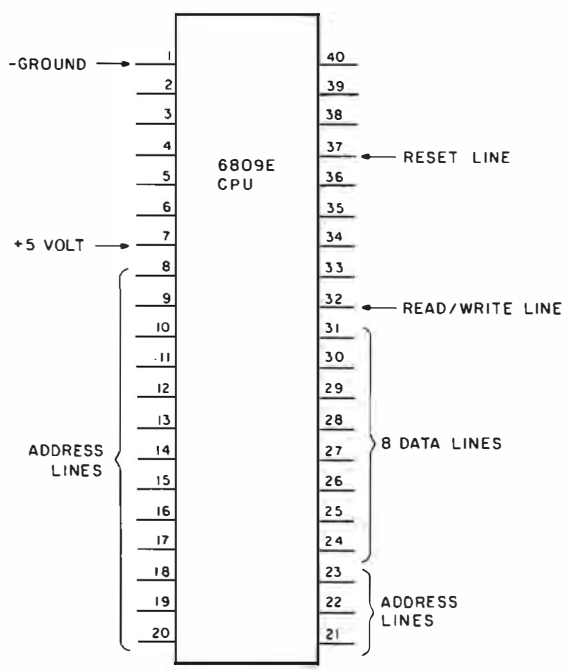


Fig. 1. 6809E Microprocessor

the 3.579545-MHz signal into four again to obtain the .89-MHz clock signal that ensures synchronization between major components. In addition, the SAM IC directs the address lines of the 6809 CPU to the RAM ICs and provides device selection for the CPU.

The 6847 VDG (Fig. 5) IC provides the standard character and graphics set to the RF (Radio Frequency) modulator, producing a video display on your TV monitor. The VDG reads data from RAM's ICs and produces characters or graphic forms from the data in memory. It also generates an external character set if necessary.

RAM IC's (Fig. 6) provide the CoCo with a temporary data-storage area for the CPU during an operation or for you while you write a program. You can read and write data into storage for retrieval later, but RAM must have a constant power source or the data is lost.

The 6821 PIAs (Figs. 7 and 8) convert data from the keyboard, joystick, cassette, and RS-232 serial ports into digital data understood by the CoCo. The PIAs also convert the CoCo's data into standard RS-232 serial output for use with devices, such as printers and modems, that require ASCII (American Standard Code for Information Interchange). The PIAs convert computer data to sound frequencies for storage on cassette tapes. In the CoCo 2, one of the PIA ICs is designated 6822p.

The original CoCo's (grey case) power supply provides a positive 5 volts, negative 5 volts, positive 12 volts, and negative 12 volts electrical output to different components. The ac transformer uses noise-immunity capacitors and a .7-amp fuse for overload protection. (See Fig. 9.)

The CoCo 2 uses a single, positive 5-volt electrical ouput to all circuits. A 300-milliamp fuse filters noise and provides overload protection. There is an unregulated 12-volt electrical output on the circuit board. (See Fig. 10.)

Although components, such as capacitors and diodes, vary on different models, the 40-pin LSIs, are the same on all models. All CoCos process data in the same way, so you follow similar procedures to find faults in components by tracing voltage levels or signals. For a complete parts list, order the technical reference manual for your CoCo. Figure 11 illustrates the lines of communication inside your CoCo.

Troubleshooting: No Image On Screen

- Is the computer receiving ac power?
- Check the transformer fuse.
- Inspect the video cable from the computer to the TV switch box.
- Check the TV switch box.
- Test the supply voltage to the RF modulator. It should be 12 volts (original CoCo) and 5 volts (CoCo 2).
- Check the video signal at the input to the RF modulator.
- Check the operation of the RF modulator

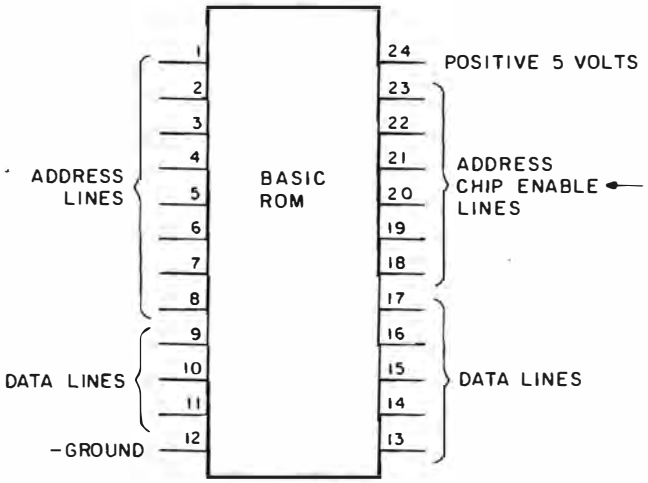


Fig. 2. Basic and Extended Basic ROM

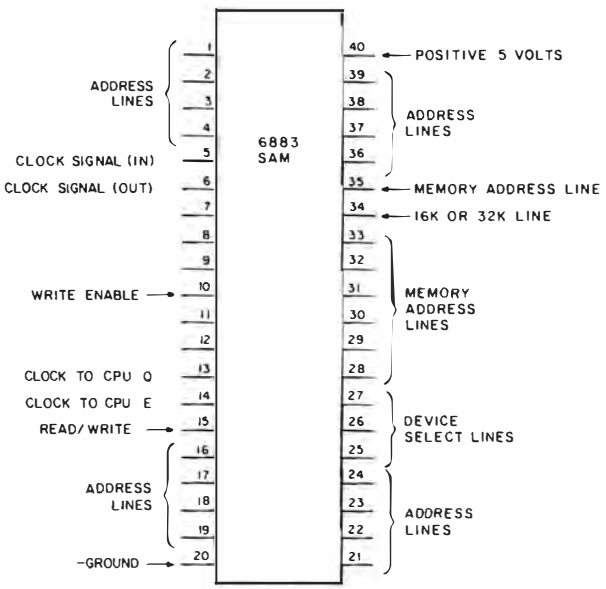


Fig. 3. System Address Multiplexer

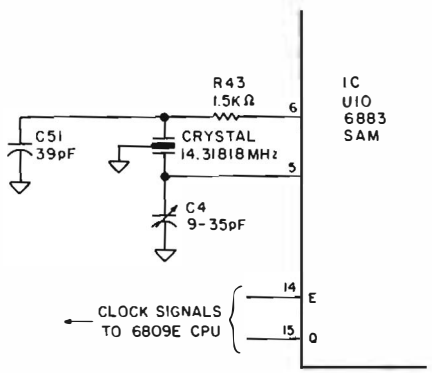


Fig. 4. Original CoCo Clock Generator

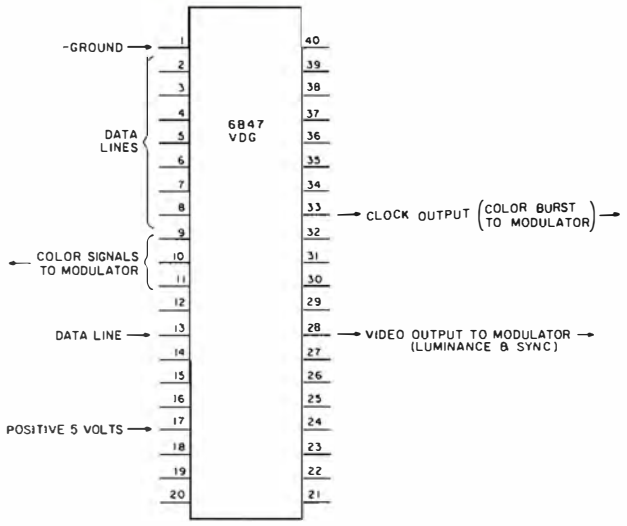


Fig. 5. Video Display Generator

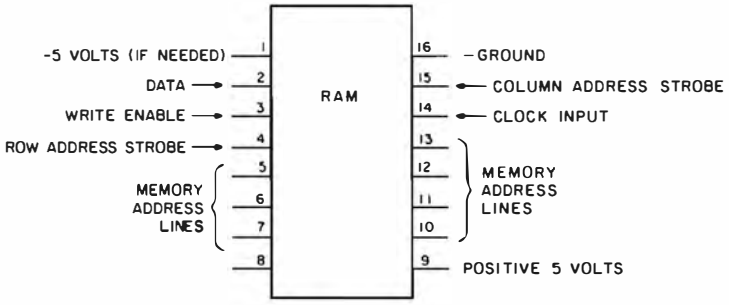


Fig. 6. Random Access Memory

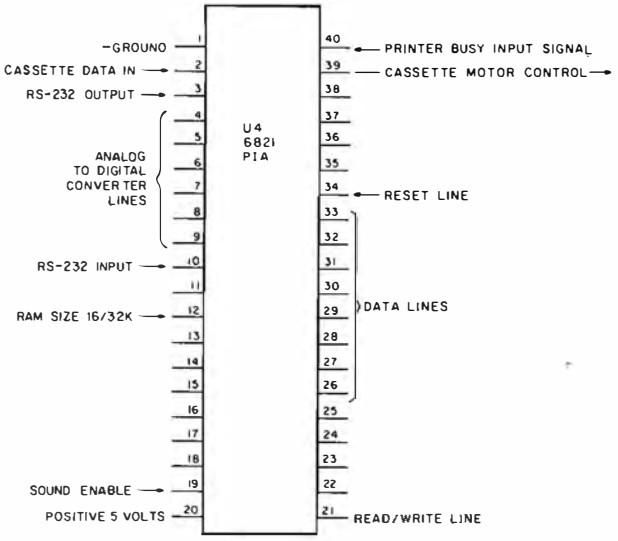


Fig. 7. Peripheral Interface Adapter (U4)

by typing in a SOUND command (SOUND 50,30); listen for a tone from the monitor.

- On original CoCo models, check the video-mixer IC, MC1372 (U12), and the transistor Q1.
- Check for a video signal at pin 28 of the VDG IC. If no signal is present, use a digital logic probe to see if the IC's data lines change as you type different characters on the keyboard.

Keyboard Does Not Output Display

Are all keys nonfunctional? If the problem is limited to one or two keys, it is a mechanical problem in the keys' switches in the keyboard assembly. If a row or column is nonfunctional, check the connector cable, socket, and pins on the PC board for proper contact and continuity. If continuity is good, check the PIA IC (U8) for proper supply voltage and output signals on pins 10 through 17.

Garbage On Screen

If the CoCo displays a stationary picture of unrelated graphics or symbols and does not respond to keyboard input, look for a short-circuit on the PC board or for foreign material on the board. Check for bent IC pins that might be shorting out a line.

Check to see if the CoCo's clock is operating from pins 34 and 35 of the SAM 6883 IC (U10). If there are no pulsating signals when you test the pins, check the oscillator components connected to the SAM IC: the 14,31818 crystal, the 1.5K resistor R43, and capacitor C1 and C4. Check for clock signals at pins 34 and 35 of the 6809E CPU IC (U1), and check pin 40 of the IC for a high state. If it is high, the processor is waiting for input from a cartridge in the cartridge connector. Check 39 of the CPU IC for a low condition and pins 2, 3, 4, 40, and 37 for a high state.

Examine the power-supply voltages going to the RAM ICs (original CoCo) and, using your logic probe, check addresses and data lines for pulsating signals.

As long as you have a working clock pulse from the SAM IC, you can trace the address and data lines from the major ICs in the machine to find where the signal terminates.

RS-232 Problems

If the CoCo freezes while printing, check for a positive 3 volts (or more) at the serial port's pin 1. If it is not present, then check the device that should be producing a signal. If there is a signal, check pin 40 of the PIA (U4) for the same signal. If the signal fails to reach the PIA IC, trace it through the comparator circuit—the LM339 IC and its resistors. Replace defective parts.

If a printer or modem does not receive an RS-232 signal from the CoCo, type in a print loop program:

```
10 PRINT#-2, "test mode"
20 GOTO 10
```

Q: Is there a quality software package that promises to give me a wide variety of easy-to-use Color Computer programs? And at a low price?

A: Yes. *instant CoCo*

Ready-to-run HOT CoCo programs on cassette!

There's just no question. ***instant CoCo*** is a terrific way to build your Color Computer software library. Each month, *instant CoCo* brings you 8–12 top-notch, original programs from the latest issue of ***HOT CoCo*** magazine—all on one high-quality cassette. Everything from action-packed games to helpful utilities. And the cost? As little as $8.33 a month when you buy a one year subscription!

instant CoCo is the ideal companion to *HOT CoCo* magazine, because ready-to-run *instant CoCo* relieves you of the frustrating, time-consuming keyboarding and debugging of the published programs. Our technical editors have done it all for you. You'll save hours of time and considerable aggravation! Just drop the tape into your cassette player, and get set to enjoy some great software!

A free bonus program in every issue!

But wait, there's more! On every monthly *instant CoCo* cassette, you'll get one or more previously unpublished bonus programs absolutely ***FREE***. These bonus programs have never appeared in *HOT CoCo*. Complete instructions for the bonus programs are included.

Bonus programs—one more reason why you should be getting *instant CoCo* each and every month!

Supplement your library with "The Best of '84"!

Complete with documentation

There's even a special collector's item edition of *instant CoCo*: "The Best of '84"—29 first-class *HOT CoCo* programs that come through 1984 with flying colors!

Graphics. Utilities. Games. Plus, applications for business, home, and school. "The Best of 84" is the perfect complement to your *instant CoCo* library, and comes complete with an easy-to-use documentation booklet.

Programs of outstanding variety and top quality. The best programs from an entire year of *HOT CoCo*, including:

- "Stock Transactions Tracker"
- "Graphics Builder"
- "Personal Money Manager"

Don't miss this one. Order your "Best of '84" today.

OK! Send me instant CoCo!

To order *instant CoCo* and "The Best of '84", just complete the information on the coupon or attached order card and drop it in the mail. For faster service, call TOLL FREE 1-800-258-5473. In NH, dial 1-924-9471. VISA, Mastercard, and American Express welcome.

Yes! I want ***instant CoCo*** to meet my programming needs!

☐ Please sign me up for a ***one year*** subscription for $99.97 ($8.33 per issue), beginning with this month's issue.

☐ Please send me the _____ issue for $11.47.
Month

☐ Please send me **"The Best of 84"** for $16.47.

☐ Payment enclosed ☐ MC ☐ VISA ☐ AE

Card # Exp. Date

Signature

Name

Address

City State Zip

Please allow 4-6 weeks for delivery
Price includes postage and handling
Foreign Air Mail please add an additional 45¢ per cassette, or $25 per subscription.
US funds drawn on US banks ONLY.

instant CoCo
80 Pine Street • Peterborough, NH • 03458

Using a logic probe, test for a pulsating signal at pin 3 of the PIA (U4). Trace this signal to the serial port of the CoCo. In original CoCo models, the signal goes through a 741 IC. Check the signal at pin 2 (input) and pin 6 (output). If the signal fails to reach the output pin, check the values of resistors R27 (100 ohms), R24 (1.5K), and R23 (3.9K).

If you do not receive incoming data from a computer or modem after checking the serial port's pin 1, use a logic probe to check for a pulsating signal at the serial port's pin 2. Check this from the inside the computer so that the cable is attached to the CoCo. If you get a signal there, trace it until it arrives at pin 10 of the PIA 6821 IC (U4). (In the original CoCo, the signal goes through a signal-detection circuit similar to the one from the serial port's pin 1—the LM339 and its resistor. This IC prevents noise or other signals from being passed on to the PIA as a high or low state.)

The CoCo uses only four of the possible 25 signals from a standard RS-232C connector cable, so be sure that the other pins or leads from the cable are connected to the proper level (high or low).

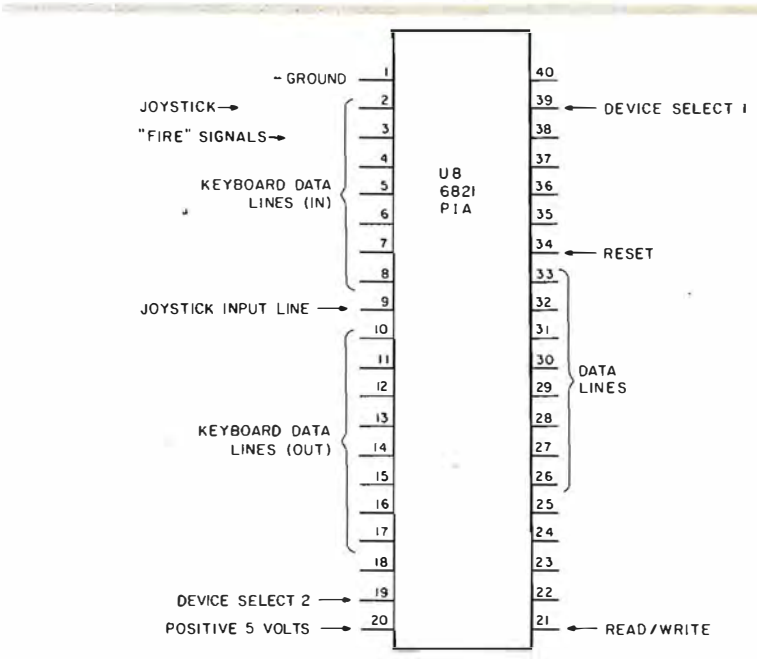


Fig. 8. Peripheral Interface Adapter (U8)

Sound Problems

If no sound emits from the TV monitor's speaker after you enter a sound command, type in AUDIO ON and play an audio tape with the recorder connected to the CoCo. If there is still no sound, try a ROM cartridge containing sound effects. If this does not produce sound either, check pin 35 of the cartridge connector for a sound signal and trace it back to pin 12 of the analog-multiplexer IC (U9). Check pin 13 of the IC for a cassette sound signal and pin 14 for a sound output signal from a Basic command. If you have not resolved the problem, check the sound-enable signal, on pin 19 of the PIA (U4), for a high state. Also check the conditions of the select lines going to pins 19 and 39 of PIA 6821 (U8). If there is a sound signal present at pin 3 of the modulator but no sound from the TV monitor, replace the modulator.

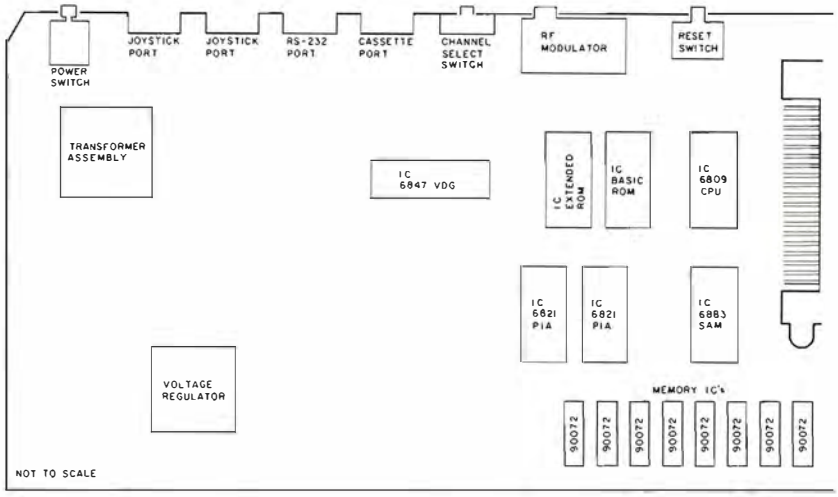


Fig. 9. Original CoCo Parts Layout

Unstable Video On Monitor

When the display monitor's picture is poor, adjust the fine-tuning, and check the switch position at the back of the CoCo to make sure it is tuned to the channel on which the RF modulator is transmitting. If you know the monitor is good, but the picture is still unstable or does not have color, check the values of resistor R43 (1.5K) and capacitor C5L (39pf) in the original CoCo.

Adjust variable resistor R21 (500 ohm) while watching the monitor. If this does not eliminate the problem, adjust the variable capacitor C4 (9-35pf) and note the change on the display. If the unstable picture persists, substitute parts in the video-mixer circuit (IC U12 and its capacitors, resistors, and diodes in the circuit) and in the video-amplifier circuit (transistors Q1 and the circuit's capacitors, resistors, and diodes). Replace the RF modulator if your substitutions are not successful.

Power-Supply Problems

In the original CoCo, check appropriate locations for correct voltage readings. Pin 2 of the cartridge connector and the RF modulator, pin 7 of the 741 IC (U15), and pin 3 of the LM339 IC (U14) should read positive 12 volts. Pin 1 of the cartridge and pin 4 of the 741 IC (U15) should read negative 12 volts. Pin 9 of the cartridge connector, pin 5 of the joystick port, pin 40 of the SAM IC (U10), pin 24 of the ROM and ROM Expansion ICs (U3 and U28), pin 20 of the PIAs (U4 and U8), and pin 17 of the VDG IC 6847 (U7) should read positive 5 volts. Pin 1 of the 4K and 16K RAM ICs (U20–U27) should read negative 5 volts. Trace levels lower than those specified to the individual regulator and rectifier components in each supply.

The CoCo 2 supplies only positive 5 volts to its components, so trace problems to the regulator, diodes, transformer, or the 300-milliamp fuse.

Joystick Problems

Test each joystick by reading the resistance level of pins 1, 2, 4, and 5 while moving the stick through its path of play. There should be rising and falling resistance between 100K and 0 ohms. Press the fire button and check the resistance of pins 3 and 4. The ohmeter should vary from infinite resistance to no resistance.

If the tests go well, but you get erratic be-

havior or no reaction when you use it with the CoCo, check pin 9 or the PIA (U8). As you move the joystick through its path, your input voltage should change. If it does not, check input pins 2 through 5 of the MC14529 IC (original CoCo). Check output at pin 9 for changing voltage levels that coincide with joystick movement.

To evaluate the condition of the fire button, press it and check pins 2 and 3 of PIA 6821 (U8) with a logic probe. There should be a momentary high state as you fire. If you get unusual readings from your circuit checks, examine capacitors C54, C55, C78, and C79 for shorted leads. If all components are fine, you will have to substitute a new PIA 6821 IC (U4 or U8) to correct the problem.

Your Turn

I have given you the basics. Now it's time for you to take over. With the above information under your belt, you should understand most hardware modifications that appear in books and magazines for the CoCo. I recommend that you obtain a technical reference manual for your CoCo so that you can expand your knowledge of this versatile machine. ■

Address correspondence to Cy Tymony, Box 2387, Beverly Hills, CA 90213.

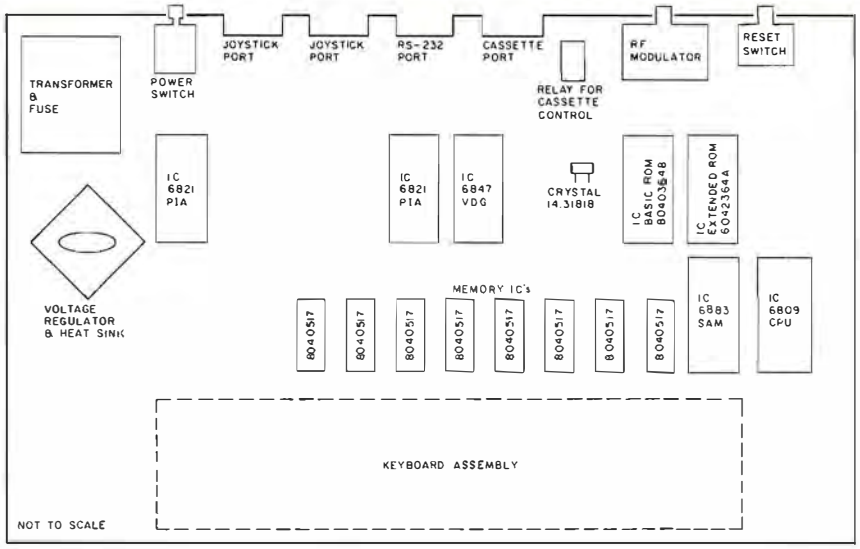


Fig. 10. CoCo 2 Parts Layout

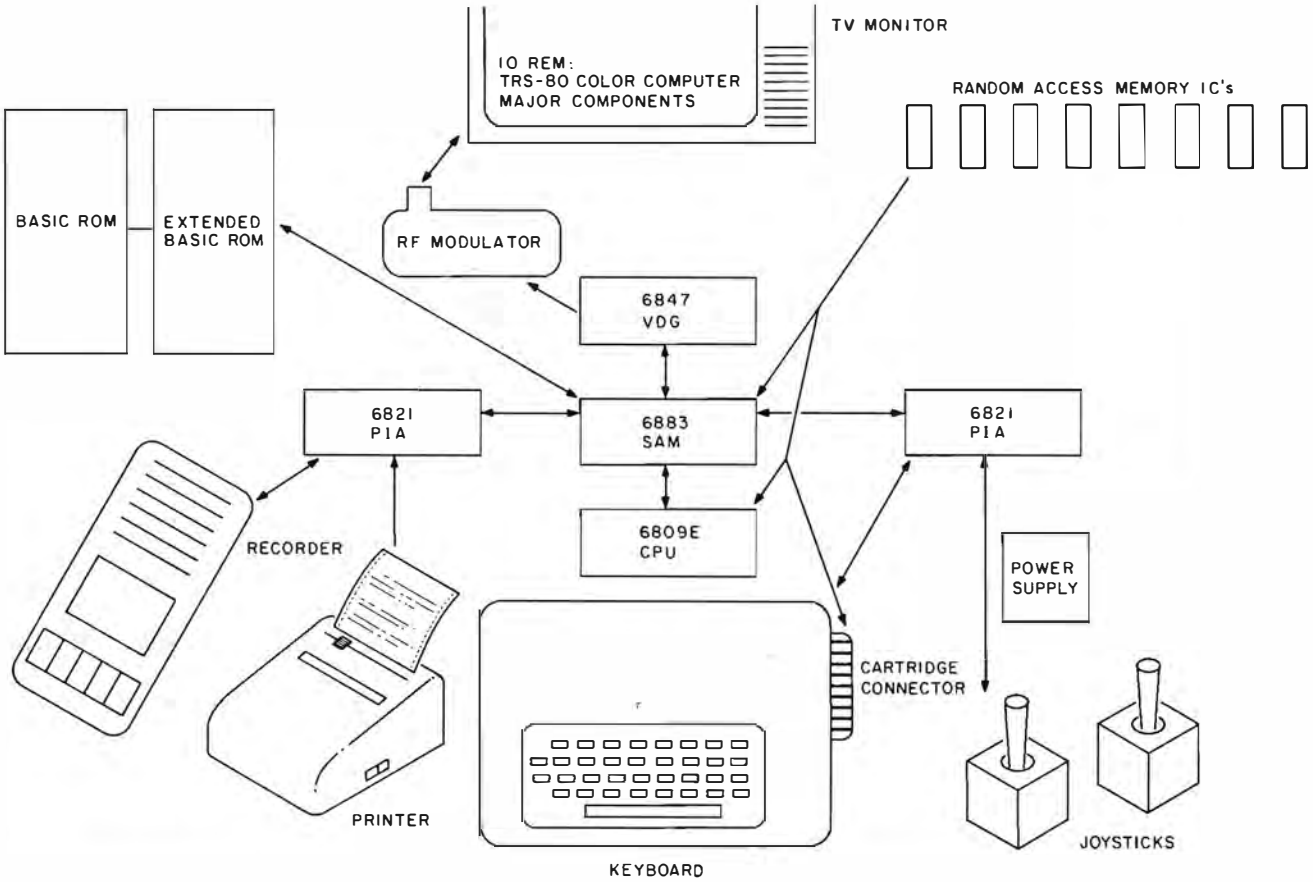


Fig. 11. How Your CoCo Works

Continued from page 32

The variables NumRight and NumWrong are given initial values of zero. The rest of the PROCEDURE GiveQuiz is a FOR. . .DO loop. A FOR. . .DO loop works almost exactly the same as a FOR. . .NEXT loop in Basic. In this case the variable Count is initialized at a value of one and will increase by increments of one to a final value of 10. Thus the loop is repeated 10 times. Either a single statement or a block of statements follows the keyword DO. As before, if a block of statements is to be repeated, they must be between the keywords BEGIN and END.

The loop begins by getting two random numbers to use in the problem. The correct type of problem is determined by the CASE structure, which is another kind of selection structure. This structure always begins:

CASE variable name OF

Following this is a list of the possible values that the given variable may have. Following each value is the program code that is to be executed if the variable is equal to the given value. The CASE statement works like a list of IF. . .THEN statements.

If the QuizType is a 1, then an addition problem is given. Note that the single quotation marks are used because QuizType is a CHAR variable and I mean the character 1 rather than the number one.

If the QuizType is 2, then a subtraction problem is given. To avoid negative results, you have to make sure that the first number is larger than the second. Since the values are random, this may not be the case. The IF. . .THEN statement checks to see if the first number is smaller than the second. If so, the two numbers are switched before displaying the problem.

Following the CASE statement, the READ statement gets the user's response. The IF. . .THEN. . .ELSE statement (a third selection structure) updates the number right or wrong and prints an appropriate message (in either case). The THEN block is executed if the response is correct. Otherwise the ELSE block is executed. This structure is similar to that of Extended Color Basic but with a couple of important differences. In Basic the entire structure must be on one program line, so the number of statements in the THEN and ELSE clauses is limited. Secondly, in Pascal you can write the structure using a format that makes it much easier to follow.

Showing The Results

The final program module prints a message giving the quiz results. Both the number right and the number wrong are reported. Listing 6 is the procedure and consists entirely of WRITELN statements. In this procedure, notice that following the variable name is a colon and a one-digit number:

NumRight:1

Normally, Pascal prints integers in a fixed-width field (typically six characters wide). You can alter the fixed-width field by entering the desired width after the colon. Actually, this value represents the minimum field width. If the value can't be printed in the designated number of characters, Pascal will use enough additional characters to finish printing the value.

I stated a field width of one. If the user gets all 10 answers correct, the program will print the value 10 even though it takes two characters. The reason for adjusting the field width is to improve the appearance of text containing both words and numbers.

```
(*-----------------------------------------
     The procedure ShowResults is called by
     the main driver and displays the number
     of correct and incorrect
responses. -----------------------------------*)

PROCEDURE ShowResults;

BEGIN
  WRITELN;
  WRITELN('OUT OF 10 QUESTIONS');
  WRITELN('YOU GOT ',NumRight:1,' CORRECT,');
  WRITELN('AND ',NumWrong:1,' WRONG.')
END; (* ShowResults *)
```

Program Listing 6. Module to Show Results

Testing And Debugging

If you are used to Basic, expect a little frustration when you start writing in Pascal. Because you have to compile the program, correcting simple errors is a bit inconvenient. You have to reload the editor, read in the text file, make the corrections, and recompile the program. For this reason, it is doubly important to carefully plan and design the program before you write it. Because the program is modular, you can test the modules as you write them. After writing the main driver, write a dummy procedure (or function) for each procedure (or function) called in the driver.

A dummy procedure consists of the normal heading (with the procedure name) but not the actual code. For example, you might include just one WRITELN statement as follows:

```
PROCEDURE GetSeed;
BEGIN
  WRITELN('Got to the GetSeed procedure')
END;
```

You will also have to assign a function a value:

```
FUNCTION UserInput:CHAR;
BEGIN
  WRITELN('Got to UserInput function);
  UserInput : = 'A'
END
```

These dummy procedures and functions will not perform the desired tasks, but they will let you compile and run the program in order to test the logic of the main driver. When you are satisfied with the main driver you can recompile the program to see if the GetSeed module does what you want.

As you begin to program in Pascal, don't be discouraged by the number of apparent errors detected by the compiler. This number is often very inflated. I have seen a program generate 40 compiler errors when in fact only one error existed. Very often a missing or misplaced semicolon will generate a large number of apparent errors; the compiler just gets confused.

When correcting errors, begin with the first detected error. Correcting the first error will often eliminate some of the other "errors" that were caused by the first one. Most important is to be careful writing and entering your programs.■

Delmar Searls teaches programming for beginners. Write him at 205 W. Main St., Wilmore, KY 40930.

by David Meredith

A Look at LOGO

Combine the fun of drawing with the satisfaction of problem solving.

Logo is a graphics computer language simple enough for children; yet Logo is a dialect of the artificial-intelligence research-language Lisp with control structures borrowed from Pascal. It is a complete computer language with more capabilities than Basic, including list and parallel processing.

In the late 1960s, Seymour Papert invented Logo at the MIT Artificial-Intelligence Laboratory to help children explore geometry by drawing figures with a robot turtle. Today many elementary and middle schools use Logo to teach programming to children.

Radio Shack sells two versions of Logo: Color Logo, which has been available for about two years, and a new upgraded version called Super Color Logo (not yet released as of press time), which includes decimal arithmetic and list processing. They are each available on cartridge ($49.95) and disk ($99.95). Network versions and teaching guides are available as well.

Fundamentals

Logo begins with the turtle, an imaginary creature that lives on your TV screen. You move the turtle with commands written in Logo. The most important commands are FD (forward), RT (right turn) and LT (left turn). You can make a square of side 50 with the commands:

```
FD 50
RT 90
FD 50
RT 90
FD 50
RT 90
FD 50
```

Listings 1 through 3 contain some simple Logo procedures and samples of their output. You enter Logo procedures into your workspace on the screen from the keyboard (or tape or disk) and then invoke them by typing their name followed by values. To draw a square of side 50, if Listing 1 is already recorded, you need only type SQUARE 50.

We never speak of a Logo program, only of Logo procedures. Logo is like Lisp or Forth in that your workspace contains not a single program but a group of related procedures. You can run a procedure, enter new procedures or change existing ones, or use the name of a procedure as a command word in another. In Listing 2, for example, the procedure FLOWER uses SQUARE to draw a pattern of squares.

If you want to move the turtle without making a line, you can lift the pen with the command PU and later drop it with PD. Listing 4 gives an example of the use of PU and PD.

Logo uses variables to store numbers. Logo variable names always begin with a colon, and they can be as long as you want. You don't have to remember that D means distance, or wonder if you already used D to mean difference. You can use mnemonic variable names like :DISTANCE and :DIFFERENCE. Mnemonic variable names make Logo procedures easy to read and understand.

Variables are assigned values with the MAKE command, as in:

```
MAKE :CIRCUMFERENCE 22*:DIAMETER/7.
```

Variable values in Color Logo are always integers between −32,768 and 32,767. Super Logo allows you to store decimal values and lists in variables.

Logo also has built-in functions that let you combine values, locate your turtle on the screen, or generate random numbers for games.

Programs in any language are more than sequences of commands. They include control statements for loops and branches. In Basic, there are three control structures: IF. . .THEN. . .ELSE, FOR, and GOTO. Logo includes IF. . .THEN. . .ELSE, but uses WHILE in place of GOTO and REPEAT instead of FOR. Since there are no line numbers in Logo, the code segment included within a control structure is delimited by parentheses. This is a flexible arrangement because the clauses after IF, ELSE, WHILE, or REPEAT can be as long as necessary.

Listings 5 and 6 demonstrate the WHILE structure. In Listing 5, WHILE 1 means continue forever or, since NOWRAP has been executed, until the turtle goes off the screen. Each time through the loop, the value of :EDGE increases by two, so eventually the turtle disappears from the screen and the procedure halts.

Listing 6 uses WHILE more cleverly. The turtle draws a polygon by repeating the commands:

```
FD :SIDE
RT :ANGLE
```

until the polygon closes. It is a theorem, called the total-turning theorem, that the polygon will be closed when the sum of the angles the turtles has turned is an integral multiple of 360 degrees. The octagon stops when the turtle has turned 360 degrees; the five-pointed star stops when the turtle has turned 720 degrees.

Procedures

According to Papert, the procedural nature of Logo is its most important feature. Procedures enable you to solve a problem by dividing it into subproblems and writing a procedure for each. You can even reuse a procedure for a different problem.

When you declare a procedure with the word TO, you can follow the procedure name with variables that are given values when you

invoke the procedure. These variables are called local variables because they are meaningful only within their own procedure. For example, FLOWER passes the value 50 to the local variable :SIDE in SQUARE (Listings 1 and 2).

If two procedures use the same local variable name, Logo creates two variables and keeps them separate. You can choose variable names without worrying about reusing one or upsetting some condition in a distant part of your program.

Local variables are not only local to their own procedure, but they are restricted to a particular invocation of the procedure. Each time FLOWER calls SQUARE, a new copy of :SIDE is created and given the value 50, and each different invocation of SQUARE refers to a different copy of :SIDE. To save memory, the space reserved for :SIDE returns to a general pool of free memory when SQUARE terminates.

Procedures can call themselves, as in Listings 7 and 8. Such procedures are called recursive. Note that each time RSPIRAL calls itself, a new copy of the variable :LENGTH is created for the new invocation of RSPIRAL. The new copy of :LENGTH has a value equal to the current value of :LENGTH plus two. When the turtle executing RSPIRAL 90 runs off the screen, RSPIRAL has called itself 92 times, and 92 copies of :LENGTH are stored in memory.

Listing 8 is more complex. A nested triangle with outside edge S is a triangular arrangement of three nested triangles, each with outside edge S/2. The IF statement prevents infinite recursion, nesting smaller and smaller triangles, *ad infinitum*.

Advanced Features

Logo has two advanced features. Only Radio Shack's Logo has multiple turtles. You can create several turtles and set them working on different procedures simultaneously. The turtles communicate with each other through a message-passing system.

Multiple turtles simulate parallel processing, computing with multiple processors simultaneously. Most authorities predict that the next significant improvement in computer speed will come through parallel processing.

The second advanced feature, list processing, is part of all except the earliest version of Color Logo. A list is a sequence of words made up of strings of alphabetic characters. You can copy a list, insert a new word in a list, test if a word is in a list, or execute a list that is a Logo procedure. In some ways, list processing is similar to string manipulation in Basic, but it is much more powerful.

List processing is the basis of Lisp, the favorite computer language of artificial-intelligence researchers. It is part of Logo because Logo grew out of Lisp at MIT. Since none of us has Super Color Logo yet, I'll say no more about it.

Logo As A Teaching Language

Papert originally intended Logo to be an environment for exploring geometry, but most teachers use Logo to teach programming concepts. Logo is powerful enough to hold the interest of beginning

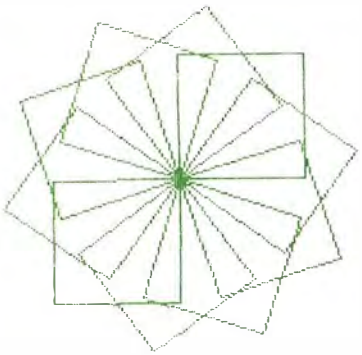
Program Listing 1. Square

SQUARE 50

```
MAKE A SQUARE OF SIDE :N

TO SQUARE :SIDE
  REPEAT 4
  [ FD :SIDE
    RT 90
  ]
END
```

Program Listing 2. Flower



FLOWER 10

```
MAKE :N SQUARES OF SIDE 50
FLOWER 10 IS NICE

TO FLOWER :N
  REPEAT :N
  [ SQUARE 50
    RT 360/:N
  ]
END
```

Program Listing 3. Circle

CIRCLE 200

```
THE MOST FAMOUS LOGO PROCEDURE OF ALL—MAKE A TURTLE GO IN A CIRCLE OF GIVEN CIRCUMFERENCE

TO CIRCLE  :CIRCUMFERENCE
  REPEAT 20
  [  FD :CIRCUMFERENCE/20
     RT 18
  ]
END
```

Program Listing 4. Circle Turtle

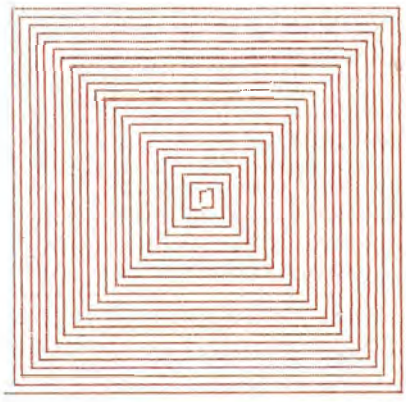
```
MAKE A CIRCLE AROUND A TURTLE OF RADIUS :R. THE IDEA IS TO MAKE A CIRCLE OF CIRCUMFERENCE 2*PI*:R AROUND THE TURTLE. AT THE END OF THE PROCEDURE, TURTLE HAS RETURNED TO ITS ORIGINAL LOCATION WITH THE PEN DOWN.

TO CIRCLE.AROUND  :RADIUS
  MOVE.TO.RIM :RADIUS
  CIRCLE (44*:RADIUS/7)
  RETURN.TO.CENTER  :RADIUS
END

TO MOVE.TO.RIM  :RADIUS
  PU
  FD :RADIUS
  RT 90
  PD
END

TO RETURN.TO.CENTER :RADIUS
  PU
  LT 90
  BK :RADIUS
  PD
END
```

Program Listing 5. Spiral



SPIRAL 90

programmers, yet simple enough to let them program complex figures with a few commands.

More important, students don't outgrow Logo; instead they progress to advanced computer concepts like structured programming, procedures, local variables, and recursion. These concepts are the basis of Pascal, C, Modula 2, and Ada. Students trained in Logo are better prepared for advanced languages than students who learn Basic. The advanced features of Logo even introduce list processing and parallel computation.

Logo, like Basic but unlike most advanced languages, is an interpreted language. Students can write and test sections of code as an application develops. Time-consuming compilations and linkings are not required after each correction.

I've used both Logo and preprogrammed educational software with primary-grade children. Learning programs are fun for kids, but nothing can replace the special satisfaction they get from writing a short Logo procedure and seeing it work.■

David Meredith is a professor of mathematics at San Francisco State University. Address correspondence to him at the Department of Mathematics, 1600 Holloway Ave., San Francisco, CA 94132.

Abelson and diSessa, ***Turtle Geometry,*** *M.I.T. Press, Cambridge, MA, 1981.* A great book on geometry from the turtle's point of view. It begins with projects simple enough for grade-school children and ends with a Logo simulation of the Theory of Relativity (no advanced math required). The authors are associated with Papert's original Logo project.

Color Logo Parent's Book *(Radio Shack no. 26-2763) and* ***Color Logo Teacher's Book,*** *(Radio Shack no. 26-2761), Radio Shack.* These are short, elementary books for adults who want to help children with Logo.

Kheriaty and Gerhold, ***Radio Shack Color Logo.*** This manual comes with Radio Shack Color Logo. It is a good introduction to this version of the language.

Papert, ***Mindstorms,*** *Basic Books: New York, 1982.* Papert's explanation of the philosophy behind Logo.

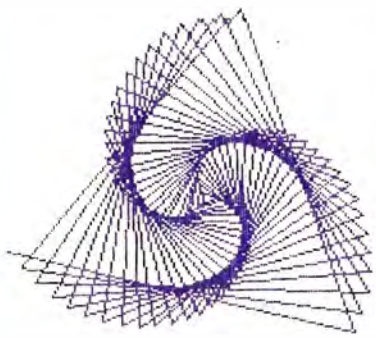
Thornberg, ***Computer Art and Animation,*** *Addison-Wesley: New York, 1983.* Look for the 1983 edition about Color Logo. It contains good ideas for designing computer-art projects.

Table 1. Annotated Bibliography

SPIRAL MAKES A SPIRAL AND STOPS WHEN IT RUNS OFF THE SCREEN. :A IS THE ANGLE OF EACH TURN. TRY SPIRAL 60, 90, 120, 92, 122

```
TO SPIRAL :ANGLE
  NOWRAP
  MAKE :EDGE 3
  WHILE 1
  (  MAKE :EDGE :EDGE + 2
     FD :EDGE
     RT :ANGLE
  )
END
```

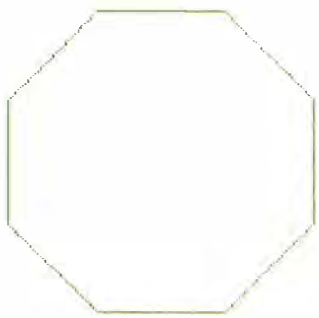
HERE'S A RECURSIVE VERSION OF SPIRAL. :ANGLE IS THE ANGLE OF TURN, :EDGE IS THE LENGTH OF THE FIRST SIDE.



SPIRAL 122

```
TO RSPIRAL :ANGLE :EDGE
  NOWRAP
  FD :EDGE
  RT :ANGLE
  RSPIRAL :ANGLE (:EDGE + 2)
END
```

Program Listing 6. Polygon



POLY 40 45

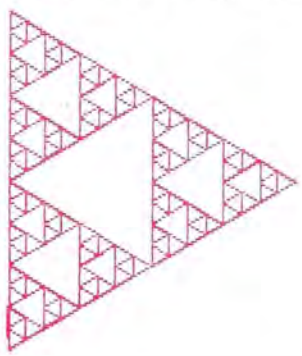
POLY 50 144

POLY 70 160

POLY MAKES A POLYGON OF SPECIFIED SIDE AND ANGLE.

```
TO POLY :SIDE :ANGLE
:REMAINDER
  MAKE :REMAINDER 360
  WHILE :REMAINDER > 0
  (  FD :SIDE
     RT :ANGLE
     MAKE :REMAINDER
       :REMAINDER – :ANGLE
     IF :REMAINDER < 0
     (  MAKE :REMAINDER
       :REMAINDER + 360)
  )
END
```

Program Listing 7. Nested Triangle



NESTED. TRIANGLE 160

NESTED.TRIANGLE DRAWS A NEST OF TRIANGLES RECURSIVELY.

```
TO NESTED.TRIANGLE :SIZE
  IF :SIZE < 10  (STOP)
  ELSE
  (  REPEAT 3
     (  NESTED.TRIANGLE (:SIZE/2)
        FD :SIZE
        RT 120
     )
  )
END
```

GAME

by Volker Meinhardt

Tricky Dice

Enjoy Yahtzee while your CoCo does the scorekeeping.

All you dice players and would-be gamblers will enjoy this game. Tricky Dice is a computerized dice game based on Yahtzee that eliminates tedious, manual scorekeeping and protects you from your impulse to cheat a little.

In Tricky Dice you have three throws of five computer-generated dice. From each of the throws you can keep or reject dice values for your score by moving a cursor.

To Play

Load the program (on 16K machines, first type PCLEAR1 and press the enter key) and run it. State the number of players and type in their names. The computer then shows a scoring list for the first player. To throw the dice, move the cursor below the red dot using the joystick.

When you hear a beeping sound, press the fire button; the computer shows the first five dice. To keep any of these dice, move the cursor below them and press the fire button. An orange cap marks each die you want to keep. Follow the same procedure to remove the block.

Between throws, the cursor jumps beside the scoring card, and by moving it up or down to the appropriate line and pressing the fire button, you record your score. (See Table 1.) For example, if you throw three dice with six spots each, and two with one spot each, move the cursor to line six for 18 points and to line one for two points. Once a score is recorded, you cannot change it.

If you have a bad throw, record it on the line labeled chance, or delete a free line. The computer marks a deleted line in violet. It also displays the throw number and player's name and automatically records the score.

When all score lines are filled, the game ends and the computer displays the winner's name and plays a song before displaying the final scores of all players. To begin a new game just press the fire button. The comment lines in the Program Listing offer a more detailed explanation.

Score Combinations And Lines

The following score combinations and score lines are used:

- Doubles: Three of a kind or four of a kind—the computer records the entire sum.
- Full House: Two of a kind and three of a kind—the computer records 25 points.
- Little Straight: Four successive dice such as three, four, five, and six, for a score of 30 points.
- Great Straight: All dice are successive for 40 points.
- Tricky Dice: Five of a kind gets you a score of 50 points. For every following Tricky Dice combination, the computer adds 100 points. You can write the sum of the dice in the appropriate line. If none is available, use any free line.
- Chance: Record any dice combination and the computer adds the score.
- Top Sum: 63 points or more on this line gives you a bonus of 35 points. ■

1-SPOTS:	_____
2-SPOTS:	_____
3-SPOTS:	_____
4-SPOTS:	_____
5-SPOTS:	_____
6-SPOTS:	_____
TOP SUM:	_____
3 SAME:	_____
4 SAME:	_____
F. HOUSE:	_____
LT. STR.:	_____
GR. STR.:	_____
TRICKY:	_____
CHANCE:	_____
TOTAL:	_____

Table 1. Screen Score Card

System Requirements

16K RAM
Extended Color Basic
Joystick

Program listing on page 50

Address correspondence to Volker Meinhardt, Am Steinicht 3, 8630 Coburg, West Germany.

Index to Advertisers

Advertising Offices: (603) 924-7138 or (800) 441-4403

The advertisers listed in bold print are new advertisers to HOT CoCo this issue.

***This advertiser prefers to be contacted directly. For further information from our advertisers, please use the reader service card.**

Pick Of The Month

The first question on our Reader Service card above asks you to pick your favorite article or feature in this issue. Write the appropriate letter in the space provided on the card. Here is the list to choose from:

A. *Doctor ASCII*, Esposito, p. 14

B. *Mindbusters*, Ramella, p. 16

C. *The Computer Room*, Norman, p. 18

D. *The Facts On CoCo Max*, Norman, p. 20

E. *Getting It Together*, Tipps, p. 23

F. *C'ing Is Believing*, Goodwin, p. 26

G. *A Short Course In Pascal*, Searls p. 27

H. *All Sorts Of Sorts*, Meredith, p. 33

I. *The John-B System, Part I*, Barbarello, p. 38

J. *CoCo Mechanic*, Tymony, p. 61

K. *A Look At Logo*, Meredith, p. 69

L. *Tricky Dice*, Meinhardt, p. 72

M. *Date Minder*, Huckabey, p. 74

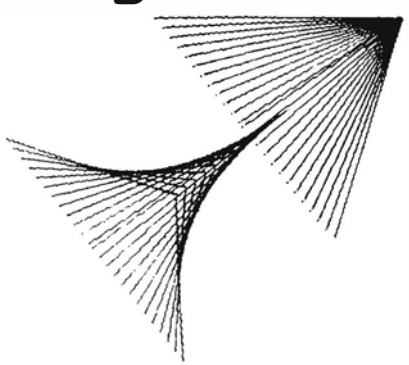
N. *Assembly 101*, Perotti and Perotti, p. 84

O. *6809 On Line*, Ballard, p. 86

P. *Product News*, Finnie, p. 90

Q. *The Learning Page*, Kipperman, p. 94

Coming Next Month



When you talk about CoCo graphics, you can cover a wide range of topics from just plain doodling to pie charts and bar graphs for business. August's *HOT CoCo* takes a look at CoCo graphics from several perspectives.

David Clements describes his program that generates a number of different bar graphs. It is a handy aid for presentations, term papers, or personal research.

Joe Finamore's Symmetry is an interesting study on creating order out of chaos. His program takes a random doodle and duplicates it into startling graphic displays.

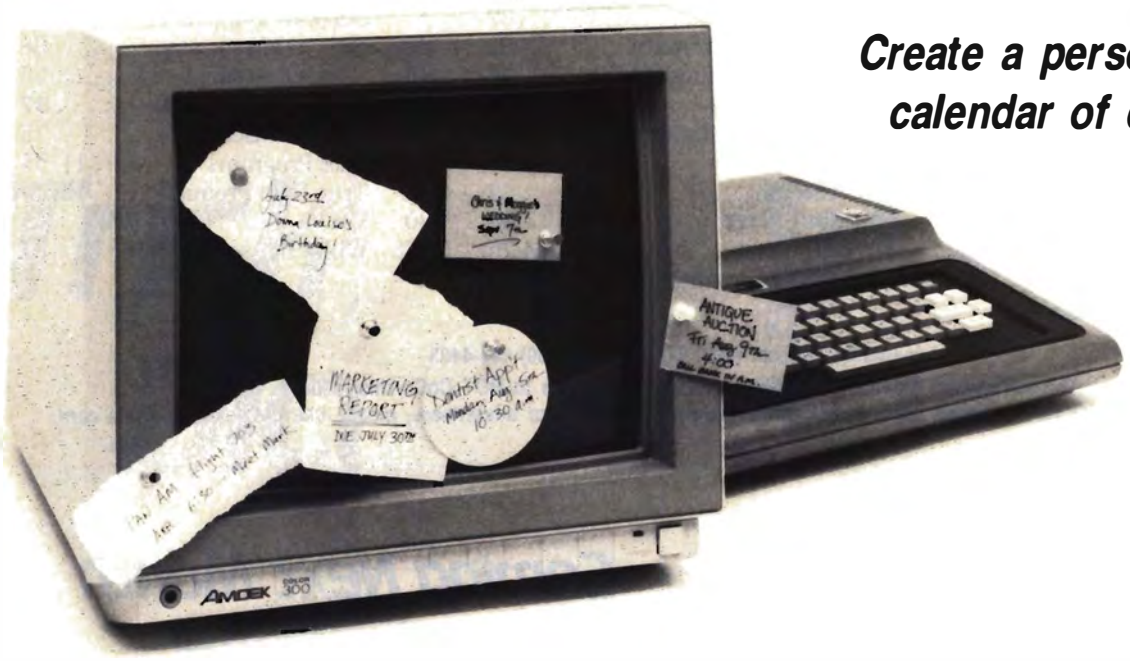
Dick Myers proves that Extended Basic is a powerful drawing tool in itself. "Cartooning and Animation" describes three Basic drawing and three animation techniques.

And there's more. We'll give you a routine that adds auto-line numbering to your CoCo and a program that analyzes potential investments. Scott Norman discusses calendar/appointment programs in his Computer Room column. And Fred Blechman test drives what promises to be one of the best printer buys around: the new Star SG-10.

by James Huckabey

Date Minder

Create a personalized calendar of events.



Have you ever overlooked a deadline, missed an appointment, or forgotten a birthday? Maybe you need Data Minder to keep you on track. The calendar lets you record six, 24-character entries per day. In addition, you can designate six special days and scan entries for 15 days. If having your CoCo remind you of upcoming events isn't enough, you can print your events schedule and use it as a reference throughout the day.

Operating Instructions

The lines in Program Listing 1 are long to save memory and speed execution. Unfortunately, it is easier to make a typing mistake in long lines, so be careful and be sure to save the program before you attempt to run it.

If you are using a disk system, you must perform a PCLEAR0 to use the space usually reserved for graphics. Type:

```
POKE25,14:POKE3584,0:NEW
```

and press the enter key. If you fail to perform a PCLEAR0, the program clears memory, and you must reload and rerun the program. You can use the normal LOAD and RUN commands or type in RUN "CALENDAR" to make the program self-starting.

With a cassette system, no special PCLEAR is necessary because the program performs a PMODE:PCLEAR1 that changes the graphic allocation without destroying memory. CLOAD and run the program in the usual way.

After the title screen appears, you are asked whether it is a leap year. Respond with a Y or an N and then provide the current date (MMDD) at the next prompt. The Main Menu (Table 1) appears. The first two options, View Data and Enter Data, produce submenus. All other options bring up prompt sequences for you to follow or run without user input.

View Menu

Press 1 to get the View Menu (Table 2). Option 1 in this menu lets you scan entries one day at a time for 15 days, beginning with the current date. To progress from day to day, press any key. If no data is available, the screen displays a timed no-data message and then brings up the next day. If you have designated a day as a Special Day, a special message appears with the other entries for that day. A musical score accompanying the entry, ensures that you do not miss the message.

After viewing 15 days, you return to the View Menu. Press 2 and update the current date to continue viewing.

While the program can accommodate up to 240 entries, it is best not to exceed 200 because too many entries slow the program down. If there is a delay between displays, do not worry. The computer is performing housekeeping tasks because you have many entries to be manipulated. The more entries

```
1 = VIEW DATA
2 = ENTER DATA
3 = SAVE TO TAPE
4 = SAVE TO DISK
5 = LOAD FROM TAPE
6 = LOAD FROM DISK
7 = PRINT CURRENT DATA
8 = EXIT PROGRAM
```

Table 1. Main Menu

```
1 = SCAN FROM TODAY'S DATE
2 = CHANGE SCAN DATE
3 = RETURN TO MAIN MENU
```

Table 2. View Menu

System Requirements
32K RAM
Extended Color Basic
Or Disk Basic
Printer Optional

Photo by Mark Corliss

you have, the more housekeeping and the greater the delay.

Data Entry Menu

Pressing 2 in the Main Menu brings up the Data Entry Menu (Table 3). You may choose a month by typing a letter from A through L, or a special day by pressing M. Press N to return to the Main Menu.

A = JANUARY
B = FEBRUARY
C = MARCH
D = APRIL
E = MAY
F = JUNE
G = JULY
H = AUGUST
I = SEPTEMBER
J = OCTOBER
K = NOVEMBER
L = DECEMBER
M = SPECIALS
N = MAIN MENU

Table 3. Data Entry Menu

To make regular calendar entries, press the letter of the appropriate month and then enter the current date for that month. The entry screen contains an entry line; an entry accumulator for the current date; edit, menu, and exit prompts; and the total-number-of-entries accumulator.

A blinking cursor marks the entry line. To the right you see the end-of-line marker, <24. If you type over any portion of this marker, you have exceeded the 24-character limit and must resubmit the line. If you use Basic commands, such as commas and colons, you get an error message and the program does not store the portion of the message after the command.

After you complete six entries for a date, the program puts you in the edit mode. If you type Y to change the data, you are prompted for the line number (1–6) or for # to change all entries. Resubmit the line or press the enter key to delete it. If you type N to leave the data intact, you return to the data entry position for the current month.

To go directly into the editing mode from the entry screen, press #. Press * to bring up the Data Entry Menu, and press @ to exit.

Press M from the Data Entry Menu to bring up the screen for special days. You see the number of special days used so far and are prompted for the date (MMDD). You can submit only one entry per special day—other entries are ignored. When you reach the limit of six special days, the program puts you in the edit mode.

You can also reach the edit mode from the special-day screen by pressing #. You see 12 lines. The date and its corresponding message are on separate lines. Edit special entries as you would a data entry. Press @ to return to the Data Entry Menu.

I$,II$ = INKEY$
I$ = Input for Special Day edit
P$,PA$,PN$,PP$,PT$,SP$,SZ$ = Various PRINT strings
S$ = 4-digit, Special-Day date (adds 0 to front of months 1–9)
DA$ = Input for day of month
**DE$** = Input for daily entries (CE$(YD,DE$))
DIM CE$(366,6) = Calendar entries
MO$(12) = Month name
DM(12) = Days in month
SD$(6) = Special Day data
SD (6) = Special Day date
TS (6) = Temporary Special Day date (Julian)
DM$ = Input for day of month
DN$ = Device name (Disk or Tape)
ND$ = Input for new date
**P1$** = STRING$ of 32 = (CHR$(61))
**P2$** = STRING$ of 31 = (CHR$(61))
**P3$** = P1$ + P2$
**P4$** = P1$ + P1$
**P6$** = STRING$ of 12> (CHR$(62))
P8$ = "NO DATA THIS DATE"
PL$ = Play string
S1$ = Input for Special Day date
S2$ = Input for Special Day data
SZ$ = Modified Special Day data (added asterisks)
**TC$** = STR$(C)
TD$ = Input for today's date
C,Y = Counters
I = ASC(I$) or ASC(I$) – 64
T,X,Y,Z,Z1,Z2 = FOR. . . NEXT loop counters
DA = Day of month
DN = Device number (1 = disk, – 1 = cassette)
D1 = Day of month (DM(n))
D2 = Day of month (DA + 14 – D1)
EF = Existing file (0 = no file in memory, 1 = file in memory)
FL = Flag for special day routine (0 = off, 1 = on)
MO = Month number
NE = Number of entries in file
PA = PRINT @ location value
PO = POKE flag (1 = disk, 0 = cassette)
PT = PRINT TAB value for printer routine
SD = VAL (SD$ or T2$)
SW = Switch counter in edit modes (FOR. . .NEXT loop)
T1$,T1 = Special Day, month number
T2$,T2 = Special Day, day of month
TC = C
TE = NE
TI = I (ASC(I) or VAL(I$))
TM = MO
TY = YD
YD = Year day (Julian)

Table 4. Program Variables

Save To Tape Or Disk

When you press 3 from the Main Menu, the program prompts you to prepare the recorder and press any key to continue. A wait message appears; when the save is complete, you return to the Main Menu.

I recommend that you leave blank space several seconds long between recordings on the tape; this makes repositioning the tape for loading easier. If you do not have a bulk tape eraser, erase your tape by rewinding the tape to the beginning and then pressing the record and play buttons simultaneously. Use the Motor On command to turn the recorder on. When you reach the end of the tape, type Motor Off.

Press 4 from the Main Menu to save to disk. The steps, which are listed on the screen, are automatic; when the save is completed you return to the Main Menu.

Load From Tape Or Disk

To load calendar entries from tape, press 5 at the Main Menu. If there is data in memory, the program deletes it, so you must press Y to confirm your desire to load data. The screen prompts you for entry. You return to the Main Menu when the tape is loaded.

Press 6 from the Main Menu to load from disk. The load-from-disk feature clears memory. Loading from disk, like saving to disk, is automatic.

Printing And Exiting

Prepare your printer and press 7 from the Main Menu to print current data. You can print entries for 15 days. To print further entries, run the program, change the current date appropriately, return to the Main Menu, and press 7.

When you press 8 from the Main Menu, the screen displays an end-of-program message, a PLAY string is played, and the program ends. To restart without losing data, type CONT and press the enter key to return to the Main Menu.

Problems And Modifications

I have commented the program so that you can easily follow program flow. Refer to Table 4 for a complete list of program variables. If you have difficulty with the program as written or need help in modifying it, please contact me. Enclose a complete description of your problems, including your printout of the program, if possible, and a stamped, self-addressed envelope. ■

Program listing on page 48

Date Minder is available on the Instant CoCo cassette (see ad on p. 65), but for $12, James Huckabey will send you a tape or disk of the program and a copy (written with Telewriter-64) of the documentation. Write to him at 3303 A Hollywood Ave., Bellingham, WA 98225.

Reader's Forum

Speed-Up Sans POKE

If your system cannot handle the "vitamin E" POKE that speeds up the computer, or even if it can but still runs too slow for you, here is a speed-up trick I picked up while programming in Commodore (shudder) Basic. Try this little program if you have Extended or Disk Basic:

```
5 CLSO:TIMER = 0:FOR I = 0*0 TO 0 + 999:J = 0:K = 0/
  13:NEXT:PRINT TIMER
```

Write down the value of TIMER returned. Now edit the line and change all the zeros to periods. Run it again. The speed increase for this single line is over a second. A longer program, of course, would benefit more from this technique. Note that only the number zero alone should be changed. When a zero appears as a digit, like in the number 2045, leave it alone. Although for demonstration purposes I used Extended and Disk Basic, the method works just as well in standard Color Basic. Any Microsoft Basic that exclusively uses floating-point math might also accept this trick.

Ronny Ong
Arlington, TX

DRAW Your Own

I have developed some CoCo graphics commands to draw any letter of the alphabet, the numbers 0–9, and various punctuation marks and symbols. Most of the characters are designed to be placed on the same horizontal line. Lowercase letters with descenders will need a few adjustments and are designed to be drawn on a PMODE 4 screen.

You can adapt these commands for use in any Extended Color Basic program. When you write a program you can use these letters to make a title screen, a scoreboard, or another screen.

Just type in the DRAW command, position the letter, and type in the commands listed beside the character you want to use. (See Table 1.) You can also place them in a string or DATA statement.

Brian Patrick
Hunstville, AL

Alphabet

A = E7; F7; H3; L8
a = H2; L2; G1; D2; F1; R2; E2; D3; U5
B = U7; R5; F1; D1; G1; L5; R5; F1; D2; G1; L5
b = U7; D3; R4; F1; D2; G1; L4
C = G1; L4; H2; U3; E2; R4; F1
c = G1: L3; H1; U3; E1; R3; F1
D = U7; R4; F2; D3; G2; L4
d = U7; D3; L4; G1; D2; F1; R4
E = U7; R6; BD3; BL2; L4; D4; R6
e = G1; L3; H1; U3; E1; R3; F1; D1; L5
F = U7; R6; L6; D3; R4
f = U6; E1; R2; F1; BD3; BL2; L4
G = H1; L4; G2; D4; F2; R4; E2; U2; L3
g = G1; L2; H1; U1; E1; R2; F1; D6; G1; L2; H1
H = U7; D3; R7; U3; D7
h = U7; D5; E2; R2; F2; D2
I = U7; R3; L6; BD7; R6
i = R2; L1; U4: L1; R2; BL; BU2; R; L2
J = D; F1: R3; E1; U6; L2; R4
j = D1; F1; R2; E1; U5; BU3; D
K = U7; D5; E5; G3; F4
k = U7; D6; E4; G2; F3
L = U7; D7; R5
l = U7; L2; BD7; R4
M = U7; F4; E4; D7
m = U3; E1; R2; F1; D3; U3; E1; R2; F1; D3
N = U7; F7; U7
n = U4; D2; E2; R2; F1; D3
O = H2; U3; E2; R2; F2; D3; G2; L2
o = H1; U2; E1; R2; F1; D2; G1; L2
P = U7; R4; F1; D2; G1; L4
p = U7; R3; F1; D2; G1; L3
Q = H2; U3; E2; R3; F2; D3; G1; F1; H3; F2; G1; L3
q = U7; L3; G1; D2; F1; R3
R = U7; R4; F1; D2; G1; L4; R2; F3
r = U4; D2; E2; R2; F1
S = F2; R3; E1; U2; H1; L3; H1; U1; E1; R3; F1
s = F1; R1: E1: U1; H1; L1; H1; U1; E1; R1; F1
T = U7; R4; L8
t = U7; D2; R3; L6
U = BU7; D5; F2; R2; E3; D3; U7
u = BU4; D3; F1; R1; E2; D2; U4
V = H5; U2; D2; F5; E5; U2
v = H4; F4; E4
W = BU7; D7; E4; F4; U7
w = BU4; D3; F1; R2; E1; U3; D3; F1; R2; E1; U3
X = E7; BD7; H7
x = E4; BD4; H4
Y = U3; H4; F4; E4
y = E6; G3; H3

Z = L7; E7; L7
z = L4; E4; L4

Numerals

0 = H1; E4; G4; H1; U2; E2; R3; F2; D2; G2; L3
1 = U7; G3; BD4; R5
2 = R5; L4; E3; U2; H1; L2; G1; D1
3 = BU1; F1; R3; E1; U1; H1; L3; E3; L5
4 = U6: G4; R6
5 = BU1; F1; R3; E1; U1; H1; L4; U3; R4
6 = R2; E1; U1; H1; L2; G1; D1; F1; H1; U4; E1; R2; F1
7 = U3; E3; L5
8 = R2; E1; U1; H1; L2; G1; D1; F1; H1; U1; E1; H1; U1; E1; R2; F1; D1; G1
9 = U6; D1; H1; L2; G1; D1; F1; R2

Punctuation marks/symbols

! = U; BU2; U4
" = BU5; U2; BR3; D2
= U7; BR4; D7; U2; BR3; L10; BU3; R10
$ = BU1; F1; R4; E1; U1; H1; L4; H1; U1; E1; R4; F1; H1; L2; BU1; D8
% = E7; BD7; U2; L2; D2; R2; BD2; BH7; U2; R2; D2; L2
& = BR1; R4; E1; G1; F; H2; U; H; L2; G; D; F; BU3; BR; H; U; E; R; F; D; G
' = BU5; U2
(= H2; U3; E2
) = E2; U3; H2
* = E6; G3; F3; H6; BD3; L1; R8; L4; U4; D8
÷ = BU3; R8; BU2; BL4; U; BD6; U
Minus Symbol " – " = BU3; R8
Addition Symbol " + " = BU3; R6; BU3; BL3; D6
/ = E7
@ = BU2; U2; R2; D2; L2; R4; U3; H; L5; G; D4; F; R5
: = BU1; U; BU3; U
; = G; E; U; BU3; U
, = G; E; U
. = U
? = U; BU2; E2; U; H; L2; G
< = BU3; E3; G3; F3
> = E3; H3
| = U7; G3; E3; F3
↓ = H3; F3; E3; G3; U7
→ = BU4; R10; H3; F3; G3
← = BU1; H3; E3; G3; R10
¢ = BU2; G1; L4; H1; U3; E1; R4; F1; H1; L2; U1; D7
Equal Symbol " = " = BU2; R8; BU3; L8

Table 1. PMODE 4 DRAW Statements for the A B C's, Numbers 0–9, and Various Punctuation Marks.

Will It Fit?

If your Color Computer has 32K or 64K of memory, and you want to know if a certain program will run on a computer with 16K, then first you must save the program on cassette or disk. Type CLEAR 200,16384. If your computer has Extended Color Basic and the program you are testing is going to run on a computer with just Color Basic, then also type POKE 25,6:NEW. Load the program into memory again. If you get an OM? ERROR, then the program takes up too much memory for 16K.

Colin Alexander
Diamond Springs, CA

1.1 Disk ROM Exec Addresses

The following is a list of EXEC addresses for the 1.1 Disk Extended Basic ROM:

Command	Hex Address	Command	Hex Address
DIR	CCA9	SAVE	C9E0
DRIVE	CEC5	WRITE	D066
FIELD	D0BC	VERIFY	D74E
FILES	D15C	UNLOAD	D233
KILL	C6EF	DSKINI	D599
LOAD	CA48	BACKUP	D262
LSET	D102	COPY	D3B9
MERGE	CA39	DSKI$	D4ED
RENAME	D01B	DSKO$	D562
RSET	D101	DOS	D6EC

Function	Hex Address	Function	Hex Address
CVN	CDF4	LOF	CE37
FREE	CE9C	MKN$	CE02
LOC	CE10	AS	B277

Kenton Fifield
Ontario, Canada

CoCo Max

This is one of those rare programs that will captivate everyone in your family.... No one can see CoCo Max and not want to try it!

We are all witnessing an exciting revolution in microcomputers: a radically new kind of computer and software that opens a whole new world of creative power to computer users.

It was inevitable that this exciting approach would be brought to the CoCo. With this in mind, Colorware chose to go all out and maximize this new concept for the color computer. That meant designing not just software but hardware too. It meant thousands of hours of pure machine language programming. Rarely has this much effort been applied to one product for the Color Computer.

UNMATCHED CAPABILITY...

Because we took the maximum approach: highly optimized machine code combined with hardware, CoCo Max truly stands above the rest as the ultimate creative tool for the Color Computer. It's unrivaled performance lets you create with more brilliance and more speed than any similar system – much more than you ever imagined possible. And, you can do it in black & white or color.

All the sophisticated power of the bigger systems is there: *Icons, Pull-Down Menus,* full *Graphic Editing, Font Styles,* and all kinds of handy tools and shortcuts.

Plug your joystick, mouse or touch pad into CoCo Max's Hi-Res Input Unit. Then use a delightfully simple ***Point-and-Click*** method to get any of CoCo Max's powerful graphic tools. It has them all:

You can ***Brush, Spray*** or ***Fill*** with any ***Color, Shading*** or ***Pattern.*** Use ***Rubber Band Lines*** and ***Shapes*** (square, rectangle, circle, elipse, etc.) to create perfect illustrations with speed and ease. There's a ***Pencil,*** an ***Eraser*** and even a selection of ***Caligraphy Brushes.*** And, as you can see, CoCo Max can do a lot with text.
All of the newest special effects are there: ***Trace Edges, Flip, Invert, Brush Mirrors,*** etc. And all of the very latest supercapabilities like: ***Undo,*** which automatically reverses your mistakes, and ***Fat Bits*** which zooms you way in on any part of your subject to allow dot-for-dot precision.

THE BIG PICTURE

The large image box in the middle of the CoCo Max screen is actually only a window on an even larger image. Use the Point-and Click "Hand" to effortlessly move your window over any portion of the larger image. You have a working area of up to 3-½ times the area of the window itself.

FLEXIBLE PRINTING...

CoCo Max gives you many ways to print. Fill a whole page with your image or condense two full CoCo screens to less than ¼ page for a finely detailed copy. "Dump" your CoCo Max screen full size or shrink it to ⅛ page size.

THE COCO MAX SYSTEM

FREEDOM TO CREATE...

Anyone who wants to create anything at all on their CoCo screen or printer will certainly be very glad to meet CoCo Max. CoCo Max's friendly yet sophisticated graphic and text capabilities let you almost instantly produce illustrations, diagrams, charts,

graphs, and computer art – for serious use or just for creative fun.

tion by using software schemes such as sliding windows. Although clever, these schemes yield sluggish and awkward results. Only CoCo Max does it the right way. The CoCo Max Hi-Res Input Unit plugs into your ROM slot and adds an entirely new joystick input to your computer – a precision one with a 49,152 point resolution to match the CoCo screen exactly.

Plug your same joystick, mouse or touch pad into this new input and you have a whole new kind of control. The difference is remarkable.

You may then use CoCo Max's graphic magic on it. The DS-69 is available as an option from Colorware from $149.95 complete with its own software on disk or tape. Using the DS-69 with a disk requires an RS multi-pak adaptor.

COCO MAX REQUIREMENTS

The CoCo Max System includes the Hi-Res Input Unit, software on disk or cassette (please specify) and user manual. It will work on any 64K Extended or non-extended Color Computer. You'll need a Radio Shack or equivalent joystick, mouse or touch pad. Disk systems require a Multi-Slot Interface or Y Branching Cable.

THE COMPLETE COCO MAX SYSTEM,
with software on ***DISK***... .... ***$69.95***

with software on ***CASSETTE*** (Available Mar '85)... .. .. ***$69.95***

Y-BRANCHING CABLE-If you have a disk system but do not have a Multi-Slot Interface, use this economical 40-pin, 1 male, 2 female cable to connect the CoCo Max Hi-Res input unit and your disk controller to your CoCo. . . ***$27.95***

AN ABSOLUTE GUARANTEE

CoCo Max is a hardware-software system that no software-only system can match. Get CoCo Max and see your CoCo perform as it never could before. If you don't agree that CoCo Max is the ultimate creative tool for the Color Computer, simply return it within 20 days for a full, courteous refund from Colorware.

THE HARDWARE...

This is the key to CoCo Max's unmatched performance. Did you know the normal joystick input built into the Color Computer only allows access to 4,096 (64 x 64) points on the CoCo screen? Yet, the Color Computer's high resolution screen

has 49,152 (256 x 192) pixels. This means that a joystick, mouse or even a touch pad can, at best, only access about one tenth of the pixels on the CoCo screen.

Most graphic programs ignore this hardware limitation of the Color Computer and give you only low-res control. Others attempt to overcome the limita-

A DIGITIZER OPTION...

We studied all the video digitizers available and picked the best of them to link with CoCo Max. The DS-69 from Micro Works was our choice. This optional device lets you capture the image from any video source (video recorder, camera, etc.) on your Color Computer.

COLORWARE

(Sorry, not compatible with JDOS)

TOLL FREE ORDER LINE:
(800) 221-0916

VISA

MasterCard

Colorware Inc.
78-03B Jamaica Ave.
Woodhaven, NY 11421
(718) 647-2864

ORDERING INFORMATION

ADD $3.00 PER ORDER FOR SHIPPING AND HANDLING.
C.O.D.'S ADD $3.00 EXTRA.
SHIPPING AND HANDLING FOR CANADA IS $5.00
WE ACCEPT VISA, MASTER CARD, M.O.'S, CHECKS.
N.Y. RESIDENTS MUST ADD SALES TAX.

DISK $44.95

NEW

Introducing The "Super Smart"

DATA PACK II

TERMINAL COMMUNICATIONS SOFTWARE

Also Supports The PBJ 80 Column "Word Pak", Deluxe RS-232 Pak, Parallel Printer Card and PBJ 2SP Pak

NEW

TAPE $34.95

28-64x24 Hi-Resolution Display

9 Unlimited Variable Length Macro Key Buffers

64K Support Up To 55K Text Buffer

BUFFER EDITOR Insert, Delete Change or Add To Buffer

"FEATURES"

- No Lost Information When Using Hi-Resolution Display On Line
- ASCII Compatible File Format
- Full Text Buffering
- Terminal Baud Rates 300 to 9600
- Automatic Word Wrap Eliminates Split Words
- Full/Half Duplex
- Automatic File Capture
- Programmable Word Length, Parity and Stop Bits
- Save and Load Text Buffer and Program Key Buffers to Tape or Disk
- 9 Hi-Resolution Display Formats, 28 to 255 x 24
- True Upper/lower Case Display
- Kill Graphics Option for an Extra 6K
- Supports Line Break
- Freeze Display and Review Information On Line
- Send Files Directly from Buffer or Disk
- Full Disk Support for Disk Version
- Send Control Codes from Keyboard
- Separate Printer Baud Rates 110-9600
- Display on Screen or Output Contents of Buffer to Printer
- Automatic Memory Sense 16-64K
- 9 Programmable Function Key Variable Length Macro Buffer
- Programmable Prompt Character or Delay to Send Next Line
- Programmable Control Character Trapping
- Programmable Open/Close Buffer Characters
- Automatic Key Repeat For Editing
- Program and Memory Status Displays

NEW!!!

"The Wait is Finally Over"

ANNOUNCING
The CBASIC COMPILER

NEW!!!

Now anyone can create fast efficient Machine Language Programs without the Drudgery of using an Assembler.

CBASIC is a fast Machine Language integer Basic Complier that can convert Color Basic programs into fast machine language programs. CBASIC features over 100 Basic Commands and functions that fully support Disk, Tape, Screen and Printer I/O, Hi & Low Resolution Graphics, Sound, Play and String Operations just like Color Basic. CBASIC also includes a powerful full featured Basic program Editor using a 51,64 or 85 by 24 line display. The Hi-Resolution display can be automatically included in your compiled program for enhanced display capability and allow mixed text and graphic displays.

Graphics Commands: CIRCLE, COLOR, CLS, DRAW, GET, LINE, PAINT, PCLS, PCOPY, PMODE, PRESET, PSET, PUT, RESET, SCREEN, SET, POINT, PPOINT

Sound Commands: PLAY, SOUND

String Functions: CHR$, LEFT$, MID$, RIGHT$, LEN, INSTR, LSET, RSET, TRM$, STR$, STRING$, INKEY$, MKN$

Numeric Functions: ABS, POS, TIMER, RND, ASC, TAB, CAL, JOYSTK, PEEK, POKE, LOC, LOF, EOF, FREE, CVN, ERR, VARPTR, SWAP

I/O Commands: OPEN, CLOSE, INPUT, LINEINPUT, PRINT WRITE, PRINT @, GET, PUT, KILL, CHAINM, FIELD, DATA, READ, RESTORE

Program Control: FOR/NEXT/STEP, GOTO/GOSUB, IF/THEN/ELSE, RETURN, STOP, RETI, ON n GOTO/GOSUB, ON ERROR, ON RESET, ON IRQ/FIRQ/NMI, ON OVR/NOVR, EXEC, LET

Directives: ORG, REM OR ', END, DIM, END, BASE, RAM, ON/OFF, RAM64K, HIRES, GENERATE, DPSET, STACK

Editor Commands: LINE EDIT, AUTO EDIT, COPY, MOVE, RENUMBER, AUTO LINE#, PRINTER, LIST, DELETE, SEARCH, REPLACE, BAUD RATE, PRINTER, CBASIC, TAPPEND, SKIP, SIZE, LOAD, SAVE, APPEND, KILL, DIR, and much, much more.

REQUIRES 32K and Disk, 64K recommended

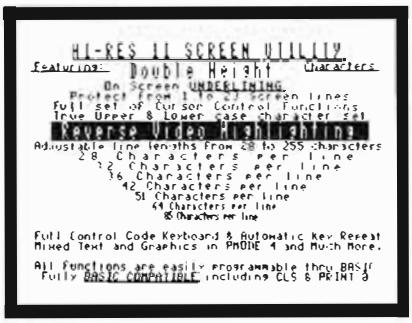
Introductory Price $119.00
Regularly $149.00

HI — RES II SCREEN UTILITY

NEW IMPROVED VERSION

- ***UP TO 85 CHARACTERS PER LINE READABILITY***
- ***ADJUSTABLE AUTOMATIC KEY REPEAT***
- ***PROPTECT 1-23 SCREEN LINES***
- ***CONTROL CODE KEYBOARD***

- FULLY BASIC COMPATIBLE
- DISPLAY FORMATS OF 28 to 255 CHARACTERS PER LINE
- FULL 96 UPPER/LOWER CASE CHARACTERS
- MIXED GRAPHICS & TEXT OR SEPARATE GRAPHIC & TEXT SCREENS
- INDIVIDUAL CHARACTER HIGHLIGHTING
- REVERSE CHARACTER HIGHLIGHT MODE
- WRITTEN IN FAST MACHINE LANGUAGE
- AUTOMATIC RELOCATES TO TOP OF 16/32K
- AUTOMATICALLY SUPPORTS 64K of RAM WITH RESET CONTROL
- REVERSE SCREEN
- ON SCREEN UNDERLINE
- DOUBLE SIZE CHARACTERS
- ERASE TO END OF LINE
- ERASE TO END OF SCREEN
- HOME CURSOR
- BELL TONE CHARACTER
- HOME CURSOR & CLEAR SCREEN
- REQUIRES ONLY 2K OF RAM
- COMPATIBLE WITH ALL TAPE & DISK SYSTEMS



$24.95 TAPE $29.95 DISK

ALL ORDERS SHIPPED FROM STOCK
ADD $2.50 POSTAGE

Circle Reader Service card #335

CER COMP

5566 Ricochet Avenue Las Vegas, Nevada 89110

(702) 452-0632

Screen Enhancement Program Comparison Chart

PROGRAM FEATURES	HI-RES II NEW	HI-RES I OLD	BRAND X
Upper/Lower case characters	Yes	Yes	Yes
Mixed Text and Graphics	Yes	Yes	Yes
Separate Text & Graphics	Yes	Yes	No
Print @ fully implemented	Yes	Yes	Yes
Print @ on all line lengths	Yes	Yes	51 only
Different line lengths	28 to 255 (9)	28 to 255 (9)	51 only (1)
Automatic Key Repeat	Yes	Yes	Yes
Adjustable Key Repeat	Yes	No	No
Auto Repeat Disable	Yes	No	No
Erase to end of line/screen	Yes	Yes	Yes
Home Cursor	Yes	Yes	Yes
Solid or Blinking Cursor	Yes	No	Yes
CLS command supported	Buff/Black	Buff/Black	Buff/Black
X,Y Coordinate Cursor Positioning	Yes	Yes	No
Double Size Characters	Yes	Yes	No
Individual/Continuous Highlighting	Yes	Yes	No
On Screen Underlining	Yes	Yes	No
Clear Key functional	Clear/L keys	Clear key	No
16 32 & 64K Supported	Yes	Yes	Yes
Green or Black Background Color	Yes	No	No
Dual Character sets for Enhanced 64 and 85 Characters per line display	Yes	No	No
Protected Screen Lines (programmable)	1 to 23	No	No
Full Control Code Keyboard for Screen control directly from the keyboard	Yes	No	No
Programmable Tab Character Spacing	Yes	No	No
Full Screen Reverse Function	Yes	Yes	No
Switch to & from the Standard 16 by 32 Screen for full compatability	Yes	No	No
On Error Goto Function	No	No	Yes
Extended Basic Required	No	Yes	Yes
All Machine Language Program	Yes	Yes	Yes
RAM Required in addition to Screen RAM	2K	2K	2K
Program Price (Tape)	$24.95	$19.95	$29.95

VISA master charge

VISA, MASTERCARD AND C.O.D. ACCEPTED

Assembly 101

by Victor and James Perotti

All Clear

If you took a class in Assembly language from a computer-science teacher, you would learn the mysterious arts of solving mathematical problems in binary. That's a tedious and difficult route, so we'll work with the Color Computer's screen memory, writing programs that affect the screen. In that way, you will get immediate feedback about what your programs are doing. Last month you learned how to put a black line on the screen; this month we'll get more sophisticated and clear the screen.

With each program we introduce, we will include further explanations of what is going on. If you are just beginning to program in Assembly language, there probably were some points in last month's lesson that you didn't understand. But as we continue along and add more information each month, things should begin to make sense. Stick with it.

Clearing The Screen

The screen is nothing but a block of memory. In text mode it uses 16 lines of 32 characters, so it is 512 bytes long; when you want to use the high-resolution screen, you'll use more screen memory. Normally, screen memory is located in RAM at addresses 1024 to 1536 in decimal, or $0400 to $0600 in hex. Whatever our programs store in this memory location will be printed on the screen. To clear the screen, the program must write to every one of the 512 screen locations.

Clearing the screen means loading screen memory with blank spaces, which are $60 hex. The Color Computer's screen memory does not use the normal ASCII values; because of its reversed lowercase letters, it uses the ASCII value plus $40 for some of the characters. The ASCII value for a space is $20; with the $40 added it becomes $60. When screen memory contains zero, the zero plus $40 prints an @, which is why PEEKs disclose large sections of memory filled with @'s.

Before you get going on the program, try experimenting with screen memory in ZBug, ABug, CBug, DCBug or any other Bug you might have. The Micro Works' products use M to display blocks of memory in hex, and ZBug uses T; therefore, M 0400 0600 will display the screen.

Now move the cursor around and start writing over the existing contents. Try zero $20, $40, $80, and so on. It is confusing, but the screen memory was written over with the contents of its memory; when you change one, it appears at the appropriate screen location, overwriting the existing numbers.

Equates

The first computer programmers worked completely with binary, and later with hex or octal, which were big improvements. Programming became easier as languages moved from low to higher levels, because programming logic became easier to understand. Assembly language is now about as low a level as you can go, but the new assemblers make programming much easier. You can now work with commands that are relatively English-like; no longer must you use cryptic numeric representations.

Labels provide a nice substitute for addresses: equates (EQU) let you write words instead of addresses or data. The syntax is as follows: word EQU number. Here are the equates you will use:

```
VIDRAM EQU $0400
VIDEND EQU $0600
POLCAT EQU $A1B1
BLANK EQU $60
BREAK EQU $03
```

We have called the top of screen memory VIDRAM ($0400) and the end of screen memory VIDEND ($0600). Instead of using the $60, the ASCII representation of a blank or space, just call it BLANK. The same goes for the break key; when you press it, it generates a $03, so make the break key equivalent to $03. Using labels and equates makes your programs readable, even when you look at them a month or so later. You will rarely write a program and run it only once. Even the best programmers write and rewrite, change and improve. When you go back to make modifications, the clarity of those English-like instructions pays off.

The Program Listing starts with the equates and then initializes the registers that will be used; hence, the label INIT. To speed things up as you store blanks in 512 locations, the program uses the 16-bit register, D. Accumulators A and B are loaded with blanks, $60; the two accumulators together form the 16-bit D accumulator. You can use A and B separately, or you can join them as D. With this approach, you can place blanks into two 8-bit memory locations at one time. Register Y will hold the address where the blanks will be stored. Register Y is initially set to VIDRAM, $0400, and will then be incremented until it holds VIDEND, $0600, which is 512 bytes later, since $200 hex equals 512 decimal.

If you're using EDTASM+, eliminate the first line (START NAM CLEAR) when you type in the listing, and switch the END and SWI instructions (so that END becomes SWI and SWI becomes END) in the last two lines of the program.

As an aside, the pound sign, (#) preceding #BLANK, #VIDRAM, and #VIDEND indicates that the register should be loaded with

System Requirements
16K RAM
Color Basic
Editor/Assembler

```
START NAM CLEAR

** EQUATES:
VIDRAM      EQU     $0400
VIDEND      EQU     $0600
POLCAT      EQU     $A1B1
BLANK       EQU     $60
BREAK       EQU     $03

** MAIN PROGRAM:
INIT    LDA     #BLANK      LOAD BLANK IN A
        LDB     #BLANK      ANOTHER IN B
        LDY     #VIDRAM     Y HAS MEM ADDR

LOOP    STD     ,Y++        D = A + B:STORE AT Y
        CMPY    #VIDEND     AT END OF MEM?
        BLO     LOOP        IF NOT, LOOP

** SCREEN CLEARED—NOW DISPLAY IT

WAIT    JSR     POLCAT      CALL ROM ROUTINE
        CMPA    #BREAK      BREAK KEY HIT?
        BNE     WAIT        IF NOT, WAIT MORE

DONE    END
        SWI
```

Program Listing. Clearing the Screen

the number itself, not the contents of the number taken as an address. Normally, LDY $400 is used; it means that Y should take on the contents of memory location $400. But that is not what you need here, so LDY #$0400 or LDY #VIDRAM loads Y with that number. LDA #BLANK puts a $60 in A, not the contents of address $60.

Main Program

With D containing two blanks, and with Y pointing at the top of screen memory, the process of storing D in screen memory begins. As the D register stores blanks into the memory locations specified by Y, Y is then automatically incremented, not once but twice with the ,Y + + instruction. D puts blanks in $0400 and $0401, Y is incremented twice to $402, and the program loops and begins placing blanks again. It clears two 8-bit memory locations during each pass. This is how the 6809 gains an enormous speed advantage over inferior 8-bit machines like the Commodore 64, which has three 8-bit registers to work with.

Controlling The Loop

While STD ,Y + + does all the work of clearing memory, it needs to operate out of a loop that exists when Y becomes $0600. The program uses the very common technique of combining a compare (CMP) statement and a branch (B). In this case, compare Y with #VIDEND and branch to LOOP if Y's value is lower (LO) than #VIDEND. The CMPY instruction subtracts $0600 (VIDEND) from the value in Y, noting whether there is a positive difference, or whether the two are equal. Various branch instructions permit branching on any number of conditions, as we will see in later columns. In this case, BLO (branch if lower than) is used, meaning branch if Y is lower than VIDEND. Every time the branch occurs, the LOOP repeats STD ,Y + + . Eventually, time flies with machine code: Y equals $0600, the looping ends, and the program moves to the WAIT module.

Waiting And Watching

If the program ended after the LOOP with an SWI, returning control to ABug or ZBug, you would never see the screen clear. Try it! The program is so fast that the "ABug" or "ZBug" message appears as you press the G key (G INIT for EDTASM +) to run it. Check the registers with R: Note that Y now has $600, and A and B have $60. The program ran! Now you need a way to halt execution and display that blank screen.

Basic's ROM provides a simple answer; it has a subroutine that reads the keyboard, waiting for you to press a key. The subroutine lives at $A1B1 and is called POLCAT; note that POLCAT was defined in one of the equates. You'll have to tell it to wait for you to press the break key, $03, while you gaze at the blank screen. When you press break, the program will end. POLCAT puts the value of the key you press into the A register; pressing break puts 03 in A. You are again using a loop that a compare and a branch control; CMPA #BREAK checks to see if you pressed break. If A is not equal to 03, then it branches to WAIT; BNE means "branch if not equal to zero". Hence, if the difference between A and 03 is not zero, the program continues to WAIT.

Assemble the program and run it. Experiment with other graphics characters in A and B, those are characters with values above 128. You do know that JSR $A928 will clear the screen, don't you? So you did a little extra work—think of all you learned. ■

Write to Victor and James Perotti at 163-D Pine Grove Heights, Athens, OH 45701.

6809 On Line

by Bobby Ballard

Language On Line

What does this month's theme, languages, have in common with telecomputing? Well, there's a wealth of language information available on line. Flex, OS-9, C, Assembly language, and Basic are all supported.

An obvious choice for CoCo owners is CompuServe's Color Computer SIG (Special Interest Group), which has a data library for Assembly language and support in Basic under various topics. However, this month I'd like to give you other CompuServe sources for on-line languages as well as some BBS phone numbers.

OS-9 SIG

Don't be fooled by the title—this group has other files in addition to OS-9. There is no extra charge for accessing the OS-9 SIG from within CompuServe and you can download information for personal use. To access the SIG, type GO PCS-18 at any prompt. The first time through, you get instructions and are asked to join by answering a few questions.

William A. Van Nest, the Sysop (system operator), encourages everyone, especially beginners, to join the SIG. If you're not a member, you must first join CompuServe by contacting them at 5000 Arlington Centre Blvd., P.O. Box 20121, Columbus, OH 43220 (800-848-8900 or 614-457-8650).

As with other CompuServe SIGs, you can exchange messages with other members, use the conference mode to chat and access a data library, and share programs. The data library has ten sections on such topics as Basic09, the C language, Assembly language, utilities, TeleCom, Flex, and OS-9/68000. There is also support for Cobol, Forth, and XEX. The general files (DL0) of the SIG's data library contain release forms for uploading software, section 6 (DL6) is for the Color Computer and section 4 (DL4) is for the OS-9 NUG (National User's Group).

OS-9 NUG

For a membership fee of $25 per year, payable by check or money order, you can use the OS-9 NUG's database. You receive a newsletter, *MOTD*, along with a disk of public-domain software to get you started on OS-9. The disk also contains a modem program for accessing CompuServe while operating OS-9. To obtain further information, write to The OS-9 National User's Group, P.O. Box 7586, Des Moines, IA 50322.

The general information files, DL0, of the OS-9 SIG Data Library have downloadable membership forms. They are formatted for a printer; you fill out the blanks and send them to NUG's headquarters.

Other CompuServe SIGs

The VAX SIG is of interest because many of its users run large OS-9 Level II and Unix systems, after which the CoCo's OS-9 is patterned. I have listed the PCS (Personal Computing Services) selections I find useful:

SIG	Location-Page
VAX	PCS-16
Programmer's	PCS-158
Software Author's	PCS-117
Pascal	PCS-55
C/PM	PCS-47
Microsoft	PCS-145

BBSes

The BBSes below support software for OS-9, Flex, and other languages. I have not checked all of them. By the time you read this, more boards might have opened and others closed. Please let me know of additions and changes that involve your favorite boards.

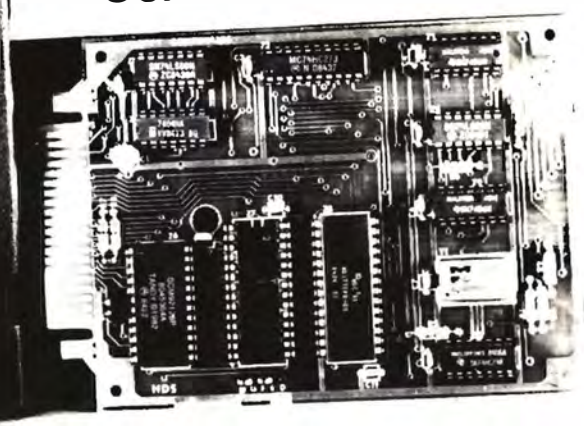
NUMBER	NAME	ADDRESS
215-277-6951	My BBS	Morristown, PA
305-274-3394	Remote OS-9	Miami, FL
312-286-9015	Skylink	Chicago, IL
312-397-8308	OS-9 User's Group	Palatine, IL
403-563-3278	T3D2 BBS	Alberta, Canada
405-728-7654	Flexnet	Oklahoma City, OK
408-984-7937	Rainbow	San Jose, CA
413-532-5631	6809er's Club	Springfield, MA
516-249-3449	CoCo Nest OS-9	Centereach, NY
803-288-0613	DLOAD OS-9 BBS	Mauldin, SC
904-378-9222	CoCo OS-9 BBS	Gainesville, FL
914-362-1422	Telemation	Pomona, NY

These BBSes are good locations to find answers to your questions. But try your local boards, too—you never know who might be using the same products in your area.

I hope the choices spur you to investigate new SIGs and find more friends, information, and clues to follow. So, get on line and get help with programming languages. And if you have on-line advice, pass it along for publication in future columns. ■

Address correspondence to Bobby Ballard, 1207 Eighth Ave. 4R, Brooklyn, NY 11215. You may also reach him on line through CompuServe (#72746,2373 or #73135,255), The Source (#BCT173), and MCI Mail (#172-3476).

2nd Generation !!!!! HDS Floppy Drive Controller Board



AS SEEN AT THE RADIO SHACK
COMPUTER SHOWCASES

Features*

- Gold Plated Edge Cards
- Dual Selectable ROM Sockets
- No POTS to adjust
- Compatible with COCO I & II
- 120 Day Warranty
- Double and Single Density

So what's so new in our second generation? We had a lot of requests for the need to use the lessor expensive 28 pin Eproms. Our 2nd generation controller allows the useage of either (two 24pin ROMS) or (one 24pin ROM and one 28pin ROM). The second feature we added was a technical one and is not apparent to the average user. Western Digital was good enough to manufacture for us a far advanced drive controller chip, called the WD1773FDC. This chip integrates the data separation method within itself allowing the cleanest data transfer to date.

Reduce your I-O errors with the Hard Drive Specialist Floppy Drive Controller for the Color Computer. Gold Edge Card Connectors. Advanced Design and the absence of potentiometers make this the best board available to date.

Completed and Tested Board with Radio Shack ROM ..... **$139.**
(Includes Case, and DOS Instructions)

Completed and Tested Board without ROM **$119**
(Includes Case)

Bare Board with instruction manual ..... **$ 40.**

Parts Kit For Bare Board without ROM ..... **$ 40.**

Radio Shack ROM ..... **$ 20.**

New Low Prices!

DISK DRIVE SYSTEMS

	SINGLE SIDED	DOUBLE SIDED
Drive 0 Complete	$249.	$289.
Drive 1 Complete	$169.	$209.
Drive 0 & 1 Dual Drive	$399.	$479.

64K UPGRADE KIT $39.95

NEW ROM

HDS has licensed the ROM from Radio Shack to be able to offer alternative operating systems pre-blown ready for installation. The first of what we hope to be a wide range of options is ADOS. ADOS is a prooduct of SpectroSytems of Miami Florida and if fully supported by the author. The HDS version of ADOS supports 2 drives, 40 track, 6ms trk-to-trk drives only, either Single Sided or Double Sided. The ADOS package comes complete with original documentation and diskette from SpectroSystems and can be installed in our Drive Controller Board at purchase time for no additional charge.

ADOS/HDS 24 pin ROM ..... **$ 50.**
(useable in all drive controllers on the market)

ADOS/HDS 28 pin ROM ..... **$ 40.**
(useable in the HDS 2nd generation board only)

Keytronics Keyboard KB500

The Fantastic Keytronic Keyboard is now being manufactured only for Hard Drive Specialist. It is the only keyboard for the Color Computer known on the market that does not use membrane switches. The KB500 uses a capacitance foam switch. This type of switch will never give keyboard bounce and last much longer than all other types. The KB500 is also the only keyboard that will fit all versions of the color computer weather it is a A, B, C, D, E, F, ET, TDP-100, COCO IIA, or COCO IIB. One keyboard fits all with out risk of getting the wrong version, and there is no need to do any modifications to your case. Additional features include a higher spring force on the break and clear key to reduce the possibility of a disastrous key-stroke, sculptured keys, low profile, 'pips' on home row keys. The 'PF' function key comes with documentation and a sample program. The Keytronics/HDS keyboard list price was $89, when it was offered through Keytronics. **Our price on it is only $69.** plus $3 for shipping.

HARD DRIVE SPECIALIST

Circle Reader Service card #455

Ordering Information
Use our Watts line to place your order via Visa, MasterCard, or Wire Transfer. Or mail your payment directly to us. Any non-certified funds will be held until proper clearance is made. COD orders are accepted as well as purchase orders from government agencies. Most items are shipped off the shelf with the exception of hard drive products that are custom built. UPS ground is our standard means of shipping unless otherwise specified. Shipping costs are available upon request.

1-713-480-6000
Order Line 1-800-231-6671
16208 Hickory Knoll,
Houston, Texas 77059

Circle Reader Service card #70

NEW!

MAROONED!

By Steve Hartford

Sitting on the back porch one afternoon, you see a strange, flashing UFO descend from the clouds & land out in the corn field. Being the curious type, you run out to investigate and find a spaceship with it's hatch open...as you step inside, the hatch closes and the ship takes off! You must find a way to get back home. A great graphics odventure! 32K & one disk drive required.

Disk or Amdek $29.95

Blackjack Dealer
Feeler Dealer

These two programs help you develop your Blackjack skill and strategy. In **Blackjack Dealer**, the computer deals the cards and plays the dealer's hand against you. **Feeler Dealer** enables you to test your strategy by playing the desired number of hands using your techniques & tendencies. A great teacher for new Blackjack players and a valuable tool for the veteran player. Both programs included. 32K extended.

Tape $24.95 Disk $29.95

EAGLE

A graphic-enhanced lunar simulator. The pilot breaks out of lunar orbit and attempts a soft landing on the lunar surface. Joysticks control thrust and craft attitude and information is continually displayed on horizontal and vertical velocities, acceleration values, vertical and horizontal distances from target, fuel consumption and much more. Diskversionallowschoice of landing site between Mars and Earth's moon. A great tool for the future astronaut or physicist. 32K, 2 joysticks required.

Tape - $24.95 Disk - $29.95

Testmaker Disk Only	$29.95
Maycode Disassembler	$24.95
TDIR Tape Directory	$24.95
Alphacopy Disk Only - 32K	$19.95
P51 Mustang 32K	$23.75
Worlds Of Flight	$23.75
Sailor Man 64K	$23.75
Trekboer 32K	$19.75
Tut's Tomb 32K	$19.75
Zookey Typing Tutor	$19.75
To Preserve Quandic Disk • 32K	$29.75
LIZPACK Disk • Stat. Anal.	$145.00
Super Screen Machine	$33.75

Add $3.00 For Disk

This Month's Special!

Sketchpad

A graphics drawing program designed to provide the computer hobbyist with easy manipulation of the powerful graphics capabilities of the CoCo. Advanced programmers can design graphics screens and characters for Basic and ML programs and games. 32K, 2 Joysticks and disk drive required.

$19.95

Reg. $29.95 - Expires 6/30/85

Amdek Dual 3" Disk Drive

NEW LOW PRICE! **$269** NEW LOW PRICE!

Includes 10 Diskettes & Cable

Amdek 3" Disks in For $25

Monitors

Amdek Color 1+	**$199**
Reconditioned - 90 Day Warranty	
Amdek Color 300	**$269**
Amdek Color 500	**$399**

Mark Data Video Driver

Works On All Cocos • No Soldering

$27.95

With Monitor Purchase - $25.95

Reverse Video Switch $11.95

$1.00 Shipping!

We'll Ship Your Order To Anywhere In The USA For $1.00

Software Only - Hardware Shipped At Cost. Outside USA - $2.00/ Program ($6.00 Max.)

Send For Our Free Catalog

Gift Certificates Available

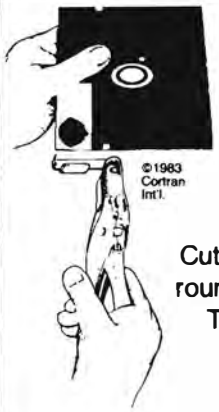
Saguaro Software

NEW ORDER LINE
602-795-4159
C.O.D. Orders Add $3.00

7331 E. Beverly • Tucson, Az. 85712
Az. Residents Add 7% Tax

Circle Reader Service card #337

DOUBLES DISKETTE CAPACITY!
CUTS YOUR DISK COST 50%!



Now! The back of 5¼" Diskettes can be used for data storage even with single head disk drives.

- ***NIBBLE NOTCH®*** Tools make it easy.
- Adds the Precise notch where it's needed.
- Doubles Diskette Space or Money Back!

NIBBLE NOTCH II

Cuts square notch and ¼" inch round "index hole." For use with TRS 80 I and III, Osborne, Kaypro, IBM and others needing an "index hole."

only **$21.90**

add $2.00 each order P&H
($5.00 foreign P&H)
Florida residents add 5% Sales Tax

SATISFACTION GUARANTEED OR YOUR MONEY BACK!

ORDER TODAY!

Toll Free 1-800-642-2536
Florida 305-493-8355
or send Check or Money Order to:

VISA • MasterCard

NIBBLE NOTCH® COMPUTER PRODUCTS
4211 NW 75th TERRACE • DEPT. 15 • LAUDERHILL, FL 33319

Circle Reader Service card #299

The complete disassembly of the Color Basic ROMs

- EASY TO UNDERSTAND • COVERS ALL ROM VERSIONS
- COMPLETE MEMORY MAP--No missing spaces, everything is defined • PROGRAMMING TRICKS EXPLAINED
- ALL FCBs, FDBs and FCCs defined
- FULLY COMMENTED--No holes

COLOR BASIC UNRAVELLED	19.95
EXTENDED BASIC UNRAVELLED	19.95
DISK BASIC UNRAVELLED	19.95
SET OF 3 UNRAVELLED BOOKS	49.95

All of the innermost operating processes are exposed. This book is an indispensable tool for the programmer seeking a full, in-depth knowledge of Basic. The UNRAVELLED SERIES will make it easy for you to write your own Basic commands or modify Basic for whatever purpose you desire.

Circle Reader Service card #299

SPECTRAL ASSOCIATES
3418 South 90th Street
1-206-581-6938
Tacoma, WA 98409

Shipping and Handling: United States add 3%, $2.00 minimum; Canada add 6%, $3.00 minimum, Foreign add 15%, $5.00 minimum. No C.O.D. orders. Washington residents add 7.8% sales tax.

YOU COULD FALL IN LOVE WITH

AUTOTERM!

IT TURNS YOUR COLOR COMPUTER INTO THE

WORLD'S SMARTEST TERMINAL

AND MOST LOVABLE

GOOD LOOKIN'

AUTOTERM shows true upper/lower case in screen widths of 32, 40, 42, 51, or 64 characters with no split words. The width of 32 has extra large letters. Scrolling is forward, backward, and fast. Block graphics pictures are displayed automatically and can be scrolled.

The screen's top line shows operating mode, unused memory size, memory on/off, and caps-lock on/off. It also gives helpful prompts.

SWEET TALKIN'

KEY-BEEP can be on/off. Unacceptable keystrokes cause a lower pitched BOP! This ERROR-BEEBOP can be on/off.

Talks to other computers with Full or Half Duplex; Baud Rate of 110, 150, 300, 600, 1200; Parity as even, odd, mark, space, none; 7 or 8 bit Word; any Stop Bits; all 128 ASCII characters; true line Break; XON/XOFF protocol; and optional line-at-a-time transmission. Able to send and receive text, block graphics, BASIC and ML programs. A 64K machine holds up to 46,600 characters (34,900 in HI-RES).

DUAL PROCESSING lets you review & edit while more data is coming in.

Fully supports D.C. Hayes and other intelligent modems.

Talks to your printer with any page size, margins, line spacing, split word avoidance. Embed your printer's control sequences for boldface, underlining, etc. Narrow text can be automatically spread out.

You'll also use Autoterm for simple word processing and record keeping

You can display directories, delete files, transmit directly from disk, and work with files larger than memory. Easily maintain a disk copy of an entire session.

Compatible with TELEWRITER (ASCII) & other word processors.

SMOOTH WALKIN'

AUTOTERM moves smoothly and quickly between word processing and intelligent terminal action. Create text, correct your typing errors; then connect to the other computer, upload your text, download information, file it, and sign-off; then edit the received data, print it in an attractive format, and/or save it on file.

Editing is super simple with the cursor. Find strings instantly, too! Any operating parameter, such as screen width, can be altered at any time. Uncompleted commands can be cancelled.

PUTTY IN YOUR HANDS

The word processor can be used to create, print, and/or save on file your personal KSMs. They let AUTOTERM act like you. For example, it can dial through your modem, sign-on, interact, perform file operations, & sign-off; an entire session without your help. KSMs can answer the phone, prompt the caller, take messages, save them, hang-up, and wait for the next call. The KSM potential is unbelievable!

NO OTHER COMPUTER IN THE WORLD CAN MATCH YOUR COCO'S AUTOMATIC TERMINAL CAPABILITIES!!!

WHAT THE REVIEWERS SAY

"AUTOTERM is the Best of Class."
Graham, *RAINBOW*, 6/83

"The AUTOTERM buffer system is the most sophisticated — and one of the easiest to use..."
Banta, *HOT CoCo*, 9/84

"Almost a full featured word processor..."
Ellers, *RAINBOW*, 11/84

RECOMMEND 32K to 64K
Tape-to-Disk Upgrade $23
You Keep the Cassette

CASSETTE $39.95
DISKETTE $49.95
Add $3 shipping and handling
MC/VISA/C.O.D.

PXE Computing
11 Vicksburg Lane
Richardson, Texas 75080
214/699-7273

Circle Reader Service card #51

Please hire the mentally retarded. They are sincere, hard working and appreciative. *Thanks! Phyllis.*

Product News

by J. Scot Finnie

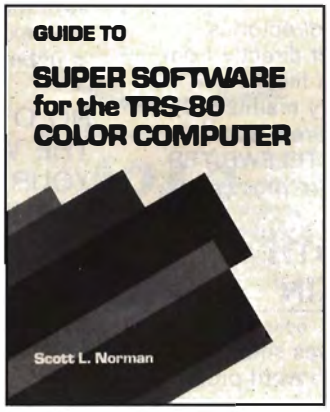
NumberJack, More Keys, Scott Norman, Morser, Try-O-Plan, MDC, Paper Catcher, Messenger Module, And More

Information printed in the Product News section is supplied by manufacturers. HOT CoCo has not tested or reviewed the products discussed here and cannot guarantee manufacturers' claims.

Topping the charts this month are two companies that have released a hardware item that could help cut your spreadsheet or database number crunching down to size. **NumberJack** from HJL Products and **More Keys** from Moreton Bay are attachable number pads for all models of the Color Computer. Both products plug in to the PIA (peripheral interface adapter) chip, providing a solderless installation.

NumberJack offers 24 keys, including the decimal and 10 numeric keys, the four cursor keys, the shift, @, comma, clear, and enter keys, and non-shifting plus, minus, multiply-by, and divide-by keys. NumberJack connects to the Color Computer by a ribbon cable that you can detach for easy movement. Its keys have bounce-proof, full-travel keyswitches with the same feel you find on the HJL-57 replacement keyboard for the CoCo. NumberJack sells for $89.95. Computers purchased after November 1984 might require a special adapter that costs an additional $3.

More Keys is a 15-key number pad that also allows rapid numeric entry. In addition to the decimal and 10 numeric keys, it offers the left-arrow, minus, comma, and enter keys. More Keys is small, light weight, and unobtrusive; it measures 6½-inches long by 4-inches wide by 3-inches high. It sells for $69.95. You must specify the model of your Color Computer when ordering.



Scott Norman's New Book

July Hot Shots

Scott Norman, a popular columnist and reviewer for *HOT CoCo*, has written a book titled ***The Guide to Super Software for the TRS-80 Color Computer.*** The book evaluates many popular applications programs along with some lesser-known offerings. The author rates each package and explains what tasks it performs well, poorly, or not at all. *The Guide to Super Software for the TRS-80 Color Computer* has 256 pages, is softbound with illustrations, and sells for $18.95. It is published by Scott, Foresman and Company.

The NumberJack From HJL Products

The Portable PC Board Layout Kit From PMC

The **MDC** is a multiple-disk-controller operating system that lets you use several kinds of disk drives with your CoCo, including 35-track, single-sided drives; 80-track, double-sided drives; and everything in between. The MDC offers several advantages and features that make it worth considering for a multi-drive application. Contact Bee Color Computer and Software for more information.

Tired of seeing ads for disk-based software that you can't use on your cassette-based system? VMC Software makes programs for cassette systems that have serious home aplications. **Color-80 PageWriter** is a cassette-based word processor with an 80-column, scrolling-text window. **Color-80 Basi-Calc** is a low-cost calculation program with a 15-row by 10-column worksheet area. These programs run on 16K CoCos in Color Basic. They sell for $18.95 each or $30 for both, plus $2 for shipping.

Try-O-Plan provides a method to help you calculate your personal finances. It can figure payments on loans, amortization tables, future values of monthly deposits, amounts needed to attain a desired annuity, and other functions. The program is manufactured by Try-O-Byte and comes on cassette or disk for $19.95 plus $2 for shipping. It requires 16K and Extended Color Basic.

The Parent-Teacher Microcomputing Sourcebook for Children 1985 is a reference work that helps parents and educators evaluate how computers

can benefit their children. It helps readers choose the best equipment and software, and find the information they need to make full use of their computers. *The Sourcebook for their Children 1985* has 846 pages and sells for $19.95. It is published by the R.R. Bowker Company.

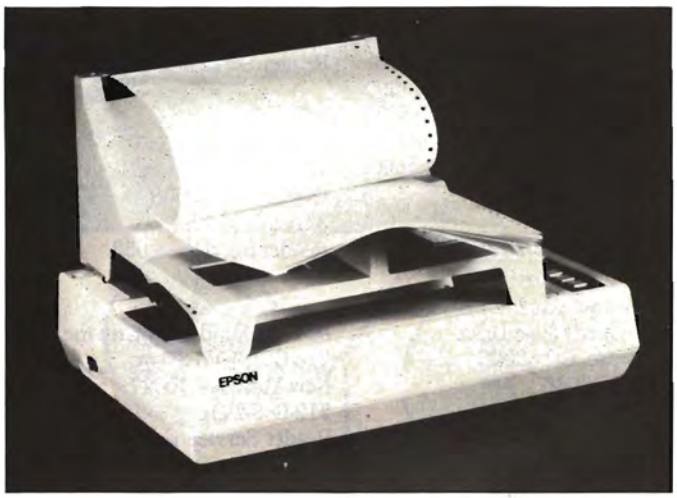
Rehab Technology has developed **Morser**, a low-cost communication system. The Morser cartridge lets you select speed of operation and change it at any time. The slowest speed allows users with limited muscle control to operate the system; the fastest speed permits printing at a greater rate than other communications devices. The Morser requires a momentary contact switch. The company sells a breath-operated switch for $50. Morser costs $195 with manual. Contact Rehab Technology for more information.

boards in the field and sells for $299. Or how about the **Paper Catcher** from Buddy Products? It stacks pages above your printer as they are printed, saving space and providing convenience. It sells for $49.95.

Aspen Ribbons is now selling replacement ribbons for Radio Shack DMP-120, 200, and 500 printers ranging in price from $5 to $10 depending on the quantity you order. The ribbons are available in red, blue, green, brown, and purple ink for an additional $2.

M. Evans and Company has released a new book for people who spend many hours at the computer screen. ***Computer Eye Stress***, by Dr. R. Anthony Hutchinson, addresses eye irritation, fatigue, and difficulty with focus and suggests methods for alleviating them. The book sells for $4.95.

Smith-Corona has come out



Buddy Product's Paper Catcher

models. But the best part about Smith-Corona's new typewriters is that the company sells a device, called the **Messenger Module**, that allows you to turn them into letter-quality printers.

Smith-Corona's XE-6000 typewriter sells for $379.

Computing Aids

Looking for something new and different? How about the **Portable PC Board Layout Kit** from PMC Industries. The PC-board kit comes with all the accessories you need to make

with a new line of electronic typewriters at a reasonable price. It offers a multiple-word, memory-correction feature that works with one keystroke per word. A 35,000-word spelling checker is an option on some

It sells for $109.95. The adapter cable (type C) for the CoCo's serial port sells for $15.95. Smith-Corona's XE line of electronic typewriters is priced from $299 through $519. Contact the company for more information.

News Bits

Imagic has relocated its corporate headquarters to 2400 Bayshore Frontage Road, Mountain View, CA 94043. The company's new telephone number is 415-940-6030.

Thundersoft has gone out of business, but wishes to honor orders previously contracted by customers. If you have sent money to Thundersoft and not received your order, contact Richard Parkey at Route A, Box 300, Silver City, NM 88061.

Spectrum Projects' president Bob Rosen has announced that his company is returning its center of business to the east coast after a year-long relocation on the west coast. Spectrum Projects has consolidated the business of all its offices to its original New York location. To contact the company, write or call Spectrum Projects Inc., 93-15 86th Drive, P.O. Box 21272, Woodhaven, NY 11421, 718-441-2807.

Computer Systems Center, makers of DynaCalc and Dynamite+, has relocated its operations to 42 Four Seasons Center,

Suite 112, Chesterfield, MO 63017, 314-576-5020. *HOT CoCo* printed an incorrect telephone number for the company with the review of Dynamite + in the May, 1985 issue.

Product News recently received notice of the startup of a new Color Computer magazine, called *Motley CoCo*. It will feature programs, classifieds, tips, and reviews. Issues will cost 40 cents each. To subscribe for five issues, send $2 to Phil Orton, 8 Lakeside Court, Grosse Pointe, MI 48230 or call 313-881-3330. *Motley CoCo* will be available to subscribers in the 48 contiguous states. ■

List Of Manufacturers

Aspen Ribbons Inc.
555 Aspen Ridge Drive
Lafayette, CO 80026
800-525-0646
800-228-3073 (in CO)
Reader Service ✔ 550

Bee Color Computer and Software
P.O. Box 5609
Aloha, OR 97006
503-649-4497
Reader Service ✔ 551

R.R. Bowker Company
205 East 42nd St.
New York, NY 10017
212-916-1600
Reader Service ✔ 552

Buddy Products
1350 South Leavitt St.
Chicago, IL 60608
312-733-6400
Reader Service ✔ 553

M. Evans and Company Inc.
216 East 49th St.
New York, NY 10017
212-688-2810
Reader Service ✔ 554

HJL Products
955 Buffalo Road
Rochester, NY 14624
716-235-8358
Reader Service ✔ 555

Moreton Bay Software
316 Castillo St.
Santa Barbara, CA 93101
805-962-3127
Reader Service ✔ 556

PMC Industries Inc.
9353 Activity Road, Suite K
San Diego, CA 92126
619-695-0645
Reader Service ✔ 557

Rehab Technology Inc.
P.O. Box 185
Aviston, IL 62216
618-228-7722
Reader Service ✔ 558

Scott, Foresman and Company
1900 East Lake Ave.
Glenview, IL 60025
312-729-3000
Reader Service ✔ 559

Smith-Corona
65 Locust Ave.
New Canaan, CT 06840
203-972-1471
Reader Service ✔ 560

Try-O-Byte
1008 Alton Circle
Florence, SC 29501
803-662-9500
Reader Service ✔ 561

VMC Software
P.O. Box 326, Cambria heights
New York, NY 11411
718-527-5298
Reader Service ✔ 562

CONTROL LIGHTS AND APPLIANCES 24 HOURS A DAY AUTOMATICALLY

Program the Household Controller Using Your Color Computer 2

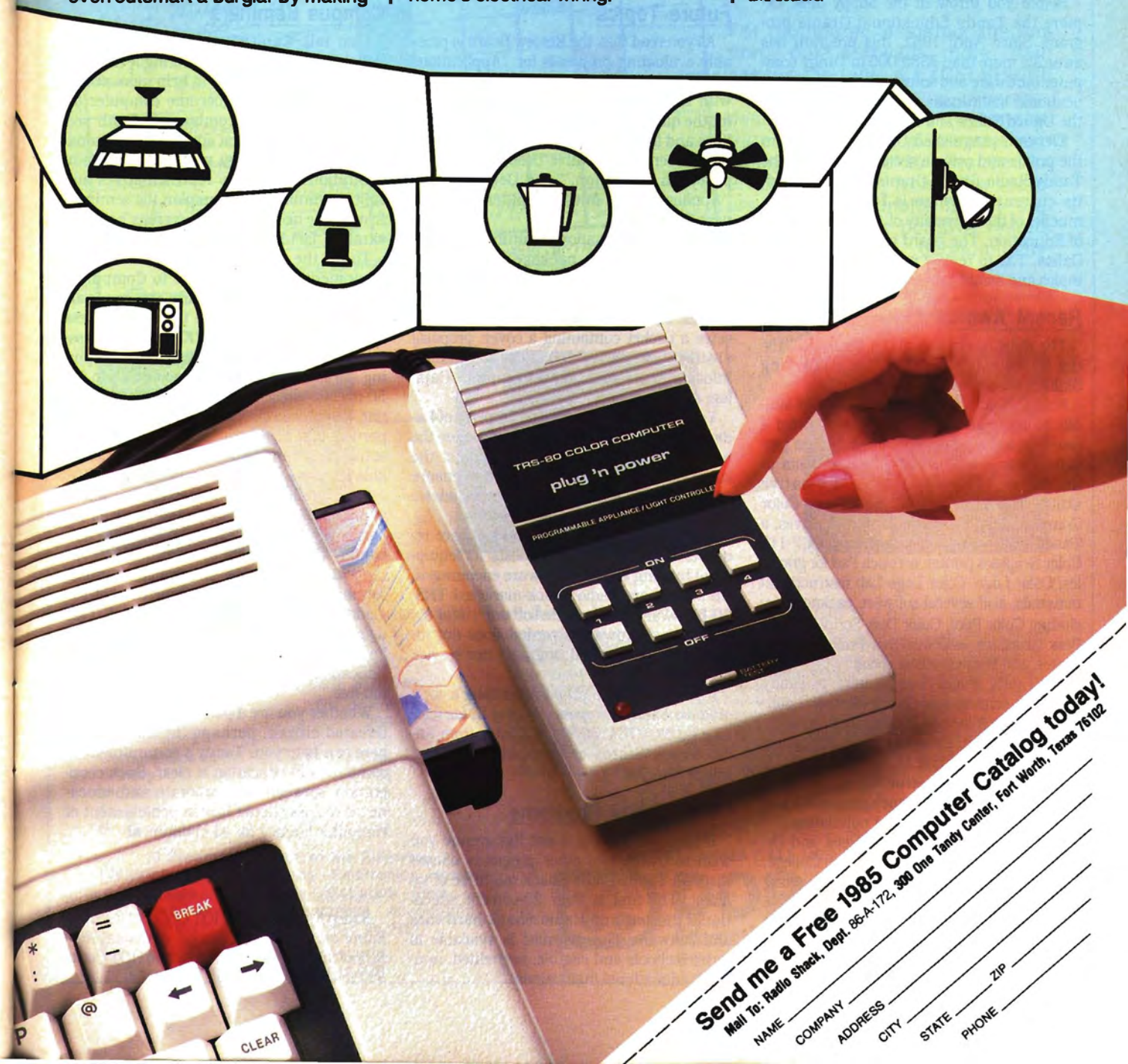
Our Appliance/Light Controller is an incredible device that makes your life easier. It can wake you to a TV or stereo, brew your coffee in the morning, pre-cool a room in the afternoon, start dinner before you leave work, and hundreds of other chores—all automatically. You can even outsmart a burglar by making your home look and sound lived in day and night.

Simply connect the Controller to your Color Computer, and enter the desired times and events. Disconnect the Controller and your computer is free to use as you wish. Then, connect lamps and appliances to Plug 'n Power™ modules (sold separately) and plug the modules into wall outlets. The Controller sends on/off signals over your home's electrical wiring.

Get the Appliance/Light Controller (26-3142, $99.95) at Radio Shack today—and computerize your home for improved security, convenience and energy savings.

Radio Shack®
The Technology Store™
A DIVISION OF TANDY CORPORATION

Circle Reader Service card #4
Prices apply at participating Radio Shack stores and dealers.



Send me a Free 1985 Computer Catalog today!
Mail To: Radio Shack, Dept. 86-A-172, 300 One Tandy Center, Fort Worth, Texas 76102

NAME _____
COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

The Learning Page

by Nancy Kipperman

Tandy's Role In Education

Picture yourself as a teacher or administrator in a nonprofit or public school. You carefully develop an idea for using microcomputer technology in your classroom or district. But due to local budget limitations, your dream might never become a reality. Is it time to give up?

Don't Quit Yet

Before you throw in the floppy disk, explore the Tandy Educational Grants program. Since April 1982, this program has awarded more than $885,000 in Tandy computer hardware and software to nonprofit educational institutions and public schools in the United States and its territories.

Eleven distinguished U.S. educators from the public and private sectors constitute the Tandy/Radio Shack Grants Review Board. Its current chairman is Dr. Lee Droegemueller of the University of Arizona's College of Education. The board meets quarterly in Dallas, TX, to review new proposals on set topics and award grants.

Recent Awards

The most recent awards on the topic, "Using Microcomputers to Develop Thinking Skills," were announced in February.

Mrs. Margaret Perry of Safety Harbor Middle School, Safety Harbor, FL, submitted a proposal to "establish a model program providing computers to aid gifted students in improving their thinking and creative skills." The award was 11 64K Tandy Color Computers with monitors and disk drives, a Tandy DMP-110 printer, a Tandy CGP-115 Color Graphics printer, a Touch Pad for graphics, Color Logo, Color Logo Lab instructional materials, and several software packages, including Color Pilot, Color Disk Scripsit, Color Disk Graphics, a Color Disk Profile, GameWriter, and Kidwriter Storybook.

Six Tandy Model 4 computers, a Tandy DMP-115 printer, and Tandy Author I software went to Aileen S. Johnson of Pan American University, Brownsville, TX, for her proposal to "evaluate the effectiveness of computer-assisted instruction designed to raise the reading competency scores of a predominately Hispanic student population."

The proposal of Dr. Ronald Reed and Dr. Mary Jane Garza of Texas Wesleyan College, Forth Worth, TX, to "increase the reasoning skills of bilingual students by developing a computer-assisted instructional program that incorporates the methodologies of Matthew Lipman" won an award of three Tandy Model 4 computers, two Tandy DWP-115 printers, and Tandy Author I software.

Winifred Green, president of the Southern Coalition for Educational Equity in Atlanta, GA, proposed to "use computer software to develop and implement instructional units for teaching critical thinking skills to minority and disadvantaged seventh and eighth graders" and received five Tandy Model 4 computers and instructional materials.

Future Topics

As you read this, the Review Board is probably evaluating proposals for "Applications of Microcomputers in the Social Sciences with Emphasis on Simulations" submitted for the quarter that ended on June 30. Deadlines and topics for the next two quarters are September 30, "Creative Uses of Microcomputers in Education," and December 30, "Applications of Microcomputers in Business Education."

To obtain information on future topics and a grant-application package, write to the Tandy Education Grants Program, Radio Shack Education Division, 1400 One Tandy Center, Fort Worth, TX 76102. You will receive a packet containing a cover, proposal abstract, evaluation schedule, submission information, Radio Shack Computer Catalog, and folder.

Your proposal becomes the property of Radio Shack and cannot be returned. Before the board sees each proposal, the name of the submitting institution is removed to ensure impartiality. Radio Shack employees advise the board in a nonvoting capacity.

The limit for equipment and software requests is $10,000, but requests for equipment totaling $8,000 or less are encouraged. All awards are Radio Shack-marketed TRS-80 hardware, courseware/software, and accessories. Hardware awarded does not include service-installed upgrades nor service contracts.

If you are new to proposal writing, a copy of *Radio Shack's Proposal Writing Guide* (catalog no. 26-2754, $9.95), available from Radio Shack Stores or Computer Centers, might help you.

Other Tandy Programs

If proposal writing is not the avenue you wish to travel, two other programs offered through Tandy/Radio Shack might be of interest. The first is their 20-percent educational discount on Radio Shack hardware and software. The discount is available to public schools and eligible, accredited, nonprofit educational institutions.

The second is free programming instruction for all educators who preregister at one of the more than 400 Radio Shack Computer Centers in major metropolitan areas. Training includes three courses of eight to ten hours each.

Obtain information on these two programs at your local retail Radio Shack outlet or Computer Center.

Campus Seminars

Last fall, Tandy/Radio Shack's Educational Division began sponsoring a two-day, free computing seminar to help more people on college campuses become computer literate. The program combines in-depth presentations on the latest advances in personal computing technology with hands-on demonstrations. Company representatives and college administrators preplan the seminars to meet the needs of each university's administrators, faculty, staff, and students.

To put the noncomputer user at ease with equipment, an Introduction to Computers session is presented during the seminar. Other formal sessions can include Business Simulations, the Model 100, Word Processing, and Spreadsheet Analysis. The computing seminar schedule may include up to 10, 50-minute presentations. During the seminar, a microcomputer display is open so that participants can try different systems and software. Tandy representatives are available to answer questions.

Small groups whose interests lie in specific applications of hardware and software for departments or specialized schools may also confer with Tandy representatives or arrange demonstrations. To arrange a campus seminar program, contact your nearest Regional Educational Coordinator or call Tandy's Educational Division at 817-390-3832. For the name and number of your Regional Educational Coordinator call 800-433-5682, toll-free.

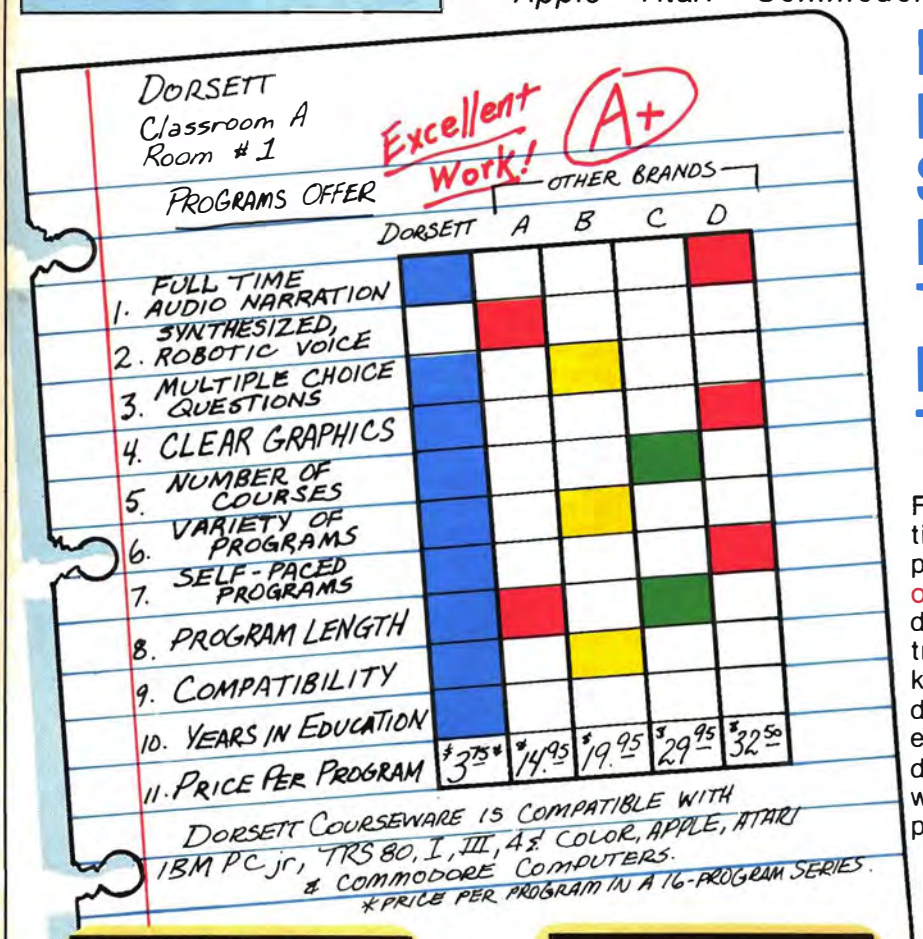
Whether you are a parent, educator, or interested citizen, perhaps the information here can help you. Tandy's commitment to computers in education is clear. Such cooperation between the corporate and educational sectors should hasten achievement of computer literacy for all children. ■

Nancy Kipperman is HOT CoCo's Education Editor and an English teacher at Conant High School in Jaffrey, NH. Write her c/o HOT CoCo, 80 Pine St., Peterborough, NH 03458.

AVAILABLE NOW FOR **IBM PC Jr.**

For Your TRS-80 Color Computer

Apple • Atari • Commodore • TRS 80 I, III, 4 & Color



DORSETT EDUCATIONAL SOFTWARE PASSES THE COST-EFFICIENCY TEST!

For computer assisted instruction, our software more than passes the test...it EXCELS! With over 1000 educational programs* designed specifically to concentrate learning for all users, from kindergarten level to advanced individual studies, from slow learners to exceptionally bright students, you'll get positive results with our quality, economically-priced courseware.

* 500 Programs Available for TRS 80 Color; 400 for IBM PC Jr.

Interactive Tutorial Programs

Over 1000 Programs with Full Time Audio Narration, Pictures & Text!

We're Your Educational Software Source for Grades K-12 and Advanced Studies

Course	No. of Programs
Reading	256
Mathematics	128
Comprehension	48
History	32
Algebra	16
Spelling	16
Government	16

16 Programs in each of the following:
Children's Tales - Carpentry - Electronics -Health Services - Office Skills - Statistics - First Aid/Safety - Economics - Business - Accounting -Psychology - AND MANY MORE!

Dealer Inquiries Welcome
Circle Reader Service card #209

For your Atari 400/600/800/1200, you will need the Atari Cassette Recorder and the Dorsett 4001 Educational Master Cartridge, **$9.95**. For your Apple II, you will need the Dorsett M402 T/T plug-in-board and M401 stereo cassette player, **$99.00**. All programs listed are available for TRS-80, I III, IV, which require the M203 speaker converter and the 401 stereo cassette player, **$99.00**. For the PC Jr., a cassette adapter cable and a good cassette recorder is required. A Radio Shack CCR-81 or CCR-82 is recommended. For your Commodore 64, you will need the Dorsett conversion kit T/T plug-in-board and a stereo cassette. **$99.00.**
$59.90 for an album containing a 16-program course (8 cassettes with 2 programs each at **$3.75** per program).
$8.00 for a 2-program cassette.

Send for a catalog of over 1000 programs for Atari, TRS 80, Apple, IBM PC Jr., Commodore, etc.

For more information, or to order call:

TOLL FREE 1-800-654-3871
IN OKLAHOMA CALL (405) 288-2301

DORSETT
Educational Systems, Inc.
Box 1226, Norman, OK 73070

MasterCard VISA

The Corner Office

by Jeff DeTray, Publisher

The Key To Programming

This month's cover feature started me thinking about my own first experiences in learning computer languages and programming techniques. Like most novice programmers, my college classmates and I were long on syntax and short on problem-solving skills. Sure, we knew the commands and statements, and after a while, writing error-free code wasn't too difficult. But there is a big difference between a program that runs without syntax errors and one that runs efficiently or even accurately. Our instructors failed to stress that the key to programming lies in discovering how to solve the problem, not writing the lines of code. In fact, once you have determined the correct approach, writing the program itself is almost trivial, albeit very time consuming.

Learning a new computer language is much like learning a new human language. With either, you begin by memorizing vocabulary and learning rules of grammar. From there, you progress to simple compositions, and eventually to more complex efforts. But just as memorizing the dictionary does not necessarily make you a writer, simply memorizing the vocabulary and syntax of a computer language does not necessarily make you a programmer. If you don't think like a novelist, no amount of vocabulary can make you one.

We beginning programmers were required to draw flowcharts to show how we intended to solve each programming problem. Like a writer's outline, a flowchart or something like it forces you to think through the solution before you write a single program line. With a good plan of attack, you can write the program itself in any computer language, assuming you have a grasp of the syntax and vocabulary. No one bothered to explain this at the time, though, and flowcharting seemed like so much busy work. Thus, the favorite technique was to plunge headlong into the programming, using brute force to come up with a working, but extraordinarily sloppy, program. Then, and only then, we would go back and draw up a flowchart based on the program we had already written. We thought that was pretty clever.

If your intent is to become proficient in a variety of programming languages, then make it a habit to develop a complete and reasonably detailed plan for every major programming project. It may seem a chore at first, but it helps you develop a set of skills that are readily transferrable from one language to another, whether it's Pascal, C, Logo, Basic, or even Assembly. Anyone can memorize the vocabulary. It's problem solving know-how that makes a programmer.

EOS Impressions

Have you made use of Express Order Software? If so, we'd like to hear from you. We want to know your opinion of this unique software distribution idea.

EOS is one way in which Tandy is attempting to make it easier for you to obtain software for your Color Computer, or any other Tandy computer, for that matter. EOS software comes from a wide variety of software companies, including some *HOT CoCo* advertisers. Rather than stocking these many programs on the shelves on each Radio Shack store and Computer Center, Tandy holds EOS products in its regional warehouses. When you walk into a Radio Shack store and ask for an EOS program, the store manager places an order with the warehouse, and the program is sent to the store, ready for your pickup a few days later.

Tandy is still refining EOS. In an effort to speed up the delivery of ordered products to the stores, alternative shipping arrangements are being tested. One problem still in need of solution is how to keep customers (you and me) apprised of what's available via EOS. With well over 200 programs already in EOS, and more being added every week, some type of constantly updated on-line catalog might be appropriate. Right now, the software companies themselves are doing their best by putting their EOS catalog numbers in their advertisements, enabling you to ask for the product by its Radio Shack number. But the stores really ought to have a master catalog or database to make the process easier.

As this is written, EOS contains only a handful of Color Computer programs, but more are on the way. If you have obtained any software through EOS, please take a moment to write and tell us about it. What did you order? Were the store personnel helpful? How long did it take you to receive the product? Would you use EOS again? Send your comments on EOS or any other topic to:

Jeff DeTray
HOT CoCo
80 Pine Street
Peterborough, NH 03458

Cards and Letters

Speaking of comments, we have received a bunch of them in response to our May issue. Many of you expressed strong opinions on the new look of *HOT CoCo*, and I'm happy to say that the verdict is favorable. Editor Mike Nadeau discusses this in greater depth in his Digressions column on page 4, and we'll be publishing your letters on this topic in coming issues.

Both Mike and I appreciate and read each and every letter. Your opinions are vital to us. They help determine what new topics we will explore and which features are most popular. You probably have a word processor, so use it! Tell us what's on your mind.

And don't forget to tell advertisers, "I saw it in *HOT CoCo*." ■

The HJL-57 Keyboard

Now available for all models, including CoCo 2.

Compare it with the rest. Then, buy the best.

If you've been thinking about spending good money on a new keyboard for your Color Computer, why not get a good keyboard for your money?

Designed from scratch, the HJL-57 Professional Keyboard is built to unlock ALL the potential performance of your Color Computer. Now, you can do real word processing and sail through lengthy listings...with maximum speed; minimum errors.

At $79.95, the HJL-57 is reasonably priced, but you can find other CoCo keyboards for a few dollars less. So, before you buy, we suggest that you compare.

Compare Design.

The ergonomically-superior HJL-57 has sculptured, low profile keycaps; and the three-color layout is identical to the original CoCo keyboard.

Compare Construction.

The HJL-57 has a rigidized aluminum baseplate for solid, no-flex mounting. Switch contacts are rated for 100 million cycles minimum, and covered by a spill-proof membrane.

Compare Performance.

Offering more than full-travel, bounce-proof keyswitches, the HJL-57 has RFI/EMI shielding that eliminates irritating noise on displays; and four user-definable function keys (one latchable), specially-positioned to avoid inadvertent actuation.

Free Function Key Program

Your HJL-57 kit includes usage instructions and decimal codes produced by the function keys, plus a free sample program that defines the function keys as follows: F1 = Screen dump to printer. F2 = Repeat key (latching). F3 = Lower case upper case flip (if you have lower case capability). F4 = Control key; subtracts 64 from the ASCII value of any key pressed. Runs on disc or tape; extended or standard Basic.

Compare Installation.

Carefully engineered for easy installation, the HJL-57 requires no soldering, drilling or gluing. Simply plug it in and drop it right on the original CoCo mounting posts. Kit includes a new bezel for a totally finished conversion.

Compare Warranties.

The HJL-57 is built so well, it carries a full, one-year warranty. And, it is sold with an exclusive 15-day money-back guarantee.

Compare Value.

You know that a bargain is a bargain only so long as it lasts. If you shop carefully, we think you will agree...The HJL-57 is the last keyboard your CoCo will ever need. And that's **real** value.

Order Today.

Only $79.95, the HJL-57 is available for immediate shipment for either the original Color Computer (sold prior to October, 1982) or the F-version and TDP-100 (introduced in October, 1982), and the new 64K CoCo. **Now also available for CoCo 2.**

Call Toll Free

1-800-828-6968

In New York 1-800-462-4891

HJL PRODUCTS

Div. of Touchstone Technology Inc.
955 Buffalo Road • P.O. Box 24954
Rochester, New York 14624

Telephone: (716) 235-8358

Ordering Information: Specify model (Original, F-version, or CoCo 2). Payment by C.O.D., check, MasterCard or Visa . Credit card customers include complete card number and expiration date. Add $2.00 for shipping ($3.50 for Canada). New York state residents add 7% sales tax.
Dealer Inquiries Invited.

Circle Reader Service card #440