COMPUTE!

The Leading Magazine Of Home, Educational, And Recreational Computing

COMPUTERS: POWERFUL NEW MUSIC MACHINES

Digital Technology Comes Home

DAZZLING GRAPHICS!

The Quick And Easy Way To Spectacular Screens

- IBM PC Math Graphics
- Color Pad For The 64
- Amiga Math Draw

Casino Blackjack
Las Vegas card game for 64, Apple II, PC, Atari, and Amiga

Out

In

Midi

www.commodore.ca

www.commodore.ca

COMPUTE!

The Leading Magazine of Home, Educational, and Recreational Computing

NOTE: See page 88 before typing in programs.

AP Apple, GS Apple IIGS, Mac Macintosh, AT Atari, ST Atari ST, AM Amiga, 64 Commodore 64, 128 Commodore 128, PC IBM PC, PCjr IBM PCjr. • General interest.

www.commodore.ca

Editor's Notes

We're just back from the Winter Consumer Electronics Show (WCES) in Las Vegas, a huge display of all the latest televisions, audio and video systems, VCRs, computers, software, electronic musical instruments, printers, and other technological gadgets that will be headed your way in catalogs and on store shelves over the next 12 months.

The computer and software section was dominated by the massive exhibit from Nintendo, the Japanese company that now controls 70 percent of the billion-dollar-plus videogame market. Only slightly smaller was the Sega booth, another videogame giant. The startling resurgence of that segment of the industry during the past year and a half is remarkable when you consider that the business was pronounced dead just four years ago. With dozens of game cartridges—many of them conversions of computer software products—Nintendo and Sega attracted a lot of attention at the show.

Despite the flashy presence of the videogame companies, there were scores of new computer software products being demonstrated at the show and in hotel suites throughout Las Vegas by dozens of companies. And there were significant displays of several IBM-compatible computer systems from companies who are offering good prices and loads of options.

The presence of Apple, Atari, and Commodore at WCES, on the other hand, was largely oblique. While virtually every major software publisher was showing new and exciting products for the 64 and the 128, Commodore sponsored no floor display. The same could be said for Atari, which had a suite away from the show floor and sponsored ads in the daily show magazines for its Atari game systems. Despite Atari's emphasis on its lucrative game machines, there was still a variety of new software packages for the ST computers and a smaller number for the Atari XL/XE systems.

Unlike Atari and Commodore, Apple Computer and IBM have never considered the CES shows necessary for distribution of their computers. It appears now that, at least for the Winter CES, Atari and Commodore are adopting the same attitude. They have apparently decided that their resources are better spent by promoting Commodore Amigas and Atari STs at the semiannual Computer Dealers Exposition (COMDEX) in Las Vegas and Atlanta, leaving their machines to be promoted at WCES through the software developed and peddled by third-party companies.

And that, in a way, is unfortunate. While Nintendo and Sega are delightful systems, they may be seizing the lion's share of the game market by default. There's no question that Apple, Atari, Commodore, and now IBM-compatible systems are all terrific game machines. What's more, these machines carry an added consumer benefit that the dedicated game systems simply can't offer: As computers, they're far more than just game-playing devices; they are *anything* machines. Not only can consumers play games on their computers, they also can take a break from gaming and use the machines for word processing, spreadsheets, paint programs, telecommunications, and all the other software applications that have made the computer such an influential piece of technology.

We would love to see Atari and Commodore return to both CES shows in the future, if only to continue to remind the industry assembled there just how good their computers are. But whatever eventually happens on that score, the more important point is that software, peripheral, and new hardware companies are fueling a consumer computer market that's still growing. In next month's issue, we'll take an in-depth look at some of the most significant stories from WCES.

Keith Ferrell and Selby Bateman

Computers:

The Powerful, New Music Machines

Last year, for the first time, electronic musical instruments outsold acoustic ones. Turn on a radio or television, and it's impossible to miss the change that is happening in the musical world: The electronic chip and the computer are joining the violin, piano, and guitar in the ranks of serious musical instruments. While some would rather ignore its presence, many musicians are embracing the technology that provides them with a new tool for composing, recording, and playing music.

Tom Netsel, Assistant Features Editor

In 1983, Americans bought more than 233,000 pianos and 392,000 digital keyboards. Two years later, piano sales had declined by almost half, while the demand for their electronic counterparts more than doubled. Last year, electronic keyboard sales topped two million. Of the $3 billion spent on musical instruments in 1987, more than half went for the purchase of electric or electronic instruments.

Digital keyboards, electric guitars, electronic violins, drum machines, synthesizers, and sound samplers are being connected together and controlled by computers to create new sounds impossible to play on traditional instruments. In addition to creating new sounds, computers can bend, shape, compress, and stretch musical notes to imitate traditional instruments or whole orchestras in ways never imagined by musicians just a few years ago.

Today's Folk Music
Jon Appleton, professor of music at Dartmouth College, is not surprised by the shift in popularity from traditional acoustic instruments to electronic ones. "Digital instruments are the folk instruments of our time," he says. "That's what people are making music on today."

Some critics contend that computers cannot make music, arguing that electronic parts and components produce an unnatural or artificial sound. In the past, musicians made music by four basic methods: striking an object (piano or drum), rubbing two items together (violin or cello), blowing air through an object (trumpet or bagpipe), or plucking a string (guitar or harpsichord). Today, these methods have been extended vastly by digital technology.

Digital Sound
When sound is recorded on a compact disc or digital tape, audible vibrations enter a microphone and are converted into numbers (digits). These numbers can be stored and played back without being affected by dust, warpage, or other problems that can plague conventional analog recordings.

Once stored, the digits can be processed, and that's what a computer does best. A computer can change the pitch, adjust the volume, or manipulate the waveform of the sound in dozens of ways, depending on its programmed instructions.

A Musical Revolution
Computers such as the Commodore 64 and the Apple IIGS have internal sound chips that lend themselves well to musical applications, but most personal computers can be used to control external digital musical instruments. This was made possible by the introduction of MIDI.

In the early 1980s, the giants of the electronic music industry got together to develop a standard that would enable digital musical instruments to exchange data. The hardware and software specifications the manufacturers agreed upon is called MIDI (Musical Instrument Digital Interface). This standard, released in August 1983, enabled musicians to control multiple synthesizers, sequencers, and sound samplers, all made by different manufacturers, from a single control unit.

Previously, if you wanted to play something on an Oberheim keyboard, a Yamaha synthesizer, and a Roland drum machine, you needed an extra set of hands or more musicians. It was impossible to connect instruments made by various manufacturers in

any productive manner. With MIDI, it's now possible to play up to 16 different synthesizers or drum machines from one controller or keyboard. This standard has permitted digital music to evolve at a tremendous rate. If the major computer makers showed such cooperation, you would be able write a program on your IBM, save it to disk, and then run it on a friend's Atari.

MIDI And Computers

While MIDI was not developed with personal computers in mind, their processing capabilities make them ideal controllers for digital instruments. Software and hardware manufacturers jumped on the MIDI bandwagon, and, before long, interfaces and programs for IBM, Commodore, Atari, and Apple were on the market. Now there are hundreds of programs for sequencers, samplers, controllers, librarians, and printers designed for the amateur musician and for the seasoned professional.

Using such a system allows one musician to compose, record, edit, and play multiple-voiced compositions that can sound like traditional instruments or have electronic voices never heard before. Professional artists, impressed with MIDI's creative flexiblity, push the technology to new limits. Peter Gabriel recently contacted Frank Foster, president of Hybrid Arts, which is a major supplier of MIDI software for Atari. Gabriel was impressed with the Atari ST, which has a built-in MIDI port, and he recently added one to his studio. Foster says Gabriel was particularly interested in Hybrid's *Android* programs, which can supply a musician with hundreds of new sounds.

Each week Jan Hammer composes and records the music for *Miami Vice* at his home studio in New York. An IBM PC XT helps Hammer control an array of synthesizers and digital instruments.

Philip Glass, Frank Zappa, Pat Metheny, Mike Pinder, Stevie Wonder, and Herbie Hancock are just a few of the musicians who have turned their talents to digital technology.

While the professionals have a wide range of powerful (and expensive) equipment at their disposal, the amateur musician or computer hobbyist can still assemble a sophisticated MIDI system for a modest cost. The Casio CZ-101 and the Yamaha DX-100 are two popular MIDI-equipped keyboards priced at $499 and $445, respectively, but which can be found discounted to about half those prices.

On the other end of the price scale is New England Digital's Synclavier Digital Music System. This top-of-the-line instrument—chosen by Stevie Wonder, Frank Zappa, Sting, Leonard Bernstein, and Chick Corea—stores, records, reproduces, or synthesizes sounds instantly at the touch of a button. As many as 304 different sound effects can be active on the keyboard at one time. Prices begin at $60,000, but more sophisticated models are priced a few dollars higher.

Synclavier And Apple

Last year, New England Digital donated a $500,000 Synclavier to Dartmouth College's Bregman Electronic Music Studio. The studio was created in 1968 when computer wizard John Kemeny, father of the BASIC programming language, was president of the college. The Synclavier is the center of one of the most advanced music instruction studios of its kind in the United States.

"Apple Computers donated a Macintosh Plus for each of 16 student workstations," says Professor Appleton, co-director of the studio, "but Apple is upgrading the gift to Macintosh IIs."

Appleton, who pioneered work on the digital synthesizer with an engineer at Dartmouth and a student who later formed New England Digital, is himself a programmer and an electronic composer with more than 30 compositions recorded on Folkways and Flying Dutchman labels. He formed the class around the Synclavier and the Macintoshes to teach students about music made with computers and digital devices and how to compose on them.

Students have complete control of the Synclavier from their workstations, which are equipped with the Mac II and a Casio CZ-1 MIDI keyboard. Compositions can be played or edited instantly using the Syncla-

www.commodore.ca

Professor Jon Appleton and students analyze a violin's waveform during a session of Dartmouth College's Electro-Acoustic Music Program.

vier's synthesized or sampled sounds. The results can be heard, but more importantly, the other students can use their monitors to see what is happening.

"Any of the 16 terminals may be designated as the master terminal, thereby enabling any of the other 15 to see exactly what the master is doing," Appleton says. "They can very rapidly move through different programs and types of music and see what everybody's done instead of having to call up one program at a time and have people huddle around one terminal."

Music students and computer students work together at Dartmouth. The goal of the computer science students is to develop the software that will make it as easy as possible for the musicians to access the synthesizer's capabilities. "More often than not," Appleton says, "musicians are not used to working with computers. It's important to develop a front end for the (Synclavier) system which is user-friendly in the best sense of the word—not simple, but one that takes into consideration the way musicians approach computers."

An Invisible Workstation
Appleton's goal is to train young composers and to introduce them to computer systems that are flexible. A professional system such as the Synclavier is powerful but very complex. The undergraduates at Dartmouth work with engineers at New England Digital to produce software that will enable a composer to use the Synclavier's power without having to devote most of his energy trying to master the system. When a composer has an idea or an elusive piece of melody that he or she wants to capture, the ideal workstation would be nearly invisible. It would allow the composer to move through the various computer tasks instantly, so his or her energy could be devoted to composing or performing.

"It doesn't really matter what instrument a person plays," Appleton says. "There are very effective MIDI interfaces for keyboards, guitars, and wind instruments. If a traditionally-trained musician comes to such a system, he should be able to generate what we call note lists in the manner most comfortable for him and not have to type in code."

Digitized music is here and its impact on music is expected to be even greater in the future. Synthesizers and electric guitars are the instruments of our times and the instruments most played by young musicians. Appletons feels young people are approaching music with a more playful and inventive bent. Whether they play a traditional instrument or a modern one, the computer and a MIDI interface are tools to help them create music, record it, and play it back flawlessly, enabling them to realize their musical potential more easily.

Goodbye, Guitar

In addition to being a tool that can free musicians from mechanical chores that stand in the way of producing music, computers with built-in voices are themselves used as musical instruments. If music is defined as a series of pleasing or harmonious sounds, then the determination of what is pleasing lies in the ear of the beholder. Lorne Strider is one musician who turned to a computer and liked what he heard.

Strider is an accomplished classical guitarist who once vowed that he had no need for a computer. He practiced his guitar in the evening, performed occasional concerts with another guitarist, and made a living selling real estate. When he decided a computer could help his business, he bought a Commodore 64. His music was never the same.

Fascinated by the 64's SID (Sound Interface Device) chip, Strider tried a variety of programs, looking for ones to demonstrate the computer's capacity for reproducing music. "I started ordering any music disks that came on the market," Strider says. "Eventually, I realized that none of them involved much effort or art."

Disappointed by the commercial programs he found, Strider took up programming as a hobby. He spent his evenings at the keyboard trying to coax Bach's fugues and Mozart's sonatas from his computer. His guitar gathered dust.

Strider's musical efforts succeeded, and he thought others might enjoy listening to classical music played on a computer. Ever the entrepeneur—he once proposed opening a branch of his real estate office on the moon—Strider launched a new enterprise called Strider's Computer Classics. He programs the works of master composers on disks that play through the computer's three voices.

Well-Tempered Computer

Strider now has ten disks that range from Renaissance music of the 1500s, through Mozart, Bach, and Clementi, to the turn-of-the-century ragtime sounds of Scott Joplin. The disks, which sell for $9.95, contain 7–15 selections and provide up to an hour of music. Each musical selection has four screens of information about the composer, his life, and his music.

Strider does the historical research as well as the musical programming. For example, the disks of J. S. Bach's *Two-Part Inventions*, *Three-Part Inventions*, and *Well-Tempered Clavier* each contain 15 selections and a total of 60 screens of text.

The real-estate broker and musicologist is somewhat surprised to find himself a software developer, albeit not one of the big guns of the industry. "It took me two or three months to design the basic structure of the disk," he says, "but once that was worked out, all I have to do is write the music files, then change the text and menus." His latest disks take about a week to complete. They are available from several distributors, including Free Spirit Software.

Strider, who calls himself more of a "whacker" than a "hacker" when it comes to programming, credits Jeff Johnson's *Virtuoso 64* for helping him with the mechanical chores of creating music on a computer. The software produces code that can be used to create sound effects or background music for any type of BASIC or machine language program. "It's a no-frills music processing utility, but it is very powerful," he says, "and it serves its purpose completely." *Virtuoso 64* is distributed through Chipmunk Software.

Strider is content with his new form of musical expression, but his former partner, Paul Brower, was disappointed when the guitar duo disbanded. Brower, who is a bit derisive about computer music, found a cartoon and sent it to his former musical partner. It shows a concert stage with two musicians bowing to the applause of the audience. One musician is holding a classical guitar, while the other holds a computer. Brower added a caption that reads, "Paul and Lorne go on tour again." ©

Chipmunk Software
P.O. Box 403
Battle Ground, WA 98604

Free Spirit Software
538 S. Edgewood
LaGrange, IL

Hybrid Arts
11920 W. Olympic Blvd.
Los Angeles, CA 90064

Strider's Computer Classics
P.O. Box 554
Laytonville, CA 95454

Buyer's Guide To Tax Preparation Software

Caroline D. Hanlon

April 15th always comes too fast. Home tax preparers with personal computers, though, have a powerful tool to aid in computing their returns. As this buyer's guide shows, there is a wide range of software available for a variety of tax situations, including a program for Canadian taxpayers. This guide includes software for home users preparing Form 1040 and its accompanying forms and schedules. Nearly all the programs incorporate the new tax laws resulting from the 1987 Tax Reform Act.

DivorceTax
Research Press
IBM PC and compatibles
Lotus 1-2-3 or compatible spread-sheet required
$65

This template is designed to help structure a divorce settlement so both parties can save on taxes. It takes into account issues such as alimony, child support, property settlements, dependents, and child care. The program must be used with a spreadsheet, and it has a money-back guarantee.

Estate Tax Planner
Randle, Coray & Associates
IBM PC
$95

ETP helps users plan the growth of their estates and then estimate the annual estate and gift taxes for the next 20 years. The program considers consumptions or gift of assets; changing estate-tax rates and credits; marital deductions; and death of first or second spouse. What-if scenarios can be created to observe the effect of certain financial decisions on estate taxes. The manual includes instructions, forms, checklists, and information on wills, trusts, gifts, and estate planning.

Everything You Need To Do Your Taxes With *Lotus 1-2-3*
Woodbury Software
IBM PC
Lotus 1-2-3 required
$39.95

This tax planning and preparation package can help users compare one to six alternate tax strategies, or evaluate taxes for one to six years. The program helps determine the tax effects of financial decisions, and aids in sorting tax records. Major IRS forms—such as the 1040 and Schedules A, B, C, D, G, SE, and W—can be completed. Help screens appear throughout the menu-driven program. The package includes a book containing instructions and examples.

EZTax-Prep 1040
EZ Ware Corporation
Apple II-series; IBM PC, PCjr, and compatibles; Macintosh
Commercial spreadsheet required, printer optional
$99.95 (Macintosh); $129.95 (all other systems)

EZTax-Prep 1040 includes templates for completing 26 IRS forms and schedules: 1040, 1040ES, 2106, 2119, 2210, 2441, 3800, 3903, 4952, 4562, 4684, 4797, 4972, 6251, 6252, 8582, 8598, 8615, and Schedules A–F, R, and SE. Computations are performed by the program and the results are automatically applied to the appropriate form. The forms can be printed in an IRS-approved format on 1040 stock or transparent overlays using standard or laser printers. The program can also be used to create what-if scenarios and plan future tax strategies. The audit-alert feature points out deductions that could possibly trigger an IRS audit, and data can be exported to EZ Ware's *EZTax-Plan* for more in-depth tax planning. *EXTax-Prep 1040* is designed to be used with *Lotus 1-2-3*, *Multiplan*, *Excel*, or *Symphony 1.2*. Annual updates that include all tax law changes are available for a fee.

EZTax-Plan, Personal Edition
EZ Ware Corporation
IBM PC, PCjr, and compatibles; Macintosh
Commercial spreadsheet required
$95

This tax preparation package is designed to be used with *Lotus 1-2-3*, *Multiplan*, *Excel*, or *Symphony*. It helps individuals project future taxes, compare different tax strategies, plan for retirement, and complete tax form 1040 and schedules A, D, R, and SE. The Audit Alert feature also signals if planned deductions may lead to an IRS audit. *EZTax-Plan* takes into account charitable contributions, self-employment taxes, credits, itemized deductions, and assets such as bonds, stocks, and real estate. Updates incorporate any changes in tax laws that may occur. The package includes a coupon for one free update. This program can be used with *EZTax-Prep 1040*, also from EZ Ware.

Future-Tax
Taxaid Software
Commodore 64 and Plus/4
$39.95

Future Tax incorporates the new tax laws of the Tax Reform Act to help users calculate income taxes for 1988 and 1989, and plan future tax strategies. The menu-driven program includes the new tax rates and automatically figures standard deductions, exemptions, limitations on IRAs, surtax, deductions for the elderly, and itemized deduction limitations. The data can also be output to the screen or printer.

Complete and Easy...
and only $16!

MONEY COUNTS®
VERSION 4.0

Parsons Technology

Complete Money Management/Accounting System . . .
- CPA-designed for home & business
- Manages cash, checking, savings & credit cards
- Smart checkbook balancer — makes short work of the toughest bank statement
- Handles up to 999 accounts & 24000 transactions per year

Prepares & Prints a Wealth of Reports . . .
- Financial statements (4 types) including net worth)
- Inquiry reports (3 types)
- General ledger & accountant's trial balance

Now Including . . .
- Automatic transactions
- Financial calculator — computes interest rates, loan payments, prints amortization schedules
- Journal entries
- And more

Plus These Important Additional Features . . .
- Budgeting
- Fast financial data base with 3-way look-up
- Pop-up calculator
- Graphics
- Works great with monochrome & color monitors
- Not copy-protected

VISA, MasterCard & COD orders call:

1-800-223-6925
(In Iowa 319/373-0224)

Here it is. You won't find software that's easier to use:
- Requires no accounting knowledge
- Completely menu-driven with on-line help
- 93-page fully indexed tutorial manual (on disk)

A Proven Track Record. Hard to believe the $16 price? Don't worry. MoneyCounts® has been marketed for three years and is continually receiving rave reviews. There's **no** catch. This is a fully functional system that compares with products selling for $99 and more.

Same-Day Shipping. Order today and own MoneyCounts® for only $16! Add $3 shipping and handling (outside North America add $7). Iowa residents please add 4% sales tax. Send check or money order payable to Parsons Technology. VISA and MasterCard orders welcome.

C-www.commodore.ca

J. K. Lasser's Your Income Tax 1988

Simon and Schuster
IBM PC and compatibles
256K, DOS 2.1, color or monochrome monitor
$69.95

This tax preparation package combines software with the J.K. Lasser Tax Institute's tax guide, to help users complete their tax returns based on the new tax laws, and learn how to save on taxes and avoid penalties. An interview feature helps users select the correct forms, and the questions can be modified for individual needs. There is an onscreen calculator to use with either the worksheets or actual forms, and data can be transferred among the forms. Various tax strategies can be input to compare liabilities. Help screens contain excerpts from the book. Individuals and professional tax preparers can use the program, and the results can be printed in IRS-approved format or on original IRS forms. Owners of the 1987 edition can buy the 1988 edition for $39.95 with proof of purchase.

MacInTax

Softview
Macintosh with 512K memory
$119

This tax preparation program displays the IRS forms on the screen, and the user fills in the appropriate lines. The program automatically performs the calculations and transfers data among the forms. There are 13 worksheets to aid in computing taxes, and itemization windows help break down any amount into a detailed listing. Instructions for each line in the forms can be displayed on the screen, and the IRS instruction book is included. The completed forms can be printed using a dot-matrix or laser printer. *MacInTax* includes forms 1040, 1040A, 1040EZ, 1040ES, 2106, 2119, 2210, 2441, 3903, 4562, 4562A, 4684, 4797, 4868, 4952, 6251, 6252, 8283, 8582, 8598, 8615, and schedules 1, A–F, R, and SE. The program operates with *MultiFinder*. Annual updates are available for $55. State tax programs to supplement *MacInTax* are available for California and New York for $65 each; state updates cost $35.

PC/TaxCut

Best Programs
IBM PC and compatibles
$75

The QuickTax feature in this program can automatically recalculate tax owed, refund due, current tax bracket, and taxable income, each time a number changes in the program. This helps users see how variations will affect the tax return. *PC/TaxCut* offers context-sensitive onscreen help for whatever topic or calculation the user may be preparing. Forms can be displayed on the screen before printing, and the 1040 can be printed on plain paper with an IBM or Epson dot-matrix printer. A telephone tax advisory service is available for registered owners of *PC/TaxCut* for $1 per minute, with a minimum of 15 minutes. There is no charge for questions concerning program operation. Updates for previous owners of *PC/TaxCut* are available for $45.

Quick Returns!

Ontek Development Corp.
Apple II; Macintosh; IBM PC and compatibles
IBM PC version requires 256K RAM and DOS 2.0 or higher and supports CGA, EGA, Hercules, and monochrome display adapters; Apple II versions require ProDOS and 80-column capability
$24.95

Designed specifically for Canadians, this program helps users in any Canadian province or territory complete their Canadian income tax returns. It is available in 11 versions. Each version includes the main return, 11 subsidiary federal schedules, and the appropriate provincial and territorial forms. The user enters the data in the highlighted areas, and the totals, deductions, and tax payable are updated. Total amounts are automatically transferred between the main return and the schedules. Over 120 screens of context-sensitive instructions provide onscreen help. Returns can be printed out in a Revenue Canada–approved format. A checklist of items to mail with the return can also be printed.

Swiftax

Timeworks
Commodore 64 and 128
$49.95

Swiftax handles tax forms 1040, 1040A, 2441, and 1040EZ and schedules A, B, C, D, G, W, and SE. It automatically calculates tax alternatives—such as income averages—to show the lowest tax to pay. Changes can be made to the tax return at a later date, using the taxpayer file. Information can be printed directly on tax forms or onto blank paper. Update disks with new tax law information and tables can be obtained each year at extra cost. The 64 version operates in 40-column mode and the 128 version in 80-column mode.

Taxaid

Taxaid Software
Commodore 64 and Plus/4
$44.95

An income tax preparation program, *Taxaid* helps users complete IRS forms 1040 and 2441, and schedules A, B, C, D, and E. It incorporates the 1987 tax law changes and tax tables, and calculations are automatic. Results can be printed on plain paper in an IRS-approved format. The program is menu-driven and includes a manual.

TaxView Planner

SoftView
Macintosh; IBM PC and compatibles
Requires *MacInTax* or *TaxView* from SoftView
$79

This tax planning program includes tax tables for 1987–1991, along with worksheets to calculate taxes, tax bracket, child care credit, itemized deductions, capital loss limitations, taxable social security benefits, and the maximum IRA deduction. Users can compare up to five alternate tax strategies against the yearly taxes. The program can compute withholding allowances for throughout the year, and print W-4 and 1040ES forms.

Tax Advantage

Arrays
Apple II; IBM PC
$69.95

A tax preparation program for Forms 1040, 2106, 2441, 4562, and 6251, and Schedules A, B, C, D, E, G, SE, and W, *Tax Advantage* automatically computes the totals, displays the forms, and prints out the data. It can handle income averaging, calculate brackets and liabilities, and provide year-round tax planning. This program works with *Home Accountant*, also from Arrays.

The Tax Advantage

Double Eagle Software
IBM PC and compatibles; Apple II-series; Atari; Atari ST; Commodore 64 and 128
$59.95

This federal tax preparation program incorporates the tax law changes for the 1987 tax year and prints out IRS-acceptable forms and supporting statements. It can help users prepare Forms 1040, 2106, 2441, 4562, and 6251, and schedules A, B, C, D, E, F, and SE. The program can also be used for tax planning. Owners of *The Tax Advantage Early Bird Edition*, which includes all changes through August 1, 1987, can purchase the final edition for $15.95. The two editions are fully compatible; the final edition includes all tax law changes through November 1987.

Tax Command Planner

Practical Programs
Commodore 64 and 128; Apple II; IBM PC
Apple version requires 128K; IBM PC version requires 128K and DOS 2.0 or higher
$69.95

With this ten-year tax planning program, users can compare six tax options for up to five years at a time. The program uses current tax tables to calculate marginal tax rates, long and short term gains and losses, and taxes from income averaging. Tax rates can be adjusted according to tax law changes, and updates will be available each year. A manual provides line-by-line instructions.

Tax Command Professional

Practical Programs
Commdore 64 and 128; Apple II; IBM PC
Apple version requires 64K; IBM PC version requires 128K and DOS 2.0 or higher
$49.95

In this program, IRS forms appear on the screen as a series of menus into which the user enters dollar amounts. A calculator and current tax tables are incorporated into the program to automatically calculate the totals. The program can complete Forms 1040, 1040ES, 2106, 2119, 2441, 3903, and 8615, and Schedules A, B, C, D, E, F, R, and SE. (The Commodore 64 and 128 versions do not include Form 1040ES or Schedule F.) It compares itemized to standard deductions and provides line-by-line tax information entry. A summary of the information can be printed for personal records, or the data can be printed directly onto IRS forms.

The program works with most printers with PICA spacing. The IBM version displays in 80-column mode. A disk drive is required and a manual is included. Annual updates are available to registered users at half price. A money-back guarantee is offered.

Tax Master

Master Software
Commodore 64
Printer optional
$30

Individuals can use this program to prepare their federal tax forms 1040 and 4562, and schedules A–F. The program includes the tax tables and a calculator function to automatically compute the amounts. Results can be transferred from one form to another, and more than one schedule can be completed for each user. *Tax Master* is a menu-driven program that can store data for more than one taxpayer, and the information can be printed to screen or printer. The package includes an instruction manual and a coupon for a discount on the 1988 version.

Tax Master

Master Software
Commodore 64 and 128
Disk drive required
$30

The 1987 version of *Tax Master* helps users complete Federal tax forms 1040 and 4562, and Schedules A, B, C, D, E, and F. The program uses a built-in calculator and tax tables to calculate taxes. Data can be transferred between forms and printed out in 80-column mode. This version incorporates the 1987 tax law changes.

Tax Preparer

HowardSoft
IBM PC and compatibles
128K or GW-BASIC with 192K
$295

Tax Preparer includes all the tax law changes through 1990 based on the Tax Reform Act, and enables users to forecast their tax liabilities beyond 1987. The user enters the data, and the program automatically performs all calculations relative to the tax year entered. The data is then transferred to over 20 IRS forms and schedules. Results are checked against the tax code to make sure limits have not been exceeded, and to make sure the forms, schedules, and worksheets can be printed in IRS-approved formats. Users can also create what-if scenarios to plan future tax strategies. Onscreen instructions are provided along with a handbook and telephone support. The program uses split screens and windows.

Tax Preparer

HowardSoft
Apple II-series
64K required
$250

Tax Preparer includes all the tax law changes through 1990 based on the Tax Reform Act, and enables users to forecast tax liabilities beyond 1987. The user enters the data, and the program automatically performs all calculations relative to the tax year entered. The data is then transferred to over 20 IRS forms and schedules. Results are checked against

the tax code to make sure limits have not been exceeded, and to make sure the forms, schedules, and worksheets can be printed in IRS-approved formats. Users can also create what-if scenarios to plan future tax strategies. Onscreen instructions are provided along with a handbook and telephone support. The program uses split screens and windows.

Tax ZipFiles

Petit Design
Apple II series
Appleworks required
$19.95

Tax ZipFiles is a template for use with *Appleworks* in completing IRS Form 1040 and Schedules A, C, D, and E. The tax-ledger feature can sort tax-deductible expenses, and the speedmaps and speedpaging features aid in using the program. A 100-page manual is included. Wisconsin residents must add 5 percent sales tax to the cost.

TAXAID

Taxaid Software
Commodore 64 and Plus/4
$44.95

An income tax preparation program written by tax accountants and designed for home use. The latest edition includes all the current changes in the tax laws for the tax year 1987. *TAXAID* prepares IRS Form 1040, Schedules A, B, C, D, E, and Form 2441 (child care). The menu-driven program calculates automatically with all tax tables built in. An editing feature allows users to make changes and revisions at any time. Printout is in IRS-approved format.

TaxEase

Park Technologies
IBM PC and compatibles
256K memory and spreadsheet required
$69.95

TaxEase helps users complete both federal and state tax forms, automatically computing the taxes and printing the results on IRS-approved forms. Values can then be transferred among the forms. Versions are available for 20 states: Arizona, California, Colorado, District of Columbia, Georgia, Illinois, Indiana, Maryland, Massachusetts, Michigan, Minnesota, Missouri, North Carolina, New Jersey, New York, Ohio, Oklahoma, Pennsylvania, Virginia, and Wisconsin. A version containing only the federal tax forms is also available. Updates are available for $34.95.

TaxView

SoftView
Apple IIGS; IBM PC and compatibles
$119

TaxView helps users complete their tax forms or create what-if scenarios to determine the best tax plan. Replicas of 27 major IRS tax forms and schedules appear on the screen, and when completed, can be printed out to a dot-matrix or laser printer. The forms and schedules can be linked together, and data can be imported from text files. The program checks for errors and any violation of an IRS rule. Instructions for each line in a form can

be called at any time, and there are 13 worksheets to help in calculations. Versions are also available for some states, including California and New York, for $65. Annual updates to the federal tax package are $55, and state updates are $35. The IBM version operates under Microsoft Windows; a run-time version of that system is included on the disk.

Taxpertise Preparer

Marnoble Software
MS-DOS or PC-DOS machines
Lotus 1-2-3 or *Symphony* required
$49.95

Taxpertise uses a fill-in-the-blanks format, so the screens look like the actual tax form. The completed forms can be printed onto a tax form, on blank paper for use with transparancies, or in an IRS-approved format. This program helps users complete the following tax forms: 1040, 1040A, 1040EZ, 1040ES, 1040X, 1116, 2106, 2119, 2210, 2210F, 2441, 3468, 3800, 3903, 3903F, 4562, 4684, 4797, 4868, 5695, 6251, 8598, 8606, and 8615, and schedules A–F, R, and SE. State forms can also be purchased for $20 for California, Georgia, Illinois, Maryland, New Jersey, New York, Ohio, Pennsylvania, Virginia, and Washington D.C. Disks are available in 5¼- or 3½-inch formats.

Taxpertise Preparer State Programs

Marnoble Software
MS-DOS or PC-DOS machines
Lotus 1-2-3 or *Symphony,* and *Taxpertise Preparer* required
$59.95

State tax returns can be completed for California, Georgia, Illinois, Maryland, New Jersey, New York, Ohio, Pennsylvania, Virginia, or Washington D.C. The program must be used in conjunction with *Taxpertise Preparer*. When purchased with the federal tax package, each state package costs $20. If purchased separately, each state package costs $59.95.

TurboTax 1988, Version 5.0

ChipSoft
IBM PC with 256K memory
$75

TurboTax is a tax preparation, planning, and recordkeeping package. With this program, users can complete over 35 IRS forms, plus supporting schedules and worksheets. Online instructions, including a 30-second manual at the beginning of the program, help answer questions about the forms, and customer support is provided by telephone or through the ChipSoft bulletin board. A data examiner option can point out anything left out of a return, and the tax window feature displays the current tax status throughout the program. Most of the math calculations are performed automatically, but a pop-up calculator and notepad are also available. The next year's tax laws are built into the planning mode of the program to facilitate tax recordkeeping and planning. The data can be printed onto supplied 1040 forms, or the forms can be printed on IBM- or Epson-compatible dot-matrix printers. The *Price Waterhouse Personal Tax Advisor* guide to 1987 and 1988 taxes is included in this software package. There are 26 integrated state tax packages available for $40 each. A version for professional tax planners is also available for $295.

Form 1040 Department of the Treasury—Internal Revenue Service **1987**
U.S. Individual Income Tax Return

Tax Preparation Software Publishers

For more information on the tax preparation software listed in the Buyer's Guide, contact the publishers listed alphabetically below.

Arrays
6711 Valjean Ave.
Van Nuys, CA 91406
818-994-1899

Best Programs
2700 S. Quincy St.
Arlington, VA 22206
703-820-9300

ChipSoft
5045 Shoreham Pl., Suite 100
San Diego, CA 92122
619-453-8722

Double Eagle Software
2210 Wilshire Blvd., Suite 875
Santa Monica, CA 90403
213-459-9748

EZ Ware Corporation
P.O. Box 620
29 Bala Ave., Suite 206
Bala Cynwyd, PA 19004
215-667-4064

HowardSoft
1224 Prospect St., Suite 150
La Jolla, CA 92037
619-454-0121

Marnoble Software
3730 Kirby Dr., 12th Floor
Houston, TX 77098
713-831-6866

Master Software
6 Hillery Court
Randallstown, MD 21133
301-922-2962

Ontek Development Corp.
#3206, 24 Mabelle Ave.
Toronto, Ontario
Canada M9A 4X8
416-234-1489

Park Technologies
Box 1317
Clifton Park, NY 12065
518-877-5881

Petit Design
2650 S. Shore Dr.
Milwaukee, WI 53207
414-483-9056

Practical Programs
P.O. Box 93104
Milwaukee, WI 53203
414-272-7227

Randle, Coray & Associates
Box 1228
Utah State University Post Office
Logan, UT 84322
801-753-2020

Research Press
4500 West 72nd Terrace
Prairie Village, KS 66208
913-362-9667

Simon and Schuster Computer Software
One Gulf + Western Plaza
New York, NY 10023
212-373-8882

Softview
4820 Adohr Ln., Suite F
Camarillo, CA 93010
805-388-2626

Taxaid Software
606 Second Ave.
Two Harbors, MN 55616
218-834-3600

Timeworks
444 Lake Cook Rd.
Deerfield, IL 60015
312-948-9200

Woodbury Software
127 White Oak Lane
CN 1001
Old Bridge, NJ 08857
201-679-0200

©

www.commodore.ca

Top Fuel Eliminator

Russ Fisher

Requirements: Commodore 64, Apple II series.

Top Fuel Eliminator is a drag-racing sim-ulation. Included in the game are time trials for practice sessions and entire racing seasons. Start with the time trials first, to become familiar with the drag-ster's characteristics and how they can be modified for various racing condi-tions. An entire racing season consists of nine events and may take 30 to 60 minutes to play, depending upon your skill and how much time you use be-tween races to make adjustments to the car. *Top Fuel Eliminator* is a combina-tion action/strategy game, where the action lasts five to seven seconds for each run down the drag strip.

With much of the newer software employing highly detailed graphics, it is surprising that the graphics here ap-pear somewhat cartoonish. It also takes time to reload the program after choos-ing either Time Trials or Play Game. It would have been more convenient to switch between playing rounds, espe-cially when first learning to play.

Competition By Computer

Competition in *Top Fuel Eliminator* is provided by eight computer-controlled drivers and their dragsters. Some of the competitors are very good indeed. There are nine different contests during the game, such as Winter Nationals, North Nationals, and Summer Nation-als, culminating in the World Event.

The racing season progresses through all four of the actual seasons, and racing conditions are affected ac-cordingly. Prior to each race, you have an opportunity to make various adjust-ments to your car in an attempt to compensate for varying track condi-tions. At each event, you can check air temperature, track temperature, hu-midity, altitude, and whether the con-dition of the track is excellent, average, or downright slippery. Some of the ad-justments, such as the supercharger set-ting, engine timing, and fuel mix, are

related to engine horsepower. At lower humidities, you might want to crank up the supercharger pressure. But setting up a very "hot" engine can lead to more engine failures during the course of a race, so be careful.

Other dragster adjustments are re-lated to traction and include the clutch, tire, and wing settings. You may vary the tire pressure, clutch slippage, and wing angle. Poor traction will result in excessively smoking tires during the race. Too much traction can have the opposite effect and bog down the car. Sometimes the track conditions change rapidly, so be sure to check them before each race. Astute adjustments are called for if you want to stay in the running.

Qualifications Required

You must qualify for each race. You have two attempts to qualify under the posted time, which is the slowest time of the nine drivers. Of course, you must qualify without jumping the "Xmas tree" yellow light count-down, and you must stay in your lane at all times. One gear shift is also required during each run. Watch the tachometer, and try to shift between 9000–9500 rpm. Going over 9500 rpm will cause your engine to blow.

After qualification, a ladder chart shows the matchups for the first round of races. The chart is shown again after each race until a winner is decided. If you do not qualify, or you lose one of the races, the completed chart shows how each of the computer players per-formed during that event.

Points are awarded for per-formance after each event, and a sum-mary of the points is displayed. You receive between 100 and 800 points at each event: 100 points for showing up, 800 for winning. You also receive 1 to 8 points for qualifying—8 for the fastest qualifier and 1 for the slowest.

Shifting Sounds

Although the actual racing times are very short, there is a lot of action hap-pening during the race. The Xmas tree is displayed at the lower-left corner of the screen, the driver's view is at the bottom center, the tachometer is at the lower-right corner, and an overhead view of the track spreads across the top

Top Fuel Eliminator, combining action with strategy, pits you against eight other dragsters in a variety of competitions.

of the screen. Watch the Xmas tree care-fully to get a quick start without red-lighting. Then pay attention to keeping your car in the lane, watch the tach for the proper shift point, and glance at the top of the screen to see what is happen-ing to your competitor. You may want to push your chair back a few feet and hope to catch a glimpse of the tach while struggling to keep the car in your lane. Fortunately, the sound effects give a clue to engine rpms; you may find yourself shifting gears when it sounds right. I've driven into the wall too many times while trying to sneak a glance at the tach.

After each race or qualifying round, a performance analysis is pre-sented. Examine this screen to see your reaction, shift, power, and traction per-formance. This screen also shows if you had an engine failure, crash, red light, or if you crossed the line. Finally, your time and top speed are posted. The attention to detail makes *Top Fuel Elim-inator* a good game choice for armchair racing fans.

Top Fuel Eliminator
Activision (Gamestar)
P.O. Box 7287
Mountain View, CA 94039
$29.95

Thexder

James V. Trunzo

Requirements: Apple IIGS with 512K RAM (reviewed here); IBM PC, XT, AT or true compatible with 256K RAM; IBM PS/2, supports MCGA color mode; Amiga.

While "Buy American" is a laudable suggestion and one worth heeding on most occasions, that isn't the case here; if IIGS owners wish to purchase the best arcade games for their machines, they've got to buy Japanese. From Game Arts of Japan via Sierra On-Line (via California) comes the first true arcade game for the Apple IIGS: *Thexder*.

A runaway best seller in Japan, *Thexder* has been converted to American machines and is certain to gain a following here. Its theme is certainly popular with the younger set: The Thexder is a super-assault vehicle—a hyper dual-armor Robot-Jet transformer that mimics the popular cartoon and toy Transformers. A single keystroke changes the Thexder from a powerful robot into a sleek jet, and in either form, it can fire heat-seeking lasers at its foes.

When the game begins, the only Thexder in existence—piloted by you, of course—is trapped inside a heinous alien complex. Twenty different enemy creatures are doing their best to be sure you don't traverse the 16 levels of play and make it to the central computer. They have good reason to want to stop you: If you make it to the computer and destroy it, you'll not only escape, you'll end the alien's only means of reproduction.

Thexder, a best-seller in Japan, is now available for U.S. machines.

Thexder is the standard shoot-'em-up arcade space game: A powerful good guy must destroy even more powerful bad guys. It's far above the standard in graphics, animation, control of the central character—the Thexder vehicle—and challenge. The screens are beautifully drawn, and the scrolling is seamless and unobtrusive. Graphically, *Thexder* delivers on its promise to be the game by which all others of its type will be judged, mainly due to its attention to details. Control of the *Thexder* is through the use of the numeric keypad (or the top number keys, if you prefer) with no mouse or joystick option. This should not be considered a drawback, however, because the keypad system works quickly and efficiently. Actually, it's similar to the multiple-button control panels of many popular full-size arcade games.

One Life To Live

As for the challenge of play, be forewarned. *Thexder* is not an easy game, and in my opinion, it can be very frustrating—especially for younger children who have short attention spans. It takes a while to get the hang of maneuvering the *Thexder* and even longer to figure out strategies. These features can be pluses for ardent gamers, but work in an opposite manner for the casual or younger gamer. Also, if the Thexder is destroyed, game play starts over, but at least it starts over from the beginning of the last level completed. In *Thexder*, you only live once.

The 20 different opponents out to thwart your mission are a paradox: Each of them is unique in its overall appearance, yet many of them are so similar in shape, they can confuse the

C-www.commodore.ca

........When you want to talk price.

DRIVES

Atari
AA314 DS/DD Disk (ST).........$199.00
AA354 SS/DD Disk (ST)..........129.00
SHD204 20 Mb ST Hard Drive 559.00
Commodore
Amiga 1020.............................189.00
Amiga 1010 3½"......................219.00
1541C......................................169.00
1571..229.00
1581 3½" External...................219.00
Indus
GT Disk Drive Atari XL/XE 179.00
Racore
Jr. Expansion Chassis.............299.00
Seagate
20 mb ST-225 Hard Drive Kit..339.00
Supra
20 Meg Hard Drive (Amiga).....739.00
20 Meg Hard Drive (ST)..........549.00
Xebec
20 Meg Hard Drive (Amiga).....799.00

MONITORS

Amdek
Video 300 Amber Composite...139.00
Commodore
Commodore 2002......................319.00
Amiga 1080 Hi-Res Color........299.00
Commodore 1802.....................199.00

Magnavox 8502 13" Composite $169

Magnavox
8505 RGB/Composite..............199.00
8562 RGB/Composite..............249.00
NEC
JC-1402P3A Multi-Sync..............Call
Princeton Graphics
MAX-12 12" Amber TTL..........139.00
Taxan
Model 124 12" Amber.............119.00
Thomson
450 15" 132 Col. Amber.........159.00
4120 RGB/Composite.............239.00

MODEMS

Anchor 1200 Baud Ext. $119

Anchor
6480 C64/128 1200 Baud..........119.00
VM520 ST520/1040 1200 Baud.129.00
Atari
SX212 (ST)..............................88.99
Best Products
1200 Baud ½ Card w/software....89.99
2400 Baud ½ Card w/software..169.00
Commodore
Amiga 1680-1200 BPS.............169.00
Everex
Evercom 1200 Baud Internal.......99.99
Hayes
Smartmodem 300.....................139.00
Smartmodem 1200 External......289.00
Novation
Parrot 1200..............................89.99
Practical Peripherals
Telecom Package.....................99.00
Supra
MPP-1064 AD/AA C64..............69.99
2400AT 2400 Baud Atari...........169.00
U.S. Robotics
1200 Baud External...................99.99
2400 Baud Internal..................179.00

DISKETTES

Maxell
MD1-M SS/DD 5¼".....................8.49
MD2-DM DS/DD 5¼"...................9.49
MF1-DDM SS/DD 3½"...............12.49
MF2-DDM DS/DD 3½"...............18.49
Sony
MD1D SS/DD 5¼".......................6.99
MD2D DS/DD 5¼".......................7.99
MFD-1DD SS/DD 3½"................11.99
MFD-2DD DS/DD 3½"................16.99
Hewlett-Packard Calculators
28C Scientific Pro...................199.99
18C Business Consultant.........139.95
12C Slim Financial....................74.99

PRINTERS

Atari
1020 XL/XE Plotter...................$31.99
XDM121 Letter Quality............159.00
XMM801 XL/XE Dot Matrix......185.00
XMM804ST Dot Matrix............179.00
Brother
M-1109 100 cps, 9 pin.............199.00
M-1409 180 cps, 9 pin.............319.00
Citizen
MSP-15 160 cps, 132-Column.269.00
Premier 35 cps Daisywheel.....479.00
C.Itoh
315P 132 Column Prowriter.....549.00

Epson LX-800 150 cps Dot Matrix $179

Epson
LX-800 150 cps, 80-column.....179.00
FX-86E 240 cps, 80-column.........Call
FX286E 240 cps, 132-column......Call
LQ-500 24 Wire, 80-column.........Call
LQ-850 330 cps, 80-column.........Call
LQ-1050 330 cps, 132-column.....Call
Hewlett Packard
Thinkjet..................................379.00
NEC
Pinwriter 2200 24 Wire............379.00
Pinwriter 660 24 Wire..............459.00
Pinwriter 760 24 Wire..............679.00
Okidata
Okimate 20 Color Printer.........129.00
ML-182 120 cps, 80-column.....219.00
ML-192 + 200 cps, 80-column.329.00
ML-193 + 200 cps, 132-column479.00
Panasonic
KX-1080i 144 cps, 80-column..179.00
KX-1091i 194 cps, 80-column..199.00
KX-P3131 22 cps Daisywheel..269.00
Star Micronics
NX-1000 144 cps, 80-column...189.00
NX-1000C 144 cps, C64 Inter..189.00
NX-15 120 cps, 132-column.....319.00
Toshiba
P-321 SL 216 cps, 24-Pin........539.00
P-351 II 300 cps, 24-Pin..........889.00

In the U.S.A. and in Canada

Call toll-free: 1-800-233-8950

Outside the U.S.A. call 717-327-9575, Telex 5106017898, Fax 717-327-1217
Educational, Governmental and Corporate Organizations call toll-free 1-800-221-4283
CMO. 477 East Third Street, Dept. A2, Williamsport, PA 17701
ALL MAJOR CREDIT CARDS ACCEPTED.

POLICY: Add 3% (minimum $7.00) shipping and handling. Larger shipments may require additional charges. Personal and company checks require 3 weeks to clear. For faster delivery use your credit card or send cashier's check or bank money order. Pennsylvania residents add 6% sales tax. All prices are U.S.A. prices and are subject to change and all items are subject to availability. Defective software will be replaced with the same item only. Hardware will be replaced or repaired at our discretion within the terms and limits of the manufacturer's warranty. We cannot guarantee compatibility. All sales are final and returned shipments are subject to a restocking fee.

C= www.commodore.ca

player during early play sessions.

It's important to know which alien form is a Tribar as opposed to an Arbalest, for example, because some aliens provide you with needed energy when destroyed, while others do not. Because you use energy when destroying aliens, fleeing or outmaneuvering aliens is a good strategy at times, especially if the aliens are not energy providers.

Start-Up

Booting *Thexder* on the GS is an experience. It requires that you use both the *Thexder* disk and your System Master each time you play. I can only hypothesize that there is so much code on the *Thexder* disk that the System info couldn't fit; or possibly it had to do with licensing agreements. In any case, it's a minor annoyance to have to go through a three-disk swapping procedure to begin play.

Thexder is one of a growing number of games that allows you to make a back-up copy but requires that the original be used at some point during the loading process. Because it is an arcade game, I can't quibble with this scheme: It gives the player a chance to protect his master, yet it prevents mass copying and distribution of copyrighted material.

[Ed. Note: The IBM version of Thexder *is hard-disk installable, requiring the use of the master* Thexder *disk as key disk during boot-up. Incidentally,* Thexder *is, according to Sierra On-Line,*

the first game to support IBM PS/2's MCGA color mode.) Finally, don't be alarmed at the paucity of instructions that come with *Thexder*. At first glance, especially after experiencing initial problems in playing the game, I thought that what appeared to be a lack of documentation was a major problem. However, it was patience that were lacking, not instructions. The six panels of instruction and the walk-through map of the first several levels provides you with the information needed to play the game; and comparing it to the instructions given on arcade machines—which *Thexder* parallels in all other phases—the directions are more than adequate.

Overall, *Thexder* jumps into the past to grab a theme that was the rage in the boom days of video arcades, and it leaps into the future with 16-bit computer technology. If you like hot-and-heavy combat action coupled with great graphics and sound, *Thexder* is sure to satisfy you while giving you excellent play value for your dollar.

Thexder
Sierra On-Line
P.O. Box 485
Coarsegold, CA 93614
$34.95
Amiga version distributed by:
Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
$34.95

ProDrive For The Amiga

Scott Thomas

Third-party hardware support for the Amiga appears to be accelerating at surprising speed. Attributable mainly to the marketing success of the Amiga 500, third-party manufacturers in recent months have come out with a panoply of hardware peripherals for the Commodore Amiga. One of these third-party hardware peripherals, the Pro-Drive from Progressive Peripherals & Software, is a good example of third-party support that benefits Amiga owners with higher quality products at less expensive prices.

Like the Amiga 1010 3½-inch external floppy drive, the ProDrive draws its power from the computer power supply. Additionally, the ProDrive is designed to be 100-percent compatible with the Amiga 1010 external drive. Similarities between the two drives, however, end there. The ProDrive is compact, consuming only about 40 percent of the space that the Amiga 1010 drive requires. Its dimensions are a mere 7 inches deep by 4¼ inches wide by 1½

inches in height. Removing the metal housing of the drive revealed the reason for its compactness. The drive is an NEC brand internal 3½-inch drive, undoubtedly developed for IBM PC clones.

Physically Distinguished

Other physical characteristics of the ProDrive that distinguish it from the Amiga 1010 drive are its sturdy metal housing, spring-loaded door to keep dust out of the drive, and its three-foot cable to connect to the computer. One woefully inadequate feature of the Amiga 1010 is its stubby cable. For Amiga 500 owners, the 1010 must sit immediately next to or on top of the computer in order for proper connection of the cable. My work station would not accommodate such an arrangement and I was forced to buy a special extension cable to connect my Amiga 1010 drive. The ProDrive's extra cable length, therefore, provides needed flexibility for your work space.

In my examination of the drive, I

found the ProDrive to be truly 100 percent compatible with the Amiga 1010. All reads and writes to the drive perform flawlessly, and all commercial software, including many with copy protection, ran on the ProDrive without a glitch. I detected no distinguishable differences in the speed of reading and writing to disk between the ProDrive and the Amiga 1010. Its quietness of operation, however, was pleasantly noticeable in comparison to the noisy operation of the 1010.

Pinned

The only drawback to the ProDrive is the absence of a 23-D-pin drive connection out of the drive to permit daisy-chaining of additional drives. As most Amiga owners are aware, the operating system of the Amiga will support up to four floppy drives. However, the power supply of the computer will support only one external floppy drive, making it impossible to connect more than one Amiga 1010 drive to the computer. That fact was undoubtedly the consideration of Progressive Peripherals when it decided to leave off the drive connection on the ProDrive. However, the absence of such connection prevents one from daisy-chaining an Amiga 1020 5¼-inch external drive. The Amiga 1020 has its own power supply and can be daisy-chained with an Amiga 1010. The absence of the drive connection on the ProDrive, therefore, is a significant consideration for any Amiga owner who is considering obtaining a 5¼-inch drive down the road.

In summary, the ProDrive is an excellent high-quality hardware peripheral for the Amiga. It is better built than the 1010 external drive and has significant feature enhancements over the Amiga drive. Its compactness, quiet operation, and sturdiness make it a very attractive peripheral. And with a list price $60 less than the 1010, it's very much a bargain. Unless you have or are considering the Amiga 1020 5¼-inch drive, I recommend serious consideration of the ProDrive as an inexpensive yet higher-quality alternative to the Amiga 1010 drive.

ProDrive
Progressive Peripherals & Software
464 Kalamath St.
Denver, CO 80204
$239.95

Dark Castle

Rhett Anderson

Requirements: Amiga (reviewed here), IBM PC and compatibles, Macintosh, Atari ST, Commodore 64. IBM version works in CGA, EGA, or Hercules Monochrome modes, or in 16-color mode on the Tandy 1000 series.

Are you the type who would enter a haunted house for fun? If you are, *Dark Castle* may be your kind of game. If you're not, *Dark Castle*'s two disks of beautiful screens, realistic animation, and blood-curdling digitized stereo sound may still be able to lure you into the danger.

In *Dark Castle*, your goal is to slay the Black Knight. Of course, the Knight knows you're after him, but he's not especially worried—after all, what chance do you stand in his heavily protected castle? Not much, as you'll find out when you begin play.

Work Of Art

Dark Castle's main achievements are to be found in its brilliant graphics, sound, and atmosphere. Each room shows lavish attention to detail. In this review, I'll focus on the Amiga version of the game—the most impressive—but most of the following comments apply to the other versions as well.

Simply put, the artwork in this game is beautiful. Each room is hand-drawn and is loaded from disk when you enter. The animation is second to none—each character in the game has a generous number of animation frames.

Each room is populated by the castle's undesirables, and each "undesirable" has its own digitized sound effects—the crows caw, the bats shriek, the rats squeak, and some strange little characters make faces at you and yell "Nya nya nya nya nya."

Dark Castle has a distinctive style that pulls the animation and sound into a coherent whole. More than any other game available for the Amiga, this game has atmosphere and class. As you travel through the rooms, you can almost feel the clammy walls and cold floors. This is one of the programs you'll use to show off your Amiga.

Game Play

In *Dark Castle*, you progress through the castle's rooms, trying to find the Black Knight in order to slay him. You face many different kinds of opponents, which, if you're accurate, can be disposed of with rocks. To do this, you first point your arm in the direction you wish to throw, and then make the toss.

Dark Castle features stunning graphics with careful attention to detail.

It doesn't take much to dispose of your character—if you trip over a small step or fall in the water, you're dead.

The most severe problem with *Dark Castle* is that game play is so difficult. Your character can be controlled by either a joystick or the combination of the mouse and keyboard. After years of playing games with joysticks, I found the keyboard/mouse duo unfathomable. Unfortunately, the many actions that need to be performed in the game become hopelessly complex under joystick control.

Even considering the trouble with controlling the character, the game seems overly difficult. For over a week now (in *Easy* mode), I've been leaping off a certain precipice, trying in vain to grab hold of a swinging rope. I've never caught it. Frankly, if I weren't so anxious to see the flying monkeys and other enemies the package illustration promises, I would have given up several days ago.

Other Problems

There are other flaws in this game which will keep it from occupying my list of favorites. The first is its copy-protection scheme. While it's understood that game producers have more to lose from piracy than other software producers (after all, you don't need a manual to play most computer games), I can't bear to hear my disk drive grinding its way through the protection sequence each time I want to play. I simply don't trust that the software respects my expensive computer equipment. In the case of *Dark Castle*, the protection is so severe that the game occasionally crashes my computer on boot-up—or worse, while loading the screen for the next room.

That brings up another problem— the frustrating wait between levels, as *Dark Castle* loads the graphics and sound data. Players with two disk drives will be annoyed. Those with one will be positively aggravated. It would be great if you could put this game on your hard drive, but the copy protection prevents that. (*Note: The IBM version is hard-disk installable.*)

Nonetheless, *Dark Castle* deserves admiration for its excellent sound and graphics. Whether it is worthy of play is another matter. *Dark Castle* has shown us what a game on the Amiga can be. We hope that future games will combine the excellent presentation of *Dark Castle* with better playability.

Dark Castle
Three-Sixty Pacific
2105 South Bascom
Campbell, CA 95008
$39.95 IBM, Amiga, and Atari ST versions
$34.95 Commodore 64 version

Macintosh version available from:
Silicon Beach Software
9770 Carroll Center Rd.
Suite J
San Diego, CA 92126
$49.95

Postcards

Ervin Bobo

Requirements: Apple II series (reviewed here), Macintosh, Commodore 64, IBM PC and compatibles.

Many years ago there was a fad for comic postcards. These cards usually carried a message of wry humor or ironic understatement, and were illustrated by rather outlandish drawings— one might show an outhouse with a moon carved in the door and the caption "Wish you were here," or an ear of corn as big as the truck bed on which it rested and the caption "You should see the big ones."

Juvenile by today's standards of humor, back then they offered a quick and easy form of communication, a way of letting the folks back home know you had not only survived this far but were actually enjoying yourself. The price was five or ten cents, and postage was a one- or two-cent stamp—I told you this was long ago—and everybody in America probably received one at some point between 1935 and 1955.

Lost now and only dimly remembered, save for a few collectors, the penny stamp and comic postcard gave way to higher postage and to the need for a flashy, full-color postcard showing the interior of Meramec Caverns or the faces on Mount Rushmore. Lost with it was a piece of Americana.

Recalling The Past

Fortunately, someone at Activision remembers older and simpler times and has brought these comic postcards back (but not the penny stamp) in a new

program called *Postcards*. The program consists of a library of line-art images, a drawing program for customizing or enhancing the artwork, and a text processor for captions. *Postcards* allows you to print out a part of the past on a dot-matrix printer, paste it on postcard stock, and mail it to someone who will either appreciate the nostalgia or be buffaloed into thinking you've invented a new art form.

Testing the Apple version, I was pleased to find that the setup program first asks you for the type of computer you're using. The list of choices includes the Laser 128 and the Franklin 500, making Activision one of the first software companies to come to grips with the newer realities of the Apple software market. Following this, you are asked for the number of disk drives and the type of printer you are using. All this information is stored on the master disk.

The working screen is really two screens. You switch between them by pressing the 1 or 2 key. Running vertically along one edge of the screen is a list of commands. Each is as simple to implement as the page command: Strike only the first letter of the command word, and you are instantly there.

Bathing Beauties On The Moon
There are ten full-screen backgrounds from which to choose and more than a hundred pieces of clip art, the latter divided into such categories as People, Edibles, Animals, and Transportation. During the heyday of these cards, a great part of the humor was in juxtaposing subjects or scale. The *Postcards* libraries and the quick reference guide to them will suggest their own incongruities—bathing beauties on the moon, a kangaroo looking at the Eiffel Tower.

Properly, you should load a background onto the palette screen, then transfer clip art or drawings onto it from the Clipboard screen. This process is made easy through the use of a "rubberband" frame that is invoked when you use the Copy command. Stretch it around the clip art until you've included only what you feel is necessary and then click the joystick button to clip it. Press Esc to take you back to the working screen; next, switch to screen 2 and press P for paste. The picture reappears, "attached" to your cursor, and can be moved wherever you wish. Once there, it can be pinned down with another click of the fire button.

There seems to be no limit to the number of elements you can include in this manner, though you'll be the best judge of how much is necessary to express whatever wacky thought you

have in mind. As an aid to keeping such built-up pictures coherent, the Copy command includes a Transparent mode, in which your background will not be blocked out by the white frame of your clip art.

The drawing portion of the program is strictly freehand. There are no premade circles or squares such as are found in stand-alone programs, but you'll probably find the drawing program adequate. While it will be possible for the more talented to draw their own pictures, the primary use of the drawing program appears to be as a means to enhance or personalize the drawings from the *Postcards* library.

Although the drawing program can be called minimal, I was pleased to find a Zoom feature which allows editing on a pixel level and a Fill command that will invoke one of six fill patterns.

Postcards offers ten different backgrounds and more than a hundred pieces of clip art for creating your own ready-to-mail postcards.

Words And Pictures
Completing the program is a text processor. You can type your message within the picture—which is framed— or outside the frame. Text is erased by using the backspace key, but that should be enough. Remember that we are creating captions, not documents.

Because we are dealing with a picture the size of a postcard, printing will be done quickly. The drawing is printed on standard paper, then cut out and pasted to the postcard stock included in the package. For those of us who can never seem to find the tools needed to do a job, a glue stick has also been included. After this, of course, you address the card, affix a stamp, and mail a piece of personalized nostalgia to someone who will appreciate it.

In bringing back an almost forgotten piece of Americana, Activision is to be commended. An outgrowth of the utility printing programs that began with *PrintShop*, *Postcards* has its own approach, its own subject, and, as evidenced by the library of drawings, its own sense of humor. As for being a utility, I think of it as more of a luxury, to

be savored in the way one might savor a fine piece of carnival glass or a plaster miniature of the Empire State building.

One problem: When loading a sheet from the library, there is no warning that you will erase whatever else is on the active screen. For that reason, it is important to develop a routine for screen use and switching. Otherwise, your carefully crafted picture on screen 2 can be erased by accidentally loading clip art onto the same screen.

With that one exception, I think *Postcards* is well thought out, well documented, easy to use, and that it will fill an empty space in many hearts.

Postcards
Activision
2350 Bayshore Pkwy.
Mountain View, CA 94043
$29.95 Apple IIgs and Macintosh
$24.95 Apple II series, Commodore 64,
IBM

Empire
Neil Randall

Requirements: Atari ST (reviewed here); IBM PC, XT, AT or true compatible; Apple II series; Amiga

WARNING! the box says. *This program is highly addictive. Considerable otherwise productive time might be lost. Play only during vacations.* Usually, of course, such warnings are best disregarded. Some games are addictive, but few as addictive as the packaging would have us believe. In the case of Interstel's *Empire*, though, the warning may be worth heeding.

Also on the box is a statement that *Empire* is an updated version of the popular game of the same name that appeared on mainframes and minicomputers as early as 1977. I never played *Empire*, and I'm glad I didn't know it existed. As it is, I had enough trouble just writing this review. See, every time I sat down to boot up my word processor, the copy of *Empire* kept staring at me. Day by day, my resistance weakened. . . .

A Dose Of Documentation
Empire comes with a 72-page Command Manual, a Command Reference Card (specific to your computer), and a disk that is not copy-protected (and is therefore copyable to a hard drive). In place of copy-protection the program opens with a Confirmation system, instructing you to type the first word from line *x* on page *x*. Like most elements of the game, even the confirmation system is friendly: Since it always specifies

the first word on the line, you need only count lines, not words. Furthermore, the request contains a message stating that distributing the program may be injurious to your conscience. For me, this works far better than the usual legal warnings—I prefer butter to guns.

The manual is complete, well-organized, and daunting. It opens with a memorandum and a short story, both of which relate the game (rather weakly) to Interstel's *StarFleet* software. The manual provides instructions on backing up the disks and copying the program onto a hard disk. Next come 24 pages on how to load, set up, and play the game, and this is followed by 26 pages describing the individual play commands. After an extremely useful 4-page section on strategy and hints, the manual gives a 7-page overview of the creation of worlds using the game's Map Generator. Last comes some peripheral information, including an interesting description of the programming history of the game.

Any 72-page manual, though, is intimidating, and this is the only unfortunate element of *Empire*. Like most players, I boot a game and start playing it before I even look at the manual, but a manual this size almost discourages such impetuosity. In *Empire*'s case, especially, the intimidation is needless. I was able to play the game after only 20 minutes or so of experimentation–and I mean really playing it, not just mindlessly moving armies around. Before my first hour was out, I was already planning devious and brilliant strategies for capturing an innocuous-looking peninsula from the green army. By the end of the second hour I was engaged in a full-scale war with that same army (their idea of "innocuous" differed from mine). Only then did I turn to the manual.

Empire is a strategy wargame, but it is much different from most computer wargames. The premise is that you have been sent by Star Fleet Command to oppose the Krellan conquest of a planet. You land in a city, and the rest of the map is black. From that city you begin producing your war machine, with the goal of recapturing all 66 cities from the Krellans. The game allows either a two-player or a three-player version, and the computer will handle as many or as few of the three commands as you wish. All three can be human players.

Strategy Is Everything
What makes *Empire* different is its strict emphasis on strategy. You create a host of armies, but their attributes are identical. So, too, are the fighters, battleships, transports, carriers, destroyers, and submarines. In other words, the game is not designed for the historical game buff. Its

purpose is to let you fight a global war quickly and easily, demanding only your best strategy. It sacrifices small detail for the sake of emphasizing strategy.

The interface is extremely well-designed. You can execute all commands from the keyboard or with a mouse, or you can combine both freely. When you capture a city, an attractive display allows you to set production for that city, and it even insures that you don't start producing battleships in a city not located on the coast. Each city produces only one type of unit at any one time, but you can easily change the production by calling up the production map, selecting the city, and making the change.

Empire *is an addictive strategy war-game (Atari ST version pictured above.)*

Moving units is what you'll be doing most, and here, the interface shines. The unit to be moved flashes, and you can either go to the (GEM) menus, type a one-letter command on the keyboard, or point with the mouse. With the mouse pointer, you simply go to the desired destination square and press the left button. A white line appears, linking the unit to the square. If you want that destination, simply release the button. The unit may take several turns to reach its destination, after which it will start flashing again, waiting for orders.

Worlds To Conquer
Not content with even this friendly a system, though, Insterstel has provided more. You can lump a bunch of units together (Group Survey command) and issue them all the same order. You can put armies on sentry duty, in which case they won't flash at all until you take them off duty. And you can command a transport unit to load all units that come adjacent to it, without having to load each one individually.

You can fight any adjacent enemy unit. To fight, simply move the unit to the enemy's square. The battle is over when one unit is destroyed. Extremely simple, the system is highly effective. You both destroy and lose many units in this game, especially in a large-scale war.

Two things about *Empire* are particularly appealing. First, since you learn about the planet only as you actually move across it, the game contains the excitement of exploration (the heart of all adventure games). Second, it truly rewards strategy. To launch a full-scale amphibious invasion, for example, demands a host of armies and transports, protection from enemy fleets, and fighters and ships to bombard the shore. But unlike most games, creating these units and using them is extremely easy; what's difficult is getting everything to the right place intact and at the right time.

Strategy gamers should find nothing to dislike here. *Empire* contains one of the best interfaces I've seen on any complex game, and its design foundation is strong. It contains a Map Generation system (it produces random maps as well), so you will never run out of new worlds to liberate. It plays as well with three players as it does with one, and the one-player game is a constant challenge. This is a superb, addictive game.

Empire
Interstel Corporation
P.O. Box 57825
Webster, TX 77598
Distributed by Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
$49.95

Casino Blackjack

John Hamilton

Win your ticket home or lose your shirt in this accurate simulation of the most popular Las Vegas card game. Originally written for the Commodore 64, we've included versions for the Atari eight-bit computers, Apple II series, IBM PC/PCjr and compatibles, and the Amiga. The PC version requires BASICA or GW-BASIC or Cartridge BASIC for the PCjr. The Apple II version works on any Apple II-series computer under either DOS 3.3 or ProDOS. The Amiga version requires 512K of RAM.

"Casino Blackjack" is *real* blackjack, played the same way it's played in the best Las Vegas casinos. Starting with a bankroll of $100, you play against an intelligent computer dealer. Just like in the casinos, you can "split" your hand, double down, or buy insurance.

The original version of Casino Blackjack was written on the Commodore 64. We've included translations for the Atari 400, 800, XL and XE, Amiga, Apple II series, and IBM PC/PCjr and compatibles. Type in the version for your computer and save a copy; then, run the program.

In The Cards

The rules for Casino Blackjack are simple to learn. The object of the game is to get more points than the dealer—without exceeding 21 points. An Ace can be counted as either 1 point or 11 points. Jacks, Queens, and Kings are all worth 10 points. All other cards count as their face value.

Before each hand, you decide how much of your bankroll you want to bet. Next, four cards are dealt—two to you, and two to the computer dealer. The dealer's bottom card is face down and his top card is face up, so you can only guess at the total number of points he has. Both your cards are dealt face up.

If you've been dealt a "blackjack" (21 points), you automatically win 1½ times your bet, unless the dealer was also dealt a blackjack, in which case no money exchanges hands.

After you've taken a look at your cards, decide whether you want to take a hit (take another card) or stand with the cards you have. If you take a hit and exceed 21 points, you have gone "bust" (lost the hand). If you take a card and still have less than 21 points,

"Casino Blackjack" for the 64—so realistic you'll think you're in Las Vegas.

The Apple II version of "Casino Blackjack" features custom-designed cards.

you're free to take another hit if you like. This process continues until you stand or bust.

If you stand, the dealer may deal himself a card. The dealer

C▪www.commodore.ca

"Casino Blackjack" for the Atari 400, 800, XL, and XE.

The IBM PC and compatible version of "Casino Blackjack."

"Casino Blackjack" for Amiga.

takes a hit if he has 16 or less points. He stands on 17 or more. If the dealer doesn't go bust, the cards are inspected to see who came closest to 21. A tie is known as a "push." In the case of a push, no money is lost, otherwise, the winner takes the money.

Special Plays

If the dealer's top card is an Ace, you may elect to insure yourself against the dealer having blackjack. You pay half your original bet for insurance. If the dealer does have blackjack, you keep your bet. You lose the insurance money, whether the dealer has blackjack or not.

If you're dealt a total of 10 or 11 points, you may double your bet. This is known as "doubling down." If you choose to double down, you're dealt one more card and then you automatically stand. If you think the dealer will win the hand, you may choose to "surrender" instead. Surrendering costs half your bet.

If you're dealt two cards of the same denomination, you may elect to "split" your hand. When you split, you double your original bet and play the two cards as two separate hands, one after the other. Each hand has a chance to beat the dealer.

Commodore 64 Version

Two programs make up the 64 version of Casino Blackjack—a BASIC program and a machine language program. Type in Program 1 (the BASIC program) and save it to disk. Do not run it yet—it attempts to load Program 2 when it runs.

Program 2 (the machine language program) must be entered with the "MLX" machine language entry program found elsewhere in this issue. Here are the addresses you need to type in the program with MLX:

Starting Address: C000
Ending Address: C247

After you've typed in all the data from Program 2, be sure to save a copy as BLACKJACK.OBJ before you leave MLX.

Apple II Version

The Apple version of Casino Blackjack is made up of two programs. Program 3 is the BASIC portion. Type it in and save it to disk. Do not run the program until you've entered and saved Program 4—it attempts to load the machine language program when it's run.

Program 4 is written in machine language. It's the same ML program used for "Canfield" in January 1988. If you've already typed in CANFIELD.ML, you can copy it to the same disk you saved Program 3 to.

If you haven't typed in CANFIELD.ML, type in Program 4 using the "Apple MLX" machine language entry program. When prompted for starting and ending addresses, respond with the following values:

Starting Address: 8000
Ending Address: 86A7

Be sure to save a copy of the program as "CANFIELD.ML" after you've finished typing in the data.

Atari 400, 800, XL, And XE Version

On the Atari, Casino Blackjack is written entirely in BASIC. Type in Program 5 and save a copy before you run the program.

IBM PC/PCjr And Compatibles

Casino Blackjack runs under both BASICA and GW-BASIC on both color and monochrome monitors.

Type in Program 6 and save a copy before you run the program.

Amiga

The Amiga version of Casino Blackjack is written in Amiga Basic and requires 512K of memory. Type in Program 7 and save a copy to disk before running the program.

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: Commodore 64 Casino Blackjack—BASIC

```
HE 10 REM COPYRIGHT 1988 COMPU
        TE! PUBLICATIONS, INC. -
        ALL RIGHTS RESERVED
MH 20 PRINT"{CLR}{2 SPACES}COP
        YRIGHT 1988 COMPUTE! PUB
        L., INC."
CA 30 PRINTTAB(11)"ALL RIGHTS
        {SPACE}RESERVED"
BH 40 IFA=0THENA=1:LOAD"BLACKJ
        ACK.OBJ",8,1
BX 50 SYS49152:POKE53272,21
GK 60 X=RND(-TI):CLR:DIMCH(13,
        4):POKE53280,5:POKE53281
        ,5:POKE53265,91:POKE5328
        3,1
JK 70 PRINT"{CLR}"TAB(15)"
        {11 DOWN}{BLK}BLACKJACK"
        :FORU=1TO1000:NEXT
PS 80 .SI$(1)="Z[{DOWN}{2 LEFT}
        £]":SI$(3)="↑↑{DOWN}
        {2 LEFT}£]":SI$(2)="$%
        {DOWN}{2 LEFT}&'"
RR 90 SI$(4)="Z[{DOWN}{2 LEFT}
        ◄#":PRINT"{CLR}":V=0:H=0
        :TY=4:POKE53272,29:FORB=
        1TO4:FORC=0TO11
PJ 100 A=C:A1=B:GOSUB1070:H=H+
        3:NEXTC:V=V+6:H=0:NEXTB
        :H=14:V=3:A=12:A1=3
BS 110 GOSUB1070:H=H+6:V=V+6:A
        1=1:GOSUB1070:H=H-6:V=V
        +6:A1=4:GOSUB1070:H=H-6
MH 120 V=V-6:A1=2:GOSUB1070:H=
        H+6:GOSUB1110:DO=100:HA
        =0
GX 130 GETA$:IFA$=""THEN130
PD 140 IFDO<1THEN1320
HR 150 HA=HA+1:POKE53272,21
GA 160 AM$="":PRINT"{CLR}{BLK}
        YOU WILL START HAND"HA
        {LEFT}.":SYS49158,0,6
SG 170 PRINT"RETURN ALONE WILL
        DISPLAY PREVIOUS HAND"
        :PRINT"{DOWN}- Q - TO Q
        UIT"
```

```
MQ 180 PRINT"{DOWN}- R - TO RE
       START GAME":PRINT"
       {DOWN}PLACE BET"DO;:INP
       UT"DOLLARS LEFT ";AM$
PR 190 IFAM$=""ANDPR=1THEN250
MA 200 IFAM$=""ANDPR=0THEN240
DD 210 IFAM$="Q"THENEND
FA 220 IFAM$="R"THENRUN
CC 230 GOTO260
DB 240 GOSUB1050:PRINTTAB(6)"
       {RVS}{RED}NO PREVIOUS H
       AND WAS PLAYED":GOSUB11
       20:GOTO160
BA 250 POKE53272,29:SYS49155,3
       :GOSUB1050:GOSUB1120:PO
       KE53272,21:GOTO160
FE 260 AM=VAL(AM$):AM=INT(100*
       (AM+.005))/100
MS 270 IFAM<.01THENGOSUB1050:P
       RINT"YOU CANNOT BET ZER
       O DOLLARS":GOSUB1120:GO
       TO160
KM 280 IFAM>DOTHENGOSUB1050:PR
       INT"YOU DO NOT HAVE THA
       T MUCH MONEY":GOSUB1120
PD 290 IFAM>DOTHEN160
SK 300 Z2=0:Z1=0:DE=0:D1=0:D2=
       0:P=0:D3=0:CL=0:JK=0:HI
       =0:BU=0:SD=0:DD=0:SP=0:
       QQ=0
DE 310 S1=0:QP=0:SP=0:B1=0:O1=
       0:O2=0:PL(0)=0:PL(1)=0:
       PL(2)=0:PL(3)=0::PL(4)=
       0
AK 320 PG(0)=AM:D=0:PR=0:O1=DO
       -AM:O2=O1:PRINT"{CLR}":
       POKE53272,29:V=0:H=0:Z2
       =DE:Z1=D
MC 330 GOSUB960:DE=Z2:D=Z1:GOS
       UB1110:D1=A:D2=A1:V=7:H
       =0:GOSUB1030:P1=A:P2=A1
FD 340 H=6:V=0:GOSUB1040:D3=A:
       H=6:V=7:GOSUB1030:P3=A
BP 350 IFP1=12ANDP3<12ANDP3>7O
       RP3=12ANDP1<12ANDP1>7TH
       EN1310
JM 360 GOSUB1150:IFIN>0THENIN=
       0:GOTO140
KF 370 GOSUB1050:PRINT"{RVS}
       {RED}H{OFF}{BLK}IT
       {RVS}{RED}S{OFF}{BLK}TA
       ND S{RVS}{RED}P{OFF}
       {BLK}LIT {RVS}{RED}D
       {OFF}{BLK}OUBLE DOWN SU
       RR{RVS}{RED}E{OFF}{BLK}
       NDER"
EC 380 GETOP$:IFOP$=""THEN380
SK 390 IFOP$="P"THENJK=3:GOTO4
       10
GK 400 JK=ABS(ASC(OP$)-77)-4:I
       FJK<0ORJK>5THEN380
XX 410 ONJKGOTO420,530,540,640
       ,690
QA 420 HI=1:GOSUB510
HJ 430 IFDD=1THENH=12:GOSUB103
       0:DD=2:SD=1:GOSUB510
JQ 440 IFSD=1THENSD=0:H=6:V=V+
       3:HI=0:GOTO470
BX 450 IFB1=1THENB1=0:GOTO370
KQ 460 H=H+6:GOSUB1030:GOSUB51
       0:B1=1:GOTO430
JD 470 IFSP>0THENSP=SP-1:S1=S1
       +1:QP=QP-1:P=0:GOTO490
DC 480 GOTO800
BX 490 IFJ(S1)=1THENJ(S1)=0:P=
       1
PP 500 GOSUB1030:B1=1:GOTO430
QE 510 IFBU=1THENSYS49158,35,V
       +1:PRINT"{RVS}{RED}BUST
       ":SD=1:DD=2:BU=0:HI=0
EG 520 RETURN
CJ 530 SYS49158,35,V+1:PRINT"
```

```
       {RVS}{RED}STAND":SD=1:G
       OTO440
FG 540 IFHI=1THENGOSUB1050:GOS
       UB620:GOSUB1120:GOTO370
FK 550 IFA<>P1THENGOSUB1050:PR
       INT"YOU CANNOT SPLIT SI
       NCE NOT OF SAME"
PK 560 IFA<>P1THENPRINT"VALUE
       {SPACE}OR DENOMINATION"
       :GOSUB1120:GOTO370
JM 570 IFO1-AM<0THENGOSUB1050:
       GOSUB630:GOSUB1120:GOTO
       370
RM 580 O1=O1-AM:SP=SP+1:QQ=QQ+
       1:QP=QP+1:PL(S1+QP)=B:P
       G(S1+QP)=AM:PL(S1)=B
FD 590 QV=V
CR 600 IFP>0THENP=1:J(S1+QP)=1
PC 610 H=0:V=7+QQ*3:GOSUB1070:
       V=QV:H=6:GOSUB1030:GOTO
       370
GS 620 PRINT"YOU CAN SPLIT ON
       {SPACE}THE FIRST TWO CA
       RDS{4 SPACES}ONLY":RETU
       RN
FC 630 PRINT"NOT ENOUGH MONEY
       {SPACE}FOR THE SPLIT":R
       ETURN
FH 640 IFHI=1THENGOSUB1050:GOS
       UB670:GOSUB1120:GOTO370
EA 650 IFDD=1THENGOSUB1050:GOS
       UB680:GOSUB1120:GOTO370
DA 660 PG(S1)=AM/2:PL(S1)=0:SD
       =1:SE=1:SYS49158,31,V+1
       :PRINT"{RVS}{RED}SURREN
       DER":GOTO430
BM 670 PRINT"YOU CANNOT SURREN
       DER SINCE YOU ALREADY
       {2 SPACES}HIT ON THIS H
       AND":RETURN
XP 680 PRINT"YOU CANNOT SURREN
       DER SINCE YOU DOUBLED
       {2 SPACES}DOWN ON THIS
       {SPACE}HAND"
SK 690 IFHI=1THENGOSUB1050:GOS
       UB770:GOSUB1120:GOTO370
RX 700 IFP>0THENAZ=PL(S1)-10:G
       OTO720
FH 710 AZ=PL(S1)
HJ 720 IFAZ<>10ANDAZ<>11THENGO
       SUB1050:GOSUB780:GOSUB1
       120:GOTO370
MC 730 IFO1-AM<0THENGOSUB1050:
       GOSUB790:GOSUB1120:GOTO
       370
MA 740 IFP>0THENPL(S1)=PL(S1)-
       10
PR 750 PG(S1)=PG(S1)+AM:SYS491
       58,29,V+1:PRINT"{RVS}
       {RED}DOUBLE DOWN":DD=1:
       O1=O1-AM:GOTO430
SC 760 STOP
SC 770 PRINT"SORRY YOU ALREADY
        HIT ON THIS HAND":RETU
       RN
SD 780 PRINT"YOU CAN ONLY DOUB
       LE DOWN ON A 10 OR 11":
       RETURN
JH 790 PRINT"NOT ENOUGH MONEY
       {SPACE}TO DOUBLE DOWN":
       RETURN
QJ 800 A=D1:A1=D2:V=0:H=0:GOSU
       B1070:H=6:IFBJ=1THENBJ=
       0:GOTO870
AJ 810 IFDE=21THENFORI=0TO3:PL
       (SG)=0:NEXT:GOTO870
BS 820 IFS1<1ANDPL(0)>21THEN87
       0
EC 830 IFS1<1ANDSE=1THENSE=0:G
       OTO870
PX 840 IFDE>16THEN860
QM 850 H=H+6:GOSUB1040:GOTO840
```

```
KM 860 IFDE>21THENSYS49158,35,
       V+3:PRINT"{RVS}{RED}BUS
       T"
FG 870 V=7:FORI=0TOS1
GR 880 IFDE>21ANDPL(I)<22ORDE<
       22ANDDE<PL(I)ANDPL(I)<2
       2THENGOSUB950
GE 890 IFDE<22ANDDE>PL(I)ORDE<
       22ANDPL(I)>21ORDE>21AND
       PL(I)>22THENGOSUB940
KH 900 IFDE<22ANDDE=PL(I)THENS
       YS49158,35,V+2:PRINT"
       {RVS}{RED}PUSH":PG(I)=0
QJ 910 DO=DO+PG(I):V=V+3:NEXT:
       DO=INT(100*(DO+.005))/1
       00
MF 920 GOSUB1050:GOSUB1120:FOR
       I=1TO13:FORJ=1TO4:CH(I,
       J)=0:NEXTJ,I:SYS49155,1
       :PR=1
FQ 930 GOTO140
AJ 940 SYS49158,35,V+2:PRINT"
       {RVS}{RED}LOSE":PG(I)=-
       PG(I):RETURN
MF 950 SYS49158,35,V+2:PRINT"
       {RVS}{RED}WIN":RETURN
HB 960 T=FRE(0):A=INT(13*RND(1
       )):A1=INT(4*RND(1)+1):I
       FCH(A,A1)=1THEN960
CX 970 CH(A,A1)=1:B=A+2:IFA+2>
       9THENB=10
SH 980 IFA+2=14THENB=11
GJ 990 IFB=11THENZ1=Z1+1
GF 1000 Z2=Z2+B:IFZ2>21ANDZ1>0
        THENZ2=Z2-10:Z1=Z1-1
JX 1010 IFZ2>21THENBU=1
SS 1020 RETURN
BK 1030 Z2=PL(S1):Z1=P:GOSUB96
        0:PL(S1)=Z2:P=Z1:GOSUB
        1070:RETURN
XB 1040 Z2=DE:Z1=D:GOSUB960:DE
        =Z2:D=Z1:GOSUB1070:RET
        URN
KE 1050 CL=22:FORZZ=CLTO24:SYS
        49158,0,ZZ
HC 1060 PRINT"{39 SPACES}{UP}"
        :NEXT:SYS49158,0,22:RE
        TURN
DE 1070 IFA1=20RA1=4THENPOKE64
        6,0:GOTO1090
CP 1080 POKE646,2
MX 1090 SYS49158,H,V:PRINT"
        {RVS}(....){DOWN}
        {6 LEFT}",CHR$(A+50)"
        {3 SPACES}{DOWN}
        {6 LEFT}","SI$(A1);
RA 1100 PRINT"{RVS}{UP} -
        {DOWN}{6 LEFT},
        {2 RIGHT} -{DOWN}
        {6 LEFT},{3 SPACES}"CH
        R$(A+50)"{DOWN}
        {6 LEFT}*////+":RETURN
KG 1110 FORZZ=0TO5:SYS49158,H,
        V+ZZ:PRINT"{RVS}{7}!!!
        !!!":NEXTZZ:RETURN
FB 1120 SYS49158,13,24:PRINT"
        {RVS}{RED}PRESS ANY KE
        Y{BLK}{4 UP}"
AM 1130 GETY$:IFY$=""THEN1130
KG 1140 RETURN
CS 1150 IFD3<>12THENRETURN
EA 1160 GOSUB1050:PRINT"{BLK}W
        ANT INSURANCE {RVS}
        {RED}Y N{OFF}"
DR 1170 GETY$:IFY$=""THEN1170
HF 1180 IFY$="Y"THEN1210
DC 1190 IFY$="N"THENRETURN
AQ 1200 GOTO1170
SQ 1210 IFO2-AM/2<0THENGOSUB10
        50:GOSUB1280:GOSUB1120
        :RETURN
GS 1220 IN=AM/2:O2=O2-AM/2:DO=
```

```
        DO-IN:IN=0
PM 1230 IFDE<>21THENGOSUB1290:
        IN=0:GOTO1270
XE 1240 A=D1:A1=D2:V=0:H=0:GOS
        UB1070:GOSUB1050
PX 1250 PRINT"{BLK}DEALER HAS
        {SPACE}BLACKJACK YOU L
        OSE YOUR{6 SPACES}INSU
        RANCE MONEY AND SAVE "
        ;
PP 1260 PRINT"BET.":GOSUB1120:
        IN=1
XB 1270 RETURN
JH 1280 PRINT"{BLK}YOU DO NOT
        {SPACE}HAVE ENOUGH MON
        EY TO INSURE{2 SPACES}
        YOUR BET":RETURN
AK 1290 GOSUB1050:PRINT"{BLK}D
        EALER DID NOT GET BLAC
        KJACK YOU LOSE
        {3 SPACES}INSURANCE ";
QG 1300 PRINT"MONEY AND HAND C
        ONTINUES":GOSUB1120:RE
        TURN
PX 1310 SYS49158,31,V+1:PRINT"
        {RVS}{RED}BLACKJACK":P
        G(S1)=1.5*AM:BJ=1:GOTO
        800
EX 1320 PRINT"{CLR}{10 DOWN}
        {RVS}{RED}YOU ARE OUT
        {SPACE}OF MONEY"
CG 1330 PRINT"{DOWN}{RVS}{RED}
        WISH TO PLAY AGAIN
        {BLK}Y N{RED}}?"
XF 1340 GETY$:IFY$=""THEN1340
HC 1350 IFY$="Y"THEN90
GM 1360 IFY$="N"THENEND
RK 1370 GOTO1340
```

Program 2: Commodore 64 Casino Blackjack—ML

```
C000:4C 09 C0 4C AE C0 4C 0E E5
C008:C1 A9 30 85 34 85 38 20 7B
C010:5E A6 A9 00 98 AA A9 0E 4F
C018:85 19 A9 DC 85 1A A9 00 8D
C020:85 1B A9 30 85 1C A9 00 53
C028:85 1D A9 D0 85 1E A9 FE EC
C030:31 19 91 19 A9 FB 25 01 DC
C038:85 01 A0 00 B1 19 1B 1B 1B
C040:C8 C0 00 D0 F7 18 E6 1C 6D
C048:E6 1E E8 E0 09 D0 EB A0 F3
C050:00 A9 04 05 01 85 01 A9 D6
C058:01 11 19 91 19 A9 1D 8D 12
C060:18 D0 A9 00 85 19 A9 30 78
C068:85 1A A9 00 AA A8 B9 2A FD
C070:C1 91 19 C0 2F F0 04 C8 74
C078:4C 6E C0 A9 18 85 19 A9 21
C080:85 1A A9 00 AA B9 8A B9 8B
C088:5A C1 91 19 C0 DF F0 04 D6
C090:C8 4C 87 C0 A9 08 85 19 18
C098:A9 31 85 1A A9 00 A8 AA D6
C0A0:B9 3A C2 91 19 C0 07 F0 C9
C0A8:04 C8 4C 4C 60 20 FD B7
C0B0:AE 20 9E AD 8D 0D C1 20 84
C0B8:E7 C0 E8 C8 AD 0D C1 C9 F6
C0C0:81 D0 0B B1 19 91 1B B1 AA
C0C8:1D 91 1F 4C D6 C0 B1 1B 1E
C0D0:91 19 B1 1F 91 1D C0 FF 0C
C0D8:D0 E1 E6 1A E6 1C E6 1E 4D
C0E0:E6 20 E0 04 D0 D4 60 A9 7E
C0E8:00 AA A0 FF A9 00 85 19 9A
C0F0:A9 04 85 1A A9 43 85 1B 1B
C0F8:A9 C2 85 1C A9 00 85 1D 1D
C100:A9 D8 85 1E A9 43 85 1F A5
C108:A9 C6 85 20 60 00 20 20 28
C110:C1 8C 29 C1 20 20 C1 98 76
C118:AA AC 29 C1 18 4C 60 FF 31
C120:20 FD AE 20 9E AD 4C AA F9
C128:B1 00 01 03 07 0F 1F 3F C6
C130:7F FF 80 C0 E0 F0 F8 FC 49
C138:FE FF FF 7F 3F 1F 0F 07 CE
C140:03 01 FF FE FC F8 F0 E0 04
C148:C0 80 00 00 00 18 3C 7E A3
C150:FF FF FF 7F 3D 19 01 03 1F
C158:07 0F FF FE BC 98 80 C0 1D
C160:E0 F0 03 07 0F 0F 07 3B 5F
C168:7D FF C0 E0 F0 F0 E0 DC BA
C170:BE FF FF FF 7F 7D 39 03 BE
C178:07 0F FF FF BE 9C C0 38
C180:E0 F0 FF 80 80 80 80 80 40
C188:80 80 FF 01 01 01 01 01 8B
C190:01 01 80 80 80 80 80 74
C198:80 FF 01 01 01 01 01 9B
C1A0:01 FF 80 80 80 80 80 44
C1A8:80 80 01 01 01 01 01 CB
C1B0:01 01 FF 00 00 00 00 F4
C1B8:00 00 00 00 00 00 00 3C
C1C0:00 FF 00 7E 42 42 42 42 0E
C1C8:42 7E 00 78 08 08 08 08 0D
C1D0:08 7E 00 7E 02 02 7E 40 35
C1D8:40 7E 00 7E 02 02 1E 02 5A
C1E0:02 7E 00 40 40 44 44 7E 23
C1E8:04 04 00 7E 40 7E 02 59
C1F0:02 7E 00 7E 40 40 7E 42 3F
C1F8:42 7E 00 7E 02 02 02 02 43
C200:02 02 00 7E 42 42 42 42 49
C208:42 7E 00 7E 42 42 7E 02 50
C210:02 7E 00 CE 4A 4A 4A 4A 7D
C218:4A EE 00 7E 08 08 08 48 1F
C220:48 78 00 7E 42 42 42 4A B9
C228:44 7A 00 42 42 44 78 44 EA
C230:42 42 00 7E 42 42 7E 42 A9
C238:42 42 CC CC 33 33 CC CC A2
C240:33 33 00 00 00 00 00 00 2C
```

Program 3: Apple II Casino Blackjack—BASIC

Version by William Chin, Editorial Programmer

```
E7 10 REM COPYRIGHT 1988 COMPUTE
      ! PUBLICATIONS, INC. - ALL
       RIGHTS RESERVED
B6 20 TEXT : HOME : PRINT " COP
      YRIGHT 1988 COMPUTE! PUBL.
      , INC."
AE 30 HTAB 11: PRINT "ALL RIGHTS
       RESERVED"
09 31 VTAB 12: HTAB 13: PRINT "A
      NY KEY TO START"
CC 32 POKE 49168,0
30 33 T = RND (1): IF PEEK (4915
      2) = 13 THEN 33
C9 35 POKE 40196,189: POKE 40197
      ,158
C5 40 PRINT CHR$ (4);"BLOAD CANF
      IELD.ML,A$8000"
0F 50 POKE 6,168: POKE 7,131: IF
       PEEK (48896) = 76 THEN PR
      INT CHR$ (4);"PR#A$8340":
      PRINT : GOTO 136
04 60 POKE 54,64: POKE 55,131: C
      ALL 1002
2C 136 DIM CH(13,4),C(255): HGR2
      : BL$ = "
                              "
7E 137 ACE = 12:DO = 100:HA = 0:
       RESTORE
59 138 FOR I = 1 TO 5: READ A$,A
      M$:C( ASC (A$)) = I:C( AS
      C (AM$)) = I: NEXT I
F8 139 DATA "H","h","S","s","P",
      "p","E","e","D","d"
4A 140 IF DO < 1 THEN 1320
1C 150 HA = HA + 1
08 160 IF PF = 1 THEN HGR2
88 168 AM$ = "": VTAB 12: PRINT
      "YOU WILL START HAND "HA
      "."
3E 170 PRINT "Q - TO QUIT"
EA 180 PRINT "R - TO RESTART GAM
      E": PRINT DO;" DOLLARS LE
      FT": PRINT "PLACE BET
      ";: HTAB 11
37 182 T = 0: GET A$: IF A$ = CH
      R$ (13) THEN 190
11 183 IF A$ < > CHR$ (8) AND A$
      < > CHR$ (127) THEN 186
44 184 IF LEN (AM$) < 2 THEN AM$
      = "": GOTO 189
FF 185 AM$ = LEFT$ (AM$, LEN (AM
      $) - 1): GOTO 189
80 186 IF A$ = "Q" OR A$ = "R" O
      R A$ = "q" OR A$ = "r" TH
      EN AM$ = A$: GOTO 190
B4 187 A = ASC (A$): IF (A > 57
      OR A < 48) AND A < > 46 T
      HEN 182
99 188 AM$ = AM$ + A$:AM$ = LEFT
      $ (AM$,10)
A2 189 HTAB 11: PRINT "
      "; HTAB 11: PRINT AM$;
      : GOTO 182
26 190 IF AM$ = "" THEN PRINT CH
      R$ (7);: GOTO 182
C0 210 IF AM$ = "Q" OR AM$ = "q"
      THEN TEXT : HOME : END
82 220 IF AM$ = "R" OR AM$ = "r"
      THEN HOME : HGR2 : GOTO
      137
18 230 GOTO 260
AA 250 GOSUB 1050: GOSUB 1120: G
      OTO 160
9F 260 AM = VAL (AM$):AM = INT (
      100 * (AM + .005)) / 100
9F 270 IF AM < .01 THEN GOSUB 10
      50: PRINT "YOU CANNOT BET
      ZERO DOLLARS": GOSUB 112
      0: GOTO 160
58 280 IF AM > DO THEN GOSUB 105
      0: PRINT "YOU DO NOT HAVE
      THAT MUCH MONEY": GOSUB
      1120: GOTO 160
02 300 Z2 = 0:Z1 = 0:DE = 0:D1 =
      0:D2 = 0:P = 0:D3 = 0:CL
      = 0:JK = 0:HI = 0:BU = 0
      :SD = 0:DD = 0:SP = 0:QQ
      = 0:SE = 0
87 310 S1 = 0:QP = 0:SP = 0:B1 =
      0:O1 = 0:O2 = 0:PL(0) =
      0:PL(1) = 0:PL(2) = 0:PL(
      3) = 0:PL(4) = 0:PF = 0
E8 320 PG(0) = AM:D = 0:PR = 0:O
      1 = DO - AM:O2 = O1: HGR2
      :V = 0:H = 0:Z2 = DE:Z1
      = D
6C 330 GOSUB 960:DE = Z2:D = Z1:
      D1 = A:D2 = A1:V = 7:H =
      0: GOSUB 1030:P1 = A:P2 =
      A1
3A 332 HCOLOR= 3:V = 0:TX = H *
      6 + 1:TY = V * 8: HPLOT T
      X,TY TO TX + 16,TY TO TX
      + 16,TY + 23 TO TX,TY + 2
      3 TO TX,TY
69 334 FOR I = 4 TO 12 STEP 4: H
      PLOT TX,TY + I TO TX + 16
      ,TY + I + 8: NEXT
75 340 H = 6:V = 0: GOSUB 1040:D
      3 = A:H = 6:V = 7: GOSUB
      1030:P3 = A
21 350 IF P1 = ACE AND P3 < ACE
      AND P3 > 7 OR P3 = ACE AN
      D P1 < ACE AND P1 > 7 THE
      N 1310
45 360 GOSUB 1150: IF IN > 0 THE
      N IN = 0: GOTO 140
2A 370 GOSUB 1050: PRINT "H HIT
      S STAND P SPLIT": PRINT
      "D DOUBLE E SURRENDER"
09 380 GET OP$:JK = C( ASC (OP$))
49 390 IF JK = 0 THEN 380
A9 410 ON JK GOTO 420,530,540,64
      0,690
4E 420 HI = 1: GOSUB 510
FB 430 IF DD = 1 THEN H = 12: GO
      SUB 1030:DD = 2:SD = 1: G
      OSUB 510
47 440 IF SD = 1 THEN SD = 0:H =
      6:V = V + 4:HI = 0: GOTO
      470
```

```
03 450 IF B1 = 1 THEN B1 = 0: GO
       TO 370
FA 460 H = H + 6: GOSUB 1030: GO
       SUB 510:B1 = 1: GOTO 430
3D 470 IF SP > 0 THEN SP = SP -
       1:S1 = S1 + 1:QP = QP - 1
       :P = 0: GOTO 490
21 480 GOTO 800
C3 490 IF J(S1) = 1 THEN J(S1) =
       0:P = 1
E8 500 GOSUB 1030:B1 = 1: GOTO 4
       30
A7 510 IF BU = 1 THEN HTAB 36: V
       TAB V + 2: PRINT "BUST":S
       D = 1:DD = 2:BU = 0:HI =
       0
19 520 RETURN
E8 530 HTAB 36: VTAB V + 2: PRIN
       T "STAND":SD = 1: GOTO 44
       0
65 540 IF HI = 1 THEN GOSUB 1050
       : GOSUB 620: GOSUB 1120:
       GOTO 370
80 550 IF A < > P1 THEN GOSUB 10
       50: PRINT "CARDS MUST BE
       THE SAME TO SPLIT.": GOSU
       B 1120: GOTO 370
1C 570 IF O1 - AM < 0 THEN GOSUB
       1050: GOSUB 630: GOSUB 1
       120: GOTO 370
66 580 O1 = O1 - AM:SP = SP + 1:
       QQ = QQ + 1:QP = QP + 1:P
       L(S1 + QP) = B:PG(S1 + QP
       ) = AM:PL(S1) = B
95 590 QV = V:PF = 1
52 600 IF P > 0 THEN P = 1:J(S1
       + QP) = 1
89 610 H = 0:V = 7 + QQ * 4: GOS
       UB 1070:V = QV:H = 6: GOS
       UB 1030: GOTO 370
B2 620 PRINT "MAY ONLY SPLIT ON
       THE FIRST TWO CARDS.": RE
       TURN
A8 630 PRINT "DO NOT HAVE ENOUGH
        MONEY FOR SPLIT.": RETUR
       N
07 640 IF HI = 1 THEN GOSUB 1050
       : GOSUB 670: GOSUB 1120:
       GOTO 370
09 660 PG(S1) = AM / 2:PL(S1) =
       0:SD = 1:SE = 1: HTAB 32:
       VTAB V + 2: PRINT "SURRE
       NDER": GOTO 430
F5 670 PRINT "YOU CAN NOT SURREN
       DER BECAUSE YOU": PRINT "
       ALREADY HIT ON THIS HAND.
       ": RETURN
21 690 IF HI = 1 THEN GOSUB 1050
       : GOSUB 770: GOSUB 1120:
       GOTO 370
75 700 IF P > 0 THEN AZ = PL(S1)
        - 10: GOTO 730
50 710 AZ = PL(S1)
9C 730 IF O1 - AM < 0 THEN GOSUB
       1050: GOSUB 790: GOSUB 1
       120: GOTO 370
F2 740 IF P > 0 THEN PL(S1) = PL
       (S1) - 10
C4 750 PG(S1) = PG(S1) + AM: HTA
       B 29: VTAB V + 2: PRINT "
       DOUBLE DOWN":DD = 1:O1 =
       O1 - AM: GOTO 430
F0 770 PRINT "SORRY YOU ALREADY
       HIT ON THIS HAND.": RETUR
       N
D6 790 PRINT "YOU DO NOT ENOUGH
       MONEY TO DOUBLE DOWN.": R
       ETURN
6A 800 A = D1:A1 = D2:V = 0:H =
       0: GOSUB 1070:H = 6: IF B
       J = 1 THEN BJ = 0: GOTO 8
       70
BA 810 IF DE = 21 THEN FOR I = 0
        TO 3:PL(I) = 0: NEXT : G
       OTO 870
4A 820 IF S1 < 1 AND PL(0) > 21
       THEN 870
F3 830 IF S1 < 1 AND SE = 1 THEN
       SE = 0: GOTO 870
32 840 IF DE > 16 THEN 860
4B 850 H = H + 6: GOSUB 1040: GO
       TO 840
8B 860 IF DE > 21 THEN HTAB 36:
       VTAB V + 3: PRINT "BUST"
46 870 V = 7: FOR I = 0 TO S1
AF 880 IF DE > 21 AND PL(I) < 22
       OR DE < 22 AND DE < PL(I
       ) AND PL(I) < 22 THEN GOS
       UB 950
7D 890 IF DE < 22 AND DE > PL(I)
       OR DE < 22 AND PL(I) > 2
       1 OR DE > 21 AND PL(I) >
       22 THEN GOSUB 940
43 900 IF DE < 22 AND DE = PL(I)
       THEN HTAB 36: VTAB V + 3
       : PRINT "PUSH":PG(I) = 0
CA 910 DO = DO + PG(I):V = V + 4
       : NEXT :DO = INT (100 * (
       DO + .005)) / 100
CF 920 GOSUB 1050: FOR I = 1 TO
       13: FOR J = 1 TO 4:CH(I,J
       ) = 0: NEXT J,I:PR = 1
9C 930 GOTO 140
3C 940 HTAB 35: VTAB V + 3: PRIN
       T " LOSE":PG(I) = - PG(I)
       : RETURN
BF 950 HTAB 35: VTAB V + 3: PRIN
       T " WIN": RETURN
C2 960 T = FRE (0):A = INT ( RND
       (1) * 13):A1 = INT (4 *
       RND (1) + 1): IF CH(A,A1)
       = 1 THEN 960
1E 970 CH(A,A1) = 1:B = A + 2: I
       F A > 7 THEN B = 10
F4 980 IF A = ACE THEN B = 11
86 990 IF B = 11 THEN Z1 = Z1 +
       1
EA 1000 Z2 = Z2 + B: IF Z2 > 21
       AND Z1 > 0 THEN Z2 = Z2
       - 10:Z1 = Z1 - 1
E8 1010 IF Z2 > 21 THEN BU = 1
D9 1020 RETURN
3F 1030 Z2 = PL(S1):Z1 = P: GOSU
       B 960:PL(S1) = Z2:P = Z1
       : GOSUB 1070: RETURN
D3 1040 Z2 = DE:Z1 = D: GOSUB 96
       0:DE = Z2:D = Z1: GOSUB
       1070: RETURN
63 1050 CL = 22: FOR ZZ = CL TO
       24: HTAB 1: VTAB ZZ
9E 1060 PRINT BL$: NEXT : HTAB 1
       : VTAB 22: RETURN
37 1070 IF A1 = 2 OR A1 = 4 THEN
       1090
41 1090 T = A1 * 13 + A - 11:TX
       = H * 6 + 1:TY = V * 8
A4 1092 IF A = ACE THEN T = A1 *
       13 - 12
85 1094 CALL 32768,T,TX,TY
D3 1100 RETURN
BF 1120 HTAB 13: VTAB 24: PRINT
       "PRESS ANY KEY"
9B 1130 GET Y$: GOTO 1050
C6 1140 REM
2F 1150 IF D3 < > 12 THEN RETURN
30 1160 GOSUB 1050: PRINT "WANT
       INSURANCE Y N"
F4 1170 GET Y$: IF Y$ = "" THEN
       1170
FD 1180 IF Y$ = "Y" OR Y$ = "y"
       THEN 1210
AD 1190 IF Y$ = "N" OR Y$ = "n"
       THEN RETURN
6C 1200 GOTO 1170
24 1210 IF O2 - AM / 2 < 0 THEN
       GOSUB 1050: GOSUB 1280:
       GOSUB 1120: RETURN
72 1220 IN = AM / 2:O2 = O2 - AM
       / 2:DO = DO - IN:IN = 0
F4 1230 IF DE < > 21 THEN GOSUB
       1290:IN = 0: GOTO 1270
E7 1240 A = D1:A1 = D2:V = 0:H =
       0: GOSUB 1070: GOSUB 10
       50
8D 1250 PRINT "DEALER HAS BLACKJ
       ACK YOU LOSE YOUR": PRIN
       T "INSURANCE MONEY AND S
       AVE YOUR BET."
AD 1260 GOSUB 1120:IN = 1
F1 1270 RETURN
22 1280 PRINT "YOU DO NOT HAVE E
       NOUGH MONEY TO INSURE":
       PRINT "YOUR BET.": RETUR
       N
AF 1290 GOSUB 1050: PRINT "DEALE
       R DID NOT GET BLACKJACK.
        YOU LOSE": PRINT "INSUR
       ANCE MONEY AND HAND CONT
       INUES."
76 1300 GOSUB 1120: RETURN
E7 1310 HTAB 31: VTAB V + 2: PRI
       NT "BLACKJACK":PG(S1) =
       1.5 * AM:BJ = 1: GOTO 80
       0
08 1320 HGR2 : VTAB 15: PRINT "Y
       OU ARE OUT OF MONEY"
FC 1330 PRINT "WISH TO PLAY AGAI
       N (Y/N)?"
6D 1340 GET Y$
8E 1350 IF Y$ = "Y" OR Y$ = "y"
       THEN HGR2 : GOTO 137
5D 1360 IF Y$ = "N" OR Y$ = "n"
       THEN TEXT : HOME : END
82 1370 GOTO 1340
```

Program 4: Apple II Casino Blackjack—ML

```
8000: 20 EC 80 B0 72 20 00 82 FD
8008: B0 6D 20 5B 82 B0 68 AD CB
8010: 26 83 F0 0F A9 7F 8D 2C A6
8018: 83 8D 2D 83 A9 07 8D 2E CE
8020: 83 D0 0F A9 2A 8D 2C 83 F6
8028: A9 55 8D 2D 83 A9 02 8D 2C
8030: 2E 83 A9 03 8D 1A 83 A9 14
8038: 02 8D 22 83 20 09 81 A9 EB
8040: 18 8D 1B 83 20 98 81 20 D2
8048: 5C 81 EE 1E 83 CE 1B 83 A8
8050: D0 F2 AD 26 83 F0 20 A0 4F
8058: 00 AD 27 83 20 88 80 A0 A6
8060: 04 AD 27 83 20 88 80 A0 B0
8068: 08 AD 26 83 20 88 80 A0 9A
8070: 0C AD 26 83 20 88 80 60 64
8078: 72 82 03 08 92 82 0A 08 EE
8080: AA 82 01 01 AA 82 0A 0F 2A
8088: 0A 0A 0A 8D 2B 83 B9 78 78
8090: 80 85 FC C8 B9 78 80 85 95
8098: FD C8 AD 28 83 8D 20 83 19
80A0: B9 78 80 C8 18 6D 29 83 85
80A8: C9 07 90 07 E9 07 EE 20 3C
80B0: 83 8D F5 8D 21 83 B9 78 3A
80B8: 80 18 6D 2A 83 8D 1E 83 62
80C0: A9 01 8D 1A 83 A9 07 8D 88
80C8: 22 83 A9 08 8D 1B 83 AC FD
80D0: 2B 83 B1 FC 8D 2C 83 D0 27
80D8: 09 81 20 98 81 20 5C 81 13
80E0: EE 1E 83 EE 2B 83 CE 1B 60
80E8: 83 D0 E4 60 20 67 82 C9 EF
80F0: 35 90 01 60 A2 00 8E 27 30
80F8: 83 C9 0E 90 07 E9 0D EE E1
8100: 27 83 B0 F5 8D 26 83 18 11
8108: 60 AD 1A 83 8D 1C 83 A8 AE
8110: A9 00 99 2C 83 AD 21 83 76
8118: 18 6D 22 83 C9 07 90 05 8F
8120: E9 07 EE 1C 83 8D 23 83 95
8128: AD 2C 83 09 7F 8D 1D 83 FD
8130: AC 21 83 F0 15 A2 00 0E 92
8138: 2C 83 BD 2C 83 0A 3E 2D 9A
8140: 83 E8 EC 1C 83 D0 F3 88 6E
8148: D0 EB AC 1C 83 B9 2C 83 E4
8150: 09 80 2D 1D 83 99 2C 83 CD
8158: 88 10 F2 AC 23 83 B9 B9
8160: 8A 81 1C 83 88 31 FE FF
8168: 19 2C 83 91 FE 88 30 0A 11
8170: F0 08 B9 2C 83 91 FE 88 D0
8178: D0 F8 AC 21 83 B9 91 81 71
8180: A0 00 31 FE 0D 2C 83 91 9B
```

```
8188: FE 60 7F 7E 7C 78 70 60 02
8190: 40 00 01 03 07 0F 1F 3F F5
8198: AD 1E 83 29 3F A8 B9 C0 CD
81A0: 81 05 AD 1E 85 FF AD 1E 83 51
81A8: 29 08 F0 02 A9 80 18 2C 2C
81B0: 1E 83 70 04 10 04 69 28 7D
81B8: 69 28 6D 20 83 85 FE 68 BA
81C0: 00 04 08 0C 10 14 18 1C A3
81C8: 00 04 08 0C 10 14 18 1C AB
81D0: 01 05 09 0D 11 15 19 1D B3
81D8: 01 05 09 0D 11 15 19 1D BB
81E0: 02 06 0A 0E 12 16 1A 1E C3
81E8: 02 06 0A 0E 12 16 1A 1E CB
81F0: 03 07 0B 0F 13 17 1B 1F DB
81F8: 03 07 0B 0F 13 17 1B 1F DB
8200: A9 00 8D 20 83 8D 21 83 A5
8208: 20 67 82 8D 1F 83 C0 90 4A
8210: 90 F0 01 60 C9 18 90 FA
8218: 01 60 A9 24 8D 20 83 A9 CA
8220: 04 8D 21 83 A9 00 8D 25 74
8228: 83 A9 6D 8D 24 83 AD 1F F7
8230: 83 CD 24 83 90 04 ED 24 BB
8238: 83 38 2E 25 83 4E 24 83 46
8240: 90 EF 18 6D 21 83 8D 21 B6
8248: 83 8D 29 83 18 AD 25 83 15
8250: 6D 20 83 8D 20 83 8D 28 AF
8258: 83 18 60 20 67 82 8D 1E B1
8260: 83 8D 2A 83 C0 C0 60 20 3A
8268: B1 00 20 05 E1 A5 A1 A4 28
8270: A0 60 7F 7F 5F 07 01 01 DF
8278: 0F 7F 7F 7F 77 55 55 57 DF
8280: 5F 7F 7F 7F 47 47 01 01 57
8288: 47 7F 7F 7F 5F 57 57 57 53
8290: 5F 7F 7F 7F 7F 7E 78 78 6C
8298: 7F 7F 7F 7F 7E 7A 7A 7E 76
82A0: 7F 7F 7F 7F 7F 7E 7A 7A 7E A2
82A8: 7F 7F 7F 7F 7F 7E 7A 7E 9E
82B0: 7F 7F 63 1C 1C 1C 00 1C F0
82B8: 1C 1C 63 1C 1F 63 7F B9
82C0: 7C 00 63 1C 1F 63 1F 1F 16
82C8: 1C 63 4F 47 43 49 4C 00 EA
82D0: 4F 4F 00 7C 7C 60 1F 1F DB
82D8: 1C 63 43 79 7C 60 1C 1C 7E
82E0: 1C 63 00 1C 1F 4F 67 73 07
82E8: 73 73 63 1C 1C 63 1C 1C 75
82F0: 1C 63 1C 1C 1C 03 1F 81
82F8: 1C 63 01 67 67 67 67 67 8A
8300: 67 67 1F 1F 1F 1F 1F 1F 3D
8308: 1C 63 63 1C 1C 1C 1C 10 BD
8310: 44 13 1C 1C 60 1C 1F F9
8318: 1C 1C 01 00 02 7F 1F 85 26
8320: 14 03 07 03 03 13 04 02 71
8328: 13 00 08 28 78 04 00 00 10
8330: 00 00 00 00 00 00 00 00 00
8338: 00 00 00 00 00 00 00 00 3F
8340: D8 78 85 45 86 46 84 47 74
8348: A6 07 0A 0A 8A 07 0A 14 E0
8350: 30 04 10 01 E8 E8 0A 86 08
8358: 1B 18 65 06 85 1A 90 02 B7
8360: E6 1B A5 28 85 08 A5 29 99
8368: 29 03 05 E6 85 09 A2 6B 15
8370: A0 00 B1 1A 24 32 30 02 EB
8378: 49 7F A4 24 91 08 E6 1A 6F
8380: D0 02 E6 1B A5 09 18 69 E9
8388: 04 05 09 CA D0 E2 A5 45 63
8390: A6 46 A4 47 58 4C F0 FD 59
8398: 00 00 00 00 00 00 00 00 9F
83A0: 00 00 00 00 00 00 00 00 A7
83A8: 00 00 00 00 00 00 00 00 AF
83B0: 08 08 08 08 08 00 08 00 8F
83B8: 14 14 14 00 00 00 00 00 51
83C0: 14 14 3E 14 3E 14 14 00 4A
83C8: 08 3C 0A 1C 28 1E 08 00 AF
83D0: 06 26 10 08 04 32 30 00 30
83D8: 04 0A 0A 04 2A 12 2C 00 D7
83E0: 08 08 08 00 00 00 00 00 EE
83E8: 08 04 02 02 02 04 08 00 85
83F0: 08 10 20 20 20 10 08 00 57
83F8: 08 2A 1C 08 1C 2A 08 00 2C
8400: 00 08 08 3E 08 08 00 00 50
8408: 00 00 00 00 08 08 04 00 79
8410: 00 00 00 3E 00 00 00 00 FC
8418: 00 00 00 00 00 08 08 00 31
8420: 00 20 10 08 04 02 00 00 DB
8428: 1C 22 32 2A 26 22 1C 00 A2
8430: 08 0C 08 08 08 08 1C 00 5A
8438: 1C 22 20 18 04 02 3E 00 2C
8440: 3E 20 10 18 20 22 1C 00 B5
```

```
8448: 10 18 14 12 3E 10 10 00 55
8450: 3E 02 1E 20 20 22 1C 00 80
8458: 38 04 02 1E 22 22 1C 00 72
8460: 3E 20 10 08 04 04 04 00 4B
8468: 1C 22 22 1C 22 22 1C 00 DF
8470: 1C 22 22 3C 20 10 0E 00 75
8478: 00 00 08 00 00 08 00 00 C2
8480: 00 00 08 00 00 08 08 04 F2
8488: 10 08 04 02 04 08 10 00 9C
8490: 00 00 3E 00 3E 00 00 00 53
8498: 04 08 10 20 10 08 04 00 52
84A0: 1C 22 10 08 08 00 08 00 13
84A8: 1C 22 2A 3A 1A 02 3C 00 82
84B0: 08 14 22 22 3E 22 22 00 E7
84B8: 1E 22 22 1E 22 22 1E 00 55
84C0: 1C 22 02 02 02 22 1C 00 91
84C8: 1E 22 22 22 22 22 1E 00 A5
84D0: 3E 02 02 1E 02 02 3E 00 30
84D8: 3E 02 02 1E 02 02 02 00 BF
84E0: 3C 02 02 02 32 22 3C 00 7B
84E8: 22 22 22 3E 22 22 22 00 91
84F0: 1C 08 08 08 08 08 1C 00 24
84F8: 20 20 20 20 20 22 1C 00 E1
8500: 22 12 0A 06 0A 12 22 00 1F
8508: 02 02 02 02 02 02 3E 00 89
8510: 22 36 2A 22 22 22 22 00 7F
8518: 22 22 26 2A 32 22 22 00 82
8520: 1C 22 22 22 22 22 1C 00 F9
8528: 1E 22 22 1E 02 02 02 00 0D
8530: 1C 22 22 22 2A 12 2C 00 2A
8538: 1E 22 22 1E 0A 12 22 00 DD
8540: 1C 22 02 1C 20 22 1C 00 A5
8548: 3E 08 08 08 08 08 08 00 66
8550: 22 22 22 22 22 22 1C 00 2D
8558: 22 22 22 22 22 14 08 00 D4
8560: 22 22 22 2A 2A 36 22 00 5A
8568: 22 22 14 08 14 22 22 00 7D
8570: 22 22 14 08 08 08 08 00 88
8578: 3E 20 10 08 04 02 3E 00 D1
8580: 3E 06 06 06 06 06 3E 00 12
8588: 00 02 04 08 10 20 00 00 16
8590: 3E 30 30 30 30 30 3E 00 8E
8598: 00 00 08 14 22 00 00 00 F6
85A0: 00 00 00 00 00 00 7F 2B
85A8: 04 08 10 00 00 00 00 00 B9
85B0: 00 00 1C 20 3C 22 3C 00 24
85B8: 02 02 1E 22 22 22 1E 00 01
85C0: 00 00 3C 02 02 02 3C 00 04
85C8: 20 20 3C 22 22 22 3C 00 A7
85D0: 00 00 3C 02 3E 02 3C 00 F3
85D8: 18 24 04 1E 04 04 04 00 93
85E0: 00 00 1C 22 22 3C 20 1C EF
85E8: 02 02 1E 22 22 22 22 00 39
85F0: 08 00 0C 08 08 08 1C 00 9A
85F8: 10 00 18 10 10 10 12 0C 01
8600: 02 02 22 12 0E 12 22 00 F0
8608: 0C 08 08 08 08 08 1C 00 37
8610: 00 00 36 2A 2A 2A 22 00 C4
8618: 00 00 1E 22 22 22 22 00 E8
8620: 00 00 1C 22 22 22 1C 00 A4
8628: 00 00 1E 22 22 1E 02 02 AA
8630: 00 00 3C 22 22 3C 20 20 49
8638: 00 00 3A 06 02 02 02 00 09
8640: 00 00 3C 02 1C 20 1E 00 92
8648: 04 04 1E 04 04 24 18 00 3D
8650: 00 00 22 22 22 32 2C 00 F5
8658: 00 00 22 22 22 14 08 00 3D
8660: 00 00 22 22 2A 2A 36 00 3A
8668: 00 00 22 14 08 14 22 00 CF
8670: 00 00 22 22 22 3C 20 1C 42
8678: 00 00 3E 10 08 04 3E 00 1B
8680: 38 0C 0C 06 0C 0C 38 00 8F
8688: 08 08 08 08 08 08 08 00 95
8690: 0E 18 18 30 18 18 0E 00 ED
8698: 2C 1A 00 00 00 00 00 00 42
86A0: 00 2A 14 2A 14 2A 00 00 A6
```

Program 5: Atari 400/800/XL/XE Casino Blackjack

Version by William Chin, Editorial Programmer

```
AD 10 REM COPYRIGHT 1988 COM
       PUTE! PUBLICATIONS, IN
       C. - ALL RIGHTS RESERV
       ED
CN 20 OPEN #1,4,0,"K:":POKE
       752,1:POKE 82,1
EL 30 PRINT CHR$(125):PRINT
       "   COPYRIGHT 1988 COMP
       UTE! PUBL., INC."
KN 40 POSITION 10,2:PRINT "A
       LL RIGHTS RESERVED"
AC 50 POSITION 15,12:? "BLAC
       KJACK"
KN 60 GOSUB 1150
IL 70 DIM CH(13,4),A$(1),Y$(
       1),OP$(1),AM$(10),C(25
       5),PL(4),PG(4),J(4),R$
       (13),S$(4)
OA 80 C(72)=1:C(83)=2:C(80)=
       3:C(69)=4:C(68)=5:C(89
       )=6:C(78)=7
HL 90 C(81)=8:C(82)=8:C(126)
       =9:C(30)=9:C(46)=10:FO
       R I=48 TO 57:C(I)=10:N
       EXT I
LL 100 FOR I=65 TO 90:C(I+32
        )=C(I):NEXT I:REM LOW
        ER CASE
CC 110 R$="23456789TJQKA":S$
        (1)=CHR$(251):S$(2)=C
        HR$(128):S$(3)=CHR$(1
        44):S$(4)=CHR$(224)
LL 120 FOR I=1 TO 13:R$(I,I)
        =CHR$(ASC(R$(I,I))+12
        8):NEXT I
EP 130 PRINT CHR$(125):V=0:H
        =0:TY=4
CB 140 DO=100:HA=0
BN 150 IF DO<1 THEN 1350
EC 160 HA=HA+1
AE 170 IF PF=1 THEN GOSUB 11
        50:? CHR$(125)
JE 180 AM$="":POSITION 1,11:
        PRINT "YOU WILL START
        HAND ";HA;"."
MP 190 PRINT "Q - TO QUIT"
NJ 200 PRINT "R - TO RESTART
        GAME":PRINT DO;" DOL
        LARS LEFT"
EL 210 PRINT "PLACE BET
        {14 SPACES}":POSITION
        15,15
AA 220 GET #1,A:IF C(A)=8 TH
        EN AM$=CHR$(A):GOTO 3
        10
AD 230 IF A=155 THEN 340
HF 240 IF C(A)<>9 THEN 270
OG 250 IF LEN(AM$)<2 THEN AM
        $="":GOTO 290
DF 260 AM$=AM$(1,LEN(AM$)-1)
        :GOTO 290
JL 270 IF C(A)<>10 THEN 220
MA 280 IF LEN(AM$)<9 THEN AM
        $(LEN(AM$)+1)=CHR$(A)
MO 290 POSITION 13,15:? "
        {9 SPACES}";:POSITION
        13,15:? AM$
GA 300 GOTO 220
BA 310 IF AM$="Q" THEN POKE
        752,0:END
MN 320 IF AM$="R" THEN RUN
GG 330 GOTO 340
HG 340 AM=VAL(AM$):AM=INT(10
        0*(AM+5.0E-03))/100
KL 350 IF AM<0.01 THEN GOSUB
        1080:PRINT "YOU CANN
        OT BET ZERO DOLLARS":
        GOSUB 1150:GOTO 170
CB 360 IF AM>DO THEN GOSUB 1
        080:PRINT "YOU DO NOT
        HAVE THAT MUCH MONEY
        ":GOSUB 1150:GOTO 170
GI 370 Z2=0:Z1=0:DE=0:D1=0:D
        2=0:P=0:D3=0:CL=0:JK=
        0:HI=0:BU=0:SD=0:DD=0
        :SP=0:QQ=0:PF=0
KE 380 S1=0:QP=0:SP=0:B1=0:O
        1=0:O2=0:PL(0)=0:PL(1
        )=0:PL(2)=0:PL(3)=0:P
        L(4)=0:SE=0
```

```
OC 390 PG(0)=AM:D=0:PR=0:O1=
       DO-AM:O2=O1:PRINT CHR
       $(125):V=1:H=2:Z2=DE:
       Z1=D
CO 400 GOSUB 990:DE=Z2:D=Z1:
       D1=A:D2=A1:V=7:H=2:GO
       SUB 1060:P1=A:P2=A1
LE 410 H=6:V=1:GOSUB 1070:D3
       =A:H=6:V=7:GOSUB 1060
       :P3=A
MM 420 FOR I=1 TO 3:POSITION
       2,I:? CHR$(160);CHR$
       (160);CHR$(160):NEXT
       I
HO 430 IF P1=12 AND P3<12 AN
       D P3>7 OR P3=12 AND P
       1<12 AND P1>7 THEN 13
       40
ON 440 GOSUB 1190:IF IN>0 TH
       EN IN=0:GOTO 150
HO 450 GOSUB 1080:PRINT " H
       HIT  S STAND  P SPLIT
       ":? "D DOUBLE  E SURR
       ENDER"
NI 460 GET #1,OP:JK=C(OP)
JO 470 IF JK=0 OR JK>5 THEN
       460
LE 480 ON JK GOTO 490,600,61
       0,700,750
PD 490 HI=1:GOSUB 580
HH 500 IF DD=1 THEN H=10:GOS
       UB 1060:DD=2:SD=1:GOS
       UB 580
BH 510 IF SD=1 THEN SD=0:H=6
       :V=V+3:HI=0:GOTO 540
CC 520 IF B1=1 THEN B1=0:GOT
       O 450
LP 530 H=H+4:GOSUB 1060:GOSU
       B 580:B1=1:GOTO 500
EA 540 IF SP>0 THEN SP=SP-1:
       S1=S1+1:QP=QP-1:P=0:G
       OTO 560
GO 550 GOTO 830
GK 560 IF J(S1)=1 THEN J(S1)
       =0:P=1
AG 570 GOSUB 1060:B1=1:GOTO
       500
ON 580 IF BU=1 THEN POSITION
       34,V:PRINT "BUST":SD
       =1:DD=2:BU=0:HI=0
HO 590 RETURN
MA 600 POSITION 33,V:PRINT "
       STAND":SD=1:GOTO 510
IC 610 IF HI=1 THEN GOSUB 10
       80:GOSUB 680:GOSUB 11
       50:GOTO 450
EH 620 IF A<>P1 THEN GOSUB 1
       080:PRINT "CARDS MUST
       BE SAME DENOMINATION
       ":? "TO SPLIT":GOSUB
       1150:GOTO 450
CN 630 IF O1-AM<0 THEN GOSUB
       1080:GOSUB 690:GOSUB
       1150:GOTO 450
DE 640 O1=O1-AM:SP=SP+1:QQ=Q
       Q+1:QP=QP+1:PL(S1+QP)
       =B:PG(S1+QP)=AM:PL(S1
       )=B
BD 650 QV=V:PF=1
GJ 660 IF P>0 THEN P=1:J(S1+
       QP)=1
PE 670 H=2:V=7+QQ*3:GOSUB 11
       00:V=QV:H=6:GOSUB 106
       0:GOTO 450
IA 680 PRINT "YOU MAY SPLIT
       WHEN FIRST TWO CARDS"
       :? "ARE THE SAME AND
       YOU HAVE NOT HIT":RET
       URN
BN 690 PRINT "NOT ENOUGH MON
       EY FOR THE SPLIT":RET
       URN
HO 700 IF HI=1 THEN GOSUB 10
       80:GOSUB 730:GOSUB 11
       50:GOTO 450
HH 710 IF DD=1 THEN GOSUB 10
       80:GOSUB 740:GOSUB 11
       50:GOTO 450
CJ 720 PG(S1)=AM/2:PL(S1)=0:
       SD=1:SE=1:POSITION 30
       ,V:PRINT "SURRENDER":
       GOTO 500
AC 730 PRINT "YOU CANNOT SUR
       RENDER SINCE YOU ALRE
       ADY":? "HIT ON THIS H
       AND":RETURN
DJ 740 PRINT "YOU CANNOT SUR
       RENDER SINCE YOU DOUB
       LED":? "DOWN ON THIS
       HAND"
IC 750 IF HI=1 THEN GOSUB 10
       80:GOSUB 810:GOSUB 11
       50:GOTO 450
AC 760 IF P>0 THEN AZ=PL(S1)
       -10:GOTO 780
OH 770 AZ=PL(S1)
CO 780 IF O1-AM<0 THEN GOSUB
       1080:GOSUB 820:GOSUB
       1150:GOTO 450
MJ 790 IF P>0 THEN PL(S1)=PL
       (S1)-10
GI 800 PG(S1)=PG(S1)+AM:POSI
       TION 27,V:PRINT "DOUB
       LE DOWN":DD=1:O1=O1-A
       M:GOTO 500
PH 810 PRINT "SORRY YOU ALRE
       ADY HIT ON THIS HAND"
       :RETURN
FK 820 PRINT "NOT ENOUGH MON
       EY TO DOUBLE DOWN":RE
       TURN
EE 830 A=D1:A1=D2:V=1:H=2:GO
       SUB 1100:H=6:IF BJ=1
       THEN BJ=0:GOTO 900
PC 840 IF DE=21 THEN FOR I=0
       TO 3:PL(SG)=0:NEXT I
       :GOTO 900
HG 850 IF S1<1 AND PL(0)>21
       THEN 900
DH 860 IF S1<1 AND SE=1 THEN
       SE=0:GOTO 900
CM 870 IF DE>16 THEN 890
GA 880 H=H+4:GOSUB 1070:GOTO
       870
IM 890 IF DE>21 THEN POSITIO
       N 34,V+1:PRINT "BUST"
GB 900 V=7:FOR I=0 TO S1
IN 910 IF DE>21 AND PL(I)<22
       OR DE<22 AND DE<PL(I
       ) AND PL(I)<22 THEN G
       OSUB 980
FM 920 IF DE<22 AND DE>PL(I)
       OR DE<22 AND PL(I)>2
       1 OR DE>21 AND PL(I)>
       22 THEN GOSUB 970
CP 930 IF DE<22 AND DE=PL(I)
       THEN POSITION 34,V+1
       :PRINT "PUSH":PG(I)=0
NH 940 DO=DO+PG(I):V=V+3:NEX
       T I:DO=INT(100*(DO+5.
       0E-03))/100
IH 950 GOSUB 1080:FOR I=1 TO
       13:FOR J=1 TO 4:CH(I
       ,J)=0:NEXT J:NEXT I:P
       R=1
GO 960 GOTO 150
LI 970 POSITION 34,V+1:PRINT
       "LOSE":PG(I)=-PG(I):
       RETURN
GO 980 POSITION 34,V+1:PRINT
       " WIN":RETURN
IB 990 T=FRE(0):A=INT(13*RND
       (1)):A1=INT(4*RND(1)+
       1):IF CH(A,A1)=1 THEN
       990
CO 1000 CH(A,A1)=1:B=A+2:IF
       A+2>9 THEN B=10
KB 1010 IF A+2=14 THEN B=11
BB 1020 IF B=11 THEN Z1=Z1+1
EF 1030 Z2=Z2+B:IF Z2>21 AND
       Z1>0 THEN Z2=Z2-10:
       Z1=Z1-1
LF 1040 IF Z2>21 THEN BU=1
KG 1050 RETURN
AL 1060 Z2=PL(S1):Z1=P:GOSUB
       990:PL(S1)=Z2:P=Z1:
       GOSUB 1100:RETURN
CE 1070 Z2=DE:Z1=D:GOSUB 990
       :DE=Z2:D=Z1:GOSUB 11
       00:RETURN
GG 1080 CL=19:FOR ZZ=CL TO 2
       2:POSITION 0,ZZ
HM 1090 PRINT "{39 SPACES}":N
       EXT ZZ:POSITION 0,CL
       :RETURN
LH 1100 POSITION H,V:? "  ":
       POSITION H,V+1:? "
       ":POSITION H,V:? R$(
       A+1,A+1);CHR$(160);C
       HR$(160)
PH 1110 POSITION H,V+1:? CHR
       $(160);S$(A1,A1);CHR
       $(160)
EO 1120 POSITION H,V+2:? CHR
       $(160);CHR$(160);R$(
       A+1,A+1)
AF 1130 REM PLACE CARD IN SP
       OT
KG 1140 RETURN
NE 1150 POSITION 13,22:PRINT
       "PRESS ANY KEY";
GA 1160 POKE 764,255
JL 1170 A=RND(1):IF PEEK(764
       )<>255 THEN 1080
MM 1180 GOTO 1170
LN 1190 IF D3<>12 THEN RETUR
       N
KA 1200 GOSUB 1080:PRINT "WA
       NT INSURANCE Y N"
HL 1210 GET #1,Y:YA=C(Y)
FI 1220 IF YA=6 THEN 1250
HC 1230 IF YA=7 THEN RETURN
NE 1240 GOTO 1210
PC 1250 IF O2-AM/2<0 THEN GO
       SUB 1080:GOSUB 1310:
       GOSUB 1150:RETURN
MA 1260 IN=AM/2:O2=O2-AM/2:D
       O=DO-IN:IN=0
KJ 1270 IF DE<>21 THEN GOSUB
       1320:IN=0:GOTO 1300
AL 1280 A=D1:A1=D2:V=1:H=2:G
       OSUB 1100:GOSUB 1080
ND 1290 PRINT "DEALER HAS BL
       ACKJACK YOU LOSE YOU
       R":? "INSURANCE MONE
       Y AND SAVE BET.":GOS
       UB 1150:IN=1
KE 1300 RETURN
DB 1310 PRINT "YOU DO NOT HA
       VE ENOUGH MONEY TO I
       NSURE":? "YOUR BET."
       :RETURN
EK 1320 GOSUB 1080:PRINT "DE
       ALER DID NOT GET BLA
       CKJACK YOU LOSE";
AJ 1330 PRINT "INSURANCE MON
       EY AND HAND CONTINUE
       S":GOSUB 1150:RETURN
BE 1340 POSITION 30,V:PRINT
       "BLACKJACK":PG(S1)=1
       .5*AM:BJ=1:GOTO 830
PG 1350 POSITION 1,15:PRINT
       "YOU ARE OUT OF MONE
       Y"
HF 1360 PRINT "WISH TO PLAY
       AGAIN (Y/N)?"
IC 1370 GET #1,Y:YA=C(Y)
CL 1380 IF YA=6 THEN 130
ND 1390 IF YA=7 THEN POKE 75
       2,0:END
MJ 1400 GOTO 1370
```

www.commodore.ca

Program 6: PC/PCjr And Compatible Casino Blackjack

Version by George Miller, Editorial Programmer

```
HE 10  ' COPYRIGHT (C) 1988 COMPU
       TE! PUBLICATIONS, INC.
FC 20  '                 ALL RIGHTS R
       ESERVED
OK 30  KEY OFF:DEF SEG=0:DEFINT A
       -Z:DEFDBL B:POKE 1047,PEEK
       (1047) OR 64:RANDOMIZE TIM
       ER:SP$=SPACE$(40)
DK 40  DIM CD$(52),V$(13),M(52)
IA 50  SCREEN 0,1:WIDTH 40:COLOR
       2,2:CLS:GOSUB 390:CLS:HA=1
NO 60  SP=0:DT1=0:DT2=0:PT1=0:PT2
       =0:COLOR 15,10:CLS:IF BA<>
       0 THEN GOTO 80 ELSE COLOR
       15,2:CLS:LOCATE 10,10:PRIN
       T"Sorry, you are broke.":L
       OCATE 12,12:PRINT"Play aga
       in? (Y/N)":GOSUB 950
QN 70  IF A$="Y" THEN GOTO 50 ELS
       E IF A$="N" GOTO 960 ELSE
       GOTO 70
EA 80  H=0:GOSUB 230:COLOR 0,2:LO
       CATE 3,13:PRINT"Playing Ha
       nd #";:PRINT HA
AC 90  LOCATE 5,5:COLOR 15:PRINT"
       Cash remaining ";:PRINT US
       ING "$$##########.##";BA
LM 100 LOCATE 10,14:COLOR 4,7:PR
       INT" Q ":LOCATE 12,14:PRI
       NT" R ":COLOR 15,2:LOCATE
       10,18:PRINT"Quit":LOCATE
       12,18:PRINT"Restart game
       ":LOCATE 15,10:INPUT"Your
       bet, please";BET$
GB 110 IF BET$="R" THEN GOTO 50
       ELSE IF BET$="Q" GOTO 960
FC 120 BET=VAL(BET$):IF BET=0 TH
       EN COLOR 4:LOCATE 17,11:P
       RINT"Please bet again.":G
       OSUB 420:GOTO 60
PM 130 IF BET>BA THEN LOCATE 17,
       7:COLOR 4:PRINT"You only
       have $";BA;" remaining.":
       LOCATE 18,12:PRINT"Please
        bet again.":GOSUB 420:GO
       TO 60
CD 140 GOSUB 360:CLS:LOCATE 25,1
       :COLOR 15,0:PRINT" it  t
       and S lit  ouble down  Su
       rr nder ";
JE 150 COLOR 4,0:LOCATE 25,2:PRI
       NT"H";:LOCATE 25,6:PRINT
       "S";:LOCATE 25,13:PRINT"p"
       ;:LOCATE 25,18:PRINT"D";:
       LOCATE 25,35:PRINT"e";:GO
       SUB 230:GOSUB 430
CA 160 IF H=1 THEN GOSUB 420:GOT
       O 60
OA 170 GOSUB 950:H=0:IF A$="H" T
       HEN GOSUB 560
YN 180 IF A$="P" THEN GOSUB 680
KO 190 IF A$="S" THEN GOSUB 590:
       GOSUB 240:GOTO 60
JC 200 IF A$="D" THEN GOSUB 830:
       GOTO 180
KP 210 IF A$="E" THEN BA=BA-BET*
       .5:GOTO 60
DA 220 IF H=1 THEN GOSUB 420:GOT
       O 60 ELSE GOTO 170
CJ 230 LOCATE 1,12:COLOR 14,4:PR
       INT" CASINO BLACKJACK ":R
       ETURN
MO 240 IF SP=1 THEN GOSUB 330
CE 250 IF DT1<=21 THEN DT=DT1 EL
       SE DT=DT2
NE 260 IF DT2>DT1 AND DT2<=21 TH
       EN DT=DT2
OA 270 IF PT1<=21 THEN PT=PT1 EL
       SE PT=PT2
PE 280 IF PT2>PT1 AND PT2<=21 TH
       EN PT=PT2
HH 290 COLOR 15,10:IF DT=PT THEN
        LOCATE 12,10:PRINT"Push,
        No winner."
LM 300 IF DT>PT AND DT<=21 THEN
       GOSUB 920:BA=BA-BET
DN 310 IF PT>DT AND PT<=21 THEN
       GOSUB 930:BA=BA+BET
HN 320 GOSUB 420:RETURN
PI 330 IF VT1<21 AND VT1>PT1 THE
       N PT1=VT1
HK 340 IF VT2<21 AND VT2>PT2 THE
       N PT2=VT2
MH 350 RETURN
MJ 360 COLOR 15,2:CLS:LOCATE 15,
       13:PRINT"Shuffling Deck..
       .":FOR I=1 TO 52:M(I)=I:N
       EXT:FOR I=1 TO 52:X=INT(5
       2*RND)+1:TP=M(I):M(I)=M(X
       ):M(X)=TP:NEXT:HA=HA+1:RE
       TURN
PO 370 IF TP$="   " THEN COLOR 0
       ,4 ELSE IF ASC(MID$(TP$,2
       ,1))<=4 THEN COLOR 4,7 EL
       SE COLOR 0,7
FK 380 LOCATE XL,YL:PRINT TP$:FO
       R Y=1 TO 2:LOCATE XL+Y,YL
       :PRINT"   ":NEXT:LOCATE X
       L+3,YL:PRINT "  ":LOCATE X
       L+3,YL+1:PRINT MID$(TP$,2
       ,1):LOCATE XL+3,YL+2:PRIN
       T LEFT$(TP$,1):RETURN
EL 390 RESTORE 970:FOR X=0 TO 3:
       FOR T=1 TO 13:READ CD$(X*
       13+T):CD$(X*13+T)=CD$(X*1
       3+T)+CHR$(X+3)+CHR$(32):N
       EXT:RESTORE 970:NEXT:REST
       ORE 970:FOR T=13 TO 1 STE
       P-1:READ V$(T):NEXT
FD 400 XL=5:YL=16:TP$=CD$(1):GOS
       UB 370:XL=5:YL=20:TP$=CD$
       (43):GOSUB 370
AE 410 COLOR 15,2:LOCATE 11,12:P
       RINT"Casino Blackjack":CO
       LOR 1,2:LOCATE 15,4:PRINT
       "Copyright 1988 COMPUTE!
       Pub., Inc.":LOCATE 16,11:
       PRINT"All Rights Reserved
       ":HA=0:PX(0)=18:PY(0)=0:P
       X(1)=5:PY(1)=-2:FOR X=0 T
       O 1:GOSUB 420:NEXT:BA=100
       :RETURN
NN 420 FOR T=1 TO 6000:NEXT:RETU
       RN
BC 430 P=0:D=0:FOR C=1 TO 4:IF C
       =2 THEN TP$="   " ELSE TP
       $=CD$(M(C))
KC 440 IF INT(C/2)<>C/2 THEN XL=
       PX(0):YL=PY(0)+2*C:PL$(P)
       =CD$(M(C)):P=P+1 ELSE XL=
       PX(1):YL=PY(1)+2*C:DL$(D)
       =CD$(M(C)):D=D+1
IJ 450 GOSUB 370:NEXT:GOSUB 860
QO 460 IF LEFT$(DL$(1),1)="A" TH
       EN LOCATE 12,10:COLOR 15,
       2:PRINT"Purchase insuranc
       e? (Y/N)" ELSE GOTO 530
NN 470 GOSUB 950
GG 480 LOCATE 12,1:PRINT SP$:IF
       A$="Y" GOTO 490 ELSE IF A
       $<>"N" GOTO 460 ELSE GOTO
        530
QO 490 IF BA-BET-BET*.5<BA THEN
       INS=1:BA=BA-BET*.5 ELSE L
       OCATE 12,1:PRINT"You can'
       t afford insurance.":GOSU
       B 420:LOCATE 12,1:PRINT S
       P$
NH 500 GOSUB 660:IF DT1=21 THEN
       GOSUB 620:GOSUB 620:GOSUB
        420:IF INS<>1 THEN BA=BA
       -BET:GOTO 60 ELSE GOTO 60
PF 510 LOCATE 12,1:PRINT"Dealer
       does not have Blackjack."
       :PRINT"Hand continues."
MK 520 GOSUB 420:LOCATE 12,1:PRI
       NT SP$:LOCATE 13,1:PRINT
       SP$
PJ 530 IF PT1=21 AND P=2 THEN LO
       CATE 19,10:COLOR 31,2:PRI
       NT"BLACKJACK!":GOSUB 670:
       GOSUB 660:IF DT1<>21 THEN
       BA=BET*1.5+BA:H=1:RETURN
CO 540 IF DT1=21 AND PT1=21 THEN
        LOCATE 6,10:COLOR 31,2:P
       RINT"BLACKJACK!":COLOR 15
       ,10:LOCATE 12,10:PRINT"Pu
       sh, no winner":H=1:RETURN
NJ 550 RETURN
EL 560 C=C+1:PL$(P)=CD$(M(C)):TP
       $=PL$(P):P=P+1:XL=PX(0):Y
       L=(PY(0)+4*(P))-2
NB 570 GOSUB 370:GOSUB 860:IF PT
       1>21 AND PT2>21 THEN LOCA
       TE PX(0)+1,(PY(0)+4*(P))+
       4:COLOR 15,2:PRINT"BUSTED
       !":IF SP=0 THEN GOSUB 920
       :BA=BA-BET:GOSUB 670:GOSU
       B 420:H=1 ELSE SP=0:RETUR
       N
NP 580 RETURN
NJ 590 GOSUB 670
DL 600 COLOR 15,2:LOCATE 19,(PY(
       0)+4*(P))+4:PRINT"Stand"
KJ 610 GOSUB 660:IF DT1=>17 THEN
        RETURN
AG 620 IF DT1=21 AND D=2 THEN LO
       CATE 6,10:COLOR 31,2:PRIN
       T"BLACKJACK!":H=1:GOSUB 4
       20:RETURN
ID 630 C=C+1:DL$(D)=CD$(M(C)):TP
       $=DL$(D):D=D+1:C=C+1:XL=P
       X(1):YL=(PY(1)+4*(D))
DO 640 GOSUB 370:GOSUB 660:IF DT
       1>21 AND DT2>21 THEN LOCA
       TE PX(1)+1,(PY(1)+4*(D))+
       4:COLOR 15,2:PRINT"BUSTED
       !":GOSUB 930:BA=BA+BET:GO
       SUB 420:H=1:RETURN
FI 650 GOTO 610
HM 660 DT1=0:DT2=0:V1=0:V2=0:FOR
        X=0 TO D-1:G$=DL$(X):GOS
       UB 870:DT1=DT1+V1:DT2=DT2
       +V2:NEXT:RETURN
JI 670 XL=5:YL=2:TP$=DL$(0):GOSU
       B 370:RETURN
PH 680 IF LEFT$(PL$(0),1)<>LEFT$
       (PL$(1),1) THEN RETURN
BH 690 SP=1:XL=PX(0):YL=PY(0)+2*
       (C-2):TP$="   ":COLOR 15,
       2:GOSUB 380
NL 700 PX(0)=PX(0)-5:XL=PX(0):YL
       =PY(0)+2*(C-4):TP$=PL$(1)
       :GOSUB 370:T=P
LP 710 TEMP=PX(0):PL$(0)=PL$(0)
       :PL$(1)=CD$(M(C)):C=C+1:
       TP$=PL$(1):YL=PY(0)+3*(P)
       :GOSUB 370:GOSUB 860
EC 720 GOSUB 790:IF PT1>21 AND P
       T2>21 THEN GOSUB 800:GOTO
        740
LJ 730 IF A$="S" THEN VT1=PT1:VT
       2=PT2:XL=14:GOSUB 810:GOT
       O 740 ELSE GOTO 720
BK 740 PX(0)=PX(0)+5:PL$(0)=TEMP
       :C=C+1:PL$(1)=CD$(M(C)):
       P=T:TP$=PL$(1):XL=PX(0):Y
       L=PY(0)+3*(P):GOSUB 370:G
       OSUB 860
DM 750 GOSUB 790:IF PT1>21 AND P
       T2>21 THEN GOSUB 800:GOTO
        780
EI 760 IF A$="S" THEN XL=19:GOSU
       B 810:GOTO 780
ID 770 GOTO 750
QK 780 GOSUB 670:GOSUB 610:GOSUB
        240:H=1:RETURN
```

```
JM 790 GOSUB 950:IF A$="H" THEN
       GOSUB 820:RETURN ELSE RET
       URN
CE 800 LOCATE PX(0)+1,(PY(0)+4*(
       P))+4:COLOR 15,2:PRINT"Bu
       sted!":RETURN
CG 810 LOCATE XL,(PY(0)+4*(P))+4
       ::COLOR 15,2:PRINT"Stand"
       :RETURN
IE 820 C=C+1:PL$(P)=CD$(M(C)):TP
       $=PL$(P):P=P+1:XL=PX(0):Y
       L=(PY(0)+4*(P))-2:PT1=0:P
       T2=0:GOSUB 370:GOSUB 860:
       RETURN
DA 830 GOSUB 860:IF PT1<10 OR PT
       1>11 THEN A$="":RETURN

MC 840 LOCATE 12,10:COLOR 31,2:P
       RINT"Double Down!":BET=BE
       T*2:GOSUB 560:A$="S":RETU
       RN
NM 850 RETURN
BD 860 PT2=0:PT1=0:FOR X=0 TO P-
       1:G$=PL$(X):GOSUB 870:PT1
       =PT1+V1:PT2=PT2+V2:NEXT:R
       ETURN
EP 870 V1=0:V2=0:FOR Y=1 TO 13:I
       F LEFT$(G$,1)=V$(Y) THEN
       V=Y+1
KP 880 NEXT:IF V=14 THEN V1=11:V
       2=1:GOTO 910
KC 890 IF V>10 THEN V1=10:V2=10:
       GOTO 910

HA 900 V1=V1+V:V2=V2+V
MF 910 RETURN
CH 920 LOCATE 6,(PY(1)+4*(D))+4:
       COLOR 31,2:PRINT"House Wi
       ns!":RETURN
PA 930 IF SP=1 THEN XL=17:YL=10
       ELSE XL=19:YL=PY(0)+4*(P)
       +4
CE 940 COLOR 31,2:LOCATE XL,YL:P
       RINT"Winner!":RETURN
IE 950 A$="":A$=INKEY$:IF A$=""
       THEN 950 ELSE RETURN
MD 960 SCREEN 0,0,0:WIDTH 80:CLS
       :END
EJ 970 DATA A,K,Q,J,T,9,8,7,6,5,
       4,3,2
```

Program 7: Amiga Casino Blackjack

Version by George Miller, Editorial Programmer

```
Blackjack:◄
'Copyright 1988 COMPUTE! Publications, Inc.◄
'All Rights Reserved◄
DEFINT a-z:DEFSNG r,g,b,cy:RANDOMIZE TIMER◄
DIM cd$(52),v$(13),m(52),sb(36,3),card(507,4)◄
sp$=SPACE$(40)◄
SCREEN 1,320,200,4,1:WINDOW 3,"",(0,0)-(311,186)
,16,1:WINDOW OUTPUT 3:COLOR 3,0◄
Start:s=1:GOSUB DoScreen:GOSUB InitiaLize:GOSUB
DispLay◄
Main:IF s<>1 THEN GOSUB HoLdscrn◄
COLOR ,10:CLS:GOSUB TitLe:h=0:sp=0:s=0◄
IF ba>0 THEN ◄
GOTO DoIt◄
ELSE◄
LOCATE 10,10:COLOR 4,10:PRINT "Sorry, you are br
oke!":LOCATE 12,12◄
PRINT "Play again? (Y/N)"◄
Answer:GOSUB GetKey◄
IF a$="Y" THEN GOTO Start ELSE IF a$="N" THEN GO
TO EndGame◄
GOTO Answer◄
END IF   ◄
DoIt:COLOR 7,10:LOCATE 6,12:◄
ba=INT(ba*100)/100◄
PRINT USING " Bank = $$#######.##";ba ◄
LOCATE 8,12:PRINT "Playing Hand #"+STR$(ha+1)◄
LOCATE 10,15:COLOR 11,15:PRINT " Q ":LOCATE 12,1
5:PRINT " R "◄
COLOR 11,10:LOCATE 10,19:PRINT "to Quit":LOCATE
12,19:PRINT "to Restart"◄
GetBet:LOCATE 16,1:PRINT sp$:COLOR 1,10:LOCATE 1
6,5◄
INPUT "Enter your bet";bet$:bet=INT(VAL(bet$)*10
0)/100◄
bet$=UCASE$(bet$)◄
IF LEFT$(bet$,1)="Q" THEN GOTO EndGame◄
IF LEFT$(bet$,1)="R" THEN CLS:GOTO Start◄
IF bet<=0 OR bet>ba THEN GOTO GetBet◄
GOSUB GameScreen:IF h=1 GOTO Main◄
ChecKey:GOSUB GetKey◄
IF a$="P" THEN GOSUB SpLit◄
IF a$="D" THEN GOSUB DoubLe◄
IF a$="H" THEN GOSUB Hit◄
IF a$="S" THEN GOSUB Stand:GOTO Main◄
IF a$="E" THEN ba=ba-bet*.5:GOTO Main◄
IF h=1 THEN h=0:GOTO Main◄
GOTO ChecKey◄
DeLay:FOR i=1 TO 5000:NEXT:RETURN◄
Stand:GOSUB ShowDeaLer:GOSUB VdeaLer◄
IF sp<>1 THEN COLOR 4,10:LOCATE py+2,px:PRINT "S
tand":GOSUB PvaLue:GOSUB VdeaLer◄
IF dt1=21 AND d=2 THEN LOCATE dy+2,dx:PRINT "Bla
ckjack!":h=1:ba=ba-bet:RETURN◄
DeaLerPLays:◄
IF dt1>21 THEN◄
LOCATE dy+2,dx:COLOR 4,10:PRINT "Busted!":ba=ba+
bet:GOSUB Pwin:h=1◄
RETURN◄
END IF◄
IF dt1<17 OR dt2<17 THEN GOSUB GetCard:GOSUB Vde
aLer:GOTO DeaLerpLays◄
GOSUB CheckCards:RETURN◄
GetCard:FOR t= 1 TO 3000:NEXT◄

c=c+1:dl$(d)=cd$(m(c)):tp$=dl$(d):d=d+1:c=c+1:x=
dx*8-1:y=dy*8-11◄
xL=dx:yL=dy:dx=dx+6:GOSUB ShowCard:RETURN◄
CheckCards:◄
IF sp=1 THEN GOSUB SpLitHand◄
IF dt1<=21 THEN dt=dt1 ELSE dt=dt2◄
IF dt2>dt1 AND dt2<=21 THEN dt=dt2◄
IF pt1<=21 THEN pt=pt1 ELSE pt=pt2◄
IF pt2>pt1 AND pt2<=21 THEN pt=pt2◄
COLOR 4,10:IF dt=pt THEN LOCATE 12,10:PRINT "Pus
h, No winner":GOSUB DeLay◄
IF dt>pt AND dt<=21 THEN GOSUB HouseWin:ba=ba-be
t◄
IF pt>dt AND pt<=21 THEN GOSUB Pwin:ba=ba+bet◄
RETURN◄
SpLitHand:◄
IF vt1<=21 AND pt1<=21 AND vt1>pt1 THEN pt1=vt1◄
IF vt2<=21 AND pt2<=21 AND vt2>pt2 THEN pt2=vt2◄
RETURN◄
Hit:c=c+1:pl$(p)=cd$(m(c)):tp$=pl$(p):p=p+1:x=px
*8-1:y=py*8-4:xL=px◄
yL=py+1:px=px+6◄
GOSUB ShowCard:GOSUB PvaLue◄
IF pt1>21 AND pt2>21 THEN◄
LOCATE py+2,px:COLOR 4,10:PRINT "Busted!"◄
IF sp=0 THEN◄
GOSUB ShowDeaLer:GOSUB HouseWin:ba=ba-bet:GOSUB
DeLay:h=1◄
ELSE◄
sp=0◄
END IF◄
END IF◄
RETURN◄
DoubLe:◄
GOSUB PvaLue:IF pt1<10 OR pt1>11 THEN a$="":RETU
RN◄
LOCATE 10,12:COLOR 4,10:PRINT "Double Down!":bet
=bet*2:GOSUB Hit:a$="S":RETURN◄
ShowDeaLer:x=1*8-1:y=5*8-11:xL=1:yL=5:tp$=dl$(0)
:GOSUB ShowCard:RETURN◄
HouseWin:LOCATE dy+2,dx:COLOR 4,10:PRINT "House
Wins!":RETURN◄
Pwin:LOCATE py+2,px:COLOR 4,10:PRINT "Winner!":R
ETURN
                                               ◄
PvaLue:pt2=0:pt1=0:FOR x=0 TO p-1:g$=pl$(x):GOSU
B VaLue:pt1=pt1+v1◄
pt2=pt2+v2:NEXT:RETURN◄
VdeaLer:dt2=0:dt1=0:FOR x=0 TO d-1:g$=dl$(x):GOS
UB VaLue:dt1=dt1+v1◄
dt2=dt2+v2:NEXT◄
IF dt1<21 THEN dt2=dt1◄
IF dt1>21 THEN dt1=dt2◄
RETURN◄
VaLue:v1=0:v2=0:FOR y=1 TO 13◄
IF LEFT$(g$,1)=v$(y) THEN v=y+1◄
NEXT◄
IF v=14 THEN v1=11:v2=1:RETURN◄
IF v>10 THEN v1=10:v2=10:RETURN◄
v1=v:v2=v◄
RETURN◄
SpLit:◄
IF LEFT$(pl$(0),1)<>LEFT$(pl$(1),1) THEN RETURN◄
LINE(55,124)-(88,160),10,bf:r=0◄
c=c+1:sp=1:px=1:py=10:tp$=pl$(1):x=px*8-1:y=py*8
-4:xL=px◄
yL=py+1:p=2:GOSUB ShowCard:px=px+6◄
temp$=pl$(0):pl$(0)=pl$(1):pl$(1)=cd$(m(c)):c=c+
```

www.commodore.ca

```
l:tp$=pl$(1)◄
x=px*8-1:y=py*8-4:xL=px:yL=py+1:GOSUB ShowCard:p
x=px+6:◄
pt1=0:pt2=0:p=2:GOSUB PvaLue◄
SpLitLup:GOSUB GetKey◄
IF a$="H" THEN GOSUB SPHit◄
IF a$="S" AND r=1 THEN  COLOR 4,10:LOCATE py+2,p
x:PRINT "Stand":RETURN◄
IF a$="S" AND r=0 THEN vt1=pt1:vt2=pt2:COLOR 4,1
0:LOCATE py+2,px:PRINT "Stand":GOSUB SecondHand◄
IF pt1>21 AND pt2>21 THEN◄
LOCATE py+2,px:COLOR 4,10:PRINT "Busted!":a$=""◄
IF r=1 THEN a$="S":RETURN◄
IF r=0 THEN vt1=pt1:vt2=pt2:GOSUB SecondHand◄
END IF◄
GOTO SpLitLup◄
SPHit:c=c+1:pl$(p)=cd$(m(c)):tp$=pl$(p):x=px*8-1
:y=py*8-4◄
xL=px:yL=py+1:px=px+6◄
GOSUB ShowCard:p=p+1:GOSUB PvaLue◄
RETURN◄
SecondHand:◄
pt1=0:pt2=0:r=1:px=7:py=16:pl$(0)=temp$:c=c+1:pl
$(1)=cd$(m(c)):p=2◄
tp$=pl$(1):x=px*8-1:y=py*8-4:xL=px:yL=py+1:px=px
+6:GOSUB ShowCard:GOSUB PvaLue:LOCATE 1,1:PRINT
pt1,pt2:RETURN◄
GameScreen:GOSUB ShuffLeCards:CLS:GOSUB TitLe◄
FOR i=0 TO 3:j=i*2:LINE(1+j,168+j)-(310-j,190-j)
,i+12,bf:NEXT◄
COLOR 2,15:LOCATE 23,3:PRINT "it   tand  ouble-do
wn S lit Surr nder"◄
COLOR 12,15:LOCATE 23,2:PRINT "H":LOCATE 23,6:PR
INT "S":LOCATE 23,12:PRINT "D"◄
LOCATE 23,25:PRINT "p":LOCATE 23,34:PRINT "e":p=
0:d=0◄
dx=1:dy=5:px=1:py=16:FOR i=1 TO 4:tp$=cd$(m(i))◄
IF i=1 OR i=3 THEN◄
x=px*8-1:y=py*8-4:xL=px:yL=py+1:px=px+6:pl$(p)=t
p$:p=p+1◄
ELSE                    ◄
x=dx*8-1:y=dy*8-11:xL=dx:yL=dy:dx=dx+6:dl$(d)=tp
$:d=d+1◄
END IF◄
IF i=2 THEN tp$="   "◄
GOSUB ShowCard:NEXT:c=4◄
Insurance:◄
IF LEFT$(dl$(1),1)="A" THEN◄
LOCATE 12,12:COLOR 4,10:PRINT "Purchase insuranc
e? (Y/N)":GOSUB GetKey◄
LOCATE 12,1:PRINT sp$:◄
IF a$="Y" THEN◄
IF bet-bet-bet*.5<ba THEN◄
ins=1:ba=ba-bet*.5◄
ELSE◄
LOCATE 12,1:PRINT "You can't afford insurance.":
GOSUB DeLay◄
LOCATE 12,1:PRINT sp$◄
END IF◄
ELSE◄
IF a$<>"N" THEN◄
GOTO Insurance◄
END IF◄
END IF◄
END IF◄
GOSUB VdeaLer:◄
IF dt1=21 THEN ◄
h=1:GOSUB ShowDeaLer:LOCATE dy+2,dx:COLOR 4,10:P
RINT "Blackjack!":GOSUB DeLay:ba=ba-bet◄
IF ins<>1 THEN ba=ba-bet:RETURN ELSE RETURN◄
ELSE◄
LOCATE 12,8:PRINT "Dealer does not have BlackJac
k":GOSUB DeLay◄
LOCATE 12,1:PRINT sp$◄
END IF◄
END IF◄
GOSUB PvaLue◄
IF pt1=21 AND p=2 THEN◄
COLOR 4,10:LOCATE py+2,px:PRINT "Blackjack!":h=1
:GOSUB ShowDeaLer:GOSUB VdeaLer◄
IF dt2<>21 THEN◄
ba=bet*1.5+ba:GOSUB DeLay:RETURN◄
ELSE◄
LOCATE dy+2,dx:PRINT "Blackjack!":LOCATE 12,10:P
RINT "Push, No winner":h=1:GOSUB DeLay◄
END IF◄
```

```
END IF◄
GOSUB DeLay◄
RETURN◄
ShowCard:◄
IF tp$="   " THEN PUT (x,y),card(0,4),PSET:RETURN
◄
j=ASC(RIGHT$(tp$,1))-65◄
PUT (x,y),card(0,j),PSET:COLOR 1,4◄
LOCATE yL,xL+1:PRINT LEFT$(tp$,1):LOCATE yL+3,xL
+4:PRINT LEFT$(tp$,1)◄
RETURN◄
TitLe:FOR i=0 TO 3:j=i*2:LINE(64+j,j)-(256-j,23-
j),i+12,bf:NEXT◄
COLOR 2,15: LOCATE 2,13:PRINT "Casino  Blackjack
":RETURN◄
GetKey:a$=INKEY$◄
IF a$<>"" THEN◄
a$=UCASE$(a$)◄
ELSE◄
GOTO GetKey◄
END IF◄
RETURN◄
ShuffLeCards:FOR i=1 TO 52:m(i)=i:NEXT:FOR i=1 T
O 52:x=INT(52*RND)+1◄
tp=m(i):m(i)=m(x):m(x)=tp:NEXT:CLS:ha=ha+1:RETUR
N◄
InitiaLize:ha=0:GOSUB InitShapes:ba=100:RESTORE
cards:FOR x=0 TO 3◄
FOR t=1 TO 13:READ cd$(x*13+t):cd$(x*13+t)=cd$(x
*13+t)+CHR$(x+65)◄
NEXT:RESTORE cards:NEXT:FOR t = 13 TO 1 STEP-1:R
EAD v$(t):NEXT:RETURN◄
cards:DATA A,K,Q,J,T,9,8,7,6,5,4,3,2◄
DispLay:WIDTH 40:CLS:COLOR ,0◄
FOR x=0 TO 3:LINE (80+x*2,40+x*2)-(225-x*2,63-x*
2),12+x,bf:NEXT◄
COLOR 2,15:LOCATE 7,12:PRINT "Casino Blackjack"◄
COLOR 2,10:LOCATE 12,4:PRINT "Copyright 1988 COM
PUTE! Pub., Inc."◄
LOCATE 13,11:PRINT "All Rights Reserved":FOR x=
1 TO 10:GOSUB DeLay:NEXT◄
COLOR 4,0:RETURN◄
InitShapes:RESTORE InitShapes:check=0◄
LINE(4,4)-(37,40),12,bf:GET (4,4)-(37,40),card(0
,4):FOR j=0 TO 3:FOR i=0 TO 30◄
READ a$:sb(i,j)=VAL("&H"+a$):check=check+sb(i,j)
:NEXT i,j◄
IF check<>178748& THEN PRINT "Error in Shape Dat
a":GOSUB DeLay:STOP◄
FOR j=0 TO 3◄
LINE(4,4)-(37,40),4,bf:PUT (14,6),sb(0,j),PSET:P
UT (17,30),sb(0,j),PSET◄
GET (4,4)-(37,40),card(0,j):CLS:NEXT j:RETURN◄
diamond:DATA B,9,3,400,E00,1F00,3F80,7FC0◄
DATA 3F80,1F00,E00,400,400,E00,1F00,3F80◄
DATA 7FC0,3F80,1F00,E00,400,FBE0,F1E0,E0E0◄
DATA C060,8020,C060,E0E0,F1E0,FBE0,0◄
CLub:DATA B,9,3,0,0,0,0,0◄
DATA 0,0,0,0,E00,1F00,1F00,7FC0◄
DATA FFE0,FFE0,75C0,E00,1F00,F1E0,E0E0,E0E0◄
DATA 8020,0,0,8A20,F1E0,E0E0,0◄
Heart:DATA B,9,3,71C0,FBE0,FFE0,FFE0,7FC0◄
DATA 3F80,1F00,E00,400,71C0,FBE0,FFE0,FFE0◄
DATA 7FC0,3F80,1F00,E00,400,8E20,400,0◄
DATA 0,8020,C060,E0E0,F1E0,FBE0,0◄
Spade:DATA B,9,3,0,0,0,0,0◄
DATA 0,0,0,400,400,E00,1F00◄
DATA 3F80,7FC0,75C0,E00,1F00,FBE0,FBE0,F1E0◄
DATA E0E0,C060,8020,8A20,F1E0,E0E0,0◄
DoScreen:WINDOW OUTPUT 3:COLOR 3,0:WIDTH 40:REST
ORE PaLetteData◄
FOR i=0 TO 15:READ r,g,b:PALETTE i,r,g,b:NEXT:RE
TURN◄
PaLetteData:DATA 0,.7,0,0,0,0,0,0,0,1,0,0,1,1,1,
0,0,0,1,0,0,1,1,1,0,0,0◄
DATA 1,0,0,0,.7,0,1,0,0,1,0,0,1,.5,.1,1,.7,.1,1
,.6,.1◄
HoLdscrn:LOCATE 12,6:COLOR 5,10:PRINT "Press any
 key to continue....":GOSUB GetKey:RETURN◄
EndGame:CLS:COLOR 1,10:LOCATE 12,10:PRINT "Thank
 you for playing":COLOR 11◄
LOCATE 14,12:PRINT "Casino Blackjack":GOSUB DeLa
y:WINDOW CLOSE 3◄
SCREEN CLOSE 1:END◄
◄
```

Centerstage: AMIGA!
The World Of Commodore Show

Neil Randall

Every December since 1983, Commodore Canada has sponsored the World of Commodore show at the International Centre in Toronto. The show spotlights Commodore computers and their wealth of third-party products. This year the showcase was seen by over 40,000 attendees. More than 85 exhibitors displayed their wares, with Commodore itself taking up the largest area.

Over the past five years, the Commodore 64 has been the World of Commodore's dominating star. In the recent 1987 show, however, the Amiga took center stage.

Amiga was everywhere, with its graphics dancing on monitors and its music booming from speakers. To find new products for the 64/128, one had to stop and search. Almost buried was the PC 10-III, Commodore's newest entry in its line of PC-compatibles. For the first time, the company heard complaints about the lack of new products for the 64/128, and for the first time, nobody speculated on the imminent demise of the Amiga.

The New Machine
Aside from its overt support for the Amiga, Commodore's biggest news was the brand new PC 10-III. An XT compatible, the PC 10-III is designed for those who want a PC compatible that anticipates the options they may later require. The MS-DOS world boasts (and in many ways, quite rightly) of its machines' expandability; but expandability, for many, is intimidating. String some wires together and you can operate a 64 or an Amiga, but PC compatibles and clones often need further purchases before they

can be considered fully loaded.

The PC 10-III includes only three expansion slots, but the machine's design implies that three is plenty. The package includes 640K RAM, monochrome and color graphics capability (for both RGB and composite monitors), Hercules and Plantronics video interfaces, a built-in clock/calendar with battery backup, a floppy drive controller, and a hard disk interface. Several of these features have to be added separately to many compatibles. Other features include a socket for an 8087 math coprocessor, software-controllable speeds of 4.77, 7.16, and 9.54 MHz, two built-in 360K floppy drives, and room for an internal half-height hard drive. The 10-III should carry a retail price under $1,200.

The Old Machines
In this case, "the old machines" include the Commodore 64 and Commodore 128. To many, these computers have always been inextricably linked, and most software is marketed as 64/128 compatible. They are treated, for the most part, as one.

Most of the 64/128 products on display at the show took the form of productivity and applications software. Berkeley Softworks displayed their newest GEOS products, GeoFiler 128, GeoPublish 64, and GeoProgrammer 64 (128 versions of the latter two products should be available soon). GeoPublish is an extremely friendly desktop publishing package aimed for home and educational use, while GeoProgrammer combines an assembler, a linker, and a debugger

for work in assembly language.

Supporting GEOS were Abacus' BeckerBASIC, for BASIC programming within the GEOS environment, and CDA's NeoFont, which gives all GEOS packages 23 new fonts in four different sizes. CDA also showed the Graphics Transformer. This package allows you to transfer pictures from one paint program to another. It supports most existing 64/128 paint programs.

Many attendees were impressed by Multi-Link, a 64/128 networking system designed for classroom use. The package links up to 48 Commodore 64s or 128s via linking hardware and an included 20-megabyte hard drive. Each computer station can interact directly with the control station (operated by the teacher), or they can operate individually as stand-alone 64s and 128s. A 10-station system, including the hard drive and the linking hardware, will cost about $3,000.

Productivity packages from Electronic Arts included Paperclip III, Outrageous Pages, and Mavis Beacon Teaches Typing. Paperclip III is the latest version of the popular word processor from EA's "Batteries Included" line, while Outrageous Pages is desktop publishing software with strong graphics tools and the ability to snake text around graphics on the page. Mavis Beacon Teaches Typing is a typing tutor whose difference is its creation of customized lessons for each user. With a highly favorable review in The New York Times, the package

seems on its way to a strong future.

Games for the 64/128 line came almost exclusively from Electronic Arts. *Halls of Montezuma* is the latest module in SSG's Battlefront series, and the delayed *Great Battles of the American Civil War* is promised soon. *Bard's Tale III* has several new features, including an automatic mapping system. *Skyfox II*, *Strike Fleet*, and *Skate or Die* round out the new selections.

The Machine That Stole The Show

What the World of Commodore showed most clearly was that the Amiga has come of age. Since its introduction, the Amiga has been the victim of Commodore's procrastination, and this, in turn, hampered its support by software developers. To judge from the dazzling new products both announced and displayed at this year's show, the Amiga is beginning to fulfill its promise of product availability.

Also evident was the growing split in the focus of Amiga software. The Amiga may have been intended as a home computer, but much of the software is squarely aimed at the professional user. Music professionals, graphics professionals, publishing professionals, and even airline professionals were well-targeted by this year's new products. Many of these products, in fact, lie well beyond the budget of the home user. But the nonprofessional had much to choose from as well, with word processors, paint programs, music programs, and games on display practically everywhere.

Amiga Hardware

Toronto-based Comspec demonstrated its new hard drive and its soon-to-be-released tape back-up system. Actually, the hard drive consists of two parts: the SCSI Host Adaptor and the Hard Drive Chassis. The software allows drive sizes ranging from 10–300 megabytes to be connected to the Adaptor, and the hard drive system allows you to boot the Amiga in eight seconds without using the Kickstart and Workbench disks. The tape back-up on display backed up 120 megabytes worth of data.

Two companies showed new RAM expansion devices. *Inboard*,

from Spirit Technology of Salt Lake City, gives the 1000 or the 500 an extra megabyte of RAM, and the board is installed internally. Like most expansion boards now, *Inboard* is autoconfigurable. Byte by Byte announced the *Byte Box*, which gives the 500 up to two megabytes of external expansion. It, too, autoconfigures, and it has its own power supply.

A British Columbia company, Hypertek/Silicon Springs, was on hand with three products. The Amiga Light Pen is designed to work with any Amiga program. Its two buttons take the place of the two mouse buttons, and it allows single-pixel drawing precision. The Deluxe Midi Interface allows a serial pass-through, and includes MIDI In, MIDI Out, and switchable Thru/Out.

Finally, the TTL High Resolution Monitor is a device that plugs into the RBG port and offers relatively flicker-free high-resolution monochrome output.

NewTek, makers of the popular *Digi-View* system, announced the *Video Toaster*, which at almost $800 offers realtime color digitizing, video effects, and broadcast-quality genlock. This full-featured professional digitizer/genlock system should provide many users with the high-quality video tools they've been waiting for.

Amiga Software

Electronic Arts displayed *Ferrari Formula One*, a new racing game in its *Sports Legends* series. Announced for the Amiga were *Bard's Tale II*, *Reach For The Stars* (third edition), Sierra On-Line's *Thexder* (an arcade game from Japan), *Space Quest II*, and *Police Quest*, the latter two from the popular *King's Quest* system. Also announced was *The Hunt For Red October*, a submarine action/strategy game from MicroProse based on the Tom Clancy book of the same name.

EA also had two announcements for its Deluxe series. *Deluxe Productions* is a hi-res graphics animation program designed for business and professional use. *Deluxe Write* is a graphics word processor that allows the importing of color graphics from programs such as *Deluxe Paint II* and *Deluxe Print II*.

Three other new word proces-

sors let users combine text and graphics. *TextPro*, from Abacus, includes a snapshot program that saves IFF graphics from programs that do not offer this option. *Becker-Text* (also from Abacus) takes *TextPro* a few steps further with up to five math columns per page and printing in up to five columns. Both programs feature a C-Source mode for help in formatting the source-code around C's necessary braces.

A third word processor, *KindWords*, from The Disc Company, looks very much like *TextCraft*, as it should have been. It merges color IFF graphics within the text, contains its own fonts, and contains a host of very friendly features. The program's appearance is very close to *TextCraft*'s, but it has dispensed with the annoyances of the Amiga's first word processor.

Of course, *WordPerfect* has taken Amiga word processing by storm, and WordPerfect Corporation was on hand to display and support its product. *WordPerfect* for the Amiga set the company's all-time first-month sales record, and users of this full-featured product have no trouble figuring out why. The company also announced *WordPerfect* 5.0 for MS-DOS owners, which integrates text and graphics and introduces several other features. Certainly 5.0 will be available for Amiga soon.

The Other Guys, from Logan Utah, demonstrated *Promise*, a 90,000-word spelling checker, *Reason*—a professional text and style analyzer for writers—and *Synthia*, a program for creating digital IFF instruments for use with music programs. An impressive synthesizer program, *Synthia* includes such features as the simulation of plucked strings, drum-kit creation, an IFF music player for users with limited RAM, and a means of simulating the natural imperfections found in real instruments.

For musicians, Sound Quest had its *Editor/Librarian* for Roland, Yamaha, and Casio CZ synthesizers. Sound Quest's stage demonstration showed the ease with which the program lets you perform multitrack recording using a synthesizer, and the ease with which you can change instrument sounds as you play. Music professionals have been enchanted by Sound Quest since its origins as *Texture* (from the now-defunct Cherry Lane Technologies).

More For The Amiga

There was more, much more. Taurus Software showed its full-color 2D *X-CAD Designer* while Haitex Resources announced *Acquisition Pioneer*, a scaling-down of its *Acquisition* relational database for less demanding users. Gold Disk showed *ComicSetter*, a comic-book creation package to be released shortly. Syndesis offered *Interchange*, a program for sharing objects between Byte By Byte's *Sculpt 3D* and Aegis Development's *VideoScape 3D*.

Software Visions demonstrated *Microfiche Filer*, a unique (and extremely easy to use) database system, while another easy solution came from Oxxi, with its *Nimbus* small-business accounting package. Meridian Software had *Zing Spell*, a 95,000-word realtime spelling checker; and *Zing Keys*, a macro and hot key program; and the company announced a significant update to *Zing*, its multifeatured Amiga utility. Finally, RGB Video Creations was on hand with *Deluxe Help* for *Deluxe Paint*, a well-designed tutorial for those intimidated by *Deluxe Paint*'s multitude of features.

Of all the products on display, though, three merit special mention. From Gold Disk comes *Professional Page*, the first Amiga desktop publishing program designed specifically for professional use. A fully featured word processor and document-creation program, *Professional Page* supports output to any PostScript laser printer or typesetting machine. Its features are far too numerous to describe, but the program is fully supported by the Toronto-area Vellum Print and Graphic Services, who produced the fine-looking *Professional Page* manual using that program exclusively. In January, Vellum will begin offering training courses on the use of *Professional Page* and other print-oriented Amiga hardware and software.

Unquestionably, the most unusual product of the show was Very Vivid's *Mandala*. Described as an interactive realtime video development system, *Mandala* joins with realtime video digitizers to let you physically interact with the Amiga's monitor. By simply standing in front of the monitor and waving your arms around, you can create some extremely interesting graphics and sound effects. The potential use of *Mandala* is only within the realm of speculation, but working it into video presentations and interactive games seem two of the most obvious uses.

Airware Solutions of Laval, Quebec was on hand with its *AWS-1000 Pairing Generator System*. Designed with extensive research into airline systems, the *AWS-1000* is the first pairing generator available to airlines. Basically, what a pairing system does is make sure that pilots and flight attendants all end up in the right places. Until now, airlines figured out the pairings in various ways, then ran them through a pairing optimizer to see if they were valid. The *AWS-1000* does all that work for the airline. Future AWS modules will include a Flight Crew Blocking System, and Aircraft Fleet Management System, a Flight Crew Scheduling System, and a Pilot Cost Evaluation Analysis. Though aimed obviously at a niche market, this software is highly impressive.

The View From Commodore

According to Stan Pagonis, Director of Marketing Operations for Commodore Canada, the emphasis on the Amiga was entirely appropriate. Although he quickly pointed out that the seminars and stage shows demonstrated the 64/128 and the PC line as well, he suggested that "because we believe the Amiga is our future, we're going to concentrate on it." For the first time, he said, "we were finally able to tell the people that the Amiga was a professional machine." For the first time, too, "we put the software question to rest."

But he insisted that the 64/128 was not about to die. The ready base of available programs attracts many new computer owners, he suggests, and that is why sales of the 64/128 remain very strong. Commodore will continue to support it and they'll continue to expand their line of PC-compatibles. Yet, to judge from the World of Commodore show, the Amiga is becoming a dominant machine. ©

The Amiga Virus

Jim Butterfield, Contributing Editor

The Amiga has been the victim of a hacker's practical joke. A group of programmers from Europe developed a small program that affects Amiga Workbench disks. This not-so-harmless virus recently caused quite a stir at The World of Commodore show in Toronto, Canada. Jim Butterfield explains.

Some call it a minor annoyance. Some call it a serious hazard. Either way, a small system program found on some Amiga disks became widespread at The World of Commodore show in Toronto, Canada. This "virus" spreads from disk to disk without the user's knowledge and can "infect" an entire library before it makes itself known.

The World of Commodore show did not spawn this software infection. The virus comes from a group in Switzerland called SCA, so it's sometimes called the SCA virus. At least one other version (a mutation) exists. It has been known in Europe for some time, and it's believed to have made its way to North America from Germany in late October 1987.

Spreading The Virus
The virus affects Amiga Workbench disks. If you boot with an infected disk, the virus program installs itself into the computer's memory. To further spread the infection, this program wedges itself into the Amiga's reboot code.

The virus waits benignly until the user gives the three-key Amiga reboot sequence. That's the one where you press the CTRL, Commodore, and Amiga keys together. On the Amiga 1000, it's the CTRL, left-Amiga, and right-Amiga keys. The virus survives the reboot and plants itself onto the boot track of the current Workbench disk.

Most users think that the three-key reboot makes the system start over, cleaning out any previous code. Not so with the virus: It stays in there, waiting to infect your next boot disk. And after a number of computer reboots, the virus prints a message such as: *A wonderful thing has happened. Your Amiga is alive! And, even better, some of your disks are infected by a virus!*

Diagnosing The Virus
Is the virus harmless? Not quite. Some commercial programs use disk copy-protection systems that live in the same area that the virus invades. If the virus spreads too far (infects too many disks), you could lose some of your most valuable programs. And there's another danger. With certain Amiga memory configurations, the virus mistakenly loads itself into the wrong place, thus causing a system failure.

How can you tell if any of your disks have the virus? There are programs available that analyze your disks, but there's an easier way: Insert the suspected Workbench disk, hold down the left mouse button, and reset your computer using the three-key reboot. If there is a virus in memory, the screen flashes green for about half a second as the computer begins to reboot.

There may be certain strains of the virus that do not exhibit this tell-tale green flash. For this reason, we offer "Vlook." Vlook is a short Amiga Basic program that checks the computer's memory for the virus. When you have typed in and saved the program, run it.

Vlook creates a program file on the Amiga's RAM disk. You may copy this file to the disk in drive 0 by activating the CLI and typing:

COPY RAM:VLOOK DF0:

To run the program and test your computer's memory, enter this instruction from the CLI:

RAM:VLOOK

If your computer is infected, the warning *Memory contains a known VIRUS* appears. Otherwise you'll see "Memory contains no signs of a known virus." If you receive the first message, one or possibly all of your boot disks are infected.

The Antidote
How can you get rid of the virus? If your disk does not contain a commercial protection scheme, the CLI command INSTALL can rid your infected boot disks of the virus. To clear the virus from the Workbench disk in drive 0, for example, enter INSTALL DF0: from the CLI. Do **not** use INSTALL if your disk has a copy-protection scheme, since this would probably render the disk useless.

Non-Workbench disks (disks that do not boot the computer) cannot be infected by the virus. Also, if you keep your Workbench disks write-protected, the virus does not affect them—an effective method of immunization.

Even after you clean up a disk, the virus remains in memory. To flush the virus from the computer's memory, shut the computer off and leave it off for at least a minute before rebooting.

Vlook

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
◄
 DATA 1,210,0,199,0,3,243,202◄
 DATA 0,0,0,0,0,0,0,62◄
 DATA 1,0,0,0,0,0,0,0◄
 DATA 0,0,0,0,48,0,0,114◄
 DATA 3,233,0,0,0,48,67,250◄
 DATA 0,114,66,64,44,120,0,4◄
 DATA 78,174,253,216,44,64,103,0◄
 DATA 0,88,78,174,255,196,46,195◄
 DATA 0,69,250,0,100,38,60,0◄
 DATA 0,0,34,78,186,0,70,12◄
 DATA 185,67,72,87,33,0,7,236◄
 DATA 8,102,0,0,16,12,185,65◄
 DATA 250,255,242,0,7,236,12,103◄
 DATA 0,0,16,69,250,0,92,38◄
 DATA 60,0,0,0,12,78,186,0◄
 DATA 28,69,250,0,90,38,60,146◄
 DATA 0,0,0,15,78,186,0,14◄
 DATA 34,78,44,120,0,4,78,174◄
 DATA 254,98,78,117,36,10,34,7◄
 DATA 78,238,255,208,100,111,115,46◄
 DATA 108,105,98,114,97,114,121,0◄
 DATA 40,32,67,111,109,112,117,116◄
 DATA 101,33,32,49,57,56,56,32◄
 DATA 41,10,77,101,109,111,114,121◄
 DATA 32,99,111,110,110,97,105,110◄
 DATA 115,32,110,111,32,115,105,103◄
 DATA 110,115,32,111,102,32,97,32◄
 DATA 107,110,111,119,110,32,86,73◄
 DATA 82,85,83,46,10,0,0,0◄
 DATA 0,0,3,242◄
p=236◄
DIM k(10),fil(p)◄
k(0)=1◄
FOR j=1 TO 8◄
 k(j)=k(j-1)*2◄
NEXT j◄
FOR j=1 TO p◄
 READ fil(j)◄
NEXT j◄
FOR j=1 TO p◄
 FOR j1=0 TO 10◄
  IF k(j1)<>j AND ((k(j1) AND j)>0) THEN◄
   k=k(j1)◄
   fil(k)=fil(k) XOR fil(j)◄
  END IF◄
 NEXT j1◄
NEXT j◄
e=0◄
FOR j=0 TO 7◄
 IF e<fil(k(j)) THEN e=fil(k(j))◄
NEXT j◄
IF e<>0 THEN◄
 a=0◄
 mult=0◄
```

```
FOR j=0 TO 7◄
 IF fil(k(j))<>0 THEN◄
  a=a+k(j)◄
  IF fil(k(j))<>e THEN mult=1◄
 END IF◄
NEXT j◄
IF mult=1 THEN◄
 PRINT "Multiple errors .. look around: "◄
ELSE ◄
 PRINT "error at: "◄
END IF◄
a=a-1 ◄
ln=INT(a/8):it=a-ln*8◄
PRINT "Line";ln+1;" Item";it+1◄
STOP◄
ELSE◄
 PRINT "writing file..."◄
 OPEN "ram:VLook" FOR OUTPUT AS #1◄
 a=0◄
 FOR j=1 TO p◄
  IF j<>k(a) THEN◄
   PRINT#1,CHR$(fil(j));◄
  ELSE◄
   a=a+1◄
  END IF◄
 NEXT j◄
 CLOSE 1◄
 PRINT "File complete."◄
END IF◄
◄
```

©

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers' Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

A Volatile BASIC

I have a Commodore 64. I want to copy the BASIC ROM to the underlying RAM (locations 40960–49151). Currently, I'm doing this with a FOR/NEXT loop:

FOR I = 40960 TO 49151:POKE I,PEEK(I):NEXT

This loop requires a little more than 30 seconds to execute. Can you show me a faster way to move BASIC?

Dan Sanderson

The fastest way to move large blocks of memory is to use the 64's native language—6502 machine language. Figure 1 is a machine language program that moves BASIC from ROM to RAM.

You'll probably want the program in the form of a BASIC loader:

```
10 FOR I = 679 TO 702:READ
   A:POKEI,A:NEXT
20 DATA 169,0,133,251,168,169,160,133
30 DATA 252,162,32,177,251,145,251,200
40 DATA 208,249,230,252,202,208,244,96
```

Type in this program and type RUN.

When you're ready to move BASIC, type SYS 679. The cursor should reappear in about one second. BASIC has been copied to RAM.

Once you've moved BASIC into RAM, you can modify it as you wish. To put your modifications into effect, you must turn off the BASIC ROM with this statement:

POKE 1, PEEK(1) AND 254

There are many interesting modifications to BASIC that you can make. You could use a FOR/NEXT loop to search for certain BASIC messages or command names.

As a sample modification, let's change the question mark used by the INPUT statement into a colon: POKE 43846,58. The ASCII value for a colon is 58. Location 43846 normally holds the number 63—the ASCII value for the question mark.

Sheldon Leemon suggests other changes in Mapping The 64, *from COMPUTE! Books.*

NEW Does Work

In recent issues of COMPUTE! magazine, the wrongful statement has been made that Atari BASIC does not clear the VNT (Variable Name Table) when NEW is typed. Actually, it does.

I have no doubt that programs of the complexity of Laser Chess and Biker Dave need to be LISTed and ENTERed, not because of left-over variables, but rather because people tend to inadvertently insert variable names while typing or editing or by "dry running" an unfinished program to see how it's doing.

Jesse Warford

We stand corrected. NEW does indeed effectively clear the variable table. The procedure we have given does work, but users may choose to type NEW after LISTing the program, rather than turning the computer off.

1541/1571 Dual-Drive System

It is not necessary to see this in print, but if you could help me, I would appreciate it very much. I have a 128 and currently two disk drives—a 1541 and a 1571.

My problem is how to set up the 128 so that every time I turn it on with two separate disk drives—my 1541 and 1571—the system will behave like my IBM PC, seeing the disk drives as separate devices. Do I have to run the DOS Shell program every time to set up a two-drive system? I am also thinking about buying a second 1571. How can I set up a two-drive system with two 1571s?

T.J. Bynum

Setting up a two-drive system with a 1541 and a 1571 is easy. You'll use the 1541 as device 8—its default—and configure the 1571 as device 9. It's easier to set the system up this way—with the 1541 as device 8 and the 1571 as 9—because one

Figure 1

02A7				*=		679	; ROUTINE LOCATION
		;	EQUATES				
02A7		BASIC		=		40960	; BASIC ROM
02A7		ZP		=		251	; FREE ZERO PAGE ADDRESS
		;					
02A7	A9	00		LDA	#<BASIC		; POINT ZP TO START OF BASIC, LOW BYTE FIRST
02A9	85	FB		STA	ZP		
02AB	A8			TAY			; AT THE SAME TIME, PUT 0 IN .Y
02AC	A9	A0		LDA	#>BASIC		; THEN HIGH BYTE
02AE	85	FC		STA	ZP+1		
02B0	A2	20		LDX	#32		; .X IS PAGE COUNTER (MOVING 32 * 256 BYTES)
02B2	B1	FB	LOOP	LDA	(ZP),Y		; MOVE A BYTE FROM ROM...
02B4	91	FB		STA	(ZP),Y		; ...TO RAM
02B6	C8			INY			; NEXT BYTE
02B7	D0	F9		BNE	LOOP		; CONTINUE FOR 256 BYTES
02B9	E6	FC		INC	ZP+1		; FOR NEXT PAGE
02BB	CA			DEX			; DROP PAGE COUNTER
02BC	D0	F4		BNE	LOOP		; AND CONTINUE FOR 32 PAGES
02BE	60			RTS			; WE'RE DONE.

of the drive's device numbers must be changed to 9, and the 1571 is much easier to change than the 1541.

To change the device number of the 1571, first look in Appendix A of the 1571 Disk Drive User's Guide that came with your drive. This section describes how to change the device number. Briefly, to configure the drive as device 9, you move the left DIP switch at the rear of the drive down. After making this adjustment, make sure the drives are chained together, and you're ready to go. Your system should now boot up and recognize the two drives as separate devices.

If you buy another 1571 and want to configure the two 1571s as a two-drive system, simply use the new drive as device 8—its default—leave the other drive configured as device 9, connect the two, and your system's all set.

Super *SpeedScript?*

I have an Atari computer, and I would like to know if "Atari XL/XE Super Editor" from the June 1987 issue of COMPUTE! can be modified to work with *SpeedScript.*

Susan Thompson

"Super Editor" is a handy editing utility for any program that uses the E: screen editing device. This includes BASIC and certain other programs. SpeedScript does not use the E: device—it has its own routines for handling keyboard entry and text displays. This allows SpeedScript to quickly handle word wrap, cursor movement, and scrolling. It also makes Speed-Script incompatible with any programs that depend on the E: device. So there is no way to make Super Editor compatible with SpeedScript.

Punctuated Numbers

I recently purchased a Tandy 1000SX computer. I have been working in BASIC, and I've noticed that when I list my program, any six-digit number has an exclamation point after it (123456!), and any eight-digit number has a pound sign after it (12345678#). To make the matter worse, these signs are also printed out when I make a hard copy listing of my program.

Could you please explain the reason for these punctuation marks?

Philip G. Crompton

Modern BASICs often have several ways to store variables. Numbers with few significant digits can be represented more compactly than numbers with more significant digits. When you type in a number, BASIC decides how much space it will need to store the number. It then places a suffix after the number to indicate which format it has used. The following

table shows the types of numeric variables available in BASICA, GW-BASIC, and Amiga Basic, and the number of bytes each format requires for storage:

%	integer (two bytes)
!	single-precision floating point (four bytes)
#	double-precision floating point (eight bytes)

These punctuation marks can be helpful. If you store a double-precision number in a single-precision variable, your calculations are likely to be less accurate than they would be if you used a double-precision variable.

Trig Scales

I own an Apple II and am working on a program that uses hi-res graphics. I'm trying to use SIN and COS to draw a line from the vertex of a circle to a point on its circumference. Here's my program:

```
10 HGR
20 HCOLOR = 7
30 HPLOT 140,70 TO 40 * COS(45) + 140,
   40 * SIN(45) + 70
```

This should draw a line from the point (140,70) to another point 40 pixels away at an angle of 45 degrees. It doesn't.

Sam Moelius

You have the right idea, but most computers don't do their trig functions in degrees—they do them in radians. To convert from degrees to radians, multiply by pi (3.14159265) and divide by 180 (or, equivalently, just divide by 0.017453312). Here is a corrected version of your program:

```
10 HGR
20 HCOLOR = 7
25 PI = 3.14159265
30 HPLOT 140,70 TO 140 − 40 * COS
   (45 * PI/180),70 − 40 * SIN(45 * PI/180)
```

DIR Power

When reading a directory on my PC, the file listing sometimes continues on for more than a full screen and it becomes impossible to read all the filenames before they are out of sight. Is it possible to stop this scrolling?

Also, on a keyboard that doesn't have a PrtSc key to print the screen, is it possible to get a printout of a directory listing?

Peter Richie

There are three ways to stop your directory listing from scrolling. The first is to press CONTROL-S to stop scrolling and to press any key to resume scrolling.

A second method is to append the switch /P to the DIR command like this

DIR /P

This pauses the directory listing between screenfuls of data.

The third option is to use the DOS filter MORE in a command line like the following:

DIR│MORE

This command will give essentially the same results as DIR /P discussed above.

While on the subject of directories and filters, there's another very useful DOS filter called SORT. With it, you can sort a directory (or any other file) on any field you wish. To create a sorted directory that will pause its display after each screen, use the following command:

DIR│SORT│MORE

Please note that you can't use the /P option to pause the directory when using the SORT filter. Also, when using filters, DOS creates a temporary file on your disk. There must be enough room on the current disk for the temporary file, and the disk cannot be write-protected.

The SORT filter can work wonders with directories. Since DOS is very rigid about the directory's format, you can tell SORT to begin sorting in certain columns and produce a directory sorted just the way you want. To sort a directory and specify a column in which to begin sorting, use a command of the form

DIR│SORT / + *n*

where *n* is the column number on which you want the sort to begin. Here are the important column numbers for the standard DOS directory:

1	filename
10	file extension
24	date
34	time
39	meridian (a.m. or p.m.)

If you want to view a directory sorted by file extension, for example, and also want the listing to pause after each screen, you would give the following command:

DIR│SORT / + 10│MORE

It's easy to obtain a printout of your directory without using a PrtSc key because of a DOS mechanism called piping. Piping allows you to redirect the output of a program—like a directory listing—to any file or device you wish. To send the output of the above directory command to the printer, all you need to do is append > PRN to the command. The > tells DOS to redirect output (you can redirect input with <) and PRN tells it to send the output to the printer.

So, the entire command would look like this:

DIR│SORT / + 10 > PRN

Notice we didn't use the MORE filter. When sending output to a diskfile or printer, pauses after each screen are unnecessary.

With DOS's filters and redirection, the simple DIR command becomes a powerful tool. ©

Microscope

Sheldon Leemon

IBM wanted to celebrate the success of its PS/2 line at the fall COMDEX show, so it proclaimed that it had shipped the one-millionth PS/2 computer and rolled out a big ad campaign based on the theme *Thanks a million*. Industry pundits quickly scrambled to determine just what *shipped* meant.

The consensus of opinion among these party-poopers was that IBM had built a million units of the new models and had sent them to dealers, but that maybe as many as a third of them still actually hadn't been sold. This technicality didn't stop IBM from flying the dealer who sold the millionth unit to Las Vegas for the show.

How could even IBM, with all of its computing power, figure out exactly which dealer sold the millionth computer? Their rather pragmatic approach was to have each of about 200 regional sales representatives pick one of their dealers, all 200 of whom were brought to the show on the premise that they had sold the lucky machine. Well, the winner was probably in the group there, somewhere.

★★★★★★★★★★★★

It wouldn't be an IBM product introduction without rumors of problems with the new machines. First, there was a minor problem with the BIOS of the Model 50 and Model 60 which caused the time to be inaccurate in some situations. This reportedly caused some dealers to conclude that the clocks were bad and to ship back all of their units as defective. IBM issued a quick DOS patch to fix that problem, but the confusion lingers.

Recently, there have been rumors of power-supply problems with the Model 50 and a high overall failure rate for the machines. IBM has issued a firm statement saying that there are no problems with the new machines.

Those of you who can remember way back to the AT introduction may recall that a number of users claimed that there were serious hard-drive problems with the machine. IBM denied all such reports. Several publications made intensive studies of the problem, leading to results that were inconclusive at best. By the time these studies were published, IBM had changed hard-drive vendors, and everybody had forgotten about the rumored problems.

★★★★★★★★★★★★

At the annual World of Commodore show in Toronto, Commodore officials announced that the company had met its goal of an installed base of 500,000 Amigas worldwide by the end of 1987. Part of the reason for this is the tremendous reception of the Amiga 500 in Europe. Over half of the Amigas sold have gone to customers outside of the U.S. Nonetheless, the 500 appears to be gaining momentum here as well. The 2000 has sold so much better than Commodore expected that the machine was unavailable for much of the Christmas selling season.

But, surprisingly enough, the Commodore 64 also appears to flourish. Despite the fact that the Amiga 500 offers ten times the power of the 64 for about a 50-percent-higher price, the name-recognition factor generated by an installed base of over ten-million machines obviously still carries some weight. Commodore chairman Irving Gould stated that he could have sold an additional fifty-thousand 64C's if he could have produced them in time for Christmas.

The continuing popularity of the 64 in Canada was particularly evident at the Toronto show, where the proportion of interest in the 64/128 as compared to the Amiga was much higher than at similar U.S. shows.

★★★★★★★★★★★★

Motorola has formally announced the 68030, the most recent addition to the 68000 family of processors used in such machines as the Macintosh, ST, and Amiga. While its predecessor, the 68020, has been likened in power to Intel's 80386 chip, the new processor is said to be two to four times as powerful. Since its instruction set is very close to those of the other 68000-series chips, software compatibility should be good.

The first name-brand machine to use the new processor will likely be the Macintosh. The Mac II already uses the 68020 and probably does not need any significant modifications in order to use the new chip. Insiders say that the 68030 version of the Mac II could show up any day now. Although such a machine would not be cheap, it would rank among the most powerful desktop units available. And though the ST and Amiga line still don't include a 68020 machine, both Commodore and Atari have stated that they intend to use the 68030 in future products.

★★★★★★★★★★★★

One of the most interesting stories to come out of the recent AppleFest was a statement from Apple that over 200,000 units of the IIGS had been sold so far. This figure is about twice as large as most previous industry estimates.

The reason so many people had underestimated the GS is that a large proportion of sales have been to schools and other institutions. Such sales may not be as visible as retail sales, but they are perhaps even more important to the continued success of the GS. Apple is well aware that large sales to schools today can generate even larger sales for home use in the future. ©

Cwww.commodore.ca

Computers, Writing, And AI

Of all the myriad applications for personal computers, I would guess that, by far, the most popular use is word processing. In the early 1980s, the availability of good word processing programs on micros virtually drove the dedicated word processors from the marketplace. Even though there has been a resurgence of dedicated word processors, there is little question that the personal computer will continue to be king in this application area.

As a writer who feels naked without his computer, I occasionally startle people when I say that I don't use a spelling checker. I thrive on the computer's ability to let me move text around to my heart's content; I enjoy the ability to format my output in any fashion I want, but I don't use the computer to check my spelling or usage.

Casting A Spell

My reluctance to use spelling checkers has nothing to do with my spelling skills; I make plenty of mistakes. My problem is that spelling checkers don't catch the kinds of errors I commonly make. Many (but not all) of my spelling errors result in other words. For example, *stop* becomes *spot*. A word-based spelling checker is useless in such cases. I would love to have a spelling checker that catches errors like these, but such programs have to be context sensitive.

Another problem with most spelling checkers is that they can lead you to wrong choices. For example, with the word *alot*, most spelling checkers would offer the choice *allot* instead of the correct split into two words, *a lot*. The word *payed* would be even harder to handle since spelling checkers that scan letter by letter would end up with *payer*, and so on, and not offer the correct past-tense form, *paid*.

The complexities of English are such that a good proofreading tool needs to be based on something far more sophisticated than a word list. It needs to have access to rules of grammar—an ability to deal with phonology, orthography, morphology, syntax, and semantics.

This is a tremendously complex task.

Because of the rule-based nature of much of English grammar, it is appropriate to seek some solutions for this problem in the domain of expert systems—sophisticated programs that are an outgrowth of years of research in the field of artificial intelligence.

A Model Emerges

A few months ago, I was privileged to visit with Dr. Robert Arn, the creator of just such a program for the ICON computer.

The ICON is manufactured by Unisys to meet the needs of educational computing in Canada. Dr. Arn designed the ICON, and he holds a doctorate in the field of linguistics. His interests in technology and linguistics came together in a project that has kept him and his colleagues busy for the past three years. During this time, he designed *English1*—a rule-based program that sits in the background of a word processor, ready to offer advice when asked.

To use *English1*, the user simply selects the text to be analyzed and requests that it be examined. Words that seem incorrect are highlighted. The user can then select any highlighted word and ask for an explanation of what may be wrong. The computer looks for spelling and usage errors and provides as much coaching and advice as it can. The reason for providing extensive coaching (as opposed to simply correcting the error) is that this program is designed to serve as a writing tutor. The goal is to help the writer avoid errors in the future, not just to correct them.

The program version I used worked with a base of rules for all but the semantic level of language. Dr. Arn is working in that area now. Using our previous example, it correctly identified that *alot* should be *a lot* and that *payed* should be *paid*. This latter error was detected by first catching the misspelling, noticing that the *-ed* ending may imply a past-tense form, and then exploring rules for forming past-tense forms.

In addition to providing an intelligent spelling checker, *English1* also examines the text for words that present usage problems for some writers, for example, *fewer/ less, seasonal/seasonable*, and so on. Consequently, this program can serve as a writing coach to a student who is working on an essay. *English1* does not make any changes in the author's text; it makes only observations and suggestions.

English 1 Is RAM Tough

In the form I saw, *English1* takes five megabytes of disk space. The new version that includes the ability to deal with semantic issues will be distributed on CD ROM since it is going to use about 500 megabytes. These sizes indicate just how complex natural languages are.

Even though Unisys is starting to sell the ICON in the U.S., it is targeted to classrooms, not businesses. Consequently, it is unlikely that we will have access to tools as sophisticated as *English1* for some time. On the other hand, the fact that *English1* exists at all proves that good proofreading tools can be created to run on microcomputers. Continued work in this area will result in programs that run on the machines we have on our desks.

Until then, I'll have to be content to do my grammatical analysis with the one computer that outstrips them all—the human brain. ©

Making Music With BASIC

I enjoy programming music on the computer. Since I am a former piano teacher, I have been able to use the computer for tutorial and drill programs. With the computer's exact tones and timing, I also use the computer to play written music so I can hear how it's supposed sound.

I would have liked to present a music program this month—however, sound commands are different among the various versions of BASIC. Each computer has its own special music features. So my advice is to read the manual that comes with your computer. Here, I'm going to review some of the more common BASIC sound commands. Unfortunately, because many early versions of BASIC do not support sound, these examples do not work on the Commodore 64 or Apple II family of computers.

Sounding Off

BEEP—available on the Amiga and IBM PC/PCjr—is probably the easiest command to use when you want to produce a sound. Some computers, such as the Apple II, Atari ST, and Commodore 128, use PRINT CHR$(7) to produce a bell or beep sound. The Atari eight-bit computers use PRINT CHR$(253). Whenever you want a short tone, such as a prompting tone, you can use BEEP. You may also use it for a game sound.

Most computers have a SOUND command, although the parameters vary from computer to computer. Here are some examples:

Amiga
SOUND *frequency,duration,volume,voice*

Atari
SOUND *voice,note,tone,loudness*

Atari ST
SOUND
voice,volume,note,octave,duration

Commodore 128
SOUND *voice,frequency,duration*

IBM PC and compatibles
SOUND *frequency,duration*

IBM PCjr with Cartridge BASIC
SOUND *frequency,duration,volume,voice*

The Commodore 128's SOUND command includes several optional parameters, such as waveform and pulse width, that are not listed above. In each case, parameters may be numeric constants or numeric variables, and there are certain limits for each parameter.

The Amiga and IBM SOUND statements use frequency numbers in cycles per second, or hertz (Hz), that produce a certain tone—frequencies with which many musicians are already familiar. For example, 440 Hz create what we hear as the note A.

Duration is a number often denoting the number of "clock ticks" that a particular sound lasts. You may need to experiment a little to see which numbers correspond to the timing you want. Because the eight-bit Atari SOUND command does not have a duration parameter, you must time the sound's duration yourself, using something like a FOR-NEXT loop. To turn the sound off with the Atari, use a SOUND command specifying a volume of 0.

Here is an example of how you can use SOUND to write music.

```
100 REM AMIGA,IBM PC/PCjr
    AND COMPATIBLES ◄
110 SOUND 523,20◄
120 SOUND 659,20◄
130 SOUND 784,20◄
◄
```

```
100 REM EIGHT-BIT ATARI
105 DURATION=90
110 FOR I=1 TO DURATION:S
    OUND 0,121,10,15:NEXT
    I
115 FOR I=1 TO 10:NEXT I
120 FOR I=1 TO DURATION:S
    OUND 0,96,10,15:NEXT
    I
125 FOR I=1 TO 10:NEXT I
130 FOR I=1 TO DURATION:S
    OUND 0,81,10,15:NEXT
    I
135 FOR I=1 TO 10:NEXT I
140 SOUND 0,0,0,0
```

```
100    REM ATARI ST - ST BASIC
110    SOUND 1,15,1,4,20
120    SOUND 1,15,5,4,20
130    SOUND 1,15,8,4,20
```

```
100 REM COMMODORE 128
110 SOUND 1, 4291, 20
120 SOUND 1, 5407, 20
130 SOUND 1, 6430, 20
```

The voice parameter is available on computers that have more than one voice—computers on which you can hear more than one note at a time. For example, here's how to play a chord using three voices:

```
100 REM AMIGA◄
110 SOUND 523,30,10,0◄
120 SOUND 659,30,20,1◄
130 SOUND 784,30,30,2◄
◄
```

```
100 REM EIGHT-BIT ATARI
105 FOR DURATION=1 TO 90
110 SOUND 0,121,10,15
120 SOUND 1,96,10,15
130 SOUND 2,81,10,15
140 NEXT DURATION
150 SOUND 0,0,0,0
160 SOUND 1,0,0,0
170 SOUND 2,0,0,0
```

```
100    REM ATARI ST - ST BASIC
110    SOUND 1,15,1,4,10
120    SOUND 2,15,5,4,20
130    SOUND 3,15,8,4,30
```

```
100 REM COMMODORE 128
110 SOUND 1, 4291, 60
120 SOUND 2, 5407, 60
130 SOUND 3, 6430, 60
```

```
100 REM PCjr WITH CARTRIDGE BAS
IC
105 SOUND ON
110 SOUND 523,30,10,0
120 SOUND 659,30,10,1
130 SOUND 784,30,10,2
```

Playing Music

The IBM PC/PCjr and compatibles have a PLAY command. (The Commodore 128 has a PLAY statement as well, but the options are quite different.) This command can be used to play simple melodies using the letter names of notes—*A, B, C, D, E, F,* and *G.* As you get more experienced, you will find that there are more options in the PLAY command.

There are certain default values for the PLAY command, but you can change them by adding options. You may add a sharp to a note by using the character # or +, and you may add a flat by using the – character. The octave number can be specified with O, such as O3, indicating the middle-C range. (Make sure you use the letter O and not the number 0.) You may also change the octave by using > to go to the next higher octave and < to go to the next lower one.

The command PLAY "MN" plays normal notes (a slight pause between notes). The string "ML" indicates legato, or notes smoothly connected; "MS" plays staccato notes.

You use L to indicate the length of notes. Actually, you indicate what fraction the note should be. For example, L1 is a whole note, L2 is a half note, L4 is a quarter note, and L8 is an eighth note. The L option affects all the notes that follow. L8C plays C as an eighth note. A shorter form is to simply follow the note name with a number and not use the L. For example, C8 also indicates C as an eighth note.

P is used for a pause or rest, and you specify the length just as in the L option. P4 is a quarter-note rest.

There is a default tempo, but if you want to go slower, use T followed by the number of quarter notes per minute, such as T88. You may change the volume by using V followed by a number.

I like to use variables in the PLAY statement. When you use a numeric variable, you need to use the equal sign before the variable name and a semicolon after the variable name. For example, PLAY "O=H; GEC" plays the notes G, E, and C in the octave specified by the variable *H*.

You may define a string variable to be a series of notes and then use that string in the PLAY command. To execute a string variable, use "X". For example, PLAY "XA$;" plays the notes defined in string A$. Note that there is a semicolon after the string name. Here is an example of PLAY using variables. The octave is set using a FOR-NEXT loop, and the actual notes are stored in S$.

```
100 REM IBM PC/PCjr AND COMPATI
BLES
110 S$="CDEFGAB"
120 FOR C=1 TO 6
130 PLAY "O=C; XS$;"
140 NEXT C
```

You may use string variables for storing musical phrases. For example, you might set up A$ for the first musical phrase, B$ for the second phrase, and C$ for the third phrase (have you ever heard of ABCA format?). This way, you can piece your music together. Here is a simple example:

```
100 REM IBM PC/PCjr AND COMPATI
BLES
110 A$="GGFE"
120 B$="EEFG"
130 C$="FEDC"
140 PLAY "XA$; XB$; XC$; XA$;"
```

©

www.commodore.ca

Computer Stew

Once each year I clean my cupboard of all the stuff that is good and worth talking about, but for one reason or another didn't make it into a column. December or January is a logical time to do this—year end and all that—but I'm late, as usual. So here, in the March issue, is a stew concocted from two of last year's left-over goodies.

The Bible As Data

Bryan Moore is an inventor and entrepreneur who developed a major piece of ophthalmological equipment. After he sold his invention to a large company, he turned his energies and talents to his passion—the study of the Bible.

The King James version of the Holy Bible was published in 1611, during the reign of James I of England. There are hundreds of editions of this book in hard-copy form and maybe a dozen computerized versions. But when Bryan Moore investigated biblical computer software, he was disappointed with what he found and thought he could do better. He did. The result is a product called *God Speed* distributed by Kingdom Age Software (3368 Governor Dr., Suite F-197, San Diego, CA 92122).

God Speed can locate any word or phrase in the 66 books of the Bible—both Old and New Testaments—in less than three seconds. This incredible speed—other software takes as long as 30 minutes—is due to a unique indexing and searching technique Moore devised just for this application. For example, you type "an eye for an eye," and in the twinkling of your own eye the screen displays the three locations where that is found. It doesn't just tell you that it's in Matthew 5:38, First Corinthians 15:52, and Exodus 21:24, but it shows you the phrase within the chapters and verses.

If you know your Bible, you are no doubt astonished and puzzled that "eye for an eye" appears in First Corinthians. And this points out one of the few weaknesses of *God Speed*. It searches and locates the words that you enter, but it doesn't require that they be consecutive within the verse. The screen shows that First Corinthians 15:52 says, ". . . in the twinkling of *an eye*, at the last trump: *for* the trumpet shall sound. . . ." The verse meets all requirements—it contains "eye," "for," and "an"—so *God Speed* reports it as a match.

Unlike some of the other software on the market, *God Speed* cannot perform sophisticated and lengthy searches based on logical ANDs, ORs, or NORs, but the program is so fast and simple to use that this is easy to forgive. You can manually toss out the verses that don't meet your criteria—and you might learn something in the process.

A menu bar across the top of the screen presents the program's four command choices: *Help*, *Find*, *Map*, and *Exit*. Press F for Find and then enter the word or words you want to look up. Press M for Map, and the program will show you a distribution for a word—"Love," for example, appears 310 times in 280 verses. You can jump immediately to any book, chapter, or verse by entering its name: GEN 1:1 takes you to the familiar "In the beginning. . . ."

God Speed is distributed on seven 5¼-inch disks and requires a hard disk with 2.5 million free bytes to install; the program uses 256K of RAM memory. If you have a color display, the program will display the search words in blue and the words of Jesus in red. A smaller, less-expensive version ($27) covers just the New Testament and can be run from a floppy-disk-based computer.

Apple And Pepsi

If you liked Lee Iacocca's book about his rescue of Chrysler—and enough of you did to keep it on the best-seller's list for several months—then you'll love John Sculley's book, *Odyssey*, about his adventures at Pepsi-Cola and Apple Computer.

Sculley began his business career as a fast-track executive at Pepsi. He was a shoo-in for the top job but was lured away to become President of Apple Computer by Steven Jobs, one of Apple's cofounders.

Sculley relates many wonderful inside stories about Pepsi and Apple. He tells how he once selected Coke over Pepsi in a blind-taste test; how he first met IBM's legendary Tom Watson, Jr., who swooped out of the clouds piloting his own stunt plane; how Big Blue sprays the apple trees at its corporate headquarters so they won't bear fruit and spoil the lawn.

More importantly, Sculley gives us an insider's view of the running of Apple Computer, how it stumbled and almost failed; and he tells of his painful rescue that cost him his friendship with Jobs. You'll be privy to the clashes in the board room and at the ad agency as Apple positioned itself to challenge IBM.

Even if you know nothing about computers, you'll enjoy this book—it's a good story, well told. Although at $21.95 (Harper & Row, 450 pages), you might want to wait for the paperback edition and use your savings to buy my book, *A Quick and Easy Guide to Dow Jones News/Retrieval*, from COMPUTE! Books ($10.95). It's not as long nor as entertaining as Mr. Sculley's, but it might help you locate the next Apple Computer. ©

Computers Of The Future

In the November 1987 issue of COMPUTE!, I challenged readers to imagine what computers would be like in the future. Many readers wrote me, and I'd like to share some of their images with you.

Wired To The Brain

The most popular image was of computers directly wired to the human brain. Georg Zimmer of Watertown, Wisconsin, wrote that "the interface will look like a headband and will plug into your computer." He says the computer won't need a monitor since its sounds and images can be piped directly into the brain. It won't need a keyboard since you'll just need to think of something, and it will go directly into the computer.

Electronic entertainment will be much more realistic. In an adventure game, for example, Zimmer says, "It would be like you were actually crawling across a smoky battlefield. You could smell the smoke, feel the pain, hear the explosions, and taste your rations (yecch!)." Georg goes on to predict that "this device means the end of TV. Why watch Magnum P. I. when you can *be* Magnum P. I.?"

A New Birthright

C. E. Deckard predicted that "direct neural hookups will allow the computer to process visual and aural information for blind and deaf people. It's only a short step from that to 'a computer in every head' as the birthright of every true American."

Deckard's line of reasoning produced some interesting predictions. He forecasted that teachers will become a thing of the past since education in the future will consist of periodic "curriculum downloads" from a school computer to students' neural and home computers; and since the greatest teachers known to man will be fab-

ricated as expert systems on silicon chips and available to everyone via broadcast or as a "whim purchase" at the local 7-11.

The publishing industry, too, according to Deckard, is in a precarious position. "CD-ROM, CD-I, and WORM are clanging the death knell for the publishing business as we know it," he wrote. "Who can afford to keep (as I do) yards and yards of shelf space for printed books, when the entire contents of my library could be handled on a very small number of optical disks and include search algorithms that are a bunch better than my weary gourd?" As traditional publishers struggle or disappear, a new "clandestine cottage industry" will emerge which will use hi-res scanners to translate printed books and magazines into computer format.

Beneath The Skin

According to reader Wally Frisbie, computers in the future will invade the cosmetics industry: "They'll clear up your skin. And that's just the beginning." Wally foresaw a computer-brain interface in the form of microchips "implanted just beneath the skin in the head."

He predicted that home computers in the future will monitor the electrical signals emitted from a person's body cells and be capable of curing any illness while a person slept by altering the electrical signals and returning them to a normal, healthy pattern.

The Sensual Computer

Lucy Valka, of Haslett, Michigan, imagined a writer and an artist's computer of the future. According to her, "Since the computer can translate and process information so well, let's give it eyes, ears, and speech. For eyes, the computer needs two color cameras. They need to have auto focus, and low

lux, maybe even infrared and heat sensitivity.

"For musical applications, we should be able to record in quadraphonic sound from an orchestra or marching band." Lucy called for pocket-size computers which writers could use to record their ideas while they were walking down the street. The little computers would upload their information into the writer's main computer which Valka said would have access to all the libraries in the world and which could "multitask when a writer has a leap of imagination."

Almost everyone mentioned that computers in the future should have artificial intelligence (AI), but no one agreed on what that should be. James Whitmore, called for RI (Real Intelligence) instead of AI.

Whitmore even came up with an entire system which he called DORIS, based on VOS (Voice Operating System). DORIS would get her input not from keyboards but from an amazing array of scanners—including sonic scanners, motion scanners, video scanners, impulse scanners, analytic scanners, and energy scanners. DORIS would not use a monitor. Instead she would use a variable beam generator to produce a "virtual screen" made of colorful 3-D holograms.

A New Attitude

Ramona Boger, an accountant in Spokane, Washington, didn't ask for her future computer to be intelligent, just forgiving. She wanted her computer to keep letting her make mistakes without complaining or mocking her. "Ten years from now," she writes, "I hope my computer will not change its attitude toward me, and will never tire of me. The programs, the speed, the versatility is bound to be exciting and productive, but the machine will still be my best friend." Ⓒ

The Trojan Wars

Telecomputing and the sharing of public domain data is a way of life for hundreds of thousands of computer users. Unfortunately, a handful of miscreants seem to want to share their misery with the rest of us as well. These antisocial folks write programs intended to wreak havoc on innocent users and their computer systems. Such programs, which masquerade as useful code, are often referred to as "Trojan horses" and have been around for years. As the popularity of modems and downloading public domain programs from computer bulletin boards and information services increases, the chances that the average user will encounter a Trojan Horse also increases.

A nondescript BASIC statement for the Commodore PET is generally credited as being the first Trojan program. The one-line program could actually damage the system's hardware, due to a design flaw in the PET. Since then, most Trojan programs have concentrated on corrupting data stored on disks.

One recent Trojan horse program was even widely rumored to have been created by a less-than-responsible copy-protection firm. The Trojan's description professed that it would remove a particular proprietary copy-protection scheme. Upon invocation, the program would prompt the user to insert the original copy-protected disk and then would erase all disk space that the system could access. After the erasure, a stern lecture on the evils of disobeying the terms of software licenses was displayed.

Computer Virus
"Viruses" are a new variant of Trojan Horse programs that are more difficult to detect and control. When virus programs are run, they typically inject a lethal bit of code into a file or disk location used by your computer's operating system. Viruses even have an "incubation period" which is controlled by a counter within the infecting code. During incubation, the virus will attempt to spread itself to any media that is inserted in or attached to the infected system.

The first microcomputer virus appeared late in 1987 and managed to infect thousands of Amiga computer systems. Thankfully, the Amiga virus was relatively benign. The Amiga virus's mission in life is to write itself to the hidden boot block on Amiga floppy disks, and, after infecting a number of disks, flash the message "Your computer has a virus" on the user's screen. Shortly after the virus was detected, a "cure" program was written that could detect and remove the virus from infected disks (for more on this topic, see "The Amiga Virus," elsewhere in this issue).

It didn't take long for a new, more virulent bug aimed at IBM PCs and compatibles to show up. The PC virus inserts itself onto one of DOS's disk files and goes into incubation. It waits until it replicates itself four times and then attempts to wipe out every file it can lay its binary fingers on.

How To Protect Yourself
So what is the average telecomputer to do? While some may choose to abstain completely from downloading or accepting public domain program files from acquaintances, such behavior is contrary to the norm of the computing community at large. If forswearing shared data is not compatible with your computing style, there are a number of precautions you can take to minimize the danger involved.

1. Know where you're getting your data. Restrict your downloading activity to commercial information services and bulletin boards that you know and trust. While dialing into new and far away systems is a lot of fun, it's often impossible to tell how closely system operators screen their files.

2. Try to be selective. Resist the temptation to immediately grab every new program that comes available. If you insist on being a downloading pioneer, be prepared for the occasional arrow in the back.

3. Take steps to protect your system. Make liberal use of write-protect tabs on your floppies. Hard disks usually cannot be physically write-protected but can often be logically protected by "anti-Trojan" programs that inhibit disk write activity or intercept updates to critical operating system files (ironically, one such "anti-Trojan" has turned out to be a cleverly disguised destructive program).

4. Backup is your best defense. Backup data regularly and keep multiple backups to insure that if you do encounter a virus, you can fall back to a clean operating environment. Keep the original disks from commercial publishers in pristine (write-protected) condition as a base to build upon if all else fails.

All warnings considered, it's important to keep the magnitude of the problem in perspective. In over ten years of regular telecomputing, using many different types of systems, I have encountered only a single Trojan horse program. While the experience of having one of my hard disks wiped clean in early 1986 was decidedly unpleasant, the would-be catastrophe was mitigated to an inconvenience by the availability of a system backup I had luckily made the day before.

Even so, this Achilles has traded his thongs for combat boots and remembers to beware of geeks bearing gifts. ©

AmigaView

Sheldon Leemon

Third-Party Hardware And Software

In my last column, I discussed the new Amiga products Commodore was showing at the fall COMDEX. There are, however, many new products on the horizon from third-party developers as well.

In the hardware category, NewTek is developing a product to rival Commodore's Professional Video Adapter for the Amiga 2000. Dubbed the Video Toaster, this product will, like the PVA, include a realtime color digitizer, genlock interface, and field store for freezing single frames from live video. In addition, NewTek's board will come with software for realtime digital special effects.

For example, NewTek demonstrated a live video image enclosed in a tumbling frame. They also showed a live video image being mapped around a sphere and other special effects like pixelization. At about $800, the Video Toaster should challenge the PVA when it appears sometime in the second quarter of 1988. NewTek is working on a number of new products, including a rumored Digi-View II— a $300 color frame grabber.

A Faster Amiga

Finally Technologies was showing off its Hurricane Accelerator board, a 68020 card for the 1000. This card plugs into the 68000 processor slot, and it provides a 14-MHz 68020 processor and 16-MHz 68881 co-processor. An optional board holds up to two megabytes of 32-bit dynamic RAM. Finally Technologies claims that the Hurricane runs all standard Amiga software packages from four to eight times normal speed. The board will be available for the 500 and 2000 as well and will cost approximately $1,000 including the 68020 and 68881 chips. The optional 32-bit memory will also be in the neighborhood of $1,000. Finally Technologies also

displayed the FourInOne expansion box for the 1000, which includes a DMA SCSI controller, 512K of fast RAM, and a clock/calendar for $700. Finally Technologies expects to sell versions of this box for the 500 and 2000, which would add to the list of hardware manufacturers making DMA SCSI controllers for the new machines. This list includes Phoenix, Pacific Peripherals, and Supra.

In another encouraging hardware development, Progressive Peripherals came out with the first third-party floppy drive for the Amiga. The ProDrive external drive is less expensive than the Amiga 1010, quieter, and quite small— only a bit larger than the bare drive itself. It also comes with a longer cable than the Amiga drive, making it more convenient for use with the 500. However, unlike the Amiga external drive, this one has no pass-through for additional drives, so it must be the last drive in the chain. This should present no problem for owners of the 500; the 500's power supply is really not meant to support more than one external drive.

The small power budget of the 500 means that owners of that machine should be careful about the peripherals they add. Low-power devices like the Digi-View digitizer and audio digitizers should work, but Commodore does not recommend use of higher-power boards, like internal memory cards which take their power from the 500. While some power supplies may have enough power to support these cards, Commodore does not guarantee that future supplies will.

To be on the safe side, owners of the 500 should look for peripherals that come with their own power supplies. One such add-on is the Byte Box, from Byte-by-Byte. This $300 (0K) memory-expansion module plugs into 500's external bus

connector and accepts 256K RAM chips for expansion from 512K to two megabytes of fast memory. Byte-by-Byte has made it very easy for the user to add memory chips to the Byte Box. They've even included a diagnostic program that displays the exact location of any chips that are bad or are improperly installed.

New Software

There's plenty of new software in store for Amiga owners as well. Progressive Peripherals was previewing *Superbase Professional*, a powerful new version of its popular database. It adds such features as keyboard macros, a database programming language, a forms editor, and a text editor with mail-merge. These new features mean that you can design a screen display that mixes data fields from separate, linked databases, and that mixes text with graphics on the same screen. Reports can be designed graphically onscreen. You can even design your own pull-down menus to replace the standard *Superbase* menus, and you can custom-tailor data-entry screens. *Superbase Professional* appears to be worth the wait.

Some familiar products have taken on a new look and/or new distributors. ITC, of *Textcraft* fame, had been working on *DeluxeWrite* for Electronic Arts but decided to market it as *KindWords* through its own subsidiary, the Disc Company. This package is like an enhanced version of *Textcraft Plus*, featuring a spell checker, graphics capabilities, and detailed graphics printer fonts.

Softwood has added a word processor to its *File* program, which will now be distributed by Brown-Wagh as *Write and File*. Brown-Wagh has also announced that they will soon release *Publisher Plus* to replace *Publisher 1000*. The new version will support laser printers and will not be copy-protected. ©

Philip I. Nelson

Snooping Into Your Disks

Admit it: Most of us treat computer disks like a magician's silk top hat. We toss a chunk of information into the hat, trusting that it will reappear when needed. When that time comes, we perform the right incantation, and—presto—changeo! The same chunk of information recombobulates itself as if by magic.

If it didn't happen so often—and so reliably—this process might seem quite remarkable. We create collections of information called files. A file can be anything from the text of this column, to a dazzling arcade game, to a digitized image of your Uncle Charlie whittling on his front porch. Those files are stored as electromagnetic patterns of ones and zeros on circular sheets of plastic. By reading those patterns from a whirling disk, the computer recreates the file in its original form.

The Veil Removed
This month's program lets you pull aside the veil and peer directly into a disk's individual sectors. It won't make you a GEMDOS expert overnight, but with this program and a decent reference book, you'll have the basic tools for learning a lot about how ST disk files are put together.

By the way, if you're queasy about tampering with disk files, you can be reassured that this program poses absolutely no danger. It can *read* exactly what's on any sector of a disk, but it can't change what's contained there. So you can use it without risking the loss of valuable information.

The program, DISKPEEK.BAS, is written in *GFA BASIC* with lots of comments to make it easy to translate into other languages. When you run the program, it prompts you to enter the track and sector (see below) you want to examine. If you enter an illegal value, the program puts up a warning and invites

Typical Sector Display

Track: 2   Sector: 1   Bytes 0-255

```
0000: 00 02 47 66 41 42 41 53 49 43 00 00 00 00 00 00   ..GfABASIC......
0010: 00 52 00 00 00 70 00 00 00 70 00 00 00 70 00 00   .R...p...p...p..
0020: 00 70 00 00 00 70 00 00 00 70 00 00 00 70 00 00   .p...p...p...p..
0030: 00 8E 00 00 00 8E 00 00 00 8E 00 00 00 8E 00 00   ................
0040: 00 8E 00 00 03 B8 00 00 07 EA 00 00 08 3E 00 00   .............>..
0050: 08 68 00 00 08 68 00 00 08 68 00 00 08 68 00 00   .h...h...h...h..
0060: 08 68 00 00 08 68 00 00 08 68 00 00 08 74 00 00   .h...h...h...t..
0070: 08 74 00 00 08 74 00 00 08 74 00 00 08 74 06 46   .t...t...t...t.F
0080: 4C 4F 50 52 44 06 44 52 56 4E 55 4D 07 53 49 44   LOPRD.DRVNUM.SID
0090: 45 4E 55 4D 08 4E 55 4D 53 45 43 54 53 03 44 55   ENUM.NUMSECTS.DU
00A0: 4D 06 54 52 4B 4E 55 4D 06 53 45 43 4E 55 4D 06   M.TRKNUM.SECNUM.
00B0: 52 45 53 55 4C 54 05 44 55 4D 4D 59 05 53 54 41   RESULT.DUMMY.STA
00C0: 52 54 06 4F 46 46 53 45 54 01 4A 01 4B 01 58 00   RT.OFFSET.J.K.X.
00D0: 04 42 4C 4F 4B 04 41 4C 52 54 06 4F 46 46 53 45   .BLOK.ALRT.OFFSE
00E0: 54 03 41 53 4B 03 42 59 54 01 58 02 4B 59 0A 4E   T.ASK.BYT.X.KY.N
00F0: 45 57 5F 53 45 43 54 4F 52 08 53 48 4F 5F 4C 4F   EW_SECTOR.SHO_LO
```

Press SPACE for bytes   0-255 of this sector
RETURN for bytes 256-512 of this sector
ESC   for a different track and sector
CTRL-SHIFT-ALT to quit

you to try again. A legal entry is rewarded by a sector display similar to the accompanying figure. Since a disk sector contains too many bytes to display on the screen at once, the program shows half of a sector at a time, letting you flip between the first and second halves by pressing a key. Press the Escape key to view a different sector.

The Basic Scheme
Knowing something about disk organization will help you understand what this program shows you. ST disks, like all others, are organized in circular, concentric tracks. There are 80 tracks in all, numbered 0–79. Each track is divided into nine 512-byte sectors, which are numbered 1–9. If you multiply these numbers (80 * 512 * 9) you come up with 368,640 (360K) bytes, the total amount of space available on a single-sided ST disk. A double-sided disk has information on both sides of the disk, and stores twice as much: 737,280 bytes, or exactly 720K. The space available for your use is somewhat smaller, since GEMDOS sets aside small areas on each disk to keep track of your files.

What you see on a given disk depends on what you've stored there. A *1ST Word* document file, for instance, contains formatting characters in addition to ordinary text. And a BASIC program looks very different on disk than it does when listed on the screen. The figure shows the very beginning of DISKPEEK.BAS as it is stored in Track 2, Sector 1 on a disk of mine. (On a different disk, of course, the same file might appear on a different track and sector.)

If you compare the figure to the program listing, you'll notice that the program's variable names appear in a bunch near the beginning of the file. With DISKPEEK.BAS, we can discover, at a glance, that GFA BASIC puts a variable table in its .BAS files, and we can examine that table in detail.

Where It All Begins
One place where every devoted disk snooper goes again and again is the directory, which begins at Track 1, Sector 3, Side 0. Each file on the disk has a 32-byte directory entry, which records its name and other information shown in the Disk Directory Entry table.

ST Disk Directory Entry

Byte number	Contents
0–10	Name and extension
11	Attribute
12–21	Reserved for future use
22–23	Time of creation
24–25	Date of creation
26–27	Beginning cluster
28–31	Size

In addition to existing files, the directory shows files that have been deleted. With a little experimentation, you ought to be able to deduce how GEMDOS marks a deleted file. (Hint: Look at the first character of the filename.)

Disk storage is a large topic, one that can easily fill a book. If you get interested in the subject, take a look at *ST Disk Drives: Inside and Out*, published by Abacus Books. I can't recommend this book wholeheartedly, since it does have its share of flaws. Nevertheless, it contains lots of information that's not easily available from any other source.

Disk Peek

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
' DISKPEEK.BAS  Examine ST disk sectors◄
'◄
floprd=8         ! XBIOS routine 8 (Floprd) reads
 a disk sector◄
drvnum=0         ! Drive 0 = A:, 1 = B:◄
sidenum=0        ! Double-sided disks have sides
0 and 1◄
numsects=1       ! Number of sectors to read with
 each Floprd call◄
dum=0            ! Dummy parameter for Floprd cal
l◄
'◄
new_sector:      ! Read the specified track and s
ector◄
CLS◄
blok$=STRING$(512,0)   ! 512-byte space for sec
tor contents◄
INPUT "Track number  (0-79)";trknum  ! Track w
e want to read◄
INPUT "Sector number (1- 9)";secnum  ! Sector
we want to read◄
' Flag illegal track/sector choice◄
IF trknum<0 OR trknum>79 OR secnum<1 OR secnum
>9 THEN◄
  alrt$="Illegal track ("+STR$(trknum)+")!or s
  ector ("+STR$(secnum)+")"◄
  ALERT 1,alrt$,1,"Oops",dum◄
  GOTO new_sector◄
ENDIF◄
' Call the XBIOS routine Floprd to read a disk
 sector.◄
result=XBIOS(floprd,L:VARPTR(blok$),L:dum,drvn
um,secnum,trknum,sidenum,numsects)◄
' Check for a disk error. Return code of zero
means no error occurred.◄
IF result<>0 THEN◄
  ALERT 1,"Disk error!!Gotta go.",1,"OK",dummy
  ◄
  END◄
ENDIF◄
start=0◄
'◄
sho_loop:  ! Display sector contents◄
CLS◄
PRINT SPACE$(10);"Track: ";trknum;SPACE$(5);"S
ector: ";secnum;◄
PRINT SPACE$(5);"Bytes ";start;"-";start+255◄
offset=start◄
PRINT◄
```

```
FOR j=0 TO 15◄
  offset$=HEX$(offset)◄
  WHILE LEN(offset$)<4◄
    offset$="0"+offset$◄
  WEND◄
  PRINT offset$;":";" ";◄
  offset=offset+16◄
  ask$=""◄
  byt$=""◄
  FOR k=0 TO 15◄
    x=PEEK(VARPTR(blok$)+start+(16*j)+k)◄
    byt$=byt$+CHR$(x)◄
    IF x<16 THEN◄
      ask$=ask$+"0"◄
    ENDIF◄
    ask$=ask$+HEX$(x)+" "◄
  NEXT k◄
  PRINT ask$;◄
  FOR k=1 TO 16◄
    x$=MID$(byt$,k,1)◄
    IF x$<CHR$(33) OR x$>"z" THEN◄
      PRINT ".";◄
    ELSE◄
      PRINT x$;◄
    ENDIF◄
  NEXT k◄
  PRINT◄
NEXT j◄
PRINT◄
PRINT "Press SPACE   for bytes    0-255 of this
 sector"◄
PRINT "          RETURN for bytes 256-512 of this
 sector"◄
PRINT "          ESC    for a different track and
 sector"◄
PRINT "          CTRL-SHIFT-ALT to quit"◄
get_a_key:◄
ky$=""◄
WHILE ky$=""  ! Wait for a keypress.◄
  ky$=INKEY$◄
WEND◄
IF ky$=CHR$(27) THEN  ! You pressed ESC.◄
  GOTO new_sector      ! Choose new track/secto
  r.◄
ENDIF◄
IF ky$=CHR$(13) AND start=0 THEN◄
  start=256  ! View bytes 256-511 of this sect
  or◄
  GOTO sho_loop◄
ENDIF◄
IF ky$=CHR$(32) AND start=256 THEN◄
  start=0  ! View bytes 0-255 of this sector◄
  GOTO sho_loop◄
ENDIF◄
GOTO get_a_key◄                          ©
```

INSIGHT: Atari

Bill Wilkinson

More On Structure

In last month's column, I discussed structured data types. In the program listing, I then used a large string and its substrings to simulate an array of records. This month, I will continue my efforts to help you write more structured programs, but first I'll make some general comments on Atari BASIC strings.

I think that last month's program demonstrates that long strings can be just as powerful as string arrays. I would expect that program to operate at least as quickly as an equivalent Microsoft BASIC program using string arrays, because the typical Microsoft implementation goes through a lot of overhead generating and reclaiming dynamic strings. In that example, we were inserting new records into our string structure. If we had been deleting records, Atari BASIC really would have shone. For example, suppose we have a string filled with 50-character pseudorecords. To delete the third such record, we could simply do this:

RECORD$(101) = RECORD$(151)

Presto! All records are moved up one spot, and the third one is gone.

A small sidetrack: Unfortunately, one failing of Atari BASIC is that it has no built-in way to conveniently save and restore such long strings to and from disk. The most common output method is to PRINT# such a string, but that doesn't help a lot, since INPUT# is limited to no more than 255 bytes per line. I have seen many users resort to using a FOR/NEXT loop to PRINT# or INPUT# the individual pseudorecords one at a time. That works, but it certainly slows down disk I/O speed. In *BASIC XL* and *BASIC XE*, we added a pair of special statements for this purpose (BPUT# and BGET#, where the *B* stands for *Block* or *Buffer*), but you can accomplish the same thing in Atari BASIC with a pair of fairly

short USR assembly language subroutines. (These routines have been published several times, and I won't repeat them this month.)

Out Of Sorts

To continue my discussion of how to achieve structured data features in Atari BASIC, I call your attention to Program 1. Study it, type it in, and run it. It is a fairly clumsy but working record-sort routine. That is, it sorts the kinds of pseudorecord strings that we also used in last month's program. (Incidentally, I have used the worst of all possible sort algorithms: the bubble sort. Please, if you are serious about sorting your data, learn a couple of other methods, such as the heap sort, Shell sort, and quicksort. Why do I use the bubble sort? Because it is the smallest and easiest to demonstrate. Or maybe because I'm just lazy.)

As you study that listing, pay attention to lines 230–260, where the tests and swaps necessary to any sort routine are made. Because I purposely organized the data in my array of pseudorecords in the worst possible way (for a bubble sort, at least), the IF test will never branch around the swap, and we will make more than 4900 of these string swaps. Surely there must be a better way.

Pointing The Finger

Time to introduce another concept from the structured languages. In *any* computer language, moving blocks of data around (whether records, pictures, disk blocks, or whatever) is time-consuming. So most programs don't actually move the data. Instead, they move pointers to the data.

What is a pointer? Quite simply, a pointer is a variable that contains the address of another variable. Atari BASIC allows only one *explicit* kind of pointer—the ADR function that gives the address of a string. And, indeed,

many programmers use ADR as a pointer when they pass the address of a string to an assembly language subroutine. (Imagine having to pass the bytes of a string through a series of POKEs.)

But there is another, hidden kind of pointer in almost any computer language: array and string indices. As an example, if the record data we are working on is *always* within a particular string, for example, then we need only know the relative position of a given record within the string in order to obtain its information. We most likely do not care about the actual physical memory address of the data.

Merge the lines shown in Program 2 to those already in place in Program 1 (deleting the three lines noted) and study the results of using implicit pointers. There is (rather obviously) little difference between the two programs. Instead of swapping actual substring pseudorecords, we are now swapping only the indices into the master string. When you run this second version, you should notice a speed improvement of almost 2:1. (I got 70.8 seconds versus 135.3 seconds, but that was done using the FAST mode of *BASIC XL*. Your times will likely be slower.)

As clever as this trick is, it does not answer all of a programmer's needs. Suppose, for example, that the data to be sorted resides in two or three different arrays. The logistics in BASIC get complicated. In a language with more data structuring capabilities, where a pointer can point to a given record type no matter where the record might be, such a division of data would probably make virtually no difference on program speed.

Enough for this month. Next month, we will go back to the acrostics puzzle of the December issue, since it generated more mail

for me than any topic in recent months. If I have managed to convince you that records and pointers are valuable tools, wait until you see my proposed solution to the acrostics problem!

Program 1

```
BD 100 DIM TEST$(10000),TEMP
       $(100),BLANK$(100)
NK 110 BLANK$=" ":BLANK$(100
       )=" ":BLANK$(2)=BLANK
       $
NI 120 FOR R1=0 TO 99:RPTR=R
       1*100+1
GO 130 TEST$(RPTR)="THIS IS
       RECORD NUMBER "
MP 140 TEMP$=STR$(199-R1):TE
       MP$(4)=BLANK$
DM 150 TEST$(RPTR+22)=TEMP$(
       2)
FJ 160 NEXT R1
IF 170 REM OUT OF ORDER...SO
       RT THEM
LF 180 POKE 20,0:POKE 19,0
BP 190 FOR R1=98 TO 0 STEP -
       1
JA 200 FOR R2=0 TO R1
JC 210 PTR1=R2*100+1
NM 220 PTR2=PTR1+100
MC 230 IF TEST$(PTR1,PTR1+99
       )<TEST$(PTR2,PTR2+99)
       THEN 270
PJ 240 TEMP$=TEST$(PTR1,PTR1
       +99)
GO 250 TEST$(PTR1,PTR1+99)=T
       EST$(PTR2,PTR2+99)
PN 260 TEST$(PTR2,PTR2+99)=T
       EMP$
FM 270 NEXT R2
NA 280 PRINT "*";
FN 290 NEXT R1
CA 300 PRINT
GF 310 TICK=PEEK(20):TOCK=PE
       EK(19):IF TICK<>PEEK(
       20) THEN 310
EP 320 TIME=TICK+256*TOCK
GC 330 PRINT "THAT TOOK ";TI
       ME/60;" SECONDS"
CE 340 PRINT "HIT RETURN TO
       SEE LIST ";
IC 350 INPUT TEMP$
NO 360 FOR R1=0 TO 99:RPTR=R
       1*100+1
DF 370 PRINT TEST$(RPTR,RPTR
       +99)
FN 380 NEXT R1
```

Program 2

```
CJ 105 DIM RPT(99)
OH 125 RPT(R1)=RPTR
MC 210 PTR1=RPT(R2)
CA 220 PTR2=RPT(R2+1)
CA 230 IF TEST$(PTR1,PTR1+99
       )>TEST$(PTR2,PTR2+99)
       THEN RPT(R2)=PTR2:RP
       T(R2+1)=PTR1
CN 240 REM DELETE
PE 250 REM THESE
EL 260 REM 3 LINES!
AO 360 FOR R1=0 TO 99:RPTR=R
       PT(R1)
```

Cursor Plus

Emmanuel Gendrano and Greg Knauss

Add even more power to the Atari 400, 800, XL, and XE editor device. Compatible with most environments and programs.

Atari computers have the most powerful screen editing features of any eight-bit computer. "Cursor Plus" extends the editor even further, adding functions to move the cursor by one word, change the case of an entire screen line, delete the remainder of the line, and more.

Cursor Plus is compatible with most programs that use the E: editor device. This includes Atari BASIC, *BASIC XL*, *MAC/65*, and other environments. It *does not* include programs that use their own editors, such as *Action!*, *Atariwriter*, and Atari's *MEDIT* editor.

Typing It In
Cursor Plus works on all Atari eight-bit computers. Type it in and save it to tape or disk. The program is written in BASIC, but it creates a machine language program when it is run.

After saving the program, type RUN. You'll be asked if you want to save a copy or install it in memory. If you have a disk system, you must choose the option to save a copy. The program asks for a filename with which to save Cursor Plus. Be sure to use a name that's different from the one you used to save the creator program. After the machine language program has been written to disk, you can install Cursor Plus by going to DOS and using the L option to load it. Alternatively, you can name the file AUTORUN.SYS, causing the program to install automatically whenever you boot up the computer.

If you have a tape system, choose the memory option.

The New Editor
When you're using Cursor Plus, all of the regular cursor movement and editing capabilities are still in effect. In addition, Cursor Plus adds the following controls:

• **Control-ESC** Go to the end of the line. This command moves the cursor to the end of the physical screen line (not the logical line).

• **Control-Shift-RETURN** Go to the beginning of the line. This command moves the cursor to the beginning of the physical screen line. It does not enter the line, as a regular RETURN would.

• **Control-Shift-(minus sign)** Go to previous word. This command moves the cursor to the word immediately to the left of the current cursor position.

• **Control-Shift-(equals sign)** Go to next word. This command moves the cursor to the word immediately to the right of the current cursor.

• **Control-Shift-SPACE** Delete to end of physical line. This command deletes everything after the current cursor position to the end of the line.

• **Control-Shift-Caps** Change case of the rest of the physical line. This command begins at the current cursor position and changes all lowercase letters to uppercase and all uppercase letters to lowercase. Single words can be changed by cursoring to the beginning of the word, changing the case of the rest of the line, cursoring to the next word, and changing the case again.

• **Control-Shift-Inverse** Change regular text to inverted text and inverted text to regular text. This command is similar in operation to the change case command. It begins at the current cursor position and changes the high bit of all of the text to the end of the physical line, thus, changing inverted letters (blue on white) to normal letters

(white on blue), and back.

• **Shift-Inverse** Change case of one letter. This command is similar to Control-Shift-Caps, except that it changes the case of only one character. This keypress also advances the cursor by one character, so you can repeatedly press Shift-Inverse to change the case of a range of characters.

• **Control-Inverse** Change one character from normal text to inverted text, or back. This command is similar to Control-Shift-Inverse; but it changes the high bit of only one character. Like Shift-Inverse, it advances the cursor, making it easy to change a range of characters.

• **Control-Clear** Move cursor to the home position (top left of screen). This command normally clears the screen. In Cursor Plus, however, it simply moves the cursor to home without clearing the screen. Use Shift-Clear to clear the screen.

• **Control-Shift-Clear** This command places the cursor in the middle of the screen.

Cursor Plus survives SYSTEM RESET. Disable it by turning off the computer. This is the only way to recover the memory used by the program.

Cursor Plus Creator

```
MH 1000 REM CURSOR PLUS
IM 1010 REM COPYRIGHT 1988 C
        OMPUTE! PUB.
LB 1020 REM ALL RIGHTS RESER
        VED
AI 1030 REM D:CURSOR.BAS
KB 1040 DIM ID$(15):OPEN #1,
        12,0,"E:"
BJ 1050 POKE 710,0:POKE 82,1
HP 1060 ? "{CLEAR}{2 DOWN}
        {TAB}{6 SPACES}CURSO
        R PLUS{7 SPACES}
        {DOWN}"
PC 1070 ? "{TAB}COPYRIGHT 19
        88 COMPUTE! {3 DOWN}
        "
PA 1080 ? " OPTION  to cre
        ate object file
        {DOWN}"
IH 1090 ? " SELECT  to ins
        tall CURSOR+{3 DOWN}
        "
MP 1100 CN=PEEK(53279):IF CN
        =7 THEN 1100
HM 1110 POKE 53279,0:REM cli
        ck
AG 1120 IF CN=3 THEN 1150:RE
        M make
GP 1130 IF CN=5 THEN 1240:RE
        M do it now
MB 1140 GOTO 1100
FP 1150 REM CREATE FILE
PH 1160 ? "Enter DEVICE:FILE
        SPEC >";
AD 1170 INPUT #1,ID$:?
CM 1180 CLOSE #3:TRAP 1320:O
        PEN #3,8,0,ID$
OP 1190 RESTORE
CM 1200 READ D:IF D<>-1 THEN
        PUT #3,D:GOTO 1200
HD 1210 CLOSE #3:? "DONE"
HK 1220 GOTO 1350:REM Wait
JN 1230 END
BD 1240 REM INSTALL IT now
KE 1250 ? :IF PEEK(145)>=96
        THEN ? "Memory confl
        ict!":GOTO 1350
BE 1260 ? "LOADING AT... ";:
        RESTORE :READ L,D
BD 1270 READ L,D:X=L+D*256:A
        =X:? X:GOTO 1290
EG 1280 READ L,D:A=L+D*256:I
        F L=-1 THEN L=USR(X+
        3)
NP 1290 READ L,D:B=L+D*256
NO 1300 FOR L=A TO B:READ D:
        POKE L,D:NEXT L
EC 1310 ? :GOTO 1280
ML 1320 REM ERROR TRAP
KB 1330 IF PEEK(195)=130 THE
        N ? "BAD DEVICE":GOT
        O 1350
PK 1340 ? "****ERROR - ";PEE
        K(195)
CC 1350 REM WAIT
DG 1360 ? :? "Press RETURN t
        o continue ";
IM 1370 INPUT #1,ID$
OP 1380 GOTO 1060:REM restar
        t
PH 1390 REM DEC DATA
DL 1400 DATA 255,255,0,96,25
        1,96,76,201,97,104,1
        69,160,72,169,0,72,1
        33,8,240,242,67,85,8
        2,83
ME 1410 DATA 79,82,43.32,32,
        86,69,82,83,73,79,78
        ,32,51,46,48,32,32,5
        6,45,49,45,56,55
HJ 1420 DATA 155,138,72,152,
        72,160,10,173,9,210,
        217,165,97,240,10,13
        6,16,248,104,168,104
        ,170,76,190
LB 1430 DATA 255,165,88,72,1
        65,89,72,166,84,240,
        16,24,169,40,101,88,
        133,88,169,0,101,89,
        133,89
LA 1440 DATA 202,208,240,240
        ,61,169,23,164,82,13
        3,84,132,85,169,143,
        208,96,160,20,169,10
        ,208,242,164
OB 1450 DATA 83,177,88,41,12
        7,208,69,136,177,88,
        41,127,208,71,196,82
        ,240,2,176,243,164,8
        3,16,61
DA 1460 DATA 169,0,133,93,16
        0,39,196,85,144,5,14
        5,88,136,16,247,56,1
        76,35,152,240,192,13
        6,240,201
CG 1470 DATA 136,240,204,136
        ,240,222,136,240,223
        ,136,240,44,136,240,
        43,136,240,70,136,24
        0,80,136,240,56
JF 1480 DATA 208,122,164,85,
        136,196,82,176,4,176
        ,172,164,83,132,85,1
        69,135,141,252,2,104
        ,133,89,104
KF 1490 DATA 133,88,104,168,
        104,170,104,64,162,1
        28,165,93,73,128,133
ML 1500 DATA 128,145,88,136,
        196,85,240,202,240,7
        2,176,241,144,196,14
        4,140,164,85,252,96,
        247,97,16,11
LA 1510 DATA 173,9,210,201,1
        03,240,196,208,181,1
        64,83,177,88,170,41,
        63,201,33,144,13,144
        ,179,144,168
NJ 1520 DATA 201,59,176,5,13
        8,73,64,145,88,196,8
        5,240,3,136,16,227,1
        38,209,88,240,211,16
        5,93,73
DM 1530 DATA 64,133,93,56,17
        6,202,176,141,208,53
        ,164,83,16,139,164,8
        5,136,48,247,177,88,
        41,63,201
EI 1540 DATA 16,176,17,136,4
        8,236,177,88,41,63,2
        01,16,176,6,196,82,1
        76,241,144,158,136,4
        8,219,177
LK 1550 DATA 88,41,63,201,16
        ,144,173,192,0,16,24
        1,48,205,144,167,164
        ,85,200,177,88,41,63
        ,201,16
LE 1560 DATA 144,18,201,26,1
        44,8,201,32,144,10,2
        01,59,176,6,196,83,1
        44,231,176,25,200,17
        7,88,41
JM 1570 DATA 63,201,16,144,1
        2,201,26,144,212,201
        ,32,144,4,201,59,144
        ,204,196,83,144,231,
        176,143
DE 1580 DATA 182,246,156,204
        ,225,167,231,103,252
        ,206,207
KL 1581 REM HOME, MIDHOME, END
        LINE, STARTLINE, ZEROL
        INE, INVERSE, INVERSEL
        INE, CAPS, CAPSLINE, WO
        RDLEFT, WORDRIGHT
GD 1585 DATA 169,0,141,8,2,1
        69,0,141,9,2,169,0,1
        33,2
HN 1590 DATA 169,0,133,3,165
        ,9,9,2,133,9,96,173,
        8,2,141,65,96,173,9,
        2,141,66,96,173
HH 1600 DATA 231,2,133,0,141
        ,177,97,173,232,2,13
        3,1,141,182,97,24,16
        5,0,105,133,141,187,
        97,165
LI 1610 DATA 1,105,1,141,191
        ,97,24,165,0,105,248
        ,97,67,98,122,141,53
        ,96,165,1,105,1,141,
        54
AJ 1620 DATA 96,160,0,185,43
        ,96,145,0,136,208,24
        8,238,7,98,230,1,173
        ,7,98,201,98,144,236
        ,24
ED 1630 DATA 169,160,109,231
        ,2,141,231,2,169,1,1
        09,232,2,141,232,2,1
        69,14,141,68,3,169,9
        6,141
BL 1640 DATA 69,3,141,73,3,1
        69,9,141,66,3,162,0,
        32,86,228,76,176,97,
        224,2,225,2,201,97,-
        1,-1
FI 1650 END :REM END    ©
```

Math Graphics For The IBM PC

Jay Seaver

If you think that math is boring, then you have a pleasant surprise in store. With these two programs you can either custom design your own dazzling Math Graphics or you can sit back, let the computer do all the work, and enjoy a beautiful graphics show. A color/graphics adapter or equivalent hardware is required, along with BASICA for the PC or GW-BASIC for compatibles.

While your computer can't create *pictures* without a paint program, a graphic *pattern*—which can be the result of a mathematical formula—is often surprisingly easy to create. In fact, if you have a flexible formula, you can create thousands of images by simply varying the formula.

One program that has done this well is "Math Graphics," versions of which have been published in COMPUTE! for three different computers: Amiga, Apple II series, and Atari ST. Because of the beautiful graphics generated by this program, I decided to design a PC version, but with a major enhancement, "Math Graphics Construction Set."

Getting Started
Since both "Demo" (Program 1) and "Construction Set" (Program 2) are written entirely in BASIC, just type each one in and save a copy of each to disk. To see the demonstration program, load it, type RUN, sit back, and enjoy.

If the demonstration program piques your interest in creating graphics of your own, you'll want

to explore the Construction Set program. When you first run the Construction Set, you'll see a large menu. Understanding this menu is the key to using the program.

Constructing Math Graphics
When you first run the Construction Set, the menu's 19 numbered options have values assigned to them. The prompt at the bottom of the screen asks you to *Enter number to modify, 20 to RUN, 21 to QUIT*. By

Several complex geometric designs are showcased in the "Math Graphics" demo program.

The "Math Graphics Construction Set" offers a menu that makes it easy to change parameters for altering images.

typing the number of one of the options and then, when prompted, the new value you wish to assign it, you change the way the final picture is formed. To see the default Math Graphic, choose option 20 from the menu. If you want to return to the menu before the Math Graphic is finished forming, simply press any key. Let's take a look at each option.

Options 1–4 are the R values, which define the shape of the graphic. Note that R1, R2, R3, and R4 modify X1, Y1, X2, and Y2, respectively. To get an idea of how to use the construction set, change the value of Option 1 to 20 and then view the graphic.

Options 5 and 6 define the limits of the theta loop in terms of *pi*. If the difference between these values is exactly 2, the picture will be drawn once and won't be retraced. If the difference is more than 2, parts or all of the picture will be retraced and will change color. If the difference is less than 2, an incomplete drawing will result.

Option 7's value is divided into *pi* to determine how quickly the theta loop proceeds from one end to the other. If Option 6 is less than Option 5, Option 7 should be a negative number. To see how this parameter changes the image, try changing the value of Option 7 to 8 and view the graphic.

Options 8–11 are the offsets for X1, Y1, X2, and Y2, respectively. Because each value is divided into *pi*, the lower the number, the larger the offset. The default for these options is 10,000, so any offset will be

imperceptible.

Options 12 and 13 are the start and end of a *second* theta loop. Use these options like 5 and 6 above.

Option 14 is the increment for the second theta loop. Use it just like Option 7.

Options 15–18 decide which theta loops X1, Y1, X2, and Y2 will use. If you don't want to use the second theta loop at all, set all of these to 1. Changing these values can produce dramatic, exciting changes in your graphics. For example, exit the program by choosing option 21 from the menu and type RUN (or Press F2), so you'll start with the Construction Set's default values in place. Now change Option 13 to 6, so you'll have a second theta-loop value, and change both Options 15 and 18 to 2, so they each use the second loop. Now you can view the graphic.

Option 19 lets you choose which color palette to use. It accepts a number from 0 to 3. See the table for the colors associated with each number.

Color Palette Numbers

Palette	Colors
0	green, maroon, brown
1	pale green, purple, white
2	green, pink, brown
3	pale green, red, white

Option 20 displays the current graphic. Option 21 returns you to BASIC.

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: Math Graphics Demo

```
DO 100  KEY OFF
OI 110  ' Mathgraphics For the IB
        M PC
HL 120  ' Copyright 1988 COMPUTE!
        Publications, Inc. -- Al
        l Rights Reserved
OJ 130  CLS:WIDTH 40:LOCATE 1,10:
        PRINT"MathGraphics Demo
GD 140  LOCATE 3, 1:PRINT"Copyrig
        ht 1988 COMPUTE! Publ., I
        nc."
LK 150  LOCATE 4,9:PRINT"All Righ
        ts Reserved"
QJ 160  LOCATE 22,8:PRINT"Press A
        ny Key To Begin"
EG 170  K$=INKEY$:IF K$="" THEN 1
        70
CB 180  GOSUB 2320
MO 190  ' Module #1 - RightOvals
IJ 200  R1=150:R2=25:R3=25:R4=85
IN 210  INC=PI/64
PL 220  FOR THETA=0 TO PI*4 STEP
        INC
CO 230    X1=FNPOLARX(R1, THETA)
FJ 240    Y1=FNPOLARY(R2, THETA)
JE 250    X2=FNPOLARX(R3, THETA+PI)
MP 260    Y2=FNPOLARY(R4, THETA+PI)
LD 270    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
GI 280    LINE (X1,Y1)-(X2,Y2),KOL
          OR
OD 290    NEXT THETA
GB 300  GOSUB 2380
FO 310  'Module #2 - SideOvals
OB 320  R1=150:R2=35:R3=65:R4=85
LN 330  INC=PI/64:OFFSET=PI/3
JD 340  FOR THETA=1 TO 6*PI STEP
        INC
CD 350    X1=FNPOLARX(R1, THETA)
FO 360    Y1=FNPOLARY(R2, THETA)
FB 370    X2=FNPOLARX(R3, THETA+OFF
          SET)
IP 380    Y2=FNPOLARY(R4, THETA)
MI 390    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
FK 400    LINE (X1,Y1)-(X2,Y2),KOL
          OR
NF 410    NEXT THETA
GG 420  GOSUB 2380
JC 430  'Module #3 - TwistedBand
OG 440  R1=150:R2=35:R3=65:R4=85
LC 450  INC=PI/64:OFFSET=PI/3
KI 460  FOR THETA=1 TO 6*PI STEP
        INC
DI 470    X1=FNPOLARX(R1, THETA)
FD 480    Y1=FNPOLARY(R2, THETA)
GJ 490    X2=FNPOLARX(R3, THETA)
HO 500    Y2=FNPOLARY(R4, THETA+OFF
          SET)
LK 510    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
FP 520    LINE (X1,Y1)-(X2,Y2),KOL
          OR
GF 530    NEXT THETA
HL 540  GOSUB 2380
BF 550  'Module #4 - TiltedOvals
PL 560  R1=150:R2=35:R3=65:R4=85
LH 570  INC=PI/64:OFFSET=PI/3
KN 580  FOR THETA=1 TO 6*PI STEP
        INC
PC 590    X1=FNPOLARX(R1, THETA+OFFS
          ET)
QM 600    Y1=FNPOLARY(R2, THETA)
AC 610    X2=FNPOLARX(R3, THETA)
DK 620    Y2=FNPOLARY(R4, THETA)
CD 630    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
HK 640    LINE (X1,Y1)-(X2,Y2),KOLO
          R
HK 650    NEXT THETA
HA 660  GOSUB 2380
DL 670  'Module #5 - StretchedOva
        ls
PA 680  R1=150:R2=35:R3=65:R4=85
MM 690  INC=PI/64:OFFSET=PI/3
JP 700  FOR THETA=1 TO 6*PI STEP
        INC
CP 710    X1=FNPOLARX(R1, THETA)
EH 720    Y1=FNPOLARY(R2, THETA+OFF
          SET)
FA 730    X2=FNPOLARX(R3, THETA)
IL 740    Y2=FNPOLARY(R4, THETA)
LE 750    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
GJ 760    LINE (X1,Y1)-(X2,Y2),KOL
          OR
OE 770    NEXT THETA
HF 780  GOSUB 2380
HO 790  'Module #6 - MultiLobes
DL 800  R1=100
EN 810  INC=PI/128
IB 820  LOBES=4
GP 830  FOR THETA=0 TO 4*PI STEP
        INC
FK 840    R2=R1*SIN(LOBES*THETA)
EL 850    X1=FNPOLARX(R2, THETA)
FD 860    Y1=FNPOLARY(R2, THETA)
MJ 870    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
PP 880    LINE (XCENTER,YCENTER)-(
          X1,Y1),KOLOR
PJ 890    NEXT THETA
GH 900  GOSUB 2380
OK 910  'Module #7 - SpiralCone
QO 920  R1=100:R2=85
AG 930  INC=PI/160
IL 940  LOBES=3
GE 950  FOR THETA=0 TO 4*PI STEP
        INC
MF 960    X1=FNPOLARX(R1, THETA*LOBE
          S)
FG 970    Y1=FNPOLARY(R2, THETA)
MM 980    KOLOR=KOLOR+1:IF KOLOR=4
          THEN KOLOR=1
PC 990    LINE (XCENTER,YCENTER)-(
          X1,Y1),KOLOR
KF 1000   NEXT THETA
LP 1010 GOSUB 2380
DI 1020 'Module #8 - SideSpiralC
        one
GN 1030 R1=130:R2=80
GK 1040 INC=PI/160
EP 1050 LOBES=3
GO 1060 FOR THETA=0 TO 4*PI STEP
        INC
NC 1070   X1=FNPOLARX(R1, THETA)
CP 1080   Y1=FNPOLARY(R2, THETA*LO
          BES)
LL 1090   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
EC 1100   LINE (XCENTER,YCENTER)-
          (X1,Y1),KOLOR
LK 1110   NEXT THETA
LE 1120 GOSUB 2380
BN 1130 'Module #9 - Circles
KD 1140 R1=115:R2=85:R3=40:R4=45
HJ 1150 INC1=PI/3:INC2=PI/20
PB 1160 FOR THETA1=0 TO PI*2 STE
        P INC1
GC 1170   FOR THETA2=0 TO PI*2 ST
          EP INC2
QO 1180     X1=FNPOLARX(R1, THETA2)
DA 1190     Y1=FNPOLARY(R2, THETA2)
QP 1200     X2=FNPOLARX(R3, THETA1)
DB 1210     Y2=FNPOLARY(R4, THETA1)
GG 1220     KOLOR=KOLOR+1:IF KOLOR
            =4 THEN KOLOR=1
NI 1230     LINE (X1,Y1)-(X2,Y2),K
            OLOR
PO 1240     NEXT THETA2
QJ 1250     NEXT THETA1
MC 1260 GOSUB 2380
KC 1270 'Module #10 - Spikes
LB 1280 R1=115:R2=85:R3=40:R4=45
CH 1290 INC1=PI/3:INC2=PI/18
FF 1300 FOR THETA1=0 TO 2*PI STE
        P INC1
MG 1310   FOR THETA2=0 TO 2*PI ST
          EP INC2
PA 1320     X1=FNPOLARX(R1, THETA2)
AG 1330     Y1=FNPOLARY(R2, THETA1)
BN 1340     X2=FNPOLARX(R3, THETA1)
GL 1350     Y2=FNPOLARY(R4, THETA2)
HE 1360     KOLOR=KOLOR+1:IF KOLOR
            =4 THEN KOLOR=1
FE 1370     LINE(X1,Y1)-(X2,Y2),KO
            LOR
QM 1380     NEXT THETA2
EB 1390 NEXT THETA1
FB 1400 ' change window size to
        equal that of AppleII's
PC 1410 WINDOW SCREEN (0,0)-(279
        ,191):XCENTER=140:YCENTE
        R=96
MK 1420 GOSUB 2380
BI 1430 'Module #11 - A Ball Of
        String
CN 1440 X1=52:Y1=59:R1=95
EK 1450 FOR LOOP=1 TO 150
FP 1460 THETA=RND*PI*2
```

```
OC 1470   X2=FNPOLARX(R1,THETA)
QN 1480   Y2=FNPOLARY(R1,THETA)
LD 1490   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
FP 1500   LINE (X1,Y1)-(X2,Y2),3
MN 1510   X1=X2:Y1=Y2
PA 1520   NEXT LOOP
MP 1530   GOSUB 2380
FC 1540 'Module #12 - LinedCircl
        e
JK 1550   THETA=2*PI*RND
OL 1560   R1=95
HG 1570   X1=FNPOLARX(R1,THETA)
JP 1580   Y1=FNPOLARY(R1,THETA)
PE 1590   FOR LOOP=1 TO 110
IB 1600   THETA=PI*2*RND
NE 1610   X2=FNPOLARX(R1,THETA)
PP 1620   Y2=FNPOLARY(R1,THETA)
KF 1630   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
LJ 1640   LINE (X1,Y1)-(X2,Y2),KO
          LOR
QL 1650   NEXT LOOP
NK 1660   GOSUB 2380
IO 1670 'Module #13 - SqureCircl
        e
MO 1680   R1=95:R2=R1^2+1
IJ 1690   FOR LOOP=0 TO R1 STEP 5
GD 1700   X1=SQR(R2-LOOP*LOOP)
BE 1710   Y1=SQR(R2-X1*X1)
KE 1720   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
DG 1730   LINE (X1+XCENTER,Y1+YCE
          NTER)-(XCENTER-X1,YCENTE
          R-Y1),KOLOR,B
QK 1740   NEXT LOOP
NJ 1750   GOSUB 2380
FP 1760 'Module #14 - CircularSp
        ikes
HG 1770   R1=20:R2=45:R3=115:R4=85
DO 1780   INC1=PI/3:INC2=PI/6
AG 1790   FOR THETA1=0 TO PI*2 STE
          P INC1
GL 1800   FOR THETA2=0 TO PI*2 ST
          EP INC2
CB 1810   X1=FNPOLARX(R3,THETA2)
FD 1820   Y1=FNPOLARY(R4,THETA2)
PK 1830   X2=FNPOLARX(R1,THETA1)
CM 1840   Y2=FNPOLARY(R2,THETA1)
IL 1850   KOLOR=KOLOR+1:IF KOLOR
          =4 THEN KOLOR=1
ON 1860   LINE (X1,Y1)-(X2,Y2),K
          OLOR
AD 1870   NEXT THETA2
BO 1880   NEXT THETA1
NH 1890   GOSUB 2380
PL 1900 'Module #14 - ExplodedCi
        rcularSpikes
HI 1910   R1=20:R2=45:R3=115:R4=85
CA 1920   INC1=PI/3:INC2=PI/6
QI 1930   FOR THETA1=0 TO PI*2 STE
          P INC1
HJ 1940   FOR THETA2=0 TO PI*2 ST
          EP INC2
DP 1950   X1=FNPOLARX(R3,THETA2)
EF 1960   Y1=FNPOLARY(R4,THETA1)
QI 1970   X2=FNPOLARX(R1,THETA1)
EG 1980   Y2=FNPOLARY(R2,THETA2)
JJ 1990   KOLOR=KOLOR+1:IF KOLOR
          =4 THEN KOLOR=1
MM 2000   LINE (X1,Y1)-(X2,Y2),K
          OLOR
OC 2010   NEXT THETA2
PN 2020   NEXT THETA1
LG 2030   GOSUB 2380
MH 2040 'Module #15 - BigMultiLo
        bes
FI 2050   R1=95:LOBES=2
KB 2060   INC=PI/128
EI 2070   FOR THETA=0 TO 2*PI STEP
          INC
BC 2080   R2=R1*SIN(LOBES*THETA)
PN 2090   X1=FNPOLARX(R2,THETA)
```

```
PM 2100   Y1=FNPOLARY(R2,THETA)
J6 2110   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
FJ 2120   LINE (XCENTER,YCENTER)-
          (X1,Y1),KOLOR
LB 2130   NEXT THETA
ML 2140   GOSUB 2380
KN 2150 'Module #16 - DoubleOval
        Cone
GA 2160   R1=120:R2=30:R3=40:R4=15
          2
II 2170   INC=PI/32
GK 2180   FOR THETA=PI TO 3*PI STE
          P INC
BK 2190   X1=XCENTER-R1*COS(THETA
          )
MH 2200   Y1=R3-R2*SIN(THETA)
NN 2210   X2=FNPOLARX(R1,THETA)
NE 2220   Y2=FNPOLARY(R2,THETA)-Y
          CENTER+R4
KO 2230   KOLOR=KOLOR+1:IF KOLOR=
          4 THEN KOLOR=1
KC 2240   LINE (X1,Y1)-(X2,Y2),KO
          LOR
MJ 2250   NEXT THETA
MD 2260   GOSUB 2380
GA 2270   CLS:WIDTH 80
EK 2280   PRINT "Press "CHR$(34)"G
          "CHR$(34)" to view again
          "
HB 2290   A$=INPUT$(1)
NC 2300   IF A$="G" OR A$="g" THEN
          RUN
KI 2310   WIDTH 80:SCREEN 0:END
MC 2320   SCREEN 1:CLS:LOCATE 1,1:
          PRINT "Mathgraphics"
HB 2330   PI=3.1415927#
IC 2340   XCENTER=151:YCENTER=93
BI 2350   DEF FNPOLARX(R,THETA)=R*
          COS(THETA)+XCENTER
NL 2360   DEF FNPOLARY(R,THETA)=R*
          SIN(THETA)+YCENTER
JH 2370   RETURN
FD 2380   TIME=TIMER
MB 2390   WHILE TIMER<TIME+15:WEND
NI 2400   CLS
JI 2410   LOCATE 1,1:PRINT "MathGr
          aphics"
JK 2420   RETURN
```

Program 2: Math Graphics Construction Set

```
OD 100 REM Copyright 1988 COMPUT
       E! Publications, Inc. --
       All Rights Reserved
PO 110   PI=3.141592654#
DF 120   DIM R(19):KEY OFF
NB 130   DEF FNPOLARX(R,THETA)=R*C
         OS(THETA)+XCENTER
JM 140   DEF FNPOLARY(R,THETA)=R*S
         IN(THETA)+YCENTER
CB 150   XCENTER=159:YCENTER=100
DD 160   FOR I=1 TO 19:READ R(I):N
         EXT I
AF 170   DATA 150,35,65,85,0,6,64,
         10000,10000,10000,10000,0
         ,0,20,1,1,1,1,1
EL 180   ON ERROR GOTO 630
ND 190   SCREEN 0,0,0
DH 200   WIDTH 80:CLS
JG 210   PRINT STRING$(78,"*")
LB 220   PRINT " MathGraphic Const
         ruction Set - Copyright 1
         988 COMPUTE! Publications
         , Inc."
KK 230   PRINT STRING$(78,"*")
IA 240   PRINT "(1) R1="R(1)
LG 250   PRINT "(2) R2="R(2)
PM 260   PRINT "(3) R3="R(3)
CC 270   PRINT "(4) R4="R(4)
EN 280   PRINT "(5) Start of Theta
          Loop (Pi * x)="R(5)
```

```
NE 290   PRINT "(6) End of Theta L
          oop (Pi * x)="R(6)
QP 300   PRINT "(7) Increment of T
          heta Loop (Pi / x)="R(7)
BK 310   PRINT "(8) X1's Offset (P
          i / x)="R(8)
FB 320   PRINT "(9) Y1's Offset (P
          i / x)="R(9)
JN 330   PRINT "(10) X2's Offset (
          Pi / x)="R(10)
OI 340   PRINT "(11) Y2's Offset (
          Pi / x)="R(11)
JG 350   PRINT "(12) Start of Thet
          a Loop #2 (Pi * x)="R(12)
PG 360   PRINT "(13) End of Theta
          Loop #2 (Pi * x)="R(13)
MO 370   PRINT "(14) Increment of
          Theta Loop #2 (Pi * x)="R
          (14)
QM 380   PRINT "(15) X1 uses Loop
          #"RIGHT$(STR$(R(15)),1)
FM 390   PRINT "(16) Y1 uses Loop
          #"RIGHT$(STR$(R(16)),1)
QJ 400   PRINT "(17) X2 Uses Loop
          #"RIGHT$(STR$(R(17)),1)
FJ 410   PRINT "(18) Y2 Uses Loop
          #"RIGHT$(STR$(R(18)),1)
LM 420   PRINT "(19) Color Palette
          #"RIGHT$(STR$(R(19)),1)
QJ 430   INPUT "Type number to mod
         ify, 20 to RUN, 21 to QUI
         T >",GO
IJ 440   IF GO=21 THEN CLS:END
DD 450   IF GO=20 THEN 500
GB 460   IF GO>19 THEN ERROR 13
HA 470   INPUT "Change to >",VA
EI 480   R(GO)=VA
HJ 490   GOTO 180
JE 500   IF R(19)=0 THEN SCREEN 1,
         0:COLOR ,0 ELSE IF R(19)=
         1 THEN SCREEN 1,0:COLOR ,
         1 ELSE IF R(19)=2 THEN SC
         REEN 1,1:COLOR ,0 ELSE IF
         R(19)=3 THEN SCREEN 1,1:
         COLOR ,1 ELSE ERROR 13
FC 510   FOR THETA1=PI*R(5) TO PI*
         R(6) STEP PI/R(7)
KF 520   FOR THETA2=PI*R(12) TO P
         I*R(13) STEP PI/R(14)
LG 530   IF R(15)=1 THEN X1=FNPO
         LARX(R(1),THETA1+PI/R(8))
         ELSE IF R(15)=2 THEN X1=
         FNPOLARX(R(1),THETA2+PI/R
         (8)) ELSE ERROR 13
AC 540   IF R(16)=1 THEN Y1=FNPO
         LARY(R(2),THETA1+PI/R(9))
         ELSE IF R(16)=2 THEN Y1=
         FNPOLARY(R(2),THETA2+PI/R
         (9)) ELSE ERROR 13
CE 550   IF R(17)=1 THEN X2=FNPO
         LARX(R(3),THETA1+PI/R(10)
         ) ELSE IF R(17)=2 THEN X2
         =FNPOLARX(R(3),THETA2+PI/
         R(10)) ELSE ERROR 13
HH 560   IF R(18)=1 THEN Y2=FNPO
         LARY(R(4),THETA1+PI/R(11)
         ) ELSE IF R(18)=2 THEN Y2
         =FNPOLARY(R(4),THETA2+PI/
         R(11)) ELSE ERROR 13
GC 570   KOLOR=KOLOR+1:IF KOLOR=
         4 THEN KOLOR=1
GF 580   LINE (X1,Y1)-(X2,Y2),KO
         LOR
OG 590   K$=INKEY$:IF K$<>"" THE
         N THETA1=PI*R(6):THETA2=P
         I*R(13):K=1
PF 600   NEXT THETA2
EB 610   NEXT THETA1
LK 615   IF K=1 THEN K=0:GOTO 190
PG 620   K$=INKEY$:IF K$="" THEN G
         OTO 620:ELSE GOTO 190
IM 630   PRINT "Bad Equation":GOTO
         190                      ©
```

G•www.commodore.ca

The Elementary Amiga
Part 4

Jim Butterfield

This month's installment focuses on the CLI and talks about special directories, logical devices, and timesaving command files. With command files, you can create your own CLI "macros" for copying fonts, deleting multiple files, or almost anything.

You may have noticed that most CLI commands reside in a directory called C (the directory has nothing to do with the language C). CLI commands are simply programs. So if I type *ZAP*, and there's a program called ZAP in my current directory, that program loads and runs. If I enter *DIR*, the computer looks for the DIR program in my current directory. Chances are, the computer will not find DIR in the current directory. In this case, it looks in the C directory.

C is just one of the Amiga's special directories. If I enter EXECUTE STARTUP–SEQUENCE, the computer knows that if it doesn't find the STARTUP–SEQUENCE in the current directory, it should look in the S directory. And if an Amiga Basic program contains a LIBRARY statement asking the system to load a specific .BMAP file, the computer looks in the LIBS directory.

A number of Amiga directories are set aside for special jobs. But how does the Amiga know to look in these directories? Whenever the Amiga needs to access certain types of information, it uses a *logical device*.

Logical Devices

Logical devices can be referred to just like hardware devices, such as DF0: for the internal disk drive or PRT: for a printer. For example, if the computer needs the system disk for some reason, it refers to it as logical device SYS:. When you boot your Amiga, the computer assigns SYS: to the Workbench disk that starts the system.

The Amiga has six other logical devices that are used to point to different disks and/or directories. These devices are explained below.

Logical device C: tells the computer where to look for CLI commands. By default, this device points to the C directory on the Workbench disk from which you booted. S: points to the S (sequence) directory, where the computer looks for command files. Files in this directory can usually be listed with the CLI's TYPE command and often make interesting reading.

DEVS: points to the DEVS directory, which holds information on devices. Information about your printer is found here, plus modules for handling other input/output operations. L: points to the L (library)

directory. This contains overlays for large commands and nonresident parts of the operating system.

LIBS: specifies the LIBS directory, where information for Open Library calls is stored. Generally, this directory doesn't contain program code, just lookup tables. Logical device FONTS: points to the FONTS directory, where character fonts are kept. If you don't use fonts, you can delete this directory with no ill effects.

There's one more logical device called T: for *temporary*. During editing and similar operations, a copy of your original file is placed here. That way, if you really botch things up you can go back to the earlier file.

To see a complete list of logical devices and what disks and/or directories they are currently linked with, enter the CLI command ASSIGN without any parameters.

Flexible Commands With EXECUTE

If there's a series of commands that you need to repeat, EXECUTE can do the job nicely. We'll start with a simple example and then work up to something more elegant.

Suppose you want to search through a series of disks to see which ones contain the ADDBUFFERS command within the C

directory. You could insert each disk into DF1: and enter LIST DF1:C/ADDBUFFERS. That seems like a lot of typing, and if you're like me, you're likely to spell something wrong. You might find it easier to create a file named Q containing this command and then EXECUTE it repeatedly. Try this:

```
ECHO >RAM:Q "LIST DF1:C/ADD-
   BUFFERS"
COPY C:EXECUTE RAM:EX
CD RAM:
```

The first line creates a file called Q on the ramdisk, containing our LIST command; the second line copies the EXECUTE command to RAM, with a shorter name (EX); and the last line switches our current directory to RAM. Now, each time we put a disk into DF1:, we need only enter EX Q, and the command LIST DF1:C/ADDBUFFERS is executed. If ADDBUFFERS is not on the disk, you get an *object not found* response. If the file exists, the computer gives you details such as ADDBUFFER's file size and date.

Let's try something more complex. Suppose we want to copy a complete font set from one disk to another. In fact, maybe we want to copy several fonts.

Font copying is tedious. First, you must make a directory with the correct name on the target disk. Then you must copy all the individual fonts into the newly created directory. Finally, you must copy the associated .FONT file into the target disk's FONTS directory. This process is hard enough with just one font, but when you have to copy several fonts, it really becomes work.

Try this: Enter the command ED S:FC to create a new file within the S directory called FC (for Font Copy). Now enter the following lines:

```
.KEY fontname/a
IF NOT EXISTS DF1:FONTS
   MAKEDIR DF1:FONTS
ENDIF
MAKEDIR DF1:FONTS/<fontname>
COPY DF0:FONTS/<fontname> TO
   DF1:FONTS/<fontname>
COPY DF0:FONTS/<fontname>.FONT
   TO DF1:FONTS
```

The first line insists on receiving a font name with this command. Lines 2–4 create a FONTS directory on the disk in DF1: if one does not already exist. Line 5 creates the font's subdirectory on the

disk in DF1:. Line 6 copies the contents of the font's subdirectory from the disk in DF0: to the disk in DF1:, and line 7 copies the associated .FONT file to disk in DF1:. When you've completed entering this command file, press ESC, X, and then RETURN to wind things up.

Now, if you want to copy a set of fonts called SAPPHIRE from a disk in DF0: to a disk in DF1:, just enter EXECUTE FC SAPPHIRE. If you have several fonts to copy, this command file makes the job easy. Don't forget that by copying this command file into the ramdisk and entering CD RAM:, you can really speed things up.

More Command Files
The versatility of ED and the power of EXECUTE can make some useful combinations. Let's look at the possibility of deleting a whole set of files from a disk.

Insert a disk from which you'd like to delete a number of files into DF1: and type

```
LIST >RAM:FLIST DF1: QUICK
```

Of course, you may list a subdirectory if that suits your purposes or use pattern matching to single out certain types of files. After entering this command, we have a list of files stored away in RAM:FLIST. Edit it with

```
ED RAM:FLIST
```

You'll see a list of files and directories, together with the header *Directory* . . . Move the cursor to this top line and press the ESC, D, and RETURN keys to delete this line.

In the same way, go through this list, removing the names of files that you don't want to delete. You can press the ESC-D sequence again, if you wish; alternatively, you may find it easier to press CTRL-G, which repeats the previous command. Either way, continue through the list until you have deleted the final line, which shows the count of files and directories.

At this point, you might like to press ESC followed by SA and then RETURN to save the file in its current state. Now, get ready for the high-powered stuff.

Press ESC and then type the following line, followed by a RETURN:

```
T; RP (E //DELETE DF1:/; N)
```

Let me explain this string of commands. The character *T* means go to the top of the document. The *RP* means repeat the following sequence in parentheses. In the repeated section, *E* means to exchange every null string (there's nothing between the first two slashes) with DELETE DF1:. The *N* commands the computer to move to the next line. Or in simpler terms, these commands mean go to the top of the text and insert the characters DELETE DF1: at the start of each line until you run out of lines.

Now we have a file to do some work for us. Exit the editor by pressing ESC, X, and RETURN. Then, assuming you really want to delete all those files on the disk in DF1:, activate the command file with

```
EXECUTE RAM:FLIST
```

You may have access to other supereditors with which you can do the same sort of thing. But if not, ED does the job neatly.

More To Come
Next time, we'll look at some more of my favorite CLI commands and take a closer look at multitasking.©

www.commodore.ca

Save Your Copies of COMPUTE!

Protect your back issues of *COMPUTE!* in durable binders or library cases. Each binder or case is custom-made in flag-blue binding with embossed white lettering. Each holds a year of *COMPUTE!*. Order several and keep your issues of *COMPUTE!* neatly organized for quick reference. (These binders make great gifts, too!)

Binders
$9.95 each;
3 for $27.95;
6 for $52.95

Cases:
$7.95 each;
3 for $21.95;
6 for $39.95

(Please add $2.50 per unit for orders outside the U.S.)

Send in your prepaid order with the attached coupon

- - - - - - - - - - - - -

≡CAPUTE!≡

Premature Prose

In last month's feature, "Superconductors: A Technology Coming Of Age" (p. 18), we incorrectly stated that researchers have superconductors working at room temperatures. This is not true. Somewhere in the editing process, a few words were dropped from the article's original text. The passage should have read: *With these recent developments, researchers believe that they'll soon have superconductors working at room temperatures.*

Softkey For Atari XL/XE

Some users may experience problems with the program "Softkey Editor" (January 1988). The program depends on the variable SCREEN$ being the first variable in the program. Unfortunately, the FOR/NEXT delay loop we added makes TT the first variable. Typing errors and immediate mode statements can also cause entries to be made in the variable name table.

To correct the problem, remove line 30 in the program. Next, LIST the program to disk with a command like LIST"D:TEMP". Do *not* use the SAVE command. Reboot your computer and type ENTER"D:TEMP". Finally, type SAVE"SKCREATE" to create a working version of the program.

Canfield For Apple II

There are a number of corrections to the Apple version of "Canfield" from the January 1988 issue. These lines need the following changes:

```
215 IF A$ = "y" THEN GOTO 40
220 IF A$ = "Y" THEN GOTO 40
1930 VTAB 18: HTAB 2: PRINT "{4
     SPACES}";: HTAB 2: PRINT
     "$"BR
2000 VTAB 22: HTAB 2: PRINT "PL
     AY AGAIN? (Y/N)"
2020 IF A$ = "y" OR A$ = "Y" TH
     EN BR = 500: POP : GOTO 40
2030 IF A$ = "n" OR A$ = "N" TH
     EN BR = 0: POP : GOTO 410
```

These changes are for both the DOS 3.3 and ProDOS versions.   ©

www.commodore.ca

Color Pad

Forrest Bentley

"Color Pad" can produce beautiful, detailed color screens such as this one.

The transfer function can be used to stamp images.

Perhaps the best Commodore 64 drawing program COMPUTE! has ever published, "Color Pad" includes many features and fast response. Features such as Fill, Square, Ellipse, Text, and Transfer—a powerful copy-and-paste option—make designing spectacular screens quick and easy. A joystick and a disk drive are required.

"Color Pad" is a full-featured drawing program that allows you to create detailed multicolor pictures using a joystick. Every program option is available through a single keypress, making the program easy to use. And Color Pad has a wide variety of functions, from Draw mode to Ellipse and Fill options.

Worth noting here is Color Pad's unique copy feature, Transfer. With Transfer, any section of the screen can be duplicated and saved to disk for later use. You can not only transfer objects from one section of the screen to another, but you can also transfer objects from one complete drawing to another.

Typing It In
Color Pad is composed of Programs 1 through 4. Program 1 is written in BASIC, so it is recommended that you use the "Automatic Proofreader," found elsewhere in this issue, to assist you in typing it in. Programs 2, 3, and 4 are each written in machine language and must be entered using "MLX," the machine language entry program found elsewhere in this issue. When you run MLX, answer the prompts as follows for each program:

Program 2
Starting address: 80E8
Ending address: 92C7

Program 3
Starting address: C000
Ending address: CEBF

Program 4
Starting address: 92E0
Ending address: 9CB7

After typing in each program, save a copy to disk. Save Program 1 with the filename COLOR PAD, Program 2 with the filename COLOR PAD.O, Program 3 with the filename COLOR PAD2.O, and Program 4 with the filename COLOR PAD3.O, all on the same disk. When run, Color Pad looks for the programs with these filenames.

Typing In Multiple Sittings
Programs 2–4 are quite long machine language programs. If you want to stop typing one of the listings at some point and resume later, press SHIFT-S and follow the screen prompts. Remember to note the line number of the last line you typed in.

When you are ready to continue typing, load MLX, answer the starting and ending address prompts, then press SHIFT-L. MLX asks for the filename you gave to the partially typed-in program. After the load is complete, press SHIFT-N and tell MLX the line number you stopped at. Now, continue typing as before. When you finish typing, MLX automatically prompts you to save the program.

Starting Up
To run Color Pad, plug a joystick into port 2, turn on your 64, and enter LOAD "COLOR PAD",8. After the program is loaded, type RUN and press RETURN. After a while, Color Pad's multicolor bit-mapped screen appears. Pressing f1 returns you to BASIC.

Once in Color Pad's bit-mapped screen, you'll notice a pen-shaped sprite indicating where you're plotting. The joystick controls the pen. To draw, the pen must be down. The joystick's fire button toggles the pen up and down. So you know the current orientation of your pen, two dark lines appear within the pen when it's down.

Color Pad contains many drawing tools for the computer artist. The following describes these tools and how to access and use each of them.

The Basics
Brushes: Color Pad offers a selection of two brushes: large and small. To select a new brush, press B. Your pen changes in size to reflect the new size of your brush.

Eraser: If you make a mistake—and we all do—you'll want an eraser. To access the eraser, press E. The eraser end of the pen opens up to indicate that Erase mode is in effect. Now, when the pen is down, you erase points rather than draw them. The eraser uses the current brush shape. To return to Draw mode, press D.

Drawing Color: This option allows you to access the 64's palette of 16 colors. To change the color your pen draws with, press the space bar. The border color indicates the new color. Pressing the space bar repeatedly cycles through all 16 different colors. To cycle backward through the colors, press SHIFT-SPACE.

The Commodore 64 allows

only four colors in any eight by eight pixel area. If you try to use more than four colors in an eight by eight block, one color will "splash" into another. Just be careful when using multiple colors in a small area.

Pen Color: To change the color of your pen, press A. Continue to press A until you find the color you want. This does *not* change the plotting color, just the color of the pen sprite. It's best to choose a pen color that contrasts well with the paint and canvas colors.

Pen Speed: You can speed up and slow down the pen's movement by pressing the plus and minus keys, respectively. Be careful when increasing your pen's speed— this program can move very fast.

Clear: The garbage can is probably an artist's most important tool. Everyone needs to throw away his or her work once in a while. The Clear option gives you this ability by disposing of your current sketch and giving you a clean slate. You may also wish to use the Clear option to begin a new drawing after saving the current one to disk.

To clear a sketch, press C. The pen vanishes, acknowledging your keypress. If you really wish to toss your sketch into the garbage can, press C again and the screen will clear. If you change your mind, press any key other than C and the pen will return, leaving your drawing intact.

Oops: Here you can dig through your garbage and pull out an old version of your drawing. The Oops option restores your drawing back to a previous state. To enter Oops mode, press O. The pen will vanish. Press O again to restore the sketch. Pressing any other key aborts the Oops process.

To specify the "previous state" that your drawing is restored to, press SHIFT-O. Whenever you press SHIFT-O, your drawing is copied into the Oops buffer, where it can be recalled by pressing O twice. The sketch is also copied into the Oops buffer whenever you use one of the function key options or display a directory.

It's a good idea to press SHIFT-O before making drastic modifications to a sketch. This way, you always have a good sketch to Oops back to.

Lines And Shapes

Lines: This option, a straight-edge tool, lets you draw accurate, straight lines.

To draw a line, move your pen to where you want the line to start and press L. A plus sign (+) appears on the screen at the location of your pen. Now, move your pen to where you want the line to end, and press the fire button. Instantly, a line is drawn using the current brush.

After the line is drawn, the plus sign moves to the current pen position (the line's endpoint), ready for another line. By allowing you to draw connecting lines like this, triangles, rectangles, and other polygons are easily created.

While in line mode, you may change your mind as to where the start of the line should be. To do this, simply press L again, and the plus sign exchanges places with the pen. To exit line mode, press RETURN.

Rays: Press R to enter ray mode. Rays work like lines, with one exception. After a ray is drawn, the + sign remains where it was before the ray was drawn. To exit ray mode, press RETURN.

Square: To draw a square, press S. Four plus signs appear, indicating the shape and size of the square. Move the joystick right or left to increase or decrease the horizontal length of the square. Similarly, move the joystick up or down to increase or decrease the vertical height of the square. Once you have the shape you want, press the fire button. Now, move the square to where you want it and press the fire button again. The square is placed onto the screen. To escape from this mode, press RETURN.

Diamond: A diamond is a square that sits on its corner. To draw a diamond, press Q. Four plus signs appear. As with a square, use the joystick to determine the shape of the diamond and press the fire button. Move to the desired location on the screen and press fire to draw the object. Pressing RETURN exits diamond mode.

Ellipse: To draw a circle or an ellipse, press I. Again, you are presented with four plus signs. Move the joystick to determine the shape of the ellipse. Once you have the shape you want, press the fire but-

ton. Now, move the plus signs to where you want the ellipse drawn and press the fire button again. An ellipse is placed onto the screen. To escape from this mode, press RETURN.

The Extras

Fill: To fill an enclosed area on your sketch, move the pen inside the area and press F. Make sure that there aren't any holes in the edges, or the paint will leak out. To terminate a Fill before it is finished, press RETURN.

After an area is filled, you may want to refill it with a different color. To do this, lift the pen up, select the new color using the space bar, and press F again. The new color covers the old color.

Fill reads the color under the pen's point and fills until it reaches a different color. If the color under the pen matches the current drawing color, the fill is stopped.

Text: With this tool, you can draw text in several different sizes. To place text onto a drawing, press T. The pen changes to an underline cursor. Now, simply type in your message. The cursor keys and the RETURN key can be used to move the cursor around the screen.

When you reach the end of the screen, the cursor does not automatically move to the next line. You must manually move to the next line using the cursor-down key or the RETURN key. The f1 key exits Text mode.

To erase a character while entering text, press f3. The cursor splits in two, indicating that you're in Erase mode. Now, use the cursor keys to move on top of the incorrect letter and retype that same letter. Just as with correctable ribbon on a typewriter, the offending letter is deleted from the screen. Press f3 again to draw characters.

Toggling the text cursor between Draw and Erase is identical to setting the pen to Draw and Erase. For instance, if you wish to enter reverse letters on a solid object, type your letters in Erase mode.

To increase the size of your characters, press the up-arrow key (↑, not the cursor-up key). Each time you press this key, the character size increases. The left-arrow key (←) reduces the character size.

To give words a three-dimensional look, type a word in black, back up to the first letter in the word using the cursor keys, press f1, move the pen a few pixels up and left, select a different paint color by pressing the space bar, press T to enter text mode, and retype the word.

Transfer: This option offers three important tools—a photocopier, scissors, and paste. You can select an area of the screen, copy it, and then paste the copy anywhere on your sketch.

To begin the transfer, move the pen to the upper-left corner of the area that you wish to select and press X. As with the square tool, four plus signs appear. Move the plus signs with the joystick to form a square that encloses the area to be transferred. Press the fire button and the enclosed area is copied. Now, move to where you want the copy to be placed and press the fire button again. The copy is pasted onto the sketch.

After you're through making all the copies you want, press RETURN and the pen reappears. If you want to retransfer the same copy later, just press N. The selection square reappears. Move to where you want the copy placed and press the fire button.

To transfer a copy to another sketch, press X, select an area, press RETURN to exit Transfer mode, press f4 to load a new drawing, and press N to transfer the copy. Transfers may also be saved to disk for later use (see below).

If you use a drawing tool that uses the plus signs (like Square or Ellipse) between the transfer and retransfer functions, the retransfer does not work properly. In this case you have to copy the original again, or save it to disk and load it back in.

Zoom: To add detail to your pictures, Color Pad offers Zoom mode. You activate Zoom mode by pressing Z. A square area defined by four plus signs appears. Move the square area over the section of the screen that you wish to edit and press fire. The selected area is enlarged to fill the entire screen.

To edit this magnified view of your sketch, simply use your pen and draw. If you wish to erase pixels, enter Erase mode by pressing E.

Using Color Pad Pictures

From Color Pad To BASIC

Programs 5 and 6 allow you to load Color Pad pictures from BASIC. You may use these pictures in your own programs, or simply create a Color Pad slide show that exhibits your finest creations.

To begin, type in Programs 5 and 6. Program 5 is written in BASIC, so be sure to use the "Automatic Proofreader" to ensure accurate entry.

Program 6 is written in machine language. You must enter Program 6 using "MLX," the machine language entry program found elsewhere in this issue. Enter the following starting and ending addresses for Program 6:

Starting address: C350
Ending address: C437

Save Program 5 and 6 to the same disk. Save Program 6 using the filename CPBASIC.O. Program 5 looks for Program 6 using this filename.

When you run Program 5, the computer requests the filename of a Color Pad file. Insert the disk containing the desired picture and enter the picture's filename. After the picture loads, it is displayed on the screen. Press any key to exit the program.

You can modify Program 5 for your own purposes. For example, by deleting line 150 and changing line 160 to

160 FL$="MYPICTURE"

the program automatically loads and displays the drawing saved as MYPICTURE.

Graphics programmers may be interested to know that Color Pad's bitmapped picture loads into memory at 8192–16383. The text screen is still located at 1024–2023. Because the multicolor drawing resides in the BASIC workspace, a POKE 52,32:POKE 56,32:CLR is necessary to keep BASIC from intruding on your picture. These POKEs, however, reduce the workspace to a little under 6K.

From Koala Pad To Color Pad

Program 7 is a machine language program that transfers files created by Koala Pad's *Koala Paint* program into Color Pad. To use Program 7, you must first type it in using MLX. Enter the following values when prompted:

Starting address: C000
Ending address: C047

When you are finished, save the program as CONVERT.O. Now, to convert the picture to Color Pad format, enter the following lines. Note that the *A* in line 10 should be changed to specify the desired Koala Pad file.

```
NEW
10 F$=CHR$(129)+"PIC A*":REM WHERE "A" IS THE PICTURE'S LETTER
20 IF L=0 THEN L=1:LOAD F$,8,1
RUN
LOAD "CONVERT.O",8,1
SYS 49152
LOAD "COLOR PAD",8
248 POKE 821,87
RUN
```

When Color Pad runs, the converted Koala Pad drawing appears on the screen, ready for you to edit or save. ©

Press D to return to Draw mode. You can change the color that your pen draws with by pressing the space bar. To move the Zoom viewing area around the sketch, use the cursor keys.

Press RETURN to restore your screen to normal. The Zoom selection square remains on the screen for you to use. If you wish to exit Zoom mode entirely, press RETURN again. Zoom mode is perfect for adding final touches to a detailed drawing.

Cwww.commodore.ca

Special Features

Exchange Color: After drawing for a while, you may want to exchange all occurrences of one color for another. To do this, lift the pen up, move the point of the pen on top of the color that you want to change, press the space bar until you find the color that you want, and press SHIFT-A. All occurrences of the old color switch with the new color.

Rapid Colors: For interesting color effects, turn on Rapid Colors by pressing SHIFT-R. When activated, the drawing color rapidly cycles between all 16 colors. Press SHIFT-R again to turn this mode off. Rapid Colors affects all drawing functions, such as Draw, Ellipse, and Fill.

Shading: Shading allows you to draw using different patterns or shades. Press SHIFT-S to activate Shading. When activated, the pen toggles quickly between Draw and Erase mode, creating a shading effect. The speed at which the pen switches between drawing and erasing depends on the speed of your pen. You can adjust the pen speed by pressing the plus and minus keys. To turn off Shading, press SHIFT-S again.

Shading can be useful for drawing dotted lines. Try using the Fill option using different pen speeds while Shading is on. You can create some very interesting graphics this way.

Screen Scroll: By using the cursor keys, you can move a drawing around on the screen. Any part of the drawing that moves off the edge of the screen is erased, so be careful. It is a good idea to create borders for your drawings last, just in case you accidentally hit a cursor key.

Swap: With Swap, you can have two sketches in memory at one time. When you press f2, the current screen "swaps" with the one stashed in memory. When loading, saving, or printing a sketch, only the current screen—the one that's visible—is affected.

Using Color Pad's Transfer feature, you can copy objects from one screen to the other. Just press X for Transfer, select an area, press RETURN to exit Transfer mode, press f2 to Swap screens, and press N to Transfer a copy to the alternate screen.

Screen Copy: Pressing f8 copies the current screen to the alternate screen. The border changes color briefly while the current screen is being duplicated. When the border color returns to normal, both screens are identical.

Help: All of Color Pad's options are documented in an online help screen. The f7 key opens the help screen and RETURN closes it.

File Commands

Directory: Pressing SHIFT-RETURN prints the current disk directory to the screen. To resume drawing, press RETURN.

Save: To save your sketch, press f3. The program prompts you for a filename. When saved, the characters *CP.* are added to the beginning of the filename to help you remember which files are Color Pad files. For this reason, you must enter a filename with 13 or fewer characters. Pressing RETURN without entering a filename exits this option.

Load: Here you can load a previously saved sketch by pressing f4. When the program prompts you, enter the filename that you used to save the sketch (without the added CP.). The sketch is loaded and ready for editing. Pressing RETURN without entering a filename returns you to Draw mode.

Save Transfer: This option allows you to save a transfer. A transfer is a selected portion of your sketch (see the Transfer description above). To save a transfer, press f5. The program prompts you for a filename. Transfer files are saved with the characters *CX.* added to the beginning of the filename. You may exit this option by pressing RETURN without entering a filename.

Load Transfer: This option allows you to load a previously saved transfer. To load a transfer, press f6. When the program prompts you, enter the filename that you used to save the sketch (without the added CX.). The transfer is loaded and ready for use. You may exit this option by pressing RETURN without entering a filename.

After the transfer is loaded, the retransfer function is automatically called and you are presented with the four + symbols. Press RETURN if you do not wish to place the transfer yet.

Quitting

Exit: To exit the program, press f1. If you exit Color Pad accidentally, entering RUN immediately restarts the program with your drawing intact.

Color Pad

```
XQ 100 REM COPYRIGHT 1988 COMP
       UTE! PUBLICATIONS, INC.
       - ALL RIGHTS RESERVED
FF 110 POKE 53269,0
MH 120 IF PEEK(52)>28 THEN POK
       E52,28:POKE56,28:POKE82
       0,87:CLR
EC 130 IF PEEK(821)=87 THEN L=
       99
ES 140 IF L>0 THEN GOTO230
AA 150 GOSUB1190
DR 160 PRINT"{15 RIGHT}{BLK}R
       {7}R{RED}R{CYN}R{PUR}R
       {8}R{GRN}R{7}R{3}R"
JD 170 PRINT"{15 RIGHT}{BLK}C
       {7}O{RED}L{CYN}O{PUR}R
       {SPACE}{GRN}P{7}A{1}D"
PM 180 PRINT"{15 RIGHT}{BLK}E
       {7}E{RED}E{CYN}E{PUR}E
       {8}E{GRN}E{7}E{1}E
       {2 DOWN}"
JS 190 PRINT"{3 RIGHT}{WHT}COP
       YRIGHT 1988 COMPUTE! PU
       BL, INC.{DOWN}"
PB 200 PRINT"{10 RIGHT}ALL RIG
       HTS RESERVED{3 DOWN}"
MF 210 PRINT"{15 RIGHT}{BLK}LO
       ADING...{CYN}"
MM 220 L=1:LOAD"COLOR PAD.O",8
       ,1
FM 230 IF L<>1 THEN GOTO270
BD 240 OPEN15,8,15:OPEN2,8,2,"
       COLOR PAD2.O,P,R":INPUT
       #15,E,E$:CLOSE2:CLOSE15
KP 250 IF E>0 THEN L=3:POKE 33
       559,96:GOTO310
PG 260 IF L=1 THEN L=2:LOAD"CO
       LOR PAD2.O",8,1
XG 270 IF L<>2 THEN GOTO300
CJ 280 OPEN15,8,15:OPEN2,8,2,"
       COLOR PAD3.O,P,R":INPUT
       #15,E,E$:CLOSE2:CLOSE15
PM 290 IF E>0 THEN L=3:POKE 49
       172,96:GOTO310
XK 300 IF L=2 THEN L=3:LOAD"CO
       LOR PAD3.O",8,1
RP 310 POKE835,1:POKE836,0:POK
       E837,0:POKE838,55
SH 320 IF DT>0 THEN GOTO350
HQ 330 DT=1:FORC=1 TO 128:READ
       D:POKE 16255+C,D:NEXT
CF 340 POKE 37465,5:POKE 37664
       ,5
SK 350 POKE840,0:PRINT"{CLR}":
       SYS33000
JF 360 POKE53280,6:POKE53281,6
       :PRINT"{CYN}{CLR}";
XD 370 IF PEEK(839)=1 THEN GOT
       O410
FE 380 IF PEEK(839)=136 THEN G
       OSUB890:GOTO350
PX 390 IFPEEK(839)=141 THEN GO
       SUB1310:SYS49152:GOSUB8
       40:GOTO350
EJ 400 GOSUB600:GOTO350
SA 410 PRINT"COLOR PAD PROCESS
       ING COMPLETE."
JX 420 END
GP 430 REM SPRITE DATA
JM 440 DATA000,000,000,000,000
```

```
QP 450 DATA000,000,000,000,000
         ,000,000,000
CJ 460 DATA000,000,000,000,000
         ,255,128,000
DP 470 DATA160,192,000,160,224
         ,000,160,192
JD 480 DATA000,255,128,000,000
         ,000,000,000
QS 490 DATA000,000,000,000,000
         ,000,000,000
FR 500 DATA000,000,000,000,000
         ,000,000,000
RJ 510 DATA000,000,000,000,000
         ,000,000,128
RX 520 DATA000,000,000,000,000
         ,000,000,000
BX 530 DATA000,000,000,000,000
         ,000,000,000
AC 540 DATA000,000,000,000,000
         ,000,003,255
BP 550 DATA000,006,005,000,014
         ,005,006
GJ 560 DATA005,000,003,255,000
         ,000,000,000
BC 570 DATA000,000,000,000,000
         ,000,000,000
FD 580 DATA000,000,000,000,000
         ,000,000,000
RS 590 DATA000,000,000,000,000
         ,000,000,128
QH 600 REM LOAD/SAVE
PK 610 FC=PEEK(839)
MM 620 IF FC=134 THEN PRINT"SA
         VE SKETCH{2 DOWN}":BC$=
         "CP."
GR 630 IF FC=138 THEN PRINT"LO
         AD SKETCH{2 DOWN}":BC$=
         "CP."
JS 640 IF FC=135 THEN PRINT"SA
         VE TRANSFER{2 DOWN}":BC
         $="CX."
JP 650 IF FC=139 THEN PRINT"LO
         AD TRANSFER{2 DOWN}":BC
         $="CX."
AQ 660 GOSUB1310
MF 670 FC$="":INPUT"ENTER THE
         {SPACE}FILE NAME";FC$:I
         F FC$="" THEN GOTO880
JE 680 FC$=BC$+LEFT$(FC$,12)
HH 690 OPEN15,8,15
QJ 700 OPEN2,8,2,FC$+",P,R":IN
         PUT#15,E,E$:CLOSE2:CLOS
         E15
JA 710 IF E>0 THEN IF FC>137 T
         HEN830
MG 720 IF E>0 AND E<>62 THEN I
         F FC<137 THEN830
FH 730 IF FC=139 THEN POKE 198
         ,1:POKE631,78
AS 740 IF FC>137 THEN PRINT"
         {3 DOWN}LOADING ";FC$:L
         OAD FC$,8,1
SM 750 POKE 840,FC:POKE841,LEN
         (FC$):PRINT"{3 DOWN}SAV
         ING ";FC$;"{3 DOWN}"
RB 760 FOR C1=1 TO LEN(FC$):PO
         KE841+C1,ASC(MID$(FC$,C
         1,1)):NEXT
FP 770 IF FC=134 THEN GOTO810
KH 780 EA=27002+INT(PEEK(27000
         )*PEEK(27001))+10
XD 790 EH=INT(EA/256):EL=EA-(E
         H*256)
HP 800 POKE860,EL:POKE861,EH
BM 810 SYS33000:IF PEEK(840)=0
          THEN PRINT"GOOD SAVE":
         GOTO840
DK 820 PRINT"BAD SAVE";PEEK(84
         0):GOTO840
HD 830 PRINTE;E$
```

```
GS 840 PRINT"{3 DOWN}PRESS RET
         URN TO CONTINUE"
AD 850 OPEN15,8,15:INPUT#15,E,
         E$:CLOSE15
HS 860 GET FC$:IF FC$<>"" THEN
          GOTO860
AB 870 GET FC$:IF FC$="" THEN
         {SPACE}GOTO870
CX 880 RETURN
FP 890 REM HELP SCREEN
XD 900 POKE53280,6:POKE53281,6
DF 910 PRINT"{CLR}{10 SPACES}
         {BLK}C[7]O{RED}L{CYN}O
         {PUR}R {GRN}P[7]A[1]D
         {RED}F{GRN}U{YEL}N{BLK}
         C{RED}T[2]I[3]O[5]N[5]S
         "
MR 920 PRINT"{BLK}CCCCCCCCCCC
         CCCCCCCCCCCCCCCCCCCCCCC
         CCCCC{UP}"
MR 930 PRINT"{UP}{12 SPACES}MI
         NOR FUNCTIONS{CYN}"
JC 940 PRINT"{RVS}FIRE{OFF}=
         {WHT}PEN UP/DOWN {CYN}
         {RVS}{3 SPACES}SHIFT-O
         {OFF}={WHT}STORE PIC.
         {CYN}"
ED 950 PRINT"{RVS} D,E{OFF}=
         {WHT}DRAW/ERASE
         {2 SPACES}{CYN}{RVS}
         {5 SPACES}SPACE{OFF}=
         {WHT}DRAW COLOR{CYN}"
JF 960 PRINT"{RVS}{3 SPACES}B
         {OFF}={WHT}BRUSH SIZE
         {2 SPACES}{CYN}{RVS}
         {3 SPACES}SHIFT-A{OFF}=
         {WHT}EXCH COLOR{CYN}"
XE 970 PRINT"{RVS}C(2){OFF}=
         {WHT}CLEAR{7 SPACES}
         {CYN}{RVS}{9 SPACES}A
         {OFF}={WHT}PEN COLOR
         {CYN}"
XH 980 PRINT"{RVS}O(2){OFF}=
         {WHT}OOPS{8 SPACES}
         {CYN}{RVS}{7 SPACES}+,-
         {OFF}={WHT}PEN SPEED
         {CYN}"
SJ 990 PRINT"{BLK}CCCCCCCCCCC
         CCCCCCCCCCCCCCCCCCCCCCC
         CCCCC{UP}"
RG 1000 PRINT"{UP}{12 SPACES}M
         AJOR FUNCTIONS{CYN}"
QF 1010 PRINT"{RVS}{3 SPACES}F
         {OFF}={WHT}FILL
         {8 SPACES}{CYN}{RVS}
         {9 SPACES}X{OFF}={WHT}
         TRANSFER{CYN}"
QM 1020 PRINT"{RVS}{3 SPACES}L
         {OFF}={WHT}LINES
         {7 SPACES}{CYN}{RVS}
         {9 SPACES}N{OFF}={WHT}
         RETRANSFER{CYN}"
XB 1030 PRINT"{RVS}{3 SPACES}R
         {OFF}={WHT}RAYS
         {8 SPACES}{CYN}{RVS}
         {9 SPACES}T{OFF}={WHT}
         TEXT{CYN}"
AH 1040 PRINT"{RVS}{3 SPACES}S
         {OFF}={WHT}SQUARE
         {6 SPACES}{CYN}{RVS}
         {3 SPACES}CURSORS{OFF}
         ={WHT}SHIFT SCREEN
         {CYN}"
PF 1050 PRINT"{UP}{RVS}
         {3 SPACES}Z{OFF}={WHT}
         ZOOM{8 SPACES}{CYN}
         {RVS}{3 SPACES}SHIFT-S
         {OFF}={WHT}SHADE
         {6 SPACES}{CYN}"
GE 1060 PRINT"{RVS}{3 SPACES}I
         {OFF}={WHT}ELLIPSE
```

```
         {5 SPACES}{CYN}{RVS}
         {3 SPACES}SHIFT-R{OFF}
         ={WHT}RAPID COLORS
         {CYN}"
ME 1070 PRINT"{UP}{RVS}
         {3 SPACES}Q{OFF}={WHT}
         DIAMOND{5 SPACES}{CYN}
         {RVS}SFT-RETURN{OFF}=
         {WHT}DIRECTORY{CYN}"
MC 1080 PRINT"{BLK}CCCCCCCCCC
         CCCCCCCCCCCCCCCCCCCCCC
         CCCCCC{UP}"
CS 1090 PRINT"{UP}{13 SPACES}F
         UNCTION KEYS{CYN}"
CA 1100 PRINT"{RVS}{2 SPACES}F
         1{OFF}={WHT}EXIT PROGR
         AM{CYN}{RVS}{8 SPACES}
         F2{OFF}={WHT}SWAP SCRE
         ENS{CYN}"
RQ 1110 PRINT"{UP}{RVS}
         {2 SPACES}F3{OFF}=
         {WHT}SAVE SKETCH {CYN}
         {RVS}{8 SPACES}F4{OFF}
         ={WHT}LOAD SKETCH{CYN}
         "
PD 1120 PRINT"{RVS}{2 SPACES}F
         5{OFF}={WHT}SAVE TRANS
         .{CYN}{RVS}{8 SPACES}
         F6{OFF}={WHT}LOAD TRAN
         S.{CYN}"
HC 1130 PRINT"{RVS}{2 SPACES}F
         7{OFF}={WHT}HELP SCREE
         N {CYN}{RVS}{8 SPACES}
         F8{OFF}={WHT}COPY SCRE
         EN{CYN}"
SR 1140 PRINT"{BLK}CCCCCCCCCC
         CCCCCCCCCCCCCCCCCCCCCC
         CCCCCC{3 UP}"
MS 1150 PRINT"{2 DOWN}
         {10 SPACES}{RVS}RETURN
          TO CONTINUE{OFF}
         {3 UP}{CYN}"
JB 1160 GET FC$:IF FC$<>"" THE
         N GOTO1160
HH 1170 GET FC$:IF FC$="" THEN
          GOTO1170
FM 1180 RETURN
GM 1190 REM PRINT BORDER
HM 1200 POKE53280,6:POKE53281,
         6:PRINT"{CLR}";
BJ 1210 BC$="{RVS} {OFF}":CR$=
         "{BLK}{WHT}{RED}{CYN}
         {PUR}{GRN}[7]{YEL}[1]
         [2][3][4][5][6][7][8]"
         :CP=1
CH 1220 FOR C=1 TO 40:GOSUB127
         0:NEXT
GC 1230 FOR C=1 TO 23:GOSUB127
         0:PRINT"{2 LEFT}{DOWN}
         ";:GOSUB1270:NEXT
HS 1240 FOR C=1 TO 39:GOSUB127
         0:NEXT
MF 1250 PRINT"{LEFT}";CHR$(148
         );:PRINT" {HOME}
         {3 DOWN}"
KS 1260 RETURN
JK 1270 REM BORDER CHARACTER
DE 1280 CP=CP+1:IF CP>16 THEN
         {SPACE}CP=1
QS 1290 PRINT MID$(CR$,CP,1);B
         C$;
AA 1300 RETURN
SA 1310 OPEN2,8,2:CLOSE2
DM 1320 IF ST<>0 THEN GOTO1350
RF 1330 OPEN15,8,15:OPEN2,8,2,
         "$,P,R":INPUT#15,E,E$:
         CLOSE2:CLOSE15
BR 1340 IF E=0 THEN GOTO1390
AR 1350 B=B+1:IF B<2 THEN GOTO
         1310
MF 1360 B=0
```

```
MR 1370 PRINT"TURN ON THE DISK
           DRIVE AND INSERT A DI
           SK"
DK 1380 GOSUB840:PRINT"{DOWN}"
        :GOTO1310
XK 1390 RETURN
```

Color Pad .O

```
80E8:AD 48 03 F0 04 20 2F 8E CF
80F0:60 20 FB 80 20 24 81 20 66
80F8:30 81 60 AD 18 D0 09 08 77
8100:8D 18 D0 AD 11 D0 09 20 C2
8108:8D 11 D0 AD 16 D0 09 10 21
8110:8D 16 D0 20 51 81 AD 15 7C
8118:D0 09 01 8D 15 D0 A9 00 FD
8120:8D 47 03 60 20 0C 82 20 78
8128:AB 82 AD 47 03 F0 F5 60 F3
8130:20 7D 8A AD 16 D0 29 EF 05
8138:8D 16 D0 AD 18 D0 29 F7 CA
8140:8D 18 D0 AD 11 D0 29 DF 03
8148:8D 11 D0 A9 00 8D 15 03 3C
8150:60 AD 35 03 C9 57 D0 06 19
8158:20 75 8A 4C A8 81 A9 57 D4
8160:8D 35 03 A9 00 85 FB A9 2A
8168:20 85 FC A9 00 98 91 FB F8
8170:20 FF 91 A5 FC C9 3F D0 6E
8178:F4 A5 FB C9 40 D0 EE AD 4C
8180:44 03 0A 0A 0A 0A 0A 0A 43 1E
8188:8D 86 92 A9 00 85 FB BE
8190:A9 04 85 FC AD 86 92 91 28
8198:FB 20 FF 91 A5 FC C9 07 76
81A0:D0 F2 A5 FB C9 E7 D0 EC B9
81A8:20 7D 8A A9 00 85 FB A9 BE
81B0:20 85 FC A9 01 8D 3E 03 1D
81B8:8D 40 03 8D 41 03 A9 00 35
81C0:8D 3F 03 8D 10 D0 A9 C0 6B
81C8:8D 42 03 AD 44 03 8D 45 EC
81D0:03 8D 20 D0 A9 F5 8D F8 03
81D8:07 A9 00 85 FD A9 04 85 46
81E0:FE A9 64 8D 67 03 20 6C 5E
81E8:89 20 3D 8A CE 86 92 D0 8F
81F0:F5 A9 00 8D 67 03 AD 9E 73
81F8:3F 09 18 8D 9E 3F AD E0 E1
8200:3F 09 18 8D 3F A9 01 1B
8208:8D 68 03 60 20 F2 91 AD F1
8210:00 DC 8D B8 92 29 10 F0 D3
8218:08 A9 01 8D 66 03 4C 3D 99
8220:82 CE 66 03 AD 66 03 D0 F4
8228:14 A9 64 8D 66 03 AD 67 09
8230:03 F0 05 A9 00 4C 3A 82 56
8238:A9 00 8D 67 03 AD 67 03 1B
8240:F0 13 A9 BE 8D 98 3F 8D 7E
8248:9E 3F A9 7D 8D DA 3F 8D 5D
8250:E0 3F 4C 65 82 A9 A0 8D FE
8258:98 3F 8D 9E 3F A9 05 8D 4D
8260:DA 3F 8D E0 3F AD 68 03 E6
8268:F0 0D A9 A0 8D 9B 3F A9 6B
8270:05 8D D0 3F 4C 7F 82 A9 1A
8278:00 8D 9B 3F 8D D0 3F AD 58
8280:B8 92 29 08 D0 03 20 6C 6B
8288:89 AD B8 92 29 04 D0 03 FB
8290:20 B6 89 AD B8 92 29 01 C2
8298:D0 03 20 00 8A AD B8 92 D9
82A0:29 02 D0 03 20 3D 8A 20 30
82A8:7F 8F 60 A5 C6 D0 01 60 93
82B0:20 E4 FF C9 44 F0 64 C9 14
82B8:45 F0 63 C9 43 F0 62 C9 12
82C0:85 F0 61 C9 41 F0 69 C9 F7
82C8:20 F0 5C C9 2D F0 5B C9 EF
82D0:2B F0 5A C9 A0 F0 71 C9 05
82D8:C1 F0 58 C9 53 F0 57 C9 79
82E0:58 F0 56 C9 4E F0 55 C9 60
82E8:46 F0 5A C9 4F F0 50 C9 DD
82F0:CF F0 4F C9 89 F0 57 C9 29
82F8:86 F0 50 C9 8A F0 4C C9 9E
8300:87 F0 48 C9 8B F0 44 C9 1F
8308:8D F0 40 C9 88 F0 3C C9 C0
8310:8C F0 41 C9 42 F0 3A 20 C8
8318:00 C0 60 4C 57 83 4C 5D DE
8320:83 4C BE 83 4C 63 83 4C 4F
8328:69 83 4C 7C 83 4C 8C 83 FF
8330:4C AE 83 4C DA 83 4C DE 9A
8338:83 4C E2 83 4C E6 83 4C F9
8340:FF 83 4C FB 83 4C F7 83 32
8348:4C 9B 83 4C 16 84 4C 1A 07
8350:84 4C 1E 84 4C 30 84 A9 8E
8358:01 8D 68 03 60 A9 00 8D B7
8360:68 03 60 A9 01 8D 47 03 D2
8368:60 EE 45 03 AD 45 03 C9 86
8370:10 D0 05 A9 00 8D 45 03 B2
8378:8D 20 D0 60 AD 46 03 69 64
8380:0A 18 C9 06 90 02 A9 F0 AB
8388:8D 46 03 60 AD 46 03 E9 C4
8390:0A 18 C9 0F B0 F2 A9 10 81
8398:4C 88 83 CE 45 03 AD 45 1C
83A0:03 C9 FF D0 05 A9 0F 8D 23
83A8:45 03 8D 20 D0 60 EE 27 D3
83B0:D0 AD 27 D0 C9 10 D0 05 B2
83B8:A9 00 8D 27 D0 60 A9 00 14
83C0:8D 15 D0 20 E4 FF F0 F6 EF
83C8:A2 01 8E 15 D0 C9 43 D0 89
83D0:08 A9 00 8D 35 03 20 51 66
83D8:81 60 20 39 84 60 20 34 6A
83E0:85 60 20 AC 87 60 A9 60 03
83E8:8D 45 86 20 06 86 A9 8D 06
83F0:8D 45 86 20 A9 88 60 20 33
83F8:42 88 60 20 7D 8A 60 42 2B
8400:00 8E 15 D0 20 E4 FF F0 E1
8408:F6 A2 01 8E 15 D0 C9 4F 0D
8410:D0 03 20 75 8A 60 8D 47 D5
8418:03 20 7E 8E 60 AD 69 61
8420:03 C9 57 F0 05 A9 57 4C E0
8428:2C 84 A9 00 8D 69 03 60 16
8430:20 8E A9 57 8D C4 92 3E
8438:60 A0 A9 B1 FB 2D 42 03 DD
8440:D0 07 AD 45 03 8D 21 D0 DE
8448:60 8D 87 92 AD 42 03 8D 09
8450:88 92 4E 88 92 4E 88 92 06
8458:AD 88 92 F0 09 4E 87 92 9E
8460:4E 87 92 4C 52 84 AD 87 11
8468:92 C9 02 F0 28 C9 01 F0 D7
8470:2E A5 FE 69 00 85 FD 88 AB
8478:A5 FE 69 D4 85 FE B1 FD 18
8480:29 0F 8D 87 92 A5 FD 38 6B
8488:E9 00 85 FD A5 FE E9 D4 E8
8490:85 FE 4C A8 84 B1 FD 29 40
8498:0F 8D 87 92 4C A8 84 B1 66
84A0:FD 4A 4A 4A 4A 8D 87 92 53
84A8:20 31 92 A9 00 85 FD A9 B6
84B0:04 85 FE B1 FD 29 0F CD 98
84B8:87 92 F0 3A B1 FD 4A 4A 50
84C0:4A 4A CD 87 92 F0 3B A5 28
84C8:FD 18 69 00 85 FD A5 FE 72
84D0:69 D4 85 FE B1 FD 29 0F 4B
84D8:CD 87 92 D0 09 B1 FD 29 3E
84E0:F0 0D 45 03 91 FD A5 FD 4C
84E8:38 E9 00 85 FD A5 FE E9 A4
84F0:D4 85 FE 4C 16 84 B1 FD 92
84F8:29 F0 0D 45 03 91 FD 4C 6F
8500:16 85 AD 45 03 0A 0A 0A DF
8508:0A 8D 90 92 B1 FD 29 0F 90
8510:0D 90 92 4C FD 84 A5 FD 28
8518:18 69 01 85 FD A5 FE 69 EF
8520:00 85 FE C9 07 D0 0A A5 3E
8528:FD C9 04 D0 04 20 3C 92 7A
8530:60 4C B3 84 20 A6 85 A5 89
8538:C6 D0 60 20 F2 91 20 56 5D
8540:86 A5 C6 D0 56 A9 01 8D C6
8548:67 03 AD 78 69 8D 95 92 44
8550:20 6C 89 20 7F 8F CE 95 27
8558:92 D0 F5 AD 79 69 8D 95 9C
8560:92 20 3D 8A 20 7F 8F CE F9
8568:95 92 D0 F5 AD 78 69 8D 20
8570:95 92 20 B6 89 20 7F 8F B5
8578:CE 95 92 D0 F5 AD 79 69 72
8580:8D 95 92 20 00 8A 20 7F 2E
8588:8F CE 95 92 D0 F5 A9 00 9C
8590:8D 67 03 A9 02 8D 66 03 4D
8598:4C 37 85 A9 01 8D 15 D0 1C
85A0:A9 00 8D 67 03 60 20 06 88
85A8:86 A5 C6 D0 53 A9 00 DC 74
85B0:8D B8 92 29 10 D0 08 A9 13
85B8:64 8D 66 03 4C 00 86 AD 73
85C0:78 69 18 C9 0A B0 0B AD D8
85C8:79 69 18 C9 0A B0 03 20 C3
85D0:F2 91 AD B8 92 29 08 D0 15
85D8:03 EE 78 69 AD B8 92 29 65
85E0:04 D0 03 CE 78 69 AD B8 EC
85E8:92 29 02 D0 03 EE 79 69 05
85F0:AD B8 92 29 01 D0 03 CE 06
85F8:79 69 20 B5 86 4C A9 85 B8
8600:A9 00 8D 67 03 60 A9 00 F6
8608:A2 40 9D 80 03 CA D0 FA 12
8610:8E 80 03 A9 10 8D 96 03 66
8618:8D 99 03 8D 9F 03 8D A2 52
8620:03 A9 7C 8D 9C 03 A9 0E D3
8628:8D F9 07 8D FA 07 8D FB 3F
8630:07 8D FC 07 AD 27 D0 8D 6D
8638:28 D0 8D 29 D0 8D 2A D0 B3
8640:8D 2B D0 A9 1E 8D 15 D0 B5
8648:A9 00 8D 67 03 A9 01 8D A0
8650:78 69 01 60 AD 27 D0 8D 28 C4
8658:F0 01 60 AD 27 D0 8D 29 01 A4
8660:D0 8D 29 D0 8D 2A D0 8D AF
8668:2B D0 AD 00 DC 8D B8 92 16
8670:29 10 F0 08 A9 01 8D 66 87
8678:03 4C 87 86 CE 66 03 D0 5A
8680:06 A9 64 8D 66 03 60 AD 0E
8688:B8 92 29 08 D0 03 20 6C 7B
8690:89 AD B8 92 29 04 D0 03 DC
8698:20 B6 89 AD B8 92 29 02 D3
86A0:D0 03 20 3D 8A AD B8 92 BD
86A8:29 01 D0 03 20 00 8A 20 0B
86B0:B5 86 4C 56 86 AD 78 69 6E
86B8:D0 05 A9 01 8D 78 69 C9 9F
86C0:A0 D0 05 A9 9F 8D 78 69 1B
86C8:AD 79 69 D0 05 A9 01 8D A3
86D0:79 69 C9 C8 D0 05 A9 C7 7D
86D8:8D 79 69 AD 78 69 8D 8C 24
86E0:92 AD 00 D0 8D 02 D0 AD 73
86E8:01 D0 8D 03 D0 AD 10 D0 BA
86F0:29 01 F0 08 AD 10 D0 09 C9
86F8:02 4C 01 87 AD 10 D0 29 2B
8700:FD 8D 10 D0 20 6C 89 20 66
8708:7F 8F CE 8C 92 D0 F5 AD CE
8710:00 D0 8D 04 D0 AD 01 D0 55
8718:8D 05 D0 AD 10 D0 29 01 3B
8720:F0 08 AD 10 D0 09 04 4C 5F
8728:2F 87 AD 10 D0 29 FB 8D 18
8730:10 D0 AD 79 69 8D 8C 92 F5
8738:20 3D 8A 20 7F 8F CE 8C 5E
8740:92 D0 F5 AD 00 D0 8D 06 CA
8748:D0 AD 01 D0 8D 07 D0 AD 30
8750:10 D0 29 01 F0 08 AD 10 E3
8758:D0 09 08 4C 63 87 AD 10 7C
8760:D0 29 F7 8D 10 D0 AD 78 91
8768:69 8D 8C 92 20 B6 89 20 59
8770:7F 8F CE 8C 92 D0 F5 AD 37
8778:00 D0 8D 08 D0 AD 01 D0 FD
8780:8D 09 D0 AD 10 D0 29 01 A4
8788:F0 08 AD 10 D0 09 10 4C DF
8790:97 87 AD 10 D0 29 EF 8D 9C
8798:10 D0 AD 79 69 8D 8C 92 5E
87A0:20 00 8A 20 7F 8F CE 8C 77
87A8:92 D0 F5 A9 47 8D C8 87 40
87B0:86 A9 46 8D C4 86 A9 56 F8
87B8:8D D3 86 A9 55 8D D7 86 06
87C0:20 A6 85 A5 C6 F0 03 4C E0
87C8:38 89 AD 45 03 8D 8D 92 5C
87D0:A5 02 8D 8E 92 A5 03 8D 8C
87D8:8F 92 A9 7A 85 02 A9 69 22
87E0:85 03 A9 00 A0 00 91 02 D7
87E8:8D 15 D0 AD 78 69 8D BD 3B
87F0:92 AD 79 69 8D BE 92 A0 A7
87F8:00 B1 FB 2D 42 03 F0 40 07
8800:CD 42 03 D0 27 A5 FD 18 D9
8808:69 00 85 FD A5 FE 69 D4 2F
8810:85 FE B1 FD 29 0F 8D 97 F1
8818:92 A5 FD 38 E9 00 85 FD 77
8820:A5 FE E9 D4 85 FE A9 97 69
8828:92 4C 42 88 29 AA F0 07 43
8830:B1 FD 29 0F 4C 42 88 B1 DD
8838:FD 4A 4A 4A 4A 4C 42 88 59
8840:A9 FF 91 02 A5 02 18 69 47
8848:01 85 02 A5 03 69 00 85 19
8850:03 CE BD 92 F0 16 A9 FF AA
8858:38 ED 42 03 8D A7 92 B1 5B
8860:FB 2D A7 92 91 FB 20 6C 02
```

```
8868:89 4C F7 87 AD 78 69 8D 78
8870:BD 92 CE BD 92 FØ Ø6 2Ø 3F
8878:8A 89 4C 72 88 CE BE 92 87
8880:FØ ØC 2Ø 3D 8A AD 78 69 4A
8888:8D BD 92 4C F7 87 AD 79 99
8890:69 8D BE 92 CE BE 92 FØ 42
8898:Ø6 2Ø 8A 4C 94 88 AD DØ
88AØ:8E 92 85 Ø2 AD 8F 92 85 C4
88A8:Ø3 A5 Ø2 8D 8E 92 A5 Ø3 CA
88BØ:8D 8F 92 AD 45 Ø3 8D 8D 78
88B8:92 8D 15 DØ 2Ø 56 95 26
88CØ:86 A5 C6 FØ Ø3 4C 38 89 A9
88C8:A9 7A 85 Ø2 A9 69 85 Ø3 1F
88DØ:AD 78 69 8D BD 92 AD 79 E9
88D8:69 8D BE 92 AØ ØØ B1 Ø2 6D
88EØ:C9 FF FØ Ø6 8D 45 Ø3 2Ø FC
88E8:8B 9Ø A5 Ø2 18 69 Ø1 85 A6
88FØ:Ø2 A5 Ø3 69 ØØ 85 Ø3 CE 4E
88F8:BD 92 FØ Ø6 2Ø 6C 89 4C 1E
89ØØ:DC 88 AD 78 69 8D BD 92 7Ø
89Ø8:CE BD 92 FØ Ø6 2Ø B6 89 FA
891Ø:4C Ø8 89 CE BE 92 FØ ØC 97
8918:2Ø 3D 8A AD 78 69 8D BD F8
892Ø:92 4C DC 88 AD 79 69 8D 67
8928:BE 92 CE BE 92 FØ Ø6 2Ø 89
893Ø:ØØ 8A 4C 2A 89 4C A9 88 6B
8938:A9 64 8D 66 Ø3 A9 ØØ 8D 9D
894Ø:67 Ø3 A9 Ø1 8D 15 DØ AD 1D
8948:8D 92 8D 45 Ø3 AD 8E 92 4B
895Ø:85 Ø2 AD 8F 92 85 Ø3 A9 AF
8958:AØ 8D CØ 86 A9 9F 8D C4 4B
896Ø:86 A9 C8 8D D3 86 A9 C7 E6
8968:8D D7 86 6Ø AD 3F Ø3 FØ 7Ø
897Ø:Ø7 AD 3E Ø3 C9 3F FØ 3D D4
8978:AD 3E Ø3 18 69 Ø2 8D 3E 8Ø
898Ø:Ø3 AD 3F Ø3 69 ØØ 8D 3F 3E
8988:Ø3 AD Ø3 4A 4A C9 ØØ 1Ø
899Ø:FØ Ø4 8D 42 Ø3 6Ø A9 CØ 1Ø
8998:8D 42 Ø3 A5 FB 18 69 Ø8 D8
89AØ:85 FB A5 FC 69 ØØ 85 FC 4D
89A8:A5 FD 18 69 Ø1 85 FD A5 67
89BØ:FE 69 ØØ 85 FE 6Ø AD 3F ØA
89B8:Ø3 DØ Ø7 AD 3E Ø3 C9 Ø1 CF
89CØ:FØ 3D AD 3E Ø3 18 69 Ø2 Ø4
89C8:8D 3E Ø3 AD 3F Ø3 69 ØØ 47
89DØ:8D 3F Ø3 AD 42 Ø3 ØA ØA F1
89D8:C9 ØØ FØ Ø4 8D 42 Ø3 6Ø ØB
89EØ:A9 ØØ 8D 42 Ø3 A5 FB 38 3E
89E8:E9 Ø8 85 FB A5 FC E9 ØØ 58
89FØ:85 FC A5 FD 38 E9 Ø1 85 8B
89F8:FD A5 FE E9 ØØ 85 FE 6Ø 67
8AØØ:AD 4Ø Ø3 C9 Ø1 FØ 35 CE FD
8AØ8:4Ø Ø3 CE 41 Ø3 DØ 2Ø A9 31
8A1Ø:Ø8 8D 41 Ø3 A5 FB 38 E9 5C
8A18:39 85 FB A5 FC 69 ØØ 85 1C
8A2Ø:FC A5 FD 38 E9 28 85 FD 59
8A28:A5 FE E9 ØØ 85 FE 6Ø A5 9B
8A3Ø:FB 38 E9 Ø1 85 FB A5 FC Ø3
8A38:E9 ØØ 85 FC 6Ø A9 Ø3 FF
8A4Ø:C9 C8 FØ 3Ø EE 4Ø Ø3 EE FA
8A48:41 Ø3 AD 41 Ø3 C9 Ø9 DØ AA
8A5Ø:2Ø A9 Ø1 8D 41 Ø3 A5 FB 36
8A58:18 69 39 85 FB A5 FC 69 2D
8A6Ø:Ø1 85 FC A5 FD 18 69 28 9C
8A68:85 FD A5 FE 69 ØØ 85 FE B9
8A7Ø:6Ø 2Ø FF 91 6Ø A9 ØØ 8D 1Ø
8A78:BF 92 4C 82 8A A9 ØØ 8D 4C
8A8Ø:BF 92 EE 2Ø DØ 2Ø 1B 92 C9
8A88:2Ø 31 92 A9 ØØ 85 FB A9 9E
8A9Ø:2Ø 85 FC A9 68 85 FD A9 5Ø
8A98:42 85 FE AØ Ø9 AD BF 92 E2
8AAØ:DØ Ø7 B1 FB 91 FD 4C AD AØ
8AA8:8A B1 FD 91 FB 2Ø FF 91 3A
8ABØ:2Ø ØD 92 A5 FC C9 CØ DØ 24
8AB8:E4 A5 FB C9 4Ø DØ DE A9 72
8ACØ:ØØ 85 FB A9 Ø4 85 FC AD 2F
8AC8:BF 92 DØ Ø7 B1 FB 91 FD 8B
8ADØ:4C D7 8A B1 FD 91 FB 2Ø BC
8AD8:FF 91 2Ø ØD 92 A5 FC C9 16
8AEØ:Ø7 DØ E4 A5 FB C9 E8 DØ 4E
8AE8:DE 2Ø ØD 92 A9 ØØ 8D 85 ED
8AFØ:FC AD BF 92 DØ Ø7 B1 FB 13
8AF8:91 FD 4C Ø1 8B B1 FD 91 AØ
```

```
8BØØ:FB 2Ø FF 91 2Ø ØD 92 A5 36
8BØ8:FC C9 DB DØ E4 A5 FB C9 18
8B1Ø:18 DØ DE 2Ø 26 92 2Ø 3C A5
8B18:92 CE 2Ø DØ AD BF 92 DØ 9F
8B2Ø:16 AD 9Ø 65 29 ØF 8D 9Ø 47
8B28:65 AD 21 DØ ØA ØA ØA ØA 25
8B3Ø:ØD 9Ø 65 8D 9Ø 65 6Ø AD FF
8B38:9Ø 65 4A 4A 4A 4A 8D 21 96
8B4Ø:DØ 6Ø A9 Ø1 8D 67 Ø3 A5 D2
8B48:FB 8D 9A 92 A5 FC 8D 9B 15
8B5Ø:92 AD 41 Ø3 8D 9C 92 AD 26
8B58:42 Ø3 8D 9D 92 AD 3E Ø3 A7
8B6Ø:8D 9E 92 AD 3F Ø3 8D 9F D3
8B68:92 AD 4Ø Ø3 8D AØ 92 A5 26
8B7Ø:FD 8D A1 92 A5 FE 8D A2 2E
8B78:92 AD 68 Ø3 8D A3 92 AD 4F
8B8Ø:69 Ø3 8D A9 92 A9 ØØ 8D 22
8B88:69 Ø3 8D 15 DØ A9 Ø1 8D A4
8B9Ø:68 Ø3 A9 ØØ 8D CØ 92 8D F3
8B98:C1 92 2Ø E8 8D 8D A4 92 46
8BAØ:29 ØF CD 45 Ø3 DØ Ø3 4C CB
8BA8:AD 8C AØ ØØ 8D DØ ØA AD 8Ø
8BBØ:3E Ø3 C9 Ø1 DØ Ø3 4C C7 E3
8BB8:8B 2Ø B6 89 2Ø E8 8D CD 9A
8BCØ:A4 92 FØ Ø6 2Ø 6C 89 A5 C6
8BC8:C6 FØ Ø3 4C AD 8C AØ ØØ 85
8BDØ:2Ø E8 8D CD A4 92 FØ Ø3 15
8BD8:4C 12 8C A9 ØØ 8D C2 92 15
8BEØ:8D C3 92 2Ø F6 8C 2Ø 6F 9D
8BE8:8D 2Ø 7F 8F AD 3F Ø3 FØ 19
8BFØ:ØA AD 3E Ø3 C9 3F DØ Ø3 6Ø
8BF8:4C 12 8C 2Ø 6C 89 2Ø E8 Ø1
8CØØ:8D CD A4 92 DØ ØC 2Ø F6 FE
8CØ8:8C 2Ø 6F 8D 2Ø 7F 8F 4C AØ
8C1Ø:EC 8B AD CØ 92 18 CD C1 96
8C18:92 DØ Ø3 4C 64 8C AE CØ 3E
8C2Ø:92 FØ F8 6Ø A5 FE 18 6Ø 7C
8C28:CA BD 6Ø 6D 85 FD CA BD 7Ø
8C3Ø:6Ø 6D 8D 4Ø Ø3 CA BD 6Ø A9
8C38:6D Ø3 3F Ø3 CA BD 6Ø 6D 6F
8C4Ø:8D 3E Ø3 CA BD 6Ø 6D 94
8C48:42 Ø3 CA BD 6Ø 6D 8D 41 8D
8C5Ø:Ø3 CA BD 6Ø 6D 85 FC CA A1
8C58:BD 6Ø 6D 85 FB CA 8E CØ 57
8C6Ø:92 4C AA 8B AE C1 92 DØ 56
8C68:Ø3 4C AD 8C BD 54 6F 85 38
8C7Ø:FE CA BD 54 6F 85 FD CA 11
8C78:BD 54 6F 8D 4Ø Ø3 CA BD AD
8C8Ø:54 6F 8D 3F Ø3 CA BD 54 58
8C88:6F 8D 3E Ø3 CA BD 54 6F 1A
8C9Ø:8D 42 Ø3 CA BD 54 6F 8D B9
8C98:41 Ø3 CA BD 54 6F 85 FC BØ
8CAØ:CA BD 54 6F 85 FB CA 8E 5Ø
8CA8:C1 92 4C AA 8B AD 9A 92 56
8CBØ:8D 9B 92 85 FC AD 4D
8CB8:9C 92 8D 41 Ø3 AD 9D 92 27
8CCØ:8D 42 Ø3 AD 9E 92 8D 3E Ø5
8CC8:Ø3 AD 9F 92 8D 3F Ø3 AD Ø9
8CDØ:AØ 92 8D 4Ø Ø3 AD A1 92 39
8CD8:85 FD AD A2 92 85 FE AD 6A
8CEØ:A3 92 8D 68 Ø3 AD A9 92 5D
8CE8:8D 69 Ø3 A9 ØØ 8D 67 Ø3 26
8CFØ:A9 Ø1 8D 15 DØ 6Ø AD 4Ø C7
8CF8:Ø3 C9 Ø1 FØ 28 2Ø ØØ 8A 81
8DØØ:2Ø E8 8D CD A4 92 FØ ØD 52
8DØ8:AD C2 92 FØ 15 A9 ØØ 8D 18
8D1Ø:C2 92 4C 22 8D AD C2 92 18
8D18:DØ Ø8 2Ø 26 8D A9 Ø1 8D A6
8D2Ø:C2 92 2Ø 3D 8A 6Ø AD CØ ØB
8D28:92 18 C9 F5 9Ø Ø1 6Ø AA 1F
8D3Ø:E8 A5 FB 9Ø 6Ø Ø6 E8 A5 B2
8D38:FC 9D 6Ø 6D E8 AD 41 Ø3 9F
8D4Ø:9D 6Ø 6D AD 42 Ø3 9D 98
8D48:6Ø Ø6 E8 AD 3E Ø3 9D 6Ø DØ
8D5Ø:6D E8 AD 3F Ø3 9D 6Ø 6D C2
8D58:E8 AD 4Ø Ø3 9D 6Ø 6D E8 BD
8D6Ø:AD 67 9D 6Ø 6D E8 A5 FE EØ
8D68:9D 6Ø 6D 8E CØ 92 6Ø AD BF
8D7Ø:4Ø Ø3 C9 C8 FØ 28 2Ø 3D D7
8D78:8A 2Ø E8 8D CD A4 92 FØ ED
8D8Ø:ØD AD C3 92 FØ 15 A9 ØØ C3
8D88:8D C3 92 4C 9B 8D AD C3 A4
8D9Ø:92 DØ Ø8 2Ø 9F 8D A9 Ø1 B3
```

```
8D98:8D C3 92 2Ø ØØ 8A 6Ø AD 58
8DAØ:C1 92 18 C9 F5 9Ø Ø1 6Ø 35
8DA8:AA E8 A5 FB 9D 54 6F E8 CD
8DBØ:A5 FC 9D 54 6F E8 AD 41 92
8DB8:Ø3 9D 54 6F E8 AD 42 Ø3 C3
8DCØ:9D 54 6F E8 AD 3E Ø3 9D 46
8DC8:54 6F E8 AD 3F Ø3 9D 54 77
8DDØ:6F E8 AD 4Ø Ø3 9D 54 6F 3E
8DD8:E8 A5 FD 9D 54 6F E8 A5 43
8DEØ:FE 9D 54 6F 8E C1 92 6Ø 65
8DE8:AØ ØØ B1 FB 2D 42 Ø3 DØ 93
8DFØ:Ø4 AD 21 DØ 6Ø CD 42 Ø3 6C
8DF8:DØ 25 A5 FD 18 69 ØØ 85 46
8EØØ:FD A5 FE 69 D4 85 FE B1 68
8EØ8:FD 29 ØF 8D A5 92 A5 FD E9
8E1Ø:38 E9 ØØ 85 FD A5 FE E9 8A
8E18:D4 85 FE AD A5 92 6Ø 29 1D
8E2Ø:2Ø AA FØ Ø5 B1 FD 29 ØF 6Ø
8E28:B1 FD 4A 4A 4A 4A 6Ø A9 71
8E3Ø:Ø5 A2 Ø8 AØ Ø5 2Ø BA FF A1
8E38:AD 49 Ø3 A2 4A AØ Ø3 2Ø Ø4
8E4Ø:BD FF 2Ø 1B 92 AD 48 Ø3 DØ
8E48:C9 86 ØF A9 68 85 FB EC
8E5Ø:A9 42 85 FC A2 78 AØ 69 F4
8E58:4C 69 8E A9 78 85 FB A9 DD
8E6Ø:69 85 FC AE 5C Ø3 AC 5D C3
8E68:Ø3 A9 FB 2Ø D8 FF 2Ø 26 2Ø
8E7Ø:92 2Ø B7 FF 8D 48 Ø3 2Ø 89
8E78:E7 FF 2Ø CC FF 9Ø Ø3 4C C4 FB
8E8Ø:92 C9 57 9Ø ØA 2Ø AØ 8E F1
8E88:2Ø ED 8E 2Ø 75 8A 6Ø 2Ø BB
8E9Ø:ED 8E A9 57 8D C4 92 A9 41
8E98:ØØ 8D 35 Ø3 2Ø 51 81 6Ø 99
8EAØ:2Ø 1B 92 2Ø 31 92 78 A5 53
8EA8:Ø1 29 FC 85 Ø1 A9 ØØ 85 BC
8EBØ:FB A9 AØ 85 FC A9 68 85 87
8EB8:FD A9 42 85 FE A9 4Ø 8D 8C
8ECØ:C5 92 A9 BF 8D C6 92 2Ø 63
8EC8:64 8F A9 ØØ 85 FB A9 EØ 81
8EDØ:85 FC A9 ØØ 8D C5 92 A9 84
8ED8:E7 8D C6 92 2Ø 64 8F A5 A6
8EEØ:Ø1 Ø9 Ø3 85 Ø1 2Ø 26 92 EØ
8EE8:2Ø 3C 92 58 6Ø 2Ø 1B 92 49
8EFØ:2Ø 31 92 6Ø 2Ø ØØ 9Ø Ø Ø 43
8EF8:FB A9 2Ø 85 FC A9 4Ø 82
8FØØ:85 FD A9 AØ 85 FE A9 4Ø 5C
8FØ8:8D C5 92 A9 3F 8D C6 92 9C
8F1Ø:2Ø 64 8F A9 ØØ 85 FB A9 9C
8F18:Ø4 85 FC A9 ØØ 85 FD A9 9Ø
8F2Ø:EØ 85 FE A9 E8 8D C5 92 27
8F28:A9 Ø7 8D C6 92 2Ø 64 8F 69
8F3Ø:A9 ØØ 85 FB A9 D8 85 FC 4D
8F38:A9 E8 8D C5 92 A9 DB 8D F4
8F4Ø:C6 92 2Ø 64 8F AD ØØ D8 BD
8F48:29 ØF 8D BF 94 92 21 DØ 19
8F5Ø:ØA ØA ØA ØA ØD 94 92 8D 46
8F58:E8 E3 2Ø 26 92 2Ø 3C 92 6B
8F6Ø:CE 2Ø DØ 6Ø AØ ØØ B1 FB 73
8F68:91 FD 2Ø FF 91 2Ø ØD 92 8D
8F7Ø:A5 FC CD C6 92 2Ø ED A5 21
8F78:FB CD C5 92 DØ E6 6Ø AD 7B
8F8Ø:69 Ø3 C9 57 DØ ØA A9 Ø2 D3
8F88:8D 87 92 AD 1D DØ Ø9 1F DA
8F9Ø:4C 9D 8F A9 ØD 8D 87 92 ØA
8F98:AD 1D DØ 29 EØ 8D 1D DØ CA
8FAØ:AD 3E Ø3 18 6D 8D 7 92 8D 44
8FA8:ØØ DØ AD 3F Ø3 69 ØØ 34
8FBØ:ØD A9 FF 8D F8 Ø7 AD 1Ø E8
8FB8:DØ 29 FE 4C C8 8F A9 FE Ø6
8FCØ:8D F8 Ø7 AD 1Ø DØ Ø9 Ø1 77
8FC8:8D 1Ø DØ 4Ø Ø3 18 69 4F
8FDØ:28 8D Ø1 DØ 67 Ø3 FØ 96
8FD8:ØA 2Ø 8B 9Ø AD 69 Ø3 C9 62
8FEØ:57 FØ Ø1 6Ø A5 FB 8D AE F4
8FE8:92 A5 FC 8D AF 92 A5 FD 44
8FFØ:8D BØ 92 A5 FE 8D B1 92 D3
8FF8:AD 41 Ø3 8D B2 92 AD 42 F5
9ØØØ:Ø3 8D B3 92 AD 3E Ø3 8D 9F
9ØØ8:B4 92 AD 3F Ø3 8D B5 92 1E
9Ø1Ø:AD 4Ø Ø3 8D B6 92 2Ø B6 48
9Ø18:89 2Ø 8B 9Ø 2Ø 3D 8A 2Ø AB
9Ø2Ø:8B 9Ø 2Ø B6 89 2Ø 8B 9Ø ØF
9Ø28:2Ø 3D 8A 2Ø 8B 9Ø 2Ø B6 91
```

```
9030:89 20 8B 90 20 3D 8A 20 C3
9038:8B 90 20 B6 89 20 8B 90 27
9040:20 3D 8A 20 8B 90 20 B6 A9
9048:89 20 8B 90 20 3D 8A 20 DB
9050:8B 90 20 B6 89 20 8B 90 3F
9058:AD AE 92 85 FB AD AF 92 2F
9060:85 FC AD B0 92 85 FD AD 98
9068:B1 92 85 FE AD B2 92 8D 92
9070:41 03 AD B3 92 8D 42 03 36
9078:AD B4 92 8D 3E 03 AD B5 D7
9080:92 8D 3F 03 AD B6 92 8D 61
9088:40 03 60 A0 00 20 47 92 42
9090:AD 68 03 D0 03 4C E1 91 AE
9098:AD 45 03 CD 21 D0 D0 03 10
90A0:4C E1 91 20 31 92 A5 FD B1
90A8:18 69 00 85 FD A5 FE 69 76
90B0:D4 85 FE B1 FD 8D 87 92 60
90B8:20 3C 92 AD 87 92 29 0F 0E
90C0:CD 45 03 F0 11 B1 FD 29 FD
90C8:0F CD 45 03 D0 10 A9 AA 82
90D0:8D 87 92 4C C4 91 A9 FF 71
90D8:8D 87 92 4C C4 91 B1 FD 87
90E0:4A 4A 4A 4A CD 45 03 D0 02
90E8:08 A9 55 8D 87 92 4C C4 DF
90F0:91 20 31 92 A5 FB 85 FD 58
90F8:A5 FC 85 FE AD 41 03 8D D2
9100:87 92 A9 00 8D 88 92 8D 02
9108:89 92 8D 8A 92 8D 8B 92 63
9110:CE 87 92 F0 10 A5 FB 38 29
9118:E9 01 85 FD A5 FE E9 00 FD
9120:85 FE 4C 10 91 B1 FD 8D 2D
9128:87 92 29 03 F0 0B C9 02 52
9130:F0 15 C9 01 F0 17 EE 8B A7
9138:92 4E 87 92 EE 87 92 AD B5
9140:87 92 F0 0F 4C 2A 91 EE F7
9148:88 92 4C 39 91 EE 89 92 5F
9150:4C 39 91 EE 8A 92 AD 8A A8
9158:92 C9 08 F0 10 A5 FD 18 72
9160:69 01 85 FD A5 FE 69 00 05
9168:85 FE 4C 25 91 20 3C 92 02
9170:AD 88 92 18 CD 89 92 B0 CA
9178:11 CD 8B 92 B0 15 B1 FD 6D
9180:29 F0 0D 45 03 91 FD 4C 11
9188:CE 90 AD 89 92 18 CD 8B A1
9190:92 90 1B 20 31 92 A5 FD A3
9198:18 69 00 85 FD A5 FE 69 68
91A0:D4 85 FE AD 45 03 91 FD A1
91A8:20 3C 92 AC D6 90 B1 FD 5C
91B0:29 0F 8D 88 92 AD 45 03 3F
91B8:0A 0A 0A 0A 0D 88 92 91 86
91C0:FD 4C E9 90 AD 42 03 2D E5
91C8:87 92 8D 87 92 A9 FF 38 F1
91D0:ED 42 03 8D 88 92 B1 FB A2
91D8:2D 88 92 0D 87 92 91 FB 7D
91E0:60 A9 FF 38 ED 42 03 8D 2E
91E8:87 92 B1 FB 2D 87 92 4C 63
91F0:DE 91 AE 46 03 AC 46 03 7C
91F8:88 D0 FD CA D0 F7 60 A5 CD
9200:FB 18 69 01 85 FB A5 FC CA
9208:69 00 85 FC 60 A5 FD 18 10
9210:69 01 85 FD A5 FE 69 00 B6
9218:85 FE 60 A5 FB 8D B9 92 42
9220:A5 FC 8D BA 92 60 AD B9 DF
9228:92 85 FB AD BA 92 85 FC 7A
9230:60 A5 FD 8D BB 92 A5 FE F9
9238:8D BC 92 60 AD BB 92 85 B2
9240:FD AD BC 92 85 FE 60 AD 27
9248:70 03 C9 57 F0 05 C9 58 9C
9250:F0 1B 60 CE 71 03 D0 14 FA
9258:A9 0A 8D 71 03 EE 45 03 FE
9260:AD 45 03 C9 10 D0 05 A9 22
9268:00 8D 45 03 60 CE 71 03 ED
9270:D0 0F AD 46 03 4A 4A 4A FB
9278:8D 71 03 A9 01 8D 68 03 CD
9280:60 A9 00 4C 7D 92 00 00 3B
9288:00 00 00 00 00 00 00 00 AD
9290:00 00 00 00 00 00 00 00 B5
9298:00 00 00 00 00 00 00 00 BD
92A0:00 00 00 00 00 00 00 00 C5
92A8:00 00 00 00 00 00 00 00 CD
92B0:00 00 00 00 00 00 00 00 D5
92B8:00 00 00 00 00 00 00 00 DD
92C0:00 00 00 00 00 00 00 00 E5
```

Color Pad 2.O

```
C000:C9 4C F0 14 C9 52 F0 13 65
C008:C9 54 F0 12 C9 49 F0 11 29
C010:C9 51 F0 10 20 E0 92 60 F3
C018:4C 27 C0 4C 30 C0 4C 39 BC
C020:C0 4C 3D C0 4C 46 C0 A9 6F
C028:00 8D 53 CE 20 4E C6 60 8C
C030:A9 01 8D 53 CE 20 4E C6 08
C038:60 20 38 CB 60 A9 05 8D F6
C040:9A CE 20 4F C0 60 A9 5A F0
C048:8D 9A CE 20 4F C0 60 20 71
C050:A6 85 A5 C6 F0 03 4C 48 1C
C058:C6 A9 00 8D 5A CE 20 56 25
C060:86 A5 C6 D0 03 4C CA C0 14
C068:20 E4 FF C9 2B F0 07 C9 C4
C070:2D F0 15 4C 9A C0 EE 5A 3C
C078:CE AD 5A CE C9 5B D0 05 67
C080:A9 5A 8D 5A CE 4C 5E C0 E9
C088:CE 5A CE AD 5A CE C9 FF 5E
C090:D0 05 A9 00 8D 5A CE 4C B0
C098:5E C0 18 C9 30 B0 03 4C AF
C0A0:AA C0 18 C9 3A B0 03 4C 2E
C0A8:AD C0 4C 48 C6 A2 00 8E 8E
C0B0:5A CE 38 E9 30 AA E0 00 A6
C0B8:F0 0D AD 5A CE 18 69 0A 05
C0C0:8D 5A CE CA 4C B6 C0 4C 31
C0C8:5E C0 A5 FB 8D 65 CE A5 63
C0D0:FC 8D 66 CE A5 FD 8D 67 95
C0D8:CE A5 FE 8D 68 CE AD 3E FB
C0E0:03 8D 69 CE AD 3F 03 8D 5F
C0E8:6A CE AD 40 03 8D 6B CE 01
C0F0:AD 42 03 8D 6D CE AD 41 56
C0F8:03 8D 6F CE AD 78 69 4A A6
C100:8D 5B CE C9 00 F0 0B 8D FE
C108:FC CE 20 6C 89 CE FC CD 97
C110:D0 F8 AD 79 69 4A 8D 5D 74
C118:CE C9 00 F0 33 8D FC CD 1C
C120:20 3D 8A CE FC CD D0 F8 FA
C128:AC 5D CE A9 00 20 91 B3 25
C130:20 0F BC A0 75 A9 00 20 9B
C138:91 B3 20 2B BA 20 0F BC 59
C140:A0 5A A9 00 20 91 B3 20 AE
C148:14 BB 20 53 B8 A1 8C 5D BC
C150:A5 FB 8D 71 CE A5 FC 8D 03
C158:72 CE A5 FD 8D 73 CE A5 DA
C160:FE 8D 74 CE AD 3E 03 8D 3C
C168:75 CE AD 3F 03 8D 76 CE 0E
C170:AD 40 03 8D 77 CE AD 42 A8
C178:03 8D 79 CE AD 41 03 8D 03
C180:7B CE AD AE A9 A8 20 A2 47
C188:BB 20 0F BC A0 B4 A9 00 CA
C190:20 91 B3 20 14 BB A2 00 D5
C198:B5 61 9D 9C CE E8 E0 05 AD
C1A0:D0 F6 AC 5D CE A9 00 20 F2
C1A8:91 B3 20 0F BC AC 5B CE F4
C1B0:A9 00 20 91 B3 20 53 B8 A3
C1B8:20 58 BC A2 00 B5 61 9D 5B
C1C0:A1 CE E8 E0 05 D0 F6 A9 F6
C1C8:8D 5F CE A9 00 8D 60 51
C1D0:CE A5 C6 F0 06 20 E4 FF 87
C1D8:4C B9 C5 AD 5F CE 8D 8F 65
C1E0:CE AD 60 CE 8D 90 CE 20 9C
C1E8:F3 C5 AD 8D CE C9 02 F0 F8
C1F0:15 C9 03 F0 1D C9 04 F0 E9
C1F8:25 C9 05 F0 2D AD 5F CE DE
C200:8D 91 CE 4C 33 C2 A9 B4 FB
C208:38 ED 5F CE 8D 91 CE 4C 9A
C210:33 C2 AD 5F CE 38 E9 B4 6B
C218:8D 91 CE 4C 33 C2 A9 68 C7
C220:38 ED 5F CE 8D 91 CE 4C B2
C228:33 C2 AD 5F CE 38 E9 68 37
C230:8D 91 CE AC 91 CE A9 00 A0
C238:20 91 B3 A2 00 BD 9C CE 9C
C240:95 69 E8 E0 05 D0 F6 A5 15
C248:61 20 2B BA 20 64 E2 20 10
C250:0F BC AC 5B CE A9 00 20 15
C258:91 B3 20 2B BA 20 0F BC 60
C260:61 9D A6 CE E8 E0 05 D0 65
C268:F6 20 2B BC 8D 89 CE 20 F2
C270:0F BC A5 61 20 2B BA A2 3D
C278:00 B5 61 9D B0 CE E8 E0 E4
C280:05 D0 F6 AC 91 CE A9 00 81
```

```
C288:20 91 B3 A2 00 BD 9C CE 22
C290:95 69 E8 E0 05 D0 F6 A5 65
C298:61 20 2B BA 20 6B E2 20 7C
C2A0:0F BC AC 5D CE A9 00 20 85
C2A8:91 B3 20 2B BA A2 00 B5 B0
C2B0:61 9D AB CE E8 E0 05 D0 56
C2B8:F6 20 0F BC A5 61 20 2B 8D
C2C0:BA A2 00 BD B0 CE 95 69 7D
C2C8:E8 E0 05 D0 F6 A5 61 20 D9
C2D0:6A B8 20 71 BF A2 00 B5 13
C2D8:61 9D B5 CE E8 E0 05 D0 BF
C2E0:F6 AD 89 CE C9 00 D0 13 6E
C2E8:A0 01 A9 00 20 91 B3 A2 85
C2F0:00 B5 61 9D A6 CE E8 E0 0D
C2F8:05 D0 F6 A2 00 BD A6 CE 51
C300:95 61 BD AB CE 95 69 E8 A5
C308:E0 05 D0 F1 20 14 BB 20 63
C310:0E E3 20 0F BC A0 B4 A9 08
C318:00 20 91 B3 20 2B BA 20 58
C320:0F BC AC 5B CE A9 A8 20 A2 30
C328:BB 20 14 BB A2 00 58 BC 20 CF
C330:AA B1 8C 7D CE 8D 7E CE 5B
C338:AD 5F CE 8D 8F CE AD 60 94
C340:CE 8D 90 CE 20 F3 C5 AD 9B
C348:8D CE C9 02 F0 17 C9 03 1E
C350:F0 27 C9 04 F0 37 AD 7D D0
C358:CE 8D 91 CE AD 7E CE 8D 5C
C360:92 CE 4C 9E C3 A9 B4 38 BC
C368:ED 7D CE 8D 91 CE A9 00 14
C370:ED 7E CE 8D 92 CE 4C 9E 48
C378:C3 A9 B4 18 6D 7D CE 8D F0
C380:91 CE A9 00 6D 7E CE 8D 4A
C388:92 CE 4C 9E C3 A9 68 38 4E
C390:ED 7D CE 8D 91 CE A9 01 3D
C398:ED 7E CE 8D 92 CE AD 91 26
C3A0:CE 18 6D 5A CE 8D 91 CE 87
C3A8:AD 92 CE 69 00 8D 92 CE 46
C3B0:AD 91 CE 8D 8F CE AD 92 CB
C3B8:CE 8D 90 CE 20 F3 C5 A9 10
C3C0:00 8D 80 CE AD 8D CE C9 B3
C3C8:02 F0 1D C9 03 F0 2F C9 D1
C3D0:04 F0 3F C9 05 F0 51 AD 57
C3D8:91 CE 8D 7F CE A9 01 8D 33
C3E0:83 CE 8D 87 CE 4C 39 C4 E6
C3E8:A9 B4 38 ED 91 CE 8D 7F BA
C3F0:CE A9 00 8D 83 CE A9 01 CE
C3F8:8D 87 CE 4C 39 C4 A9 91 91
C400:CE 38 E9 B4 8D 7F CE A9 39
C408:00 8D 83 CE 8D 87 CE 4C C6
C410:39 C4 A9 68 38 ED 91 CE 8E
C418:8D 7F CE A9 01 8D 83 CE 8D
C420:A9 00 8D 87 CE 4C 39 C4 87
C428:AD 91 CE 38 E9 68 8D 7F D5
C430:CE A9 01 8D 83 CE 8D 87 7E
C438:CE AC 7F CE A9 00 20 91 50
C440:B3 A2 00 BD 9C CE 95 69 DC
C448:E8 E0 05 D0 F6 A5 61 20 5D
C450:2B BA 20 64 E2 A2 00 BD C7
C458:B5 CE 95 69 E8 E0 05 D0 5F
C460:F6 A5 61 20 2B BA 20 AA 2C
C468:B1 8C 81 CE AC 7F CE A9 B5
C470:00 20 91 B3 A2 00 BD 9C 9C
C478:CE 95 69 E8 E0 05 D0 F6 3E
C480:A5 61 20 2B BA 20 6B E2 FB
C488:A2 00 BD B5 CE 95 69 E8 FE
C490:E0 05 D0 F6 A5 61 20 2B 73
C498:BA 20 0F BC A0 5A A9 00 F6
C4A0:20 91 B3 20 2B BA 20 0F AA
C4A8:BC A0 75 A9 00 20 91 B3 59
C4B0:20 14 BB 20 AA B1 8C 85 83
C4B8:CE AD 71 CE 85 FB AD 72 1A
C4C0:CE 85 FC AD 73 CE 85 FD 6D
C4C8:AD 74 CE 85 FE AD 75 CE E0
C4D0:8D 3E 03 AD 76 CE 8D 3F 35
C4D8:03 AD 77 CE 8D 40 03 AD 4C
C4E0:79 CE 8D 42 03 AD 7B CE 45
C4E8:8D 41 03 AD 81 CE F0 13 01
C4F0:AD 83 CE D0 06 20 B6 89 C0
C4F8:4C FE C4 20 6C 89 CE 81 AB
C500:CE D0 ED AD 85 CE F0 13 1C
C508:AD 87 CE D0 06 20 3D 8A BF
C510:4C 16 C5 20 20 8A CE 85 4F
C518:CE D0 ED AD 5F CE D0 3A E9
```

www.commodore.ca

```
C520:AD 60 CE DØ 35 A5 FB 8D 47
C528:2E CE A5 FC 8D 2F CE A5 6F
C530:FD 8D 30 CE A5 FE 8D 31 86
C538:CE AD 3E 03 8D 32 CE AD ØF
C540:3F 03 8D 33 CE AD 40 03 C1
C548:8D 34 CE AD 42 03 8D 36 CB
C550:CE A5 41 03 8D 35 CE 4C 32
C558:98 C5 AD 2E CE 8D 37 CE 24
C560:AD 2F CE 8D 38 CE AD 30 C9
C568:CE 8D 39 CE AD 31 CE 8D 30
C570:3A CE AD 32 CE 8D 3B CE 97
C578:AD 33 CE AD 3C CE 8D 34 07
C580:CE 8D 3D CE AD 36 CE 8D DC
C588:40 CE AD 35 CE 8D 3F CE EA
C590:A9 00 8D 41 CE 20 2F C8 D4
C598:AD 5F CE 18 6D 9A CE 8D 2F
C5A0:5F CE AD 60 CE 69 00 8D F4
C5A8:60 CE 18 C9 6C BØ 03 4C D1
C5B0:CE 18 C9 6C BØ 03 4C D1 A5
C5B8:C1 AD 65 CE 85 FB AD 66 08
C5C0:CE 85 FC AD 67 CE 85 FD ØF
C5C8:AD 68 CE 85 FE AD 69 CE C7
C5D0:8D 3E 03 AD 6A CE 8D 3F D6
C5D8:03 AD 6B CE 8D 40 03 AD CC
C5E0:6D CE 8D 42 03 AD 6F CE 29
C5E8:8D 41 03 A9 1E 8D 15 DØ A8
C5F0:4C 59 CØ AD 90 CE DØ 10 5D
C5F8:AD 8F CE 18 C9 5A BØ 08 BB
C600:A9 01 8D 8C CE 4C 47 C6 2A
C608:AD 90 CE DØ 10 AD 8F CE 9C
C610:18 C9 B4 BØ 08 A9 02 8D 36
C618:8D CE 4C 47 C6 AD 90 CE FA
C620:FØ 08 8F CE AD 90 CE ØE 4F
C628:08 A9 03 8D 8D CE 4C 02
C630:47 C6 8F CE 18 C9 68 94
C638:BØ 08 A9 04 8D 8D CE 4C 22
C640:47 C6 A9 05 8D 8D CE 60 49
C648:A9 01 8D 15 DØ 60 20 06 3C
C650:86 AD 27 DØ 8D 28 DØ A9 D6
C658:03 8D 15 DØ 20 10 C8 A5 F2
C660:FB 8D 2E CE A5 FC 8D 2F 8E
C668:FC 8D 2F CE A5 FD 8D 3B
C670:FD 8D 30 CE 8D 39 CE A5 E7
C678:FE 8D 31 CE 8D 3A CE AD 9C
C680:3E 03 8D 32 CE 8D 3B CE B4
C688:AD 3F 03 8D 33 CE 8D 3C 22
C690:CE AD 40 03 8D 34 CE 8D 91
C698:3D 03 8D 42 03 8D 36 CE DB
C6A0:8D 40 CE AD 41 03 8D 35 20
C6A8:CE 8D 3F CE A9 00 8D 67 A5
C6B0:03 8D 41 CE 20 F2 91 AD D5
C6B8:77 02 FØ 06 A5 C6 DØ 06 98
C6C0:20 E4 FF 4C FF C6 4C 29 39
C6C8:C8 AD 00 DC 8D 42 CE 29 30
C6D0:10 FØ 08 A9 01 8D 66 03 4C
C6D8:4C E2 C7 66 03 FØ 03 4F
C6E0:4C E2 C7 A9 64 8D 66 03 ØA
C6E8:20 2F C8 AD 53 CE DØ ØC C9
C6F0:A9 01 85 C6 A9 4C 8D 77 C1
C6F8:02 4C 26 C6 4C E2 C7 AD ØB
C700:41 CE DØ 03 4C 76 C7 A5 9F
C708:FB 8D 37 CE A5 FC 8D 38 41
C710:CE A5 FD 8D 39 CE A5 FE 58
C718:8D 3A CE AD 3E 03 8D 3B 06
C720:CE AD 3F 03 8D 3C CE AD 43
C728:40 03 8D 3D CE AD 41 03 DØ
C730:8D 3F CE AD 42 03 8D 40 84
C738:CE AD 2E CE 85 FB AD 2F F3
C740:CE 85 FC AD 30 CE 85 FD D8
C748:AD 31 CE 85 FE AD 32 CE ØF
C750:8D 3E 03 AD 33 CE 8D 3F AØ
C758:03 AD 34 CE 8D 40 03 AD 69
C760:35 CE 8D 41 03 AD 36 CE ØE
C768:8D 42 03 A9 00 8D 41 CE D1
C770:20 10 C8 4C E2 C7 A5 FB 6F
C778:8D 2E CE A5 FC 8D 2F CE D9
C780:A5 FD 8D 30 CE A5 FE 8D AF
C788:31 CE AD 3E 03 8D 32 CE 7F
C790:AD 3F 03 8D 33 CE AD 40 70
C798:03 8D 34 CE AD 41 03 8D 86
C7A0:35 CE AD 42 03 8D 36 CE E1
C7A8:AD 37 CE 85 FB AD 38 CE E4
C7B0:85 FC AD 39 CE 85 FD AD C1

C7B8:3A CE 85 FE AD 3B CE 8D 3F
C7C0:3E 03 AD 3C CE 8D 3F 03 D7
C7C8:AD 3D CE 8D 40 03 AD 3F D9
C7D0:8D 41 03 AD 40 CE 8D 1D
C7D8:42 03 A9 01 8D 41 CE 20 BE
C7E0:10 C8 AD 42 CE 29 08 DØ 80
C7E8:03 20 65 89 AD 42 CE 29 65
C7F0:04 DØ 03 20 B6 89 AD 42 92
C7F8:CE 29 01 DØ 03 20 00 8A 8A
C800:AD 42 CE 29 02 DØ 03 20 DE
C808:3D 8A 20 7F 8F 4C B4 C6 B4
C810:AD 10 DØ 29 01 FØ 02 A9 A2
C818:03 8D 10 DØ AD 00 DØ 8D 3A
C820:02 DØ AD 01 DØ 8D 03 DØ 40
C828:60 A9 01 8D 15 DØ 60 20 1A
C830:7F 8F A9 01 8D 67 03 AD 68
C838:15 DØ 8D 43 CE A9 00 8D 19
C840:15 DØ AD 41 CE FØ 35 A5 A4
C848:FB 8D 37 CE A5 FC 8D 38 83
C850:CE A5 FD 8D 39 CE A5 FE 9A
C858:8D 3A CE AD 3E 03 8D 3B 48
C860:CE AD 3F 03 8D 3C CE AD 85
C868:40 03 8D 3D CE AD 42 03 15
C870:8D 40 CE AD 41 03 8D 3F FD
C878:CE 4C AE C8 A5 FB 8D 2E 4D
C880:CE A5 FC 8D 2F CE A5 FD 59
C888:8D 30 CE A5 FE 8D 31 CE 80
C890:AD 3E 03 8D 32 CE AD 3F 29
C898:03 8D 33 CE AD 40 03 8D 64
C8A0:34 CE AD 42 03 8D 36 CE 63
C8A8:AD 41 03 8D 35 CE 8D 34 ØF
C8B0:CE 18 CD 3D CE BØ 12 A9 44
C8B8:00 8D 51 CE AD 3D CE 38 FC
C8C0:ED 34 CE 8D 46 CE 4C D8 E7
C8C8:C8 A9 01 8D 51 CE AD 34 77
C8D0:CE 38 ED 3D CE 8D 46 CE 71
C8D8:AD 33 CE CD 3C CE FØ 07 CA
C8E0:C9 01 27 4C FØ C8 32 CE 82
C8E8:32 CE 18 CD 3B CE BØ 1B B8
C8F0:A9 01 8D 52 CE AD 3B CE EØ
C8F8:38 ED 32 CE 8D 44 CE AD 1E
C900:3C CE 38 ED 33 CE 8D 45 5C
C908:4C 23 C9 A9 00 8D 52 CE 08
C910:AD 32 CE 38 ED 3B CE 8D EB
C918:44 CE AD 33 CE ED 3C CE DF
C920:8D 45 CE A5 2E CE 85 FB 34
C928:AD 2F CE 85 FC AD 30 CE 5E
C930:85 FD AD 31 CE 85 FE AD 07
C938:32 CE 8D 3E 03 AD 33 CE 32
C940:8D 3F 03 AD 34 CE 8D 40 DD
C948:03 AD 36 CE 8D 42 03 AD A5
C950:35 CE 8D 41 03 A9 00 8D 44
C958:48 CE 8D 49 CE 8D 47 CE 14
C960:8D 4D CE 8D 4A CE 8D 4B B4
C968:CE 8D 4C CE 8D 4E CE AD 2E
C970:45 CE 8D 45 CE AD 44 CE DØ
C978:44 CE 6A 8D 44 CE AD 45 06
C980:CE DØ 11 AD 44 CE 18 CD 08
C988:46 CE BØ 08 A9 01 8D 50 46
C990:CE 4C 99 C9 A9 00 8D 50 27
C998:CE 20 7F 8F A5 C6 FØ 08 B6
C9A0:A9 01 8D 4A CE 8D 4B CE B1
C9A8:AD 4D CE CD 45 CE DØ ØD 31
C9B0:AD 4C CE CD 44 CE DØ 05 88
C9B8:A9 01 8D 4A CE AD 4E CE 50
C9C0:CD 46 CE DØ 05 A9 01 8D 12
C9C8:4B CE AD 4A CE FØ 08 AD 08
C9D0:4B CE FØ 03 4C BD CA AD A8
C9D8:50 CE FØ 03 4C 4E CA EE B6
C9E0:4C DØ 03 EE 4D CE AD 90
C9E8:52 CE FØ 06 20 6C 89 4C E9
C9F0:F5 C9 20 B6 89 AD 46 CE BF
C9F8:DØ 03 4C 99 C9 AD 48 3D
CA00:18 6D 46 CE 8D 48 CE AD 8B
CA08:49 CE 69 00 8D 49 CE 20 20
CA10:45 CE FØ 07 C9 00 DØ ØF 89
CA18:4C 99 C9 AD 48 CE 18 CD C9
CA20:44 CE BØ 03 4C 99 C9 EE 1D
CA28:4E CE AD 51 CE FØ 06 20 C9
CA30:00 8A 4C 38 CA 20 3D 8A 51
CA38:AD 48 CE 38 ED 44 CE 8D BF
CA40:48 CE AD 49 CE ED 45 CE 7F
CA48:8D 49 CE 4C 99 C9 EE 4E B5

CA50:CE AD 51 CE FØ 06 20 00 AF
CA58:8A 4C 5F CA 20 3D 8A AD 97
CA60:44 CE DØ 03 4C 99 C9 AD 20
CA68:48 CE 18 6D 44 CE 8D 48 70
CA70:CE AD 49 CE 6D 45 CE 8D 9A
CA78:49 CE CD 47 CE FØ 07 C9 A6
CA80:00 DØ ØF 4C 99 C9 AD 48 88
CA88:CE 18 CD 46 CE BØ 03 4C 35
CA90:99 C9 EE 4C CE DØ 03 EE B6
CA98:4D CE AD 52 CE FØ 06 20 C9
CAA0:6C 89 4C A8 CA 20 B6 89 BØ
CAA8:AD 48 38 ED 46 CE 8D 38
CAB0:48 CE AD 49 CE E9 00 8D 14
CAB8:49 CE 4C 99 C9 AD 41 CE 20
CAC0:FØ 03 4C FA CA AD 2E CE 01
CAC8:85 FB AD 2F CE 85 FC AD FC
CAD0:30 CE 85 FD AD 31 CE 85 18
CAD8:FE AD 32 CE 8D 3E 03 AD A5
CAE0:33 CE AD 3F 03 AD 34 CE 6F
CAE8:8D 40 03 AD 36 CE 8D 42 DA
CAF0:03 AD 35 CE 8D 41 03 4C CA
CAF8:2C CB AD 37 CE 85 FB AD F2
CB00:38 CE 85 FC AD 39 CE 85 5D
CB08:FD AD 3A CE 85 FE AD 3B FC
CB10:CE 8D 3E 03 AD 3C CE 8D F3
CB18:3F 03 AD 3D CE 8D 40 03 F9
CB20:AD 40 CE 8D 42 03 AD 3F ØA
CB28:CE 8D 41 03 AD 43 CE 8D 88
CB30:15 DØ A9 00 8D 67 03 60 2C
CB38:AD 91 CE 8D 54 CE 3F 8D F4
CB40:FC CD A5 40 8D FD CD 30 30
CB48:69 03 8D 06 CE A9 00 8D 12
CB50:69 03 A2 40 9D 80 03 CA 75
CB58:DØ FA A9 ØE 8D 19 F9 07 39
CB60:EØ 8D 9D 03 8D AØ 03 AD 52
CB68:27 DØ 8D 28 DØ A9 02 8D BA
CB70:15 DØ A9 00 8D 67 03 20 2C
CB78:7F 8F A9 68 03 FØ 05 A9 7F
CB80:1F 4C 86 CB A9 18 8D 9C AD
CB88:03 8D 9F 03 AD 00 DØ 8D C5
CB90:02 DØ AD 01 DØ 8D 03 DØ B6
CB98:AD 10 DØ 29 01 FØ 02 A9 31
CBA0:03 8D 10 DØ 20 E4 FF FØ B1
CBA8:CE C9 5E DØ 10 EE 54 CE A6
CBB0:A2 14 EC 54 CE DØ 14 8E F1
CBB8:54 CE 4C CB CB C9 5F DØ 89
CBC0:ØA CE 54 CE DØ 05 A2 01 69
CBC8:8E 54 CE C9 85 DØ 03 4C F4
CBD0:D6 CD C9 91 FØ 1D C9 1D 3A
CBD8:FØ 1C C9 1D FØ 1B C9 9D 2D
CBE0:FØ 1A C9 86 FØ ØA C9 ØD 69
CBE8:FØ 03 4C AØ CC 4C FF CB BD
CBF0:4C 1C CC 4C 2E CC 4C 49 9A
CBF8:CC 4C 64 CC 4C 82 CC A9 13
CC00:AØ 8D 55 CE 20 B6 89 CE A2
CC08:55 CE DØ F8 A9 AØ 8D 55 E9
CC10:CE 20 B6 89 CE 55 CE DØ C2
CC18:F8 4C 49 CC AD 68 03 FØ 3D
CC20:08 A9 00 8D 68 03 4C 77 60
CC28:CB A9 01 4C 23 CC AD 54 F2
CC30:CE 8D 57 CE A9 ØA 8D 55 52
CC38:CE 20 00 8A CE 55 CE DØ 24
CC40:F8 CE 57 CE DØ EE 4C 77 34
CC48:CB AD 54 CE 8D 57 CE A9 BB
CC50:ØA 8D 55 CE 20 3D 8A CE C3
CC58:55 CE DØ F8 CE 57 CE DØ 3C
CC60:EE 4C 77 CB AD 54 CE 8D 1A
CC68:56 CE A9 08 8D 55 CE 20 16
CC70:6C 89 CE 55 CE DØ F8 CE 4C
CC78:56 CE DØ EE 20 6C 89 4C ØC
CC80:77 CB AD 54 CE 8D 56 CE EB
CC88:A9 08 8D 55 CE 20 B6 89 ED
CC90:CE 55 CE DØ F8 CE 56 CE 4C
CC98:DØ EE 20 B6 89 4C 77 CB FD
CCA0:18 C9 20 BØ 03 4C 77 CB CB
CCA8:18 C9 5B BØ 00 DØ C9 40 BØ F7
CCB0:03 4C CA CC 38 E9 40 4C 3B
CCB8:CA CC 18 C9 C1 BØ 03 4C AD
CCC0:77 CB 18 C9 DB BØ D6 38 30
CCC8:E9 80 8D 58 CE DØ 55 CE D4
CCD0:A9 00 85 3F A9 D8 85 40 DF
CCD8:A5 3F 18 69 08 85 3F A5 29
CCE0:40 69 00 85 40 CE 55 CE 04
```

```
9030:89 20 8B 90 20 3D 8A 20 C3
9038:8B 90 20 B6 89 20 8B 90 27
9040:20 3D 8A 20 8B 90 20 B6 A9
9048:89 20 8B 90 20 3D 8A 20 DB
9050:8B 90 20 B6 89 20 8B 90 3F
9058:AD AE 92 85 FB AD AF 92 2F
9060:85 FC AD B0 92 85 FD AD 98
9068:B1 92 85 FE AD B2 92 8D 92
9070:41 03 AD B3 92 8D 42 03 36
9078:AD B4 92 8D 3E 03 AD B5 D7
9080:92 8D 3F 03 AD B6 92 8D 61
9088:40 03 60 A0 00 20 47 92 42
9090:AD 68 03 D0 03 4C E1 91 AE
9098:AD 45 03 CD 21 D0 D0 03 10
90A0:4C E1 91 20 31 92 A5 FD B1
90A8:18 69 00 85 FD A5 FE 69 76
90B0:D4 85 FE B1 FD 8D 87 92 60
90B8:20 3C 92 AD 87 92 29 0F 0E
90C0:CD 45 03 F0 11 B1 FD 29 FD
90C8:0F CD 45 03 D0 10 A9 AA 82
90D0:8D 87 92 4C C4 91 A9 FF 71
90D8:8D 87 92 4C C4 91 B1 FD 02
90E0:4A 4A 4A 4A CD 45 03 D0 02
90E8:08 A9 55 8D 87 92 4C C4 DF
90F0:91 20 31 92 A5 FB 85 FD 58
90F8:A5 FC 85 FE AD 41 03 8D D2
9100:87 92 A9 00 8D 88 92 8D 02
9108:89 92 8D 8A 92 8D 8B 92 63
9110:CE 87 92 F0 10 A5 FD 38 29
9118:E9 01 85 FD A5 FE E9 00 FD
9120:85 FE 4C 10 91 B1 FD 8D 2D
9128:87 92 29 03 F0 0B C9 02 52
9130:F0 15 C9 01 F0 17 EE 8B A7
9138:92 4E 87 92 4E 87 92 AD B5
9140:87 92 0F 4C 2A 91 EE EE F7
9148:88 92 4C 39 91 EE 89 92 5F
9150:4C 39 91 EE 8A 92 AD 8A 8D
9158:92 C9 08 F0 10 A5 FD 18 72
9160:69 01 85 FD A5 FE 69 00 05
9168:85 FE 4C 25 91 20 3C 92 02
9170:AD 88 92 18 CD 89 92 B0 CA
9178:11 20 31 92 8D 88 92 B1 FD 60
9180:29 F0 0D 45 03 91 FD 4C 11
9188:CE 90 AD 89 92 18 CD 8B A1
9190:92 90 1B 20 31 92 A5 FD A3
9198:18 69 00 85 FD A5 FE 69 68
91A0:D4 85 FE AD 45 03 91 FD A1
91A8:20 3C 92 4C D6 90 B1 FD 5C
91B0:29 0F 8D 88 92 AD 45 03 3F
91B8:0A 0A 0A 0A 0D 88 92 91 86
91C0:FD 4C E9 90 AD 42 03 2D E5
91C8:87 92 8D 87 92 A9 FF 38 F1
91D0:ED 42 03 8D 88 92 B1 FB A2
91D8:2D 88 92 8D 87 92 91 FB 7D
91E0:60 A9 FF 38 ED 42 03 8D 2E
91E8:87 92 B1 FB 2D 87 92 4C 63
91F0:DE 91 4E 46 03 AD 46 03 7C
91F8:88 D0 FD CA D0 F7 60 A5 CD
9200:FB 18 69 01 85 FB A5 FC CA
9208:69 00 85 FC 60 A5 FD 18 10
9210:69 01 85 FD A5 FE 69 00 B6
9218:85 FE 60 A5 FB 8D B9 92 42
9220:A5 FC 8D BA 92 60 AD B9 DF
9228:92 85 FB AD BA 92 85 FC 7A
9230:60 A5 FD 8D BB 92 A5 FE F9
9238:8D BC 92 60 AD BB 92 85 B2
9240:FD AD BC 92 85 FE 60 AD 27
9248:70 03 C9 57 F0 05 C9 58 9C
9250:F0 1B 60 CE 71 03 D0 14 FA
9258:A9 0A 8D 71 03 EE 45 03 FE
9260:AD 45 03 C9 10 D0 05 A9 22
9268:00 8D 45 03 60 CE 71 03 ED
9270:D0 0F AD 46 03 4A 4A 4A FB
9278:8D 71 03 A9 01 8D 68 03 CD
9280:60 A9 00 4C 7D 92 00 00 3B
9288:00 00 00 00 00 00 00 00 AD
9290:00 00 00 00 00 00 00 00 B5
9298:00 00 00 00 00 00 00 00 BD
92A0:00 00 00 00 00 00 00 00 C5
92A8:00 00 00 00 00 00 00 00 CD
92B0:00 00 00 00 00 00 00 00 D5
92B8:00 00 00 00 00 00 00 00 DD
92C0:00 00 00 00 00 00 00 00 E5
```

Color Pad 2.O

```
C000:C9 4C F0 14 C9 52 F0 13 65
C008:C9 54 F0 12 C9 49 F0 11 29
C010:C9 20 10 20 60 92 60 F3
C018:4C 27 C0 4C 30 C0 4C 39 BC
C020:C0 4C 3D C0 4C 46 C0 A9 6F
C028:00 8D 53 CE 20 4E C6 60 8C
C030:A9 01 8D 53 CE 20 4E C6 08
C038:60 20 38 CB 60 A9 05 8D F6
C040:9A CE 20 4F C0 60 A9 5A F0
C048:8D 9A CE 20 4F C0 60 20 71
C050:A6 85 A5 C6 F0 03 4C 48 1C
C058:C6 A9 00 8D 5A CE 20 56 25
C060:86 A5 C6 D0 03 4C CA C0 14
C068:20 E4 FF C9 2B F0 07 C9 C4
C070:2D F0 15 4C 9A C0 EE 5A 3C
C078:CE AD 5A CE C9 5B D0 05 67
C080:A9 5A 8D 5A CE 4C 5E C0 E9
C088:CE 5A CE AD 5A CE C9 FF 5E
C090:D0 05 A9 00 8D 5A CE 4C B0
C098:5E C0 18 C9 30 B0 03 4C AF
C0A0:AA C0 18 C9 3A B0 03 4C 2E
C0A8:AD C0 4C 48 C6 A2 00 8E 8E
C0B0:5A CE 38 E9 30 AA E0 00 A6
C0B8:F0 0D AD 5A CE 18 69 0A 05
C0C0:8D 5A CE CA 4C B6 C0 4C 31
C0C8:5E C0 A5 FB 8D 65 CE A5 63
C0D0:FC 8D 66 CE A5 FD 8D 67 95
C0D8:CE A5 FE 8D 68 CE AD 3E FB
C0E0:03 8D 69 CE AD 3F 03 8D 5F
C0E8:6A CE AD 40 03 8D 6B CE 01
C0F0:AD 42 03 8D 6D CE AD 41 56
C0F8:03 8D 6E CE AD 78 69 4A A6
C100:8D 5B CE C9 00 F0 0B 8D FE
C108:FC CD 20 6C 89 CE FC CD 97
C110:D0 F8 AD 79 69 4A 8D 5D 74
C118:CE C9 00 F0 03 8D FC CD 1C
C120:20 3D 8A CE FC CD D0 F8 FA
C128:AC 5D CE A9 00 20 91 B3 25
C130:20 0F BC A0 75 A9 00 20 9B
C138:91 B3 20 2B BA 20 0F BC 59
C140:A0 5A A9 00 20 91 B3 20 AE
C148:14 BB 20 AA B1 8C 5D CE BC
C150:A5 FB 8D 71 CE A5 FC 8D 03
C158:72 CE A5 FD 8D 73 CE A5 DA
C160:FE 8D 74 CE AD 3E 03 8D 3C
C168:75 CE AD 3F 03 8D 76 CE 0E
C170:AD 40 03 8D 77 CE AD 42 A8
C178:03 8D 79 CE AD 41 03 8D 03
C180:7B CE A0 AE A9 A8 20 A2 47
C188:BB 20 0F BC A0 B4 A9 00 CA
C190:20 91 B3 20 14 BB A2 00 D5
C198:B5 61 9D 9C CE E8 E0 05 AD
C1A0:D0 F6 AC 5D CE A9 00 20 F2
C1A8:91 B3 20 0F BC AC 5B CE F4
C1B0:A9 00 20 91 B3 20 53 B8 A3
C1B8:20 58 BC A2 00 B5 61 9D 5B
C1C0:A1 CE E8 E0 05 D0 F6 A9 F6
C1C8:00 8D 5F CE A9 00 8D 60 51
C1D0:CE A5 C6 F0 06 20 E4 FF 87
C1D8:4C B9 C5 AD 5F CE 8D 8F 65
C1E0:CE AD 60 CE 8D 90 CE 20 9C
C1E8:F3 C5 AD 8F CE 8D 8D F0 F8
C1F0:15 C9 03 F0 1D C9 04 F0 E9
C1F8:25 C9 05 F0 2D AD 5F CE DE
C200:8D 91 CE 4C 33 C2 A9 B4 FB
C208:38 ED 5F CE 8D 91 CE 4C 9A
C210:33 C2 AD 5F CE 38 E9 64 6B
C218:8D 91 CE 4C 33 C2 A9 68 C7
C220:38 ED 5F CE 8D 91 CE 4C B2
C228:33 C2 AD 5F CE 38 E9 68 37
C230:8D 91 CE AC 91 CE A9 00 A0
C238:20 91 B3 A2 00 BD 9C CE D1
C240:95 69 E8 E0 05 D0 F6 A5 15
C248:61 20 2B BA 20 64 E2 20 10
C250:0F BC AC 5B CE A9 00 20 15
C258:91 B3 20 2B BA A2 00 B5 60
C260:61 9D A6 CE E8 E0 05 D0 65
C268:F6 20 2B BC 8D 89 CE 20 F2
C270:0F BC A5 61 20 2B BA A2 3D
C278:00 B5 61 9D B0 CE E8 E0 E4
C280:05 D0 F6 AC 91 CE A9 00 81
```

```
C288:20 91 B3 A2 00 BD 9C CE 22
C290:95 69 E8 E0 05 D0 F6 A5 65
C298:61 20 2B BA 20 6B E2 20 7C
C2A0:0F BC AC 5D CE A9 00 20 85
C2A8:91 B3 20 2B BA A2 00 B5 B0
C2B0:61 9D AB CE E8 E0 05 D0 56
C2B8:F6 20 0F BC A5 61 20 2B 8D
C2C0:BA A2 00 BD B0 CE 95 69 7D
C2C8:E8 E0 05 D0 F6 A5 61 20 D9
C2D0:6A B8 20 71 BF A2 00 B5 13
C2D8:61 9D B5 CE E8 E0 05 D0 BF
C2E0:F6 AD 89 CE C9 00 D0 13 6E
C2E8:A0 01 A9 00 20 91 B3 A2 85
C2F0:00 B5 61 9D A6 CE E8 E0 0D
C2F8:05 D0 F6 A2 00 BD A6 CE 51
C300:95 61 BD AB CE 95 69 E8 A5
C308:E0 05 D0 F1 20 14 BB 20 63
C310:0E E3 20 0F BC A0 B4 A9 03
C318:00 20 91 B3 20 2B BA 20 58
C320:0F BC A0 AE A9 A8 20 A2 30
C328:BB 20 14 BB 20 58 BC 20 CF
C330:AA B1 8C 7D CE 8D 7E CE 5B
C338:AD 5F CE 8D 8F CE AD 60 94
C340:CE 8D 90 CE 20 F3 C5 AD 9B
C348:8D CE C9 04 F0 37 AD 7D D0
C350:F0 27 C9 04 F0 17 C9 03 1E
C358:CE 8D 91 CE AD 7E CE 8D 5C
C360:92 CE 4C 9E C3 A9 B4 38 BE
C368:ED 7D CE 8D 91 CE A9 00 14
C370:ED 7E CE 8D 92 CE 4C 9E 48
C378:C3 A9 B4 18 6D 7D CE 8D F0
C380:91 CE A9 00 6D 7E CE 8D 4A
C388:92 CE 4C 9E C3 A9 68 38 4E
C390:ED 7D CE 8D 91 CE A9 01 3D
C398:ED 7E CE 8D 92 CE AD 91 26
C3A0:CE 18 6D 5A CE 8D 91 CE 87
C3A8:AD 92 CE 69 00 8D 92 CE 46
C3B0:AD 91 CE 8D 8F CE AD 92 CB
C3B8:CE 8D 90 CE 20 F3 C5 A9 10
C3C0:00 8D 80 CE AD 8D CE D0 B3
C3C8:8D 02 F0 1D C9 03 F0 2F C9 D1
C3D0:04 F0 3F C9 05 F0 51 AD 57
C3D8:91 CE 8D 7F CE A9 01 8D 33
C3E0:83 CE 8D 87 CE 4C 39 C4 BA
C3E8:A9 B4 38 ED 91 CE 8D 7F BA
C3F0:CE A9 00 8D 83 CE A9 01 CE
C3F8:8D 87 CE 4C 39 C4 AD 91 91
C400:CE 38 E9 B4 8D 7F CE A9 39
C408:00 8D 83 CE 8D 87 CE 4C C6
C410:39 C4 A9 68 38 ED 91 CE 8E
C418:8D 7F CE A9 01 8D 83 CE D0
C420:A9 00 8D 87 CE 4C 39 C4 87
C428:AD 91 CE 38 E9 68 8D 7F D5
C430:CE A9 01 8D 83 CE 8D 87 7E
C438:CE AC 7F CE A9 00 20 91 50
C440:B3 A2 00 BD 9C CE 95 69 DC
C448:E8 E0 05 D0 F6 A5 61 20 5D
C450:2B BA 20 64 E2 A2 00 BD C7
C458:B5 CE 95 69 E8 E0 05 D0 5F
C460:F6 A5 61 20 2B BA 20 AA 2C
C468:B1 8C 81 CE AC 7F CE A9 B5
C470:00 20 91 B3 A2 00 BD 9C 9C
C478:CE 95 69 E8 E0 05 D0 F6 3E
C480:A5 61 20 2B BA 20 6B E2 8D
C488:A2 00 BD B5 CE 95 69 E8 FE
C490:E0 05 D0 F6 A5 61 20 2B 73
C498:BA 20 0F BC A0 5A A9 00 F6
C4A0:20 91 B3 20 2B BA A9 0F AA
C4A8:BC A0 75 A9 00 20 91 B3 59
C4B0:20 14 BB 20 AA B1 8C 85 83
C4B8:CE A0 71 CE 85 FB AD 72 1A
C4C0:CE 85 FC AD 73 CE 85 FD 6D
C4C8:AD 74 CE 85 FE AD 75 CE E0
C4D0:8D 3E 03 AD 76 CE 8D 3F 35
C4D8:03 AD 77 CE 8D 40 03 4C 4C
C4E0:79 CE AD 42 03 8D 7B CE 45
C4E8:8D 41 03 AD 81 CE F0 13 01
C4F0:AD 83 CE D0 06 20 B6 89 C0
C4F8:4C FE C4 20 6C 89 CE 81 AB
C500:CE D0 ED AD 85 CE F0 13 1C
C508:AD 87 CE D0 06 20 3D 8A E8
C510:4C 16 C5 20 00 8A CE 85 4F
C518:CE D0 ED AD 5F CE D0 3A E9
```

www.commodore.ca

```
C520:AD 60 CE D0 35 A5 FB 8D 47      C7B8:3A CE 85 FE AD 3B CE 8D 3F      CA50:CE AD 51 CE F0 06 20 00 AF
C528:2E CE A5 FC 8D 2F CE A5 6F      C7C0:3E 03 AD 3C CE 8D 3F 03 D7      CA58:8A 4C 5F CA 20 3D 8A AD 97
C530:FD 8D 30 CE A5 FE 8D 31 86      C7C8:AD 3D CE 8D 40 03 AD 3F D9      CA60:44 CE D0 03 4C 99 C9 AD 20
C538:CE AD 3E 03 8D 32 CE AD 0F      C7D0:03 8D 41 03 A9 40 8D 8D 1D      CA68:48 CE 18 6D 44 CE 8D 48 70
C540:3F 03 8D 33 CE AD 40 03 C1      C7D8:42 03 A9 01 8D 41 CE 20 BE      CA70:CE AD 49 CE 6D 45 CE 8D 9A
C548:8D 34 CE AD 42 03 8D 36 CB      C7E0:10 C8 AD 42 CE 29 08 D0 80      CA78:49 CE CD 47 CE F0 07 C9 A6
C550:CE AD 41 03 8D 35 CE 4C 32      C7E8:03 20 6C 89 AD 42 CE 29 65      CA80:00 D0 0F 4C 99 C9 AD 48 88
C558:98 C5 AD 2E CE 8D 37 CE 24      C7F0:04 D0 03 20 B6 89 AD 42 92      CA88:CE 18 CD 46 CE 8D 03 4C 35
C560:AD 2F CE 8D 38 CE AD 30 C9      C7F8:CE 29 01 D0 03 20 00 8A 8A      CA90:99 C9 EE 4C CE D0 03 EE B6
C568:CE 8D 39 CE AD 31 CE 8D 30      C800:AD 42 CE 29 02 D0 03 20 DE      CA98:4D CE AD 52 CE F0 06 20 C9
C570:3A CE AD 32 CE 8D 3B CE 97      C808:3D 8A 20 7F 8F 4C 84 C6 B4      CAA0:6C 89 4C A8 CA 20 B6 89 B0
C578:AD 33 CE 8D 3C CE AD 34 07      C810:AD 10 D0 29 01 F0 02 A9 A2      CAA8:AD 48 CE 38 ED 46 CE 8D 38
C580:CE 8D 3D CE AD 36 CE 8D DC      C818:03 8D 10 D0 AD 00 D0 8D 3A      CAB0:48 CE AD 49 CE E9 00 8D 14
C588:40 CE AD 35 CE 8D 3F CE EA      C820:02 D0 AD 01 D0 8D 03 D0 40      CAB8:49 CE 4C 99 C9 AD 41 CE 20
C590:A9 00 8D 41 CE 20 2F C8 D4      C828:60 A9 01 8D 15 D0 60 20 1A      CAC0:F0 03 4C FA CA AD 2E CE 01
C598:AD 5F CE 18 6D 9A CE 8D 2F      C830:7F 8F A9 01 8D 67 03 A8 68      CAC8:85 FB AD 2F CE 85 FC AD FC
C5A0:5F CE AD 60 CE 69 00 8D F4      C838:15 D0 8D 43 CE A9 00 8D 19      CAD0:30 CE 85 FD AD 31 CE 85 18
C5A8:60 CE A9 01 D0 08 AD 5F C2      C840:15 D0 AD 41 CE F0 35 A5 A4      CAD8:FE AD 32 CE 8D 3E 03 AD A5
C5B0:CE 18 C9 6C B0 03 4C D1 A5      C848:FB 8D 37 CE A5 FC 8D 38 83      CAE0:33 CE 8D 3F 03 AD 34 CE 6F
C5B8:C1 AD 65 CE 85 FB AD 66 08      C850:CE A5 FD 8D 39 CE A5 FE 9A      CAE8:8D 40 03 AD 36 CE 8D 42 DA
C5C0:CE 85 FC AD 67 CE 85 FD 0F      C858:8D 3A CE AD 3E 03 8D 3B 48      CAF0:03 AD 35 CE 8D 41 03 4C CA
C5C8:AD 68 CE 85 FE AD 69 CE C7      C860:CE AD 3F 03 8D 3C CE AD 85      CAF8:2C CB AD 37 CE 85 FB AD F2
C5D0:8D 3E 03 AD 6A CE 8D 3F D6      C868:40 03 8D 3D CE AD 42 03 15      CB00:38 CE 85 FC AD 39 CE 85 5D
C5D8:03 AD 6B CE 8D 40 03 AD CC      C870:8D 40 CE AD 41 03 8D 3F FD      CB08:FD AD 3A CE 85 FE AD 3B FC
C5E0:6D 8D 42 03 AD 6F CE 29      C878:CE 4C AE C8 A5 FB 8D 2E 4D      CB10:CE 8D 3E 03 AD 3C CE 8D F3
C5E8:8D 41 03 A9 1E 8D 15 D0 A8      C880:CE A5 FC 8D 2F CE A5 FD 59      CB18:3F 03 AD 3D CE 8D 40 03 C9
C5F0:4C 59 C0 90 CE D0 10 5D      C888:8D 30 CE A5 FE 8D 31 CE 80      CB20:AD 40 CE 8D 42 03 AD 3F 0A
C5F8:AD 8F CE 18 C9 5A B0 08 BB      C890:AD 3E 03 8D 32 CE AD 3F 29      CB28:CE 8D 41 03 AD 43 CE 8D 88
C600:A9 01 8D CE CE 4C 47 C6 C2      C898:03 8D 33 CE AD 40 03 8D 64      CB30:15 D0 A9 00 8D 67 03 60 2C
C608:AD 90 CE D0 10 AD 8F CE 9C      C8A0:34 CE AD 41 03 8D 36 CE 63      CB38:A9 01 8D 54 CE A5 3F 8D F4
C610:18 C9 B4 B0 08 A9 02 8D 36      C8A8:AD 41 03 8D 35 CE AD 34 0F      CB40:FC CD A5 40 8D FD CD AD 30
C618:8D CE 4C 47 C6 AD 90 CE FA      C8B0:CE 18 CD 3D CE B0 12 A9 44      CB48:69 03 8D 06 CE A9 00 8D 12
C620:F0 08 AD 8F CE 18 C9 0E 4F      C8B8:00 8D 51 CE AD 3D CE 38 FC      CB50:69 03 A2 40 9D 80 03 CA 75
C628:B0 08 A9 03 8D 8D CE 4C 02      C8C0:ED 34 CE 8D 46 CE 4C D8 E7      CB58:D0 FA A9 0E 8D F9 07 A9 39
C630:47 C6 AD 8F CE 18 C9 68 94      C8C8:C8 A9 01 8D 51 CE AD 34 77      CB60:E0 8D 9D 03 8D A0 03 8D 52
C638:B0 08 A9 04 8D 8D CE 4C 22      C8D0:CE 38 ED 3D CE 8D 46 CE 71      CB68:27 D0 8D 28 D0 A9 02 8D BA
C640:47 C6 A9 05 8D 8D CE 60 49      C8D8:33 CE 8D 3C CE F0 07 CA      CB70:15 D0 A9 00 8D 67 03 20 2C
C648:A9 01 8D 15 D0 60 20 06 3C      C8E0:C9 01 F0 27 4C C8 AD 8D      CB78:7F 8F AD 68 03 F0 05 A9 7F
C650:86 AD 27 D0 8D 28 D0 A9 D6      C8E8:32 CE 18 CD 3B CE B0 1B B8      CB80:1F 4C 86 CB A9 18 8D 9C AD
C658:03 8D 15 D0 20 10 C8 A5 F2      C8F0:A9 01 8D 52 CE AD 3B CE E0      CB88:03 8D 9F 03 AD 00 D0 8D C5
C660:FB 8D 2E CE 8D 37 CE A5 8E      C8F8:38 ED 32 CE 8D 44 CE AD 1E      CB90:02 D0 AD 01 D0 8D 03 D0 B6
C668:FC 8D 2F CE 8D 38 CE A5 3B      C900:3C ED 33 CE 8D 45 CE 5C      CB98:AD 10 D0 29 01 F0 02 A9 31
C670:FD 8D 30 CE 8D 39 CE A5 E7      C908:4C 23 C9 A9 00 8D 52 CE 08      CBA0:03 8D 10 D0 20 E4 FF F0 B1
C678:FE 8D 31 CE 8D 3A CE AD 9C      C910:AD 32 CE 38 ED 3B CE 8D EB      CBA8:CE C9 5E D0 10 EE 54 CE A6
C680:3E 03 8D 32 CE 8D 3B CE B4      C918:44 CE AD 33 CE ED 3C CE DF      CBB0:A2 14 EC 54 CE D0 14 8E F1
C688:AD 3F 03 8D 33 CE 8D 3C 22      C920:8D 45 CE AD 2E CE 85 FB 34      CBB8:54 CE 4C CB CB C9 5F D0 89
C690:CE AD 40 03 8D 34 CE 8D 91      C928:AD 2F CE 85 FC AD 30 CE 5E      CBC0:0A CE 54 CE D0 05 A2 01 69
C698:3D CE AD 42 03 8D 36 CE DB      C930:85 FD AD 31 CE 85 FE AD 07      CBC8:8E 54 CE C9 85 D0 03 4C F4
C6A0:8D 40 CE AD 41 03 8D 35 20      C938:32 CE 8D 3E 03 AD 33 CE 32      CBD0:D6 CD C9 91 F0 1D C9 11 3A
C6A8:CE 8D 3F CE A9 00 8D 67 A5      C940:8D 3F 03 AD 34 CE 8D 40 37      CBD8:F0 1C C9 1D F0 1B C9 9D 20
C6B0:03 8D 41 CE 20 F2 91 AD D5      C948:03 AD 36 CE 8D 42 03 AD A5      CBE0:F0 1A C9 86 F0 0A C9 0D 69
C6B8:77 02 F0 08 AD D0 06 90 98      C950:35 CE 8D 41 03 A9 00 8D 44      CBE8:F0 03 4C A0 CC 4C FF CB 89
C6C0:20 E4 FF 4C FF C6 4C 29 39      C958:48 CE 8D 49 CE 8D 47 CE 14      CBF0:4C 1C CC 4C 2E CC 4C 49 9A
C6C8:C8 AD 00 DC 8D 42 CE 29 30      C960:8D 4D CE 8D 4A CE 8D 4B B4      CBF8:CC 4C 64 CC 4C 82 CC A9 13
C6D0:10 F0 08 A9 01 8D 66 03 4C      C968:CE 8D 4C CE 8D 4E CE AD 2E      CC00:A0 8D 55 CE 20 B6 89 CE A2
C6D8:4C E2 C7 66 03 F0 03 4F      C970:45 CE 18 6A AD 45 CE AD D0      CC08:55 CE D0 F8 A9 A0 8D 55 E9
C6E0:4C E2 C7 A9 64 8D 66 03 0A      C978:44 CE 6A 44 CE AD 45 06      CC10:CE 20 B6 89 CE 55 D0 C2
C6E8:20 2F C8 AD 53 CE D0 0C C9      C980:CE D0 11 AD 44 CE 18 CD 08      CC18:F8 4C 49 CC AD 68 03 F0 3D
C6F0:A9 01 85 C6 A9 4C 8D 77 C1      C988:46 CE B0 08 A9 01 8D 50 46      CC20:08 A9 00 8D 68 03 4C 77 60
C6F8:02 4C C6 4C 6C 4C E2 C7 AD 0B      C990:CE 4C 99 C9 A9 00 8D 50 27      CC28:CB A9 01 4C 23 CC 4C 54 F2
C700:41 CE D0 03 4C 76 C7 A5 9F      C998:CE 20 7F 8F A5 C6 F0 08 B6      CC30:CE 8D 57 CE A9 0A 8D 55 52
C708:FB 8D 37 CE A5 FC 8D 38 41      C9A0:A9 01 8D 4A CE 8D 4B CE B1      CC38:CE 20 00 8A CE 55 CE D0 24
C710:CE A5 FD 8D 39 CE A5 FE 58      C9A8:AD 4D CE CD 45 CE D0 0D 31      CC40:F8 CE 57 CE D0 EE 4C 77 34
C718:8D 3A CE AD 3E 03 8D 3B 06      C9B0:AD 4C CE CD 44 CE D0 05 88      CC48:CB AD 54 CE 8D 57 CE A9 BB
C720:CE AD 3F 03 8D 3C CE AD 43      C9B8:A9 01 8D 4A CE AD 4E CE 50      CC50:0A 8D 55 CE 20 3D 8A CE 77
C728:40 03 8D 3D CE AD 41 03 D0      C9C0:CD 46 CE D0 05 A9 01 8D 12      CC58:55 CE D0 F8 CE 57 CE D0 3C
C730:8D 3F CE AD 42 03 8D 40 84      C9C8:4B CE AD 4A CE F0 08 AD 08      CC60:EE 4C 77 CB AD 54 CE 8D 1A
C738:CE AD 2E CE 85 FB AD 2F F3      C9D0:4B CE F0 03 4C BD CA AD A8      CC68:56 CE A9 08 8D 55 CE 20 16
C740:CE 85 FC AD 30 CE 85 FD D8      C9D8:50 CE F0 03 4C 4E CA EE B6      CC70:6C 89 CE 55 CE D0 F8 CE 4C
C748:AD 31 CE 85 FE AD 32 CE 0F      C9E0:4C CE D0 03 EE 4D CE AD 90      CC78:56 CE D0 EE 20 6C 89 4C 0C
C750:8D 3E 03 AD 33 CE 8D 3F AD      C9E8:52 CE 20 6C 89 4C E9      CC80:77 CB AD 54 CE 8D 56 CE EB
C758:03 AD 34 CE 8D 40 03 AD 69      C9F0:F5 C9 20 B6 89 AD 46 CE BF      CC88:A9 08 8D 55 CE 20 B6 89 ED
C760:35 CE 8D 41 03 AD 36 CE 0E      C9F8:D0 03 4C 99 C9 AD 48 CE 3D      CC90:CE 55 CE D0 F8 CE 56 CE 4C
C768:8D 42 03 A9 00 8D 41 CE D1      CA00:18 6D 46 CE 8D 48 CE AD 8B      CC98:D0 EE 20 B6 89 4C 77 CB FD
C770:20 10 C8 4C E2 C7 A5 FB 6F      CA08:49 CE 69 00 8D 49 CE 20 20      CCA0:18 C9 20 B6 89 4C 77 CB 28
C778:8D 2E CE A5 FC 8D 2F CE D9      CA10:45 CE F0 07 C9 00 D0 0F 89      CCA8:18 C9 5B B0 0D C9 40 B0 F7
C780:A5 FD 8D 30 CE A5 FE 8D AF      CA18:4C 99 C9 AD 48 CE 18 CD C9      CCB0:03 4C CA CC 38 E9 40 4C 3B
C788:31 CE AD 3E 03 8D 32 CE 7F      CA20:44 CE B0 03 4C 99 C9 EE 1D      CCB8:CA CC 18 C9 C1 B0 03 4C AD
C790:AD 3F 03 8D 33 CE AD 40 70      CA28:4E CE AD 52 CE F0 06 20 C9      CCC0:77 CB 18 C9 DB B0 D6 38 30
C798:03 8D 34 CE AD 41 CE 8D 86      CA30:00 8A 4C 38 CA 20 3D 8A 51      CCC8:E9 80 8D 58 CE 8D 55 CE D4
C7A0:35 CE AD 42 03 8D 36 CE E1      CA38:AD 48 CE 38 ED 44 CE 8D BF      CCD0:A9 00 85 3F A9 D8 85 40 DF
C7A8:AD 37 CE 85 FB AD 38 CE E4      CA40:48 CE AD 49 CE ED 45 CE 7F      CCD8:A5 3F 18 69 08 85 3F A5 29
C7B0:85 FC AD 39 CE 85 FD AD C1      CA48:8D 49 CE 4C 99 C9 EE 4E B5      CCE0:40 69 00 85 40 CE 55 CE 04
```

```
CCE8:D0 EE AD 54 CE 8D 57 CE CB
CCF0:8D 59 CE A9 08 8D 55 CE 0C
CCF8:20 00 8A CE 55 CE D0 F8 61
CD00:CE 57 CE D0 EE AD 54 CE 65
CD08:8D 57 CE CE 57 CE F0 06 E4
CD10:20 3D 8A 4C 0B CD A9 08 0C
CD18:8D CA 54 CE 8D 56 CE D3
CD20:CE 8D 57 CE A0 00 AD 0E CC
CD28:DC 29 FE 8D 0E DC A5 01 65
CD30:29 FB 85 01 B1 3F 8D 58 1E
CD38:CE A5 01 09 04 85 01 AD 3B
CD40:0E DC 09 01 8D 0E DC A9 53
CD48:80 8D 40 CE AD 58 CE 2D 16
CD50:40 8D 60 CE A9 01 8D 67 82
CD58:03 20 7F 8F A9 00 8D 67 36
CD60:03 20 6C 89 CE 56 CE D0 E9
CD68:E3 AD 54 CE 8D 56 CE 4E 8A
CD70:40 CE D0 D8 CE 59 CE D0 D1
CD78:0E AD 54 CE 8D 59 CE CE 3C
CD80:55 CE D0 03 4C BC CD AD 63
CD88:54 CE 8D 56 CE A9 08 8D D3
CD90:40 CE 20 B6 89 CE 40 CE 46
CD98:D0 F8 CE 56 CE D0 EE 20 D1
CDA0:3D 8A AD 54 CE 8D 56 CE A0
CDA8:CE 57 CE AD CD AD 54 CE F6
CDB0:8D 57 CE E6 3F D0 02 E6 59
CDB8:40 4C 24 CD 20 6C 89 20 CE
CDC0:3D 8A AD 54 CE 8D 57 CE C2
CDC8:CE 57 CE F0 06 20 00 8A C5
CDD0:4C C8 CD 4C 77 CB A9 01 82
CDD8:8D 15 D0 AD FC CD 85 3F DE
CDE0:AD FD CD 85 40 AD 06 CE 78
CDE8:8D 69 03 60 00 00 00 00 0C
CDF0:00 00 00 C1 D6 C1 C9 CC C6
CDF8:C1 C2 CC C5 00 00 00 00 1C
CE00:00 00 00 00 00 00 00 00 9D
CE08:00 00 00 00 00 00 00 00 A5
CE10:00 00 00 00 00 00 00 00 AD
CE18:00 00 00 00 00 00 00 00 B5
CE20:00 00 00 00 00 00 00 00 BD
CE28:00 00 00 00 00 A4 B0 00 AD
CE30:00 00 00 00 00 00 00 00 CD
CE38:00 00 00 00 00 00 00 00 D5
CE40:00 00 00 00 00 00 00 00 DD
CE48:00 00 00 00 00 00 00 00 E5
CE50:00 00 00 00 00 00 00 00 ED
CE58:00 00 00 00 00 00 00 00 F5
CE60:00 00 00 00 00 00 00 00 FD
CE68:00 00 00 00 00 00 00 00 06
CE70:00 00 00 00 00 00 00 00 0E
CE78:00 00 00 00 00 00 00 00 16
CE80:00 00 00 00 00 00 00 00 1E
CE88:00 00 00 00 00 00 00 00 26
CE90:00 00 00 00 00 00 00 00 2E
CE98:00 00 00 00 00 00 00 00 36
CEA0:00 00 00 00 00 00 00 00 3E
CEA8:00 00 00 00 00 00 00 00 46
CEB0:00 00 00 00 00 00 00 00 4E
CEB8:00 00 00 00 00 00 00 00 56
```

Color Pad 3.0

```
92E0:AE 47 03 E0 8D F0 1D C9 D1
92E8:D2 F0 1C C9 5A F0 1B C9 6A
92F0:D3 F0 1A C9 91 F0 19 C9 68
92F8:11 F0 15 C9 1D F0 11 C9 BA
9300:9D F0 0D 60 4C 51 98 4C FE
9308:13 93 4C 3A 93 4C 2E 93 88
9310:4C 7F 99 AD 70 03 C9 57 E5
9318:F0 0B A9 57 8D 70 03 A9 03
9320:0A 8D 71 03 60 AD 45 03 55
9328:8D 20 04 4C 1C 93 AD 70 F7
9330:03 C9 58 F0 F0 A9 58 4C 90
9338:1C 93 20 06 86 A9 00 8D 1F
9340:69 03 8D B5 9C A5 3F 8D 71
9348:5E 9C A5 40 8D 5F 9C A5 47
9350:41 8D 60 9C A5 42 8D 61 04
9358:9C A9 03 8D B2 9C A9 14 E0
9360:8D 78 69 A9 19 8D 79 69 8F
9368:8A 85 3F A9 04 85 40 B1
9370:A9 18 85 41 A9 79 85 42 B7
9378:A9 E8 8D B3 9C A9 07 8D C2
9380:B4 9C 20 22 98 A9 00 85 40
```

```
9388:3F A9 D8 85 40 A9 E8 8D 35
9390:B3 9C A9 DB 8D B4 9C 20 44
9398:22 98 A5 FD 18 69 18 85 A7
93A0:FD A5 FE 69 75 85 FE A9 10
93A8:E8 8D 09 88 8D 1D 88 8D D0
93B0:AA 98 A9 91 A9 03 8D 73
93B8:0F 88 8D 23 88 8D B0 90 D9
93C0:8D A0 91 A9 60 8D 11 94 93
93C8:20 DF 93 A9 A9 8D 11 94 3F
93D0:20 56 86 A5 C6 F0 08 A9 7C
93D8:01 8D B5 9C 4C 4D 97 A5 90
93E0:FB 8D 93 9C A5 FC 8D 94 76
93E8:9C A5 FD 8D 95 9C A5 FE C9
93F0:8D 96 9C AD 41 03 8D 97 BB
93F8:9C 42 03 8D 98 9C AD 08
9400:3E 03 8D 99 9C AD 3F 03 71
9408:8D 9A 9C AD 40 03 8D 9B D1
9410:9C A9 00 8D 67 03 A9 B1 17
9418:8D 64 88 A9 60 8D A9 88 E1
9420:20 CA 87 20 33 81 A9 91 83
9428:8D 64 88 A5 A5 8D A9 88 1C
9430:A9 02 8D 15 D0 A9 FF 8D 6C
9438:F9 07 AD 27 D0 8D 28 D0 26
9440:A9 7A 85 3F A9 95 85 40 BF
9448:A9 00 85 41 A9 04 85 42 B5
9450:A0 00 A9 A0 91 41 C8 91 BD
9458:41 88 A5 41 18 69 00 85 F8
9460:41 A5 42 69 04 85 42 B1 65
9468:3F C9 FF D0 03 AD 21 D0 92
9470:91 41 C8 91 41 88 A5 41 9D
9478:38 E9 FE 85 41 A5 42 E9 7F
9480:D3 85 42 A5 3F 18 69 01 C5
9488:85 3F A5 40 69 00 85 40 93
9490:C9 6B D0 BE A5 3F C9 6E AB
9498:D0 BA 8D B1 9C C9 57 F0 D4
94A0:25 A9 15 8D 02 D0 A9 2C 15
94A8:8D 03 D0 AD 10 D0 29 FD 62
94B0:8D 10 D0 A9 01 8D 90 9C 55
94B8:8D 01 D0 A5 3F A9 79 9D 2B
94C0:04 85 40 4C 29 95 AD 9D B2
94C8:9C 85 FB AD 9E 9C 85 FC 6B
94D0:AD 9F 9C 85 FD AD A0 9C 29
94D8:85 FE A6 A1 9C 8D 41 03 F4
94E0:AD A2 9C 8D 42 03 AD A3 13
94E8:9C 8D 3E 03 AD A4 9C 8D 82
94F0:3F 03 AD A5 9C 8D 40 03 29
94F8:AD A6 9C 85 3F AD A7 9C 2B
9500:85 40 AD A8 9C 8D 02 D0 2E
9508:AD A9 9C 8D 03 D0 AD AA 42
9510:9C 8D 90 9C AD AB 9C 8D AB
9518:91 9C AD AC 9C 8D 10 D0 BF
9520:29 02 F0 05 A9 FE 8D F9 2D
9528:07 A2 73 8A A8 88 D0 FD 9D
9530:CA F0 F9 A5 C6 F0 0A 20 BC
9538:E4 FF A2 00 86 C6 4C 47 59
9540:96 AD 00 DC 8D 92 9C 29 09
9548:10 D0 03 4C D8 96 AD 92 E3
9550:9C 29 08 D0 3F EE 90 9C 95
9558:AD 90 9C C9 15 D0 08 A9 54
9560:14 8D 90 9C 4C 94 95 20 D4
9568:6C 89 A9 01 8D B2 9C AD 8F
9570:02 D0 18 69 10 8D 02 D0 F5
9578:90 0D AD 10 D0 09 02 8D 22
9580:10 D0 A9 FE 8D F9 07 A5 15
9588:3F 18 69 02 85 3F A5 40 5B
9590:69 00 85 40 92 9C 29 3F 3F
9598:04 D0 3D CE 90 9C AD 90 71
95A0:9C D0 08 A9 01 8D 90 9C E5
95A8:4C D8 96 A5 3F 20 B6 89 01 15
95B0:8D B2 9C AD 02 D0 38 E9 6B
95B8:10 8D 02 D0 B0 0D AD 10 C1
95C0:D0 29 FD 8D 10 D0 A9 FF 4B
95C8:8D F9 07 A5 3F 38 E9 02 25
95D0:85 3F A5 40 E9 00 85 40 E1
95D8:AD 92 9C 29 01 D0 2E CE 1C
95E0:91 9C AD 91 9C D0 08 A9 E4
95E8:01 8D 91 9C 4C 0D 96 20 D7
95F0:00 8A A9 01 8D B2 9C AD 22
95F8:03 D0 38 E9 08 8D 03 D0 CC
9600:A5 3F 38 E9 28 85 3F A5 F0
9608:40 E9 00 85 40 92 9C A2
9610:29 02 D0 30 EE 91 9C AD 14
9618:91 9C C9 1A D0 08 A9 19 23
```

```
9620:8D 91 9C 4C 44 96 20 3D CA
9628:8A A9 01 8D B2 9C AD 03 64
9630:D0 18 69 08 8D 03 D0 A5 39
9638:3F 18 69 28 85 3F A5 40 6F
9640:69 00 85 40 4C 29 95 C9 D2
9648:20 F0 23 C9 44 F0 25 C9 9F
9650:41 F0 27 C9 44 F0 2C C9 E3
9658:45 F0 3B C9 91 F0 3D C9 FC
9660:11 F0 39 C9 1D F0 35 C9 F6
9668:9D F0 31 4C 4D 97 20 69 FD
9670:83 4C 29 95 20 9B 83 4C B3
9678:29 95 20 AE 83 8D 28 D0 02
9680:4C 29 95 20 57 83 A9 60 4F
9688:8D 7F 82 20 65 82 A9 AD E4
9690:8D 7F 82 4C 29 95 20 5D B6
9698:83 4C 86 96 8D 70 9C 20 5C
96A0:8F 97 A9 A9 8D 6F 9C AD 22
96A8:70 9C C9 91 F0 0E C9 11 EB
96B0:F0 10 C9 1D F0 12 20 B6 2C
96B8:9C 4C CB 96 20 3D 8A 4C 05
96C0:CB 96 20 3D 8A 4C CB 96 05
96C8:20 6C 89 CE 6F 9C D0 D7 A6
96D0:A9 00 8D B1 9C 4C D3 93 F0
96D8:CF B2 9C F0 03 4C 4E 95 38
96E0:A9 03 8D B2 9C A5 FB 81
96E8:9D 9C A5 FC 8D 9E 9C A5 56
96F0:FD 8D 9F 9C A5 FE 8D A0 23
96F8:9C AI 03 8D A1 9C AD 12
9700:42 03 8D A2 9C AD 3E 03 08
9708:8D A3 9C AD 3F 03 8D A4 1B
9710:9C AD 40 03 8D A5 9C A5 13
9718:3F 8D A6 9C A5 40 8D A7 D9
9720:9C AD 02 D0 8D A8 9C AD 4C
9728:03 D0 8D A9 9C AD 90 9C B2
9730:8D AA 9C AD 91 9C 8D AB 05
9738:9C AD 10 D0 8D AC 9C 20 32
9740:57 8D B1 9C 20 8B 90 20 EE
9748:EF 97 4C DF 93 A9 18 85 D5
9750:3F A9 79 85 40 A9 00 85 3F
9758:41 A9 04 85 42 A9 00 8D B1
9760:B3 9C A9 7D 8D B4 9C 20 36
9768:22 98 A9 00 85 41 A9 D8 61
9770:85 42 A9 88 8D B3 9C 20 D4
9778:80 8D B4 9C 20 22 98 A9 10
9780:00 8D 09 88 8D 1D 88 8D 3C
9788:AA 90 8D 9A 91 A9 D4 8D F6
9790:0F 88 8D 23 88 8D B0 90 B9
9798:8D A0 91 20 EF 97 A5 FD 12
97A0:38 E9 18 85 FD A5 FE E9 30
97A8:75 85 FE AD 5E 9C 85 3F 5E
97B0:AD 5F 9C 65 40 AD 60 9C 90
97B8:85 41 AD 61 9C 85 42 A9 EF
97C0:00 8D 67 03 A9 00 8D B1 8A
97C8:9C A9 01 8D 15 D0 AD 18 09
97D0:D0 09 08 8D 18 D0 AD 11 F4
97D8:D0 09 20 8D 11 D0 AD 16 CC
97E0:D0 09 10 8D 16 D0 AD B5 9A
97E8:9C D0 03 4C 3A 93 60 AD 4E
97F0:93 9C 85 FB AD 94 9C 85 FF
97F8:FC AD 95 9C 85 FD AD 96 A4
9800:9C 85 FE AD 97 9C 8D 41 27
9808:03 AD 98 9C 8D 42 03 AD 2C
9810:99 9C 8D 3E 03 AD 9A 9C 6B
9818:8D 3F 03 AD 9B 9C 8D 40 C5
9820:03 60 A0 00 B1 3F 91 41 ED
9828:A5 3F 18 69 01 85 3F A5 D7
9830:40 69 00 85 40 A5 41 18 67
9838:69 01 85 41 A5 42 69 00 2C
9840:85 42 A5 40 CD B4 9C D0 C8
9848:D9 A5 3F CD B3 9C D0 D2 19
9850:60 A2 00 BD 8E 9C 29 7F EE
9858:90 86 9C E8 E0 02 D0 F3 0F
9860:A9 01 A2 08 A0 00 20 BA 7B
9868:FF A2 86 A0 9C A9 02 20 CC
9870:BD FF 20 C0 FF A2 01 20 3D
9878:C6 FF A9 93 20 D2 FF A9 71
9880:06 20 D2 FF 8D 21 D0 A9 65
9888:9F 20 D2 FF A9 12 20 D2 94
9890:FF 20 CF FF 20 CF FF 20 24
9898:CF FF 20 CF FF 20 CF FF 32
98A0:8D 4F 9C 20 CF FF 8D 50 EB
98A8:9C AC 4F 9C AD 50 9C 20 0F
98B0:91 B3 20 0F BC A0 64 A9 67
```

```
98B8:00 20 91 B3 20 14 BB 20 48
98C0:AA B1 8C 6D 9C 98 18 69 FC
98C8:30 20 D2 FF AC 6D 9C A5 6E
98D0:1E 20 91 B3 20 0F BC A0 DD
98D8:64 A9 00 20 91 B3 A5 61 B0
98E0:20 2B BA 20 AA B1 8C 6D E8
98E8:9C 8D 6E 9C AD 4F 9C 38 7F
98F0:ED 6D 9C 8D 4F 9C A8 AD CC
98F8:50 9C ED 6E 9C 8D 50 9C 76
9900:20 91 B3 20 0F BC A0 0A D6
9908:A9 00 20 91 B3 20 14 BB 2F
9910:20 AA B1 8C 6D 9C 98 18 24
9918:69 30 20 D2 FF AC 6D 9C 67
9920:A9 00 20 91 B3 20 0F BC 3E
9928:A0 0A A9 00 20 91 B3 A5 B7
9930:61 20 2B BA 20 AA B1 8C C8
9938:6D 9C AD 4F 9C 38 ED 6D 03
9940:9C 18 69 30 20 D2 FF A9 ED
9948:20 20 D2 FF 20 CF FF 8D BB
9950:5E 9C 20 B7 FF C9 00 F0 71
9958:03 4C E2 99 AD 5E 9C C9 FD
9960:22 D0 E9 20 D2 FF 20 CF BE
9968:FF C9 22 F0 16 8D 5E 9C A1
9970:20 B7 FF C9 00 D0 09 AD 41
9978:5E 9C 20 D2 FF 4C 66 99 CA
9980:4C E2 99 20 D2 FF 20 CF 6E
9988:FF C9 20 F0 F6 8D 5E 9C 88
9990:A9 0D 20 D2 FF A9 91 20 F6
9998:D2 FF A9 16 8D 72 9C A9 E4
99A0:1D 20 D2 FF CE 72 9C F0 0F
99A8:F6 AD 5E 9C 20 D2 FF 20 C4
99B0:CF FF 8D 5E 9C 20 B7 FF 38
99B8:D0 28 AD 5E 9C C9 00 D0 D6
99C0:EB A9 0D 20 D2 FF A5 C6 A0
99C8:F0 0E 20 E4 FF C9 51 F0 05
99D0:20 A5 C6 F0 FC 20 E4 FF 97
99D8:20 B7 FF C9 00 D0 12 4C 5A
99E0:97 98 A2 00 BD 55 9C 29 FF
99E8:7F 20 D2 FF E8 E0 09 D0 EB
99F0:F3 20 CC FF A9 01 20 C3 15
99F8:FF A9 0D 20 D2 FF 60 20 B1
9A00:90 9B C9 91 F0 0B C9 11 0F
9A08:F0 0A C9 1D F0 09 4C 1A A1
9A10:9A 4C 42 9A 4C 6A 9A 4C 25
9A18:92 9A A9 04 8D B6 9C A9 DC
9A20:00 85 3F A9 20 85 40 A9 7A
9A28:08 85 41 A9 20 85 42 A9 CA
9A30:00 85 45 A9 04 85 46 A9 76
9A38:01 85 47 A9 04 85 48 4C E5
9A40:B7 9A A9 01 8D B6 9C A9 67
9A48:00 85 3F A9 20 85 40 A9 A2
9A50:40 85 41 A9 21 85 42 A9 17
9A58:00 85 45 A9 04 85 46 A9 9E
9A60:28 85 47 A9 04 85 48 4C A1
9A68:B7 9A A9 02 8D B6 9C A9 DC
9A70:38 85 3F A9 3F 85 40 A9 DF
9A78:F8 85 41 A9 3D 85 42 A9 7C
9A80:E7 85 45 A9 07 85 46 A9 D2
9A88:BF 85 47 A9 07 85 48 4C AD
9A90:B7 9A A9 03 8D B6 9C A9 D7
9A98:38 85 3F A9 3F 85 40 A9 08
9AA0:30 85 41 A9 3F 85 42 A9 50
9AA8:E7 85 45 A9 07 85 46 A9 FA
9AB0:E6 85 47 A9 07 85 48 A9 C6
9AB8:01 8D 90 9C 8D 91 9C A0 3A
9AC0:00 B1 41 91 3F AD 90 9C 12
9AC8:C9 28 F0 03 4C DE 9A AD FB
9AD0:B6 9C C9 01 F0 08 C9 02 0F
9AD8:F0 04 A9 00 91 3F AD 91 33
9AE0:9C C9 19 F0 03 4C F7 9A DC
9AE8:AD B6 9C C9 03 F0 08 C9 88
9AF0:04 F0 04 A9 00 91 3F C8 0D
9AF8:C0 08 F0 03 4C C1 9A A0 1E
9B00:00 B1 47 91 45 A5 45 18 09
9B08:69 00 85 45 A5 46 69 D4 E6
9B10:85 46 A5 47 18 69 00 85 B0
9B18:47 A5 48 69 D4 85 48 B1 FA
9B20:47 91 45 A5 45 38 E9 00 41
9B28:85 45 A5 46 E9 D4 85 46 80
9B30:A5 47 38 E9 00 85 47 A5 FB
9B38:48 E9 D4 85 48 EE 90 9C BC
9B40:AD 90 9C C9 29 D0 12 A9 FC
9B48:01 8D 90 9C EE 91 9C AD E3
9B50:91 9C C9 1A D0 03 4C 8C 0A
9B58:9B AD B6 9C C9 01 F0 0B A8
9B60:C9 02 F0 0A C9 03 F0 09 41
9B68:4C 86 9B 4C 74 9B 4C 7A C4
9B70:9B 4C 80 9B 20 E2 9B 4C 62
9B78:89 9B 20 17 9C 4C 89 9B 95
9B80:20 17 9C 4C 89 9B 20 E2 C3
9B88:9B 4C BF 9A 20 B9 9B 60 C1
9B90:A6 3F 8E 5E 9C A6 40 8E 31
9B98:5F 9C A6 41 8E 60 9C A6 65
9BA0:42 8E 61 9C A6 45 8E 62 5C
9BA8:9C A6 46 8E 63 9C A6 47 AB
9BB0:8E 64 9C A6 48 8E 65 9C 2A
9BB8:60 AD 5E 9C 85 3F AD 5F 05
9BC0:9C 85 40 AD 60 9C 85 41 4C
9BC8:AD 61 9C 85 42 AD 62 9C 45
9BD0:85 45 AD 63 9C 85 46 AD 3D
9BD8:64 9C 85 47 AD 65 9C 85 50
9BE0:48 60 A5 3F 18 69 08 85 F8
9BE8:3F A5 40 69 00 85 40 A5 04
9BF0:41 18 69 08 85 41 A5 42 3B
9BF8:69 00 85 42 A5 45 18 69 95
9C00:01 85 45 A5 46 69 00 85 7B
9C08:46 A5 47 18 69 01 85 47 D9
9C10:A5 48 69 00 85 48 60 A5 0F
9C18:3F 38 E9 08 85 3F A5 40 77
9C20:E9 00 85 40 A5 41 38 E9 8F
9C28:08 85 41 A5 42 E9 00 85 88
9C30:42 A5 45 38 E9 01 85 45 C3
9C38:46 E9 00 85 46 A5 47 EA
9C40:38 E9 01 85 47 A5 48 E9 D3
9C48:00 85 48 60 00 00 00 00 F1
9C50:00 00 00 00 00 C1 D6 C1 FF
9C58:C9 CC C1 C2 CC C5 00 00 8B
9C60:00 00 00 00 00 00 00 00 99
9C68:00 00 00 00 00 00 00 00 A1
9C70:00 00 00 00 00 00 00 00 A9
9C78:00 00 00 00 00 00 00 00 B1
9C80:00 00 00 00 00 00 00 00 B9
9C88:00 00 00 00 00 00 A4 B0 BB
9C90:00 00 00 00 00 00 00 00 C9
9C98:00 00 00 00 00 00 00 00 D1
9CA0:00 00 00 00 00 00 00 00 D9
9CA8:00 00 00 00 00 00 00 00 E1
9CB0:00 00 00 00 00 00 00 00 E9
```

BASIC Sketch Loader

```
JS 100 POKE52,32:POKE56,32:CLR
PH 110 IF (PEEK(49998)=87) OR(
       PEEK(49998)=88) THEN GO
       TO140
AJ 120 PRINT"{CLR}LOADING CPBA
       SIC.O{3 DOWN}"
XS 130 POKE49998,87:LOAD"CPBAS
       IC.O",8,1
HC 140 IF PEEK(49998)=88 THEN
       {SPACE}GOTO190
KG 150 PRINT"PLEASE ENTER THE
       {SPACE}COLOR PAD"
SF 160 INPUT"FILE NAME (EXCLUD
       E CP.)";FL$
ED 170 PRINT"{3 DOWN}LOADING C
       P.";FL$
QA 180 POKE49998,88:LOAD"CP."+
       FL$,8,1
ED 190 POKE49999,87:SYS50000
GP 200 GET IN$:IF IN$<>"" THEN
       GOTO200
DX 210 GET IN$:IF IN$="" THEN
       {SPACE}GOTO210
RP 220 SYS50000:PRINT"{CLR}"
AF 230 POKE49998,0
```

CPBASIC .O

```
C350:AD 4F C3 D0 03 4C D6 C3 C2
C358:A9 00 8D 4F C3 AD 20 D0 41
C360:8D 31 C4 AD 21 D0 8D 32 08
C368:C4 A9 68 85 FB A9 42 85 B2
C370:FC A9 00 85 FD A9 20 85 95
C378:FE A9 A8 85 2F C4 A9 61 19
C380:8D 30 C4 20 00 C4 AD 18 FB
C388:D0 09 08 8D 18 D0 AD 11 05
C390:D0 09 20 8D 11 D0 AD 16 DC
C398:D0 09 10 8D 16 D0 A9 00 EC
C3A0:85 FD A9 04 85 FE A9 90 EB
C3A8:8D 2F C4 A9 65 8D 30 C4 7C
C3B0:20 00 C4 A9 00 85 FD A9 37
C3B8:D8 85 FE A9 78 8D 2F C4 A5
C3C0:A9 69 8D 30 C4 20 00 C4 97
C3C8:AD 90 65 4A 4A 4A 4A 8D 3A
C3D0:20 D0 8D 21 D0 60 A9 57 13
C3D8:8D 4F C3 AD 18 D0 29 F7 9C
C3E0:8D 18 D0 AD 11 D0 29 DF 28
C3E8:8D 11 D0 AD 16 D0 29 EF A6
C3F0:8D 16 D0 AD 31 C4 8D 20 91
C3F8:D0 AD 32 C4 8D 21 D0 60 D9
C400:A0 00 B1 FB 91 FD A5 FB 9B
C408:18 69 01 85 FB A5 FC 69 4A
C410:00 85 FC A5 FD 18 69 01 19
C418:85 FD A5 FE 69 00 85 FE DD
C420:A5 FC CD 30 C4 D0 D9 A5 3B
C428:FB CD 2F C4 D0 D2 60 00 E7
C430:00 00 00 00 00 00 00 00 B9
```

CONVERT .O

```
C000:A9 00 85 FB A9 60 85 FC 9D
C008:A9 68 85 FD A9 42 85 FE 69
C010:A0 00 B1 FB 91 FD E6 FB 26
C018:D0 02 E6 FC E6 FD D0 02 02
C020:E6 FE A5 FC C9 87 D0 EA 52
C028:A5 FB C9 10 D0 E4 AD 10 3B
C030:87 0A 0A 0A 0A 85 FB AD E5
C038:90 65 29 0F 05 FB 8D 90 35
C040:65 60 00 00 00 00 00 00 8C
```

"Math Draw" is a menu-driven program that can create graphic webs, seashells, and wheels.

Math Draw For Amiga

Rhett Anderson

Create complex geometric patterns with this menu-driven graphics program for the Amiga. When you've finished, you can save your picture as an IFF graphics file for use with Deluxe Paint II and other Amiga paint programs. Requires 512K.

"Math Draw" lets you create intricate geometric designs and patterns, even if you don't understand the mathematics behind them. Use the special menu to select from a palette of 32 different colors and several different drawing options. When you've finished your picture, you can save it in IFF ILBM format, which is used by most paint programs.

Math Draw is written in Amiga Basic. Type it in and save it to disk. When you're ready to use the program, load and run it.

Math Draw starts drawing immediately. After you've seen the pattern it draws, activate the Amiga menus by pressing the right mouse button. Math Draw adds an Option menu to Amiga Basic's menus. While holding the right button, slide the mouse pointer up to the word Option to see the new menu options. To choose an option, release the button while pointing to the desired option. Here is a list of the new options:

• **Color Control.** When you activate this function, you'll be presented with a chart which shows 32 colors, each labeled with a number. Choose a new drawing color by entering a number. The 32 colors are fixed; they cannot be changed from within the program. If you wish to change the colors, you'll need to modify the numbers in the first set of DATA statements in the program.

• **Radius Control.** Math Draw works by moving an imaginary circle inside or outside another imaginary circle. The drawing pen is "attached" to the edge of the moving circle. You change the size of these circles (and thus the scale of your drawing) by choosing the Radius Control menu selection. You'll be asked for new radius values for the circles.

• **Growth Control.** If you choose, the imaginary circles used to move the drawing pen can grow at a fixed rate. This feature can be used to create spirals and shell designs. Normally, you'll want to keep this parameter at its default value, 0. A reasonable growth rate is 0.01. A negative number will make the circles shrink.

• **Type Control.** This option lets you choose between drawing epicycloids and hypocycloids. In an epicycloid, the moving circle moves around the outside of the stationary circle. In a hypocycloid, the moving circle moves around the inside of the stationary one.

• **Pause.** This option halts drawing. After selecting this option, the drawing pen freezes. This is useful when you want to make several changes to the pen at once. For instance, you may want to change the pen color and the outer radius at the same time. Select Pause and then use the menu options to select the new color and the new radii. Continue is used to start the pen moving again.

• **Continue.** Ends Pause mode.

• **Save Picture.** Saves your picture as an IFF file compatible with *Deluxe Paint II, Digi-Paint, Aegis Images,* and other Amiga paint programs. Since the save routine is written entirely in BASIC, it takes a few minutes to save a picture. Be sure not to drop any menus down while the picture is being saved, or else you'll see ghosts of the menus in your saved pictures.

• **Clear.** Clears the screen. Since your work could be lost if you accidently select this option, you'll be asked whether you're sure you want to clear the screen.

• **Quit.** Exits to BASIC cleanly. You should always use this menu option to stop the program. If you quit another way, type MENU RESET to disable the Math Draw menu and SCREEN CLOSE 1 to free up the screen created by Math Draw. Since

your work could be lost if you select this option accidently, you'll be asked whether you're sure you want to quit.

Your First Drawing

Let's create a picture. First, select Pause to freeze the drawing pen. Now select Clear to erase any picture that might be on the screen. Select Color Control and pick a color. Now select Radius Control. Type 30 at the first prompt, 8 at the second. Select Continue, and a pattern will be drawn. When the pattern begins to repeat, select Pause again and then change the color. Change the radii to 50 and 10. Repeat this process until you're satisfied with your picture. If you wish to save it, select the Save option.

When you're ready for a new effect, turn on the Growth control. A moderate value for Growth is 0.01, but try values like 0.001, 1, and 10 to see how Growth works.

The following table has six sets of parameters that produce attractive designs. After seeing what they do, try combining them.

Radius 1	Radius 2	Growth	Type
120	35	0	2
75	25	−.1	1
10	45	0	2
80	46	0	2
30	16	0	1
40	16	0	1

After you have saved a couple of pictures, go to a paint program and fill in different areas with different colors. From within the paint program, you can modify the palette colors if you like.

Math Draw

For instructions on entering this program, please refer to ''COMPUTE!'s Guide to Typing In Programs'' elsewhere in this issue.

```
'Math Draw◄
'Copyright 1988◄
'COMPUTE! Publications, Inc.◄
'All Rights Reserved◄
◄
CLEAR 32767◄
DEFSNG a-z : DEFDBL g,h,theta◄
TRUE% = -1 : FALSE% =0 : PI = 3.141593 : speed =
 PI/100◄
DIM cmap$(31)◄
◄
coLours:◄
 DATA 0,0,0, 10,10,10, 6,0,0, 9,0,0, 12,0,0, 15,
0,0◄
 DATA 0,6,0, 0,9,0, 0,12,0, 0,15,0, 0,0,6, 0,0,9
◄
 DATA 0,0,12, 0,0,15, 4,4,0, 7,7,0, 10,10,0, 13,
13,0◄
 DATA 4,0,4, 7,0,7, 10,0,10, 13,0,13, 6,3,0, 8,4
,0◄
 DATA 10,5,0, 12,6,0, 4,8,12, 8,4,12, 4,12,8, 12
,8,4◄
 DATA 12,4,8, 8,12,4◄
◄
ON BREAK GOSUB NoMore◄
BREAK ON◄
SCREEN 1,320,200,5,1:WINDOW 3,"",(0,0)-(311,186)
,16,1:WINDOW OUTPUT 3◄
RESTORE coLours◄
FOR i%=0 TO 31◄
 READ r,g,b◄
 PALETTE i%,r/15,g/15,b/15◄
 cmap$(i%)=CHR$(r*17)+CHR$(g*17)+CHR$(b*17)◄
NEXT i%◄
MENU 3,0,1,"Options"◄
MENU 3,1,1,"Color Control"◄
```

```
MENU 3,2,1,"Radius Control"◄
MENU 3,3,1,"Growth Control"◄
MENU 3,4,1,"Type Control"◄
MENU 3,5,1,"Save Picture"◄
MENU 3,6,1,"Pause"◄
MENU 3,7,1,"Continue"◄
MENU 3,8,1,"Clear"◄
MENU 3,9,1,"Quit"◄
ON MENU GOSUB checkmenu◄
a=100:b=20:type%=1:coLour=10:deLtaA=0:deLtaB=0◄
starting = TRUE% : pause = FALSE%◄
nextpoint:◄
MENU ON◄
WHILE pause:WEND◄
MENU STOP◄
 g = (a+b)/b : h = (a-b)/b◄
 theta = theta+speed◄
 a = a+deLtaA◄
 b = b+deLtaB◄
 oLdx = x: oLdy = y◄
 IF type% = 1 THEN GOSUB EpicycLoid ELSE GOSUB H
ypocycLoid◄
 IF NOT starting THEN LINE (160+oLdx,100+oLdy*.8
8)-(160+x,100+y*.88),coLour◄
 starting = FALSE%◄
GOTO nextpoint◄
◄
EpicycLoid:◄
 x = (a+b)*COS(theta)-b*COS(g*theta)◄
 y = (a+b)*SIN(theta)-b*SIN(g*theta)◄
RETURN◄
◄
HypocycLoid:◄
 x=(a-b)*COS(theta)+b*COS(h*theta)◄
 y=(a-b)*SIN(theta)-b*SIN(h*theta)◄
RETURN ◄
◄
checkmenu:◄
 menuid=MENU(0)◄
 menuitem=MENU(1)◄
 IF menuid=3 THEN◄
  IF menuitem = 1 THEN◄
   WINDOW 4,"",(0,0)-(260,82),18,1:WINDOW OUTPUT
4◄
    FOR xx%=0 TO 7◄
     FOR yy%=0 TO 3◄
      LINE (xx%*32+16,yy%*16+8)-(xx%*32+24,yy%*16
+16),xx%+yy%*8,bf◄
      LOCATE yy%*2+2,xx%*4+1◄
      PRINT RIGHT$(STR$(xx%+yy%*8),2);◄
     NEXT yy%◄
    NEXT xx%◄
    LOCATE 10,4◄
    INPUT "Choose a Color";coLour : coLour=coLour
MOD 32◄
    WINDOW OUTPUT 3◄
    WINDOW CLOSE 4◄
  END IF◄
  IF menuitem = 2 THEN◄
   WINDOW 4,"",(0,0)-(280,65),18,1:WINDOW OUTPUT
4◄
    PRINT "Current Radius":PRINT" of Stationary C
ircle:";a◄
    INPUT "New Radius";a◄
    PRINT "Current Radius":PRINT" of Moving Circl
e:";b◄
    INPUT "New Radius";b◄
    a=ABS(a):b=ABS(b)◄
    IF a=0 THEN a=1◄
    IF b=0 THEN b=1◄
    starting = TRUE%◄
    WINDOW OUTPUT 3◄
    WINDOW CLOSE 4◄
  END IF◄
  IF menuitem = 3 THEN◄
   WINDOW 4,"",(0,0)-(300,50),18,1:WINDOW OUTPUT
4◄
    PRINT "Current Growth Rate:";deLtaA◄
    INPUT "New Rate";deLtaA:IF a<>0 THEN deLtaB=d
eLtaA*b/a◄
    starting = TRUE%◄
    WINDOW OUTPUT 3◄
    WINDOW CLOSE 4◄
  END IF◄
  IF menuitem = 4 THEN◄
```

```
    WINDOW 4,"",(0,0)-(250,50),18,1:WINDOW OUTPUT
4
    PRINT "Choose a Type:"
    PRINT "1. Epicycloid (Outside)"
    PRINT "2. Hypocycloid (Inside)"
    PRINT:PRINT"Current Type =";type%
    INPUT "New Type";type%
    IF type% = 2 THEN speed = PI/50 ELSE speed =
PI/100
    starting = TRUE%
    WINDOW OUTPUT 3
    WINDOW CLOSE 4
  END IF
  IF menuitem = 5 THEN
    WINDOW 4,"",(0,0)-(300,40),18,1:WINDOW OUTPUT
4
    INPUT "Filename For Save";fiL$
    WINDOW OUTPUT 3 : WINDOW CLOSE 4
    IF fiL$<>"" THEN GOSUB savepic
  END IF
  IF menuitem = 6 THEN pause = TRUE%
  IF menuitem = 7 THEN pause = FALSE%
  IF menuitem = 8 THEN
    WINDOW 4,"",(0,0)-(250,50),18,1:WINDOW OUTPUT
4
    PRINT "Are You Sure You Want":PRINT" To Erase
the Screen?"
    INPUT a$
    WINDOW OUTPUT 3 : WINDOW CLOSE 4
    IF UCASE$(LEFT$(a$,1))="Y" THEN CLS : startin
g = TRUE
  END IF
  IF menuitem = 9 THEN
    WINDOW 4,"",(0,0)-(250,50),18,1 : WINDOW OUTP
UT 4
    PRINT "Are You Sure You Want To Quit?"
    INPUT a$
    IF UCASE$(LEFT$(a$,1))="Y" THEN GOTO NoMore
    WINDOW OUTPUT 3 : WINDOW CLOSE 4
  END IF
 END IF
RETURN

savepic:
  rastport& = WINDOW(8)
  bitmap& = PEEKL(rastport&+4)
  topLine = 0
  topadd = topLine * 40
  FOR i% = 0 TO 4
    pLane&(i%) = PEEKL(bitmap& + 8 + i%*4)+topadd
  NEXT
  bottomLine = 200
  Lines = bottomLine - topLine
  OPEN fiL$ FOR OUTPUT AS 1
  a$ = MKL$(Lines * 40 * 5 + 144)
  PRINT#1,"FORM";a$;"ILBMBMHD";MKL$(20);
  PRINT#1,MKI$(320);MKI$(Lines);MKL$(0);
  PRINT#1,CHR$(5);MKI$(0);CHR$(0);
  PRINT#1,MKI$(0);CHR$(10);CHR$(11);
  PRINT#1,MKI$(320);MKI$(200);
  PRINT#1,"CMAP";MKL$(96);
  FOR i% = 0 TO 31
    PRINT#1,cmap$(i%);
  NEXT
  PRINT#1,"BODY";MKL$(Lines * 40 * 5);
  FOR i% = 1 TO Lines
    FOR p% = 0 TO 4
      FOR j% = 0 TO 39 STEP 4
        PRINT#1,MKL$(PEEKL(pLane&(p%) + j%));
      NEXT j%
      pLane&(p%) = pLane&(p%) + 40
    NEXT p%
  NEXT i%
  CLOSE
  PALETTE 0,1,1,1:SOUND 440,20,200,0:PALETTE 0,0,
0,0
RETURN

NoMore:
  WINDOW CLOSE 3 : WINDOW CLOSE 4 : SCREEN CLOSE
1
  MENU RESET
  MENU OFF
END
```

Catalog To Text File

Jason Fuller

Apple II users can turn disk directories into text files with this BASIC program. Saved as text, disk catalogs can be edited, and then printed with a word processor or your own BASIC program. Works with DOS 3.3 disks only.

Unlike many computers, the Apple II doesn't provide a way to make text files from disk catalogs. "Catalog to Text File" is a BASIC program that solves this problem. When run, Catalog to Text File reads in a disk's directory and creates a text file that you can print and/or edit using almost any word processor.

By modifying the program to suit your needs, Catalog to Text File can be used to read the number of free sectors on a disk, to count the number of files on a particular volume, or to print out selected filenames in an easy-to-read format. This program works with unprotected DOS 3.3 disks only.

Getting Started

To begin, type in Programs 1 and 2 using the "Automatic Proofreader" program found elsewhere in this issue. When you're done, be sure to save a copy of each program.

Next, load and run Program 1. Program 1 writes a short machine language file to disk. This file is BLOADed into memory by Program 2, so it is important that the machine language file and Program 2 are saved to the same disk.

Creating A Text File

Load and run Program 2. The program prompts you for the volume number (or name) of a disk. Enter anything you wish, including letters. The characters that you enter become part of the filename used to save the disk's catalog.

When the computer prompts you, insert the disk that you wish to catalog and press Return. After the computer has read the catalog into memory, you are asked to insert the destination disk. Insert a disk and press Return. The catalog is written to disk with the filename CAT-VOL followed by the volume name that you entered earlier.

Using The Text File

The text file created by this program can be loaded by most word processors. From there you may edit it, reformat it, and print it out. When viewed, the catalog text file appears in the following format:

Volume number, as entered by the user
Total number of used sectors
Total number of files on the disk
Filenames
.
.
.
Filenames
The word *END*

It is possible to use this text file from within your own BASIC programs. To read the file, use the following subroutine:

```
5B 1000 TEXT : HOME
8F 1010 INPUT "WHAT VOLUME NUMBE
        R TO LOAD ";VOL$
2B 1020 D$ = CHR$ (4): REM CTRL-
        D
1C 1030 PRINT D$;"OPEN CAT-VOL "
        ;VOL$
B1 1040 PRINT D$;"READ CAT-VOL "
        ;VOL$
77 1050 INPUT VOL$
53 1060 INPUT USED
45 1070 INPUT NUM
11 1080 FOR X = 1 TO NUM
2E 1090 INPUT NAME$(X)
7C 1100 NEXT X
C3 1110 PRINT D$;"CLOSE CAT-VOL
        ";VOL$
DB 1120 RETURN
```

If you would like see the catalog text file sent to the screen, add these lines to the subroutine above.

```
8F 100 DIM NAME$(75): GOSUB 1000
A2 110 FOR X = 1 TO NUM
87 120 PRINT NAME$(X)
6B 130 NEXT X
91 140 END
```

Catalog To Text File

For instructions on entering these programs, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

Program 1: ML File Creator

```
5F 10 T = 0
52 20 D$ = CHR$ (4)
2D 30 FOR X = 1 TO 52
75 40 READ D
4C 50 T = T + D
BC 60 POKE 767 + X,D
E0 70 NEXT X
BE 80 IF T < > 1717 THEN PRINT "
      ERROR IN DATA STATEMENTS.
      PLEASE CHECK.": END
A9 90 PRINT D$;"BSAVE IOB,A 768,
      L 53"
89 100 END
29 200 DATA 169,3,160,16,32,217,
       3,96,0,0
7C 210 DATA 0,0,0,0,0,0,1,96,1,0
       ,17,15
CB 220 DATA 48,3,0,32,0,0,1,0,25
       4,96,1
54 230 DATA 0,0,0,0,0,0,0,0
F1 240 DATA 0,0,0,0,0,0,0,0,1,23
       9,216
```

Program 2: Main Program

```
D6 10 REM COPYRIGHT 1988 COMPUTE
      ! PUBLICATIONS, INC.
69 20 REM ALL RIGHTS RESERVED.
C7 100 ONERR GOTO 1600
CA 105 HIMEM: 8192
54 110 D$ = CHR$ (4)
5B 120 TEXT : HOME
B1 130 IF PEEK (768) = 169 THEN
       150
```

www.commodore.ca

```
64 140 PRINT D$;"BLOAD IOB,D1"
FF 150 INVERSE : FOR X = 1 TO 6:
       FOR Y = 1 TO 40: PRINT "
       ";: NEXT : NEXT
4A 160 A$ = "PRINT CATALOG TO A
       TEXT FILE": GOSUB 1200
72 170 VTAB 2: HTAB HT: PRINT A$
1D 180 A$ = "COPYRIGHT '88 COMPU
       TE!": GOSUB 1200: INVERSE
77 190 VTAB 4: HTAB HT: PRINT A$
10 200 POKE 34,8
DB 210 DIM FE$(100),NB$(35)
C9 220 NORMAL
A6 230 HOME : PRINT "WHAT IS VOL
       UME NUMBER OF THE DISK YO
       U": PRINT "WANT CATALOGUE
       D?"
61 240 INPUT "";VOL$
CA 250 IF LEN (VOL$) > 24 THEN 2
       30
33 260 INVERSE
F4 270 HOME : PRINT "DISK VOLUME
       ";VOL$
D5 280 NORMAL
D6 290 PRINT : PRINT : PRINT : P
       RINT "IS THIS CORRECT?";
54 300 GET S$
D6 310 IF S$ = "Y" OR S$ = "y" T
       HEN 340
2B 320 IF S$ = "N" OR S$ = "n" T
       HEN 230
93 330 GOTO 300
4D 340 HOME : PRINT "PLEASE INSE
       RT": PRINT : PRINT "DISK
       VOLUME ";VOL$: PRINT : PR
       INT "AND PRESS RETURN.";
D1 350 GOSUB 1400
C9 400 A$ = "PLEASE WAIT. READIN
       G CATALOG TO MEMORY": GOS
       UB 1200
6D 410 HOME : HTAB HT: PRINT A$
14 420 SS = 0:TR = 17:SE = 15:BU
       = 8192:OP = 1: GOSUB 150
       0
E1 430 STA = 12046:EN = 12256:LL
       = 0:KK = 0:USED = 0:NB$
       = "":FE$ = ""
9E 440 FOR X = STA TO EN STEP 35
B0 450 IF PEEK (X) = 0 THEN KK =
       LL: GOTO 700: REM UNUSE
       D FILE
44 460 IF PEEK (X - 3) = 255 THE
       N 560: REM DELETED FILE
50 470 E = PEEK (X - 1): GOSUB 1
       000: REM TYPE
CD 480 LY = PEEK (X + 30): GOSUB
       1300:FE$(LL) = FE$(LL) +
       FIN$:FE$(LL) = FE$(LL) +
       "-": REM LENGTH OF FILE
       IN SECTORS
16 490 USED = USED + LY: REM CO
       UNTER FOR SECTORS USED
74 500 NB$ = ""
E5 510 FOR C = X TO X + 29: IF P
       EEK (C) > 0 AND PEEK (C)
       < = 31 OR PEEK (C) > 127
       AND PEEK (C) < = 159 THEN
       NB$ = NB$ + " ": GOTO 53
       0
3D 520 NB$ = NB$ + CHR$ ( PEEK (
       C))
E4 530 NEXT C
6E 540 FE$(LL) = FE$(LL) + NB$
77 550 LL = LL + 1
75 560 NEXT X
49 570 STA = EN - 466:EN = STA +
       210
24 580 GOTO 440
BF 700 HOME : PRINT "PLEASE INSE
       RT A DISK WITH WHICH"
A9 710 PRINT "YOU WANT TO STORE
       THE TEXT FILE AND"
DB 720 PRINT "PRESS RETURN";

D1 730 GOSUB 1400
EF 740 PRINT : PRINT D$
48 750 PRINT D$;"OPEN CAT-VOL ";
       VOL$
1A 760 PRINT D$;"WRITE CAT-VOL "
       ;VOL$
C3 770 PRINT VOL$
B1 780 PRINT USED
5C 790 PRINT KK
94 800 FOR X = 0 TO (LL - 1): PR
       INT FE$(X): NEXT
B9 810 PRINT "*END*"
12 820 PRINT D$;"CLOSE CAT-VOL "
       ;VOL$
2B 830 TEXT : HOME : PRINT "DONE
       ": END
E5 998 :
7B 999 REM   TYPE OF FILE
B5 1000 IF E = 0 OR E = 128 THEN
        FE$(LL) = "T-": GOTO 10
        80
D9 1010 IF E = 1 OR E = 129 THEN
        FE$(LL) = "I-": GOTO 10
        80
99 1020 IF E = 2 OR E = 130 THEN
        FE$(LL) = "A-": GOTO 10
        80
C0 1030 IF E = 4 OR E = 132 THEN
        FE$(LL) = "B-": GOTO 10
        80
EC 1040 IF E = 8 OR E = 136 THEN
        FE$(LL) = "S-": GOTO 10
        80
8C 1050 IF E = 10 OR E = 144 THE
        N FE$(LL) = "R-": GOTO 1
        080
D5 1060 IF E = 20 OR E = 160 THE
        N FE$(LL) = "A1-": GOTO
        1080
9F 1070 IF E = 40 OR E = 192 THE
        N FE$(LL) = "B1-": GOTO
        1080
F1 1080 RETURN
6D 1198 :
64 1199 REM   CENTER TEXT
5C 1200 HT = ((40 - LEN (A$)) /
        2) + 1
41 1210 NORMAL
DD 1220 RETURN
6F 1298 :
CA 1299 REM   LENGTH OF FILE
3C 1300 GG$ = STR$ (LY)
7A 1310 IF LEN (GG$) = 1 THEN FI
        N$ = "00" + STR$ (LY): R
        ETURN
CC 1320 IF LEN (GG$) = 2 THEN FI
        N$ = "0" + STR$ (LY): RE
        TURN
E3 1330 RETURN
71 1398 :
18 1399 REM   GET RETURN
5C 1400 GET S$
2B 1410 IF S$ < > CHR$ (13) THEN
        1400
E1 1420 RETURN
73 1498 :
89 1499 REM   ACCESS DISK
B4 1500 FOR SA = SS TO SE
99 1510 POKE 788,TR: POKE 789,SA
        : POKE 796,OP
39 1520 HB = INT (BU / 256):LB =
        BU - (HB * 256)
B6 1530 POKE 792,LB: POKE 793,HB
F7 1540 CALL 768:BU = BU + 256:
        NEXT SA
EF 1550 RETURN
75 1598 :
E1 1599 REM   ERROR ROUTINE
DC 1600 EC = PEEK (222)
7D 1610 LN = PEEK (218) + PEEK (
        219) * 256
AD 1620 POKE 216,0
A5 1630 PRINT D$

BB 1640 FOR X = 1 TO 7: PRINT CH
        R$ (7): NEXT
97 1650 IF EC = 4 THEN 1720
A1 1660 IF EC = 6 THEN 1740
AD 1670 IF EC = 9 THEN 1760
B1 1680 IF EC = 8 THEN 1780
9B 1690 PRINT "ERROR #:";EC
27 1700 PRINT "IN LINE #:";LN
A1 1710 STOP
5D 1720 INVERSE : PRINT "REMOVE
        WRITE-PROTECT TAB.": NOR
        MAL
74 1730 GOTO 1800
1F 1740 INVERSE : PRINT "FILE NO
        T FOUND.": NORMAL
7C 1750 GOTO 1800
34 1760 INVERSE : PRINT "THE DIS
        K IS FULL.^": NORMAL
84 1770 GOTO 1800
E8 1780 INVERSE : PRINT "I/O ERR
        OR WITH CURRENT DISK.":
        NORMAL
8C 1790 GOTO 1800
86 1800 INVERSE : PRINT "ERROR I
        N LINE ";LN
36 1810 PRINT : PRINT "PLEASE IN
        SERT THE CATALOG UTILITY
        DISK AND PRESS RETURN."
        ;
53 1820 GOSUB 1400
98 1830 CLEAR : TEXT : HOME : RU
        N
                                ©
```

COMPUTE!'s Guide To Typing In Programs

Computers are precise—type the program *exactly* as listed, including necessary punctuation and symbols, except for special characters noted below. We have provided a special listing convention as well as a set of programs to check your typing—"The Automatic Proofreader."

Programs for the IBM and those in ST BASIC for Atari ST models should be typed exactly as listed; no special characters are used. Programs for Commodore, Apple, and Atari 400/800/XL/XE computers may contain some hard-to-read special characters, so we have a listing system that indicates these control characters. You will find these characters in curly braces; *do not type the braces*. For example, {CLEAR} or {CLR} instructs you to type the character which clears the screen on the Atari or Commodore machines. A complete list of these symbols is shown in the tables below. For Commodore, Apple, and Atari, a single symbol by itself within curly braces is a control key or graphics key. If you see {A}, hold down the CONTROL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple.

For Commodore computers, graphics characters entered with the Commodore logo key are enclosed in a special bracket: [<A>]. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is {SHIFT-SPACE}. When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, repeat the character the indicated number of times. For example, {5 RIGHT}, {6 S}, and [<8 Q>], mean, respectively, that you should enter five cursor rights, six shifted S's, and eight Commodore-Q's. On the Atari, inverse characters (white on black) should be entered with the inverse vid-

Atari 400/800/XL/XE

When you see	Type	See	
{CLEAR}	ESC SHIFT <	↰	Clear Screen
{UP}	ESC CTRL –	↑	Cursor Up
{DOWN}	ESC CTRL =	↓	Cursor Down
{LEFT}	ESC CTRL +	←	Cursor Left
{RIGHT}	ESC CTRL *	→	Cursor Right
{BACK S}	ESC DELETE	◀	Backspace
{DELETE}	ESC CTRL DELETE	▯◀	Delete character
{INSERT}	ESC CTRL INSERT	▮▷	Insert character
{DEL LINE}	ESC SHIFT DELETE	▯	Delete line
{INS LINE}	ESC SHIFT INSERT	▮	Insert line
{TAB}	ESC TAB	▶	TAB key
{CLR TAB}	ESC CTRL TAB	◀	Clear tab
{SET TAB}	ESC SHIFT TAB	▶	Set tab stop
{BELL}	ESC CTRL 2	▯	Ring buzzer
{ESC}	ESC ESC	▮	ESCape key

Commodore PET/CBM/VIC/64/128/16/+4

When You Read:	Press:		See:	When You Read:	Press:		See:
{CLR}	SHIFT	CLR/HOME	▧	[1]	COMMODORE	1	♣
{HOME}		CLR/HOME	▧	[2]	COMMODORE	2	◤
{UP}	SHIFT	↑ CRSR ↓	▮	[3]	COMMODORE	3	▨
{DOWN}		↑ CRSR ↓	▯	[4]	COMMODORE	4	▢
{LEFT}	SHIFT	← CRSR →	▮	[5]	COMMODORE	5	✚
{RIGHT}		← CRSR →	▯	[6]	COMMODORE	6	▮
{RVS}	CTRL	9	�R	[7]	COMMODORE	7	◒
{OFF}	CTRL	0	▮	[8]	COMMODORE	8	▦
{BLK}	CTRL	1	▮	{ F1 }		f1	▨
{WHT}	CTRL	2	E	{ F2 }	SHIFT	f1	▨
{RED}	CTRL	3	▦	{ F3 }		f3	▨
{CYN}	CTRL	4	▨	{ F4 }	SHIFT	f3	▨
{PUR}	CTRL	5	▨	{ F5 }		f5	▨
{GRN}	CTRL	6	↑	{ F6 }	SHIFT	f5	▨
{BLU}	CTRL	7	←	{ F7 }		f7	▨
{YEL}	CTRL	8	Ⅲ	{ F8 }	SHIFT	f7	▨
				◀		←	▨

C·www.commodore.ca

eo key (Atari logo key on 400/800 models).

Whenever more than two spaces appear in a row, they are listed in a special format. For example, {6 SPACES} means press the space bar six times. Our Commodore listings never leave a single space at the end of a line, instead moving it to the next printed line as {SPACE}.

Amiga program listings and Atari ST program listings in GFA BASIC contain only one special character, the left arrow (←) symbol. This character marks the end of each program line. Wherever you see a left arrow, press RETURN to enter that line into memory. (For the Amiga, you can also enter the line simply by moving the cursor off the line.) Don't try to type in the left arrow symbol; it's there only as a marker to indicate where each program line ends.

The Automatic Proofreader

Type in the appropriate program listed below, then save it for future use. The Commodore Proofreader works on the Commodore 128, 64, Plus/4, 16, and VIC-20. Don't omit any lines, even if they contain unfamiliar commands or you think they don't apply to your computer. When you run the program, it installs a machine language program in memory and erases its BASIC portion automatically (so be sure to save several copies before running the program for the first time). If you're using a Commodore 128, Plus/4 or 16, do *not* use any GRAPHIC commands while the Proofreader is active. You should disable the Commodore Proofreader before running any other program. To do this, either turn the computer off and on or enter SYS 64738 (for the 64), SYS 65341 (128), SYS 64802 (VIC-20), or SYS 65526 (Plus/4 or 16). To reenable the Proofreader, reload the program and run it as usual. Unlike the original VIC/64 Proofreader, this version works the same with disk or tape.

The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate. Be sure to leave Caps Lock on, except when typing lowercase characters.

On the Atari, run the Proofreader to activate it (the Proofreader remains active in memory as a machine language program); you must then enter NEW to erase the BASIC loader. Pressing SYSTEM RESET deactivates the Atari Proofreader; enter PRINT USR (1536) to reenable it.

The Apple Proofreader erases the BASIC portion of itself after you run it, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a hexadecimal number (on the Apple) or a pair of letters (on the Commodore, Atari, or IBM) appears. The number or pair of letters is called a *checksum*.

Compare the value displayed on the screen by the Proofreader with the checksum printed in the program listing in the magazine. The checksum is given to the left of each line number. Just type in the program a line at a time (without the printed checksum), press RETURN or Enter, and compare the checksums. If they match, go on to the next line. If not, check your typing; you've made a mistake. Because of the checksum method used, do not type abbreviations, such as ? for PRINT. On the Atari and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Atari Proofreader does not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. The Commodore Proofreader catches transposition errors and ignores spaces unless they're enclosed in quotation marks. The IBM Proofreader detects errors in spacing and transposition.

IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you enter NEW, the Proofreader prompts you to press Y to be especially sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to resave it to disk. This will shorten it on disk and make it load faster, but it can no longer be edited with the Proofreader. If you want to convert an existing BASIC program to Proofreader format, save it to disk with SAVE *"filename",A*.

Program 1: Atari Proofreader
By Charles Brannon

```
100 GRAPHICS Ø
110 FOR I=1536 TO 1700:RE
    AD A:POKE I,A:CK=CK+A
    :NEXT I
120 IF CK<>19072 THEN ? "
    Error in DATA Stateme
    nts.  Check Typing.":
    END
130 A=USR(1536)
140 ? :? "Automatic Proof
    reader Now Activated.
    "
150 END
160 DATA 104,160,0,185,26
    ,3,201,69,240,7
170 DATA 200,200,192,34,2
    08,243,96,200,169,74
180 DATA 153,26,3,200,169
    ,6,153,26,3,162
190 DATA 0,189,0,228,157,
    74,6,232,224,16
200 DATA 208,245,169,93,1
    41,78,6,169,6,141
210 DATA 79,6,24,173,4,22
    8,105,1,141,95
220 DATA 6,173,5,228,105,
    0,141,96,6,169
230 DATA 0,133,203,96,247
    ,238,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,205,238
250 DATA 0,0,0,0,0,32,62,
    246,8,201
260 DATA 155,240,13,201,3
    2,240,7,72,24,101
270 DATA 203,133,203,104,
    40,96,72,152,72,138
280 DATA 72,160,0,169,128
    ,145,88,200,192,40
290 DATA 208,249,165,203,
    74,74,74,74,24,105
300 DATA 161,160,3,145,88
    ,165,203,41,15,24
310 DATA 105,161,200,145,
    88,169,0,133,203,104
320 DATA 170,104,168,104,
    40,96
```

Program 2: Commodore Proofreader
By Philip Nelson

```
10 VEC=PEEK(772)+256*PEEK(773)
   :LO=43:HI=44
20 PRINT "AUTOMATIC PROOFREADE
   R FOR ";:IF VEC=42364 THEN
   {SPACE}PRINT "C-64"
30 IF VEC=50556 THEN PRINT "VI
   C-20"
40 IF VEC=35158 THEN GRAPHIC C
   LR:PRINT "PLUS/4 & 16"
50 IF VEC=17165 THEN LO=45:HI=
   46:GRAPHIC CLR:PRINT"128"
60 SA=(PEEK(LO)+256*PEEK(HI))+
   6:ADR=SA
70 FOR J=0 TO 166:READ BYT:POK
   E ADR,BYT:ADR=ADR+1:CHK=CHK
   +BYT:NEXT
80 IF CHK<>20570 THEN PRINT "*
   ERROR* CHECK TYPING IN DATA
   STATEMENTS":END
90 FOR J=1 TO 5:READ RF,LF,HF:
   RS=SA+RF:HB=INT(RS/256):LB=
   RS-(256*HB)
100 CHK=CHK+RF+LF+HF:POKE SA+L
    F,LB:POKE SA+HF,HB:NEXT
```

```
110 IF CHK<>22054 THEN PRINT "
    *ERROR* RELOAD PROGRAM AND
    {SPACE}CHECK FINAL LINE":EN
    D
120 POKE SA+149,PEEK(772):POKE
    SA+150,PEEK(773)
130 IF VEC=17165 THEN POKE SA+
    14,22:POKE SA+18.23:POKESA+
    29,224:POKESA+139,224
140 PRINT CHR$(147);CHR$(17);"
    PROOFREADER ACTIVE":SYS SA
150 POKE HI,PEEK(HI)+1:POKE (P
    EEK(LO)+256*PEEK(HI))-1,0:N
    EW
160 DATA 120,169,73,141,4,3,16
    9,3,141,5,3
170 DATA 88,96,165,20,133,167,
    165,21,133,168,169
180 DATA 0,141,0,255,162,31,18
    1,199,157,227,3
190 DATA 202,16,248,169,19,32,
    210,255,169,18,32
200 DATA 210,255,160,0,132,180
    ,132,176,136,230,180
210 DATA 200,185,0,2,240,46,20
    1,34,208,8,72
220 DATA 165,176,73,255,133,17
    6,104,201,32,208
230 DATA 7,165,176,208,3,104,2
    08,226,104,166,180
240 DATA 24,165,167,121,0,2,13
    3,167,165,168,105
250 DATA 0,133,168,202,208,239
    ,240,202,165,167,69
260 DATA 168,72,41,15,168,185,
    211,3,32,210,255
270 DATA 104,74,74,74,74,168,1
    85,211,3,32,210
280 DATA 255,162,31,189,227,3,
    149,199,202,16,248
290 DATA 169,146,32,210,255,76
    ,86,137,65,66,67
300 DATA 68,69,70,71,72,74,75,
    77,80,81,82,83,88
310 DATA 13,2,7,167,31,32,151,
    116,117,151,128,129,167,136
    ,137
```

Program 3: IBM Proofreader

By Charles Brannon

```
10 'Automatic Proofreader Ver
    sion 3.0 (Lines 205,206 ad
    ded/190 deleted/470,490 ch
    anged from V2.0)
100 DIM L$(500),LNUM(500):COL
    OR 0,7,7:KEY OFF:CLS:MAX=
    0:LNUM(0)=65536!
110 ON ERROR GOTO 120:KEY 15,
    CHR$(4)+CHR$(70):ON KEY(1
    5) GOSUB 640:KEY (15) ON:
    GOTO 130
120 RESUME 130
130 DEF SEG=&H40:W=PEEK(&H4A)
140 ON ERROR GOTO 650:PRINT:P
    RINT"Proofreader Ready."
150 LINE INPUT L$:Y=CSRLIN-IN
    T(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:PO
    KE 1052,34:POKE 1054,0:PO
    KE 1055,79:POKE 1056,13:P
    OKE 1057,28:LINE INPUT L$
    :DEF SEG:IF L$="" THEN 15
    0
170 IF LEFT$(L$,1)=" " THEN L
    $=MID$(L$,2):GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND
    MID$(L$,3,1)=" " THEN L$
    =MID$(L$,4)
200 IF ASC(L$)>57 THEN 260 'n
    o line number, therefore
    command
205 BL=INSTR(L$," "):IF BL=0
    THEN BL$=L$:GOTO 206 ELSE
    BL$=LEFT$(L$,BL-1)
206 LNUM=VAL(BL$):TEXT$=MID$(
    L$,LEN(STR$(LNUM))+1)
210 IF TEXT$="" THEN GOSUB 54
    0:IF LNUM=LNUM(P) THEN GO
    SUB 560:GOTO 150 ELSE 150
220 CKSUM=0:FOR I=1 TO LEN(L$
    ):CKSUM=(CKSUM+ASC(MID$(L
    $,I))*I) AND 255:NEXT:LOC
    ATE Y,1:PRINT CHR$(65+CKS
    UM/16)+CHR$(65+(CKSUM AND
    15))+" "+L$
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT$:GOTO 15
    0 'replace line
240 GOSUB 580:GOTO 150 'inser
    t the line
260 TEXT$="":FOR I=1 TO LEN(L
    $):A=ASC(MID$(L$,I)):TEXT
    $=TEXT$+CHR$(A+32*(A>96 A
    ND A<123)):NEXT
270 DELIMITER=INSTR(TEXT$," "
    ):COMMAND$=TEXT$:ARG$="":
    IF DELIMITER THEN COMMAND
    $=LEFT$(TEXT$,DELIMITER-1
    ):ARG$=MID$(TEXT$,DELIMIT
    ER+1) ELSE DELIMITER=INST
    R(TEXT$,CHR$(34)):IF DELI
    MITER THEN COMMAND$=LEFT$
    (TEXT$,DELIMITER-1):ARG$=
    MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN
    410
290 OPEN "scrn:" FOR OUTPUT A
    S #1
300 IF ARG$="" THEN FIRST=0:P
    =MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-")
    :IF DELIMITER=0 THEN LNUM
    =VAL(ARG$):GOSUB 540:FIRS
    T=P:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELI
    MITER)):LAST=VAL(MID$(ARG
    $,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIRS
    T=P:LNUM=LAST:GOSUB 540:I
    F P=0 THEN P=MAX-1
340 FOR X=FIRST TO P:N$=MID$(
    STR$(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN A$="":GO
    TO 370
360 CKSUM=0:A$=N$+L$(X):FOR I
    =1 TO LEN(A$):CKSUM=(CKSU
    M+ASC(MID$(A$,I))*I) AND
    255:NEXT:A$=CHR$(65+CKSUM
    /16)+CHR$(65+(CKSUM AND 1
    5))+" "
370 PRINT #1,A$+N$+L$(X)
380 IF INKEY$<>"" THEN X=P
390 NEXT :CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN
    OPEN "lpt1:" FOR OUTPUT A
    S #1:GOTO 300
420 IF COMMAND$="CHECK" THEN
    CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN
    450
440 GOSUB 600:OPEN ARG$ FOR O
    UTPUT AS #1:ARG$="":GOTO
    300
450 IF COMMAND$<>"LOAD" THEN
    490
460 GOSUB 600:OPEN ARG$ FOR I
    NPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INP
    UT #1,L$:BL=INSTR(L$," ")
    :BL$=LEFT$(L$,BL-1):LNUM(
    P)=VAL(BL$):L$(P)=MID$(L$
    ,LEN(STR$(VAL(BL$)))+1):P
    =P+1:WEND
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure";L$:IF LEFT$(L$,
    1)="y" OR LEFT$(L$,1)="Y"
    THEN MAX=0:LNUM(0)=65536
    !:GOTO 130:ELSE 130
500 IF COMMAND$="BASIC" THEN
    COLOR 7,0,0:ON ERROR GOTO
    0:CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO
    130
540 P=0:WHILE LNUM>LNUM(P) AN
    D P<MAX:P=P+1:WEND:RETURN
560 MAX=MAX-1:FOR P=P TO MAX:
    LNUM(X)=LNUM(X+1):L$(X)=L
    $(X+1):NEXT:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+
    1 STEP -1:LNUM(X)=LNUM(X-
    1):L$(X)=L$(X-1):NEXT:L$(
    P)=TEXT$:LNUM(P)=LNUM:RET
    URN
600 IF LEFT$(ARG$,1)<>CHR$(34
    ) THEN 520 ELSE ARG$=MID$
    (ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34
    ) THEN ARG$=LEFT$(ARG$,LE
    N(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,"
    .")=0 THEN ARG$=ARG$+".BA
    S"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"S
    topped.":RETURN 150
650 PRINT "Error #";ERR:RESUM
    E 150
```

Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```
10 C = 0: FOR I = 768 TO 768
    + 68: READ A:C = C + A: PO
    KE I,A: NEXT
20 IF C < > 7258 THEN PRINT "
    ERROR IN PROOFREADER DATA
    STATEMENTS": END
30 IF PEEK (190 * 256) < > 76
    THEN POKE 56,0: POKE 57,3
    : CALL 1002: GOTO 50
40 PRINT CHR$ (4);"IN#A$300"
50 POKE 34,0: HOME : POKE 34,
    1: VTAB 2: PRINT "PROOFREA
    DER INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,14
    1
110 DATA 208,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,125,255
140 DATA 1,105,0,72,202,208
150 DATA 238,104,170,41,15,9
160 DATA 48,201,58,144,2,233
170 DATA 57,141,1,4,138,74
180 DATA 74,74,74,74,41,15,9
190 DATA 48,201,58,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96
```

See bottom of page

C www.Commodore.ca

www.commodore.ca

MLX Machine Language Entry Program For Commodore 64

Ottis Cowper, Technical Editor

"MLX" is a labor-saving utility that allows almost fail-safe entry of Commodore 64 machine language programs.

Type in and save some copies of MLX—you'll want to use it to enter future machine langauge (ML) programs from COMPUTE!. When you're ready to enter an ML program, load and run MLX. It asks you for a starting address and an ending address. These addresses appear in the article accompanying the MLX-format program listing you're typing.

If you're unfamiliar with machine language, the addresses (and all other values you enter in MLX) may appear strange. Instead of the usual decimal numbers you're accustomed to, these numbers are in *hexadecimal*—a base 16 numbering system commonly used by ML programmers. Hexadecimal—hex for short—includes the numerals 0–9 and the letters A–F. But don't worry—even if you know nothing about ML or hex, you should have no trouble using MLX.

After you enter the starting and ending addresses, you'll be offered the option of clearing the workspace. Choose this option if you're starting to enter a new listing. If you're continuing a listing that's partially typed from a previous session, don't choose this option.

A functions menu will appear. The first option in the menu is ENTER DATA. If you're just starting to type in a program, pick this. Press the E key, and type the first number in the first line of the program listing. If you've already typed in part of a program, type the line number where you left off typing at the end of the previous session (be sure to load the partially completed program before you resume entry). In any case, make sure the address you enter corresponds to the address of a line in the listing you are entering. Otherwise, you'll be unable to enter the data correctly. If you pressed E by mistake, you can return to the command menu by pressing RETURN alone when asked for the address. (You can get back to the menu from most options by pressing RETURN with no other input.)

Entering A Listing

Once you're in Enter mode, MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight data bytes and a check-

sum. Although an MLX-format listing appears similar to the "hex dump" listings from a machine language monitor program, the extra checksum number on the end allows MLX to check your typing.

When you enter a line, MLX recalculates the checksum from the eight bytes and the address and compares this value to the number from the ninth column. If the values match, you'll hear a bell tone, the data will be added to the workspace area, and the prompt for the next line of data will appear. But if MLX detects a typing error, you'll hear a low buzz and see an error message. The line will then be redisplayed for editing.

Invalid Characters Banned

Only a few keys are active while you're entering data, so you may have to unlearn some habits. You *do not* type spaces between the columns; MLX automatically inserts these for you. You *do not* press RETURN after typing the last number in a line; MLX automatically enters and checks the line after you type the last digit.

Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), you'll hear a warning buzz. To simplify typing, a numeric keypad is now incorporated in the listing. The keypad is active only while entering data. Addresses must be entered with the normal letter and number keys. The figure below shows the keypad configuration:

7	8	9	0
4 U	5 I	6 O	F P
1 J	2 K	3 L	E :
A M	B ,	C •	D /
	0 Space		

MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, MLX will catch your mistake. There is one error that can slip past MLX: Because of the checksum formula used, MLX won't notice if you accidentally type FF in place of 00, and vice

versa. And there's a very slim chance that you could garble a line and still end up with a combination of characters that adds up to the proper checksum. However, these mistakes should not occur if you take reasonable care while entering data.

Editing Features

To correct typing mistakes before finishing a line, use the INST/DEL key to delete the character to the left of the cursor. (The cursor-left key also deletes.) If you mess up a line really badly, press CLR/HOME to start the line over. The RETURN key is also active, but only before any data is typed on a line. Pressing RETURN at this point returns you to the command menu. After you type a character of data, MLX disables RETURN until the cursor returns to the start of a line. Remember, you can press CLR/HOME to quickly get to a line number prompt.

More editing features are available when correcting lines in which MLX has detected an error. To make corrections in a line that MLX has redisplayed for editing, compare the line on the screen with the one printed in the listing, then move the cursor to the mistake and type the correct key. The cursor left and right keys provide the normal cursor controls. (The INST/DEL key now works as an alternative cursor-left key.) You cannot move left beyond the first character in the line. If you try to move beyond the rightmost character, you'll reenter the line. During editing, RETURN is active; pressing it tells MLX to recheck the line. You can press the CLR/HOME key to clear the entire line if you want to start from scratch, or if you want to get to a line number prompt to use RETURN to get back to the menu.

Display Data

The second menu choice, DISPLAY DATA, examines memory and shows the contents in the same format as the program listing (including the checksum). When you press D, MLX asks you for a starting address. Be sure that the starting address you give corresponds to a line number in the listing. Otherwise, the checksum display will be meaningless. MLX displays program lines until it reaches the end of the program, at which point the menu is redisplayed. You can pause the display by pressing the space bar. (MLX finishes printing the current line before halting.) Press space again to

restart the display. To break out of the display and get back to the menu before the ending address is reached, press RETURN.

Other Menu Options

Two more menu selections let you save programs and load them back into the computer. These are SAVE FILE and LOAD FILE; their operation is quite straightforward. When you press S or L, MLX asks you for the filename. You'll then be asked to press either D or T to select disk or tape.

You'll notice the disk drive starting and stopping several times during a load or save. Don't panic; this is normal behavior. MLX opens and reads from or writes to the file instead of using the usual LOAD and SAVE commands. Disk users should also note that the drive prefix 0: is automatically added to the filename (line 750), so this should not be included when entering the name. This also precludes the use of @ for Save-with-Replace, so remember to give each version you save a different name.

Remember that MLX saves the entire workspace area from the starting address to the ending address, so the save or load may take longer than you might expect if you've entered only a small amount of data from a long listing. When saving a partially completed listing, make sure to note the address where you stopped typing so you'll know where to resume entry when you reload.

MLX reports the standard disk or tape error messages if any problems are detected during the save or load. (Tape users should bear in mind that Commodore computers are never able to detect errors during a save to tape.) MLX also has three special load error messages: INCORRECT STARTING ADDRESS, which means the file you're trying to load does not have the starting address you specified when you ran MLX; LOAD ENDED AT *address*, which means the file you're trying to load ends before the ending address you specified when you started MLX; and TRUNCATED AT ENDING ADDRESS, which means the file you're trying to load extends beyond the ending address you specified when you started MLX. If you see one of these messages and feel certain that you've loaded the right file, exit and rerun MLX, being careful to enter the correct starting and ending addresses.

The QUIT menu option has the obvious effect—it stops MLX and enters BASIC. The RUN/STOP key is disabled, so the Q option lets you exit the program without turning off the computer. (Of course, RUN/STOP–RESTORE also gets you out.) You'll be asked for verification; press Y to exit to BASIC, or any other key to return to the menu. After quitting, you

can type RUN again and reenter MLX without losing your data, as long as you don't use the clear workspace option.

The Finished Product

When you've finished typing all the data for an ML program and saved your work, you're ready to see the results. The instructions for loading and using the finished product vary from program to program. Some ML programs are designed to be loaded and run like BASIC programs, so all you need to type is LOAD "filename",8 for disk or LOAD "filename" for tape, and then RUN. Such programs will usually have a starting address of 0801 for the 64. Other programs must be reloaded to specific addresses with a command such as LOAD "filename",8,1 for disk or LOAD "filename",1,1 for tape, then started with a SYS to a particular memory address. On the Commodore 64, the most common starting address for such programs is 49152, which corresponds to MLX address C000. In either case, you should always refer to the article which accompanies the ML listing for information on loading and running the program.

An Ounce Of Prevention

By the time you finish typing in the data for a long ML program, you may have several hours invested in the project. Don't take chances—use our "Automatic Proofreader" to type the new MLX, and then test your copy *thoroughly* before first using it to enter any significant amount of data. Make sure all the menu options work as they should. Enter fragments of the program starting at several different addresses, then use the Display option to verify that the data has been entered correctly. And be sure to test the Save and Load options several times to ensure that you can recall your work from disk or tape. Don't let a simple typing error in the new MLX cost you several nights of hard work.

MLX For Commodore 64

```
SS 10 REM VERSION 1.1: LINES 8
       30,950 MODIFIED, LINES 4
       85-487 ADDED
EK 100 POKE 56,50:CLR:DIM IN$,
       I,J,A,B,A$,B$,A(7),N$
DM 110 C4=48:C6=16:C7=7:Z2=2:Z
       4=254:Z5=255:Z6=256:Z7=
       127
CJ 120 FA=PEEK(45)+Z6*PEEK(46)
       :BS=PEEK(55)+Z6*PEEK(56
       ):H$="0123456789ABCDEF"
SB 130 R$=CHR$(13):L$="{LEFT}"
       :S$=" ":D$=CHR$(20):Z$=
       CHR$(0):T$="{13 RIGHT}"
CQ 140 SD=54272:FOR I=SD TO SD
       +23:POKE I,0:NEXT:POKE
       {SPACE}SD+24,15:POKE 78
       8,52
FC 150 PRINT"{CLR}"CHR$(142)CH
       R$(8):POKE 53280,15:POK
```

```
E 53281,15
EJ 160 PRINT T$" {RED}{RVS}
       {2 SPACES}{8 @}
       {2 SPACES}{SPC(28)"
       {2 SPACES}{OFF}{BLU} ML
       X II {RED}{RVS}
       {2 SPACES}"SPC(28)"
       {12 SPACES}{BLU}"
FR 170 PRINT"{3 DOWN}
       {3 SPACES}COMPUTE!'S MA
       CHINE LANGUAGE EDITOR
       {3 DOWN}"
JB 180 PRINT"{BLK}STARTING ADD
       RESS{4}";:GOSUB300:SA=A
       D:GOSUB1040:IF F THEN18
       0
GF 190 PRINT"{BLK}{2 SPACES}EN
       DING ADDRESS{4}";:GOSUB
       300:EA=AD:GOSUB1030:IF
       {SPACE}F THEN190
KR 200 INPUT"{3 DOWN}{BLK}CLEA
       R WORKSPACE [Y/N]{4}";A
       $:IF LEFT$(A$,1)<>"Y"TH
       EN220
PG 210 PRINT"{2 DOWN}{BLU}WORK
       ING...";:FORI=BS TO BS+
       EA-SA+7:POKE I,0:NEXT:P
       RINT"DONE"
DR 220 PRINTTAB(10)"{2 DOWN}
       {BLK}{RVS} MLX COMMAND
       {SPACE}MENU {DOWN}{4}":
       PRINT T$"{RVS}E{OFF}NTE
       R DATA"
BD 230 PRINT T$"{RVS}D{OFF}ISP
       LAY DATA":PRINT T$"
       {RVS}L{OFF}OAD FILE"
JS 240 PRINT T$"{RVS}S{OFF}AVE
       FILE":PRINT T$"{RVS}Q
       {OFF}UIT{2 DOWN}{BLK}"
JH 250 GET A$:IF A$=N$ THEN250
HK 260 A=0:FOR I=1 TO 5:IF A$=
       MID$("EDLSQ",I,1)THEN A
       =I:I=5
FD 270 NEXT:ON A GOTO420,610,6
       90,700,280:GOSUB1060:GO
       TO250
EJ 280 PRINT"{RVS} QUIT ":INPU
       T"{DOWN}{4}ARE YOU SURE
       [Y/N]";A$:IF LEFT$(A$,
       1)<>"Y"THEN220
EM 290 POKE SD+24,0:END
JX 300 IN$=N$:AD=0:INPUTIN$:IF
       LEN(IN$)<>4THENRETURN
KF 310 B$=IN$:GOSUB320:AD=A:B$
       =MID$(IN$,3):GOSUB320:A
       D=AD*256+A:RETURN
PP 320 A=0:FOR J=1 TO 2:A$=MID
       $(B$,J,1):B=ASC(A$)-C4+
       (A$>"@")*C7:A=A*C6+B
JA 330 IF B<0 OR B>15 THEN AD=
       0:A=-1:J=2
GX 340 NEXT:RETURN
CH 350 B=INT(A/C6):PRINT MID$(
       H$,B+1,1);:B=A-B*C6:PRI
       NT MID$(H$,B+1,1);:RETU
       RN
RR 360 A=INT(AD/Z6):GOSUB350:A
       =AD-A*Z6:GOSUB350:PRINT
       ":";
BE 370 CK=INT(AD/Z6):CK=AD-Z4*
       CK+Z5*(CK>Z7):GOTO390
PX 380 CK=CK*Z2+Z5*(CK>Z7)+A
JC 390 CK=CK+Z5*(CK>Z5):RETURN
QS 400 PRINT"{DOWN}STARTING AT
       {4}";:GOSUB300:IF IN$<>
       N$ THEN GOSUB1030:IF F
       {SPACE}THEN400
EX 410 RETURN
HD 420 PRINT"{RVS} ENTER DATA
       {SPACE}":GOSUB400:IF IN
       $=N$ THEN220
JK 430 OPEN3,3:PRINT
SK 440 POKE198,0:GOSUB360:IF F
```

```
                THEN PRINT IN$:PRINT"
                {UP}{5 RIGHT}";
GC 450  FOR I=0 TO 24 STEP 3:B$
        =S$:FOR J=1 TO 2:IF F T
        HEN B$=MID$(IN$,I+J,1)
HA 460  PRINT"{RVS}"B$L$;:IF I<
        24THEN PRINT"{OFF}";
HD 470  GET A$:IF A$=N$ THEN470
FK 480  IF(A$>"/"ANDA$<":")OR(A
        $>"@"ANDA$<"G")THEN540
GS 485  A=-(A$="M")-2*(A$=",")-
        3*(A$=".")-4*(A$="/")-5
        *(A$="J")-6*(A$="K")
FX 486  A=A-7*(A$="L")-8*(A$=":
        ")-9*(A$="U")-10*(A$="I
        ")-11*(A$="O")-12*(A$="
        P")
CM 487  A=A-13*(A$=S$):IF A THE
        N A$=MID$("ABCD123E456F
        0",A,1):GOTO540
MP 490  IF A$=R$ AND((I=0)AND(J
        =1)OR F)THEN PRINT B$;:
        J=2:NEXT:I=24:GOTO550
KC 500  IF A$="{HOME}" THEN PRI
        NT B$:J=2:NEXT:I=24:NEX
        T:GOTO440
MX 510  IF(A$="{RIGHT}")ANDF TH
        ENPRINT B$L$;:GOTO540
GK 520  IF A$<>L$ AND A$<>D$ OR
        ((I=0)AND(J=1))THEN GOS
        UB1060:GOTO470
HG 530  A$=L$+S$+L$:PRINT B$L$;
        :J=2-J:IF J THEN PRINT
        {SPACE}L$;:I=I-3
QS 540  PRINT A$;:NEXT J:PRINT
        {SPACE}S$;
PM 550  NEXT I:PRINT:PRINT"{UP}
        {5 RIGHT}";:INPUT#3,IN$
        :IF IN$=N$ THEN CLOSE3:
        GOTO220
QC 560  FOR I=1 TO 25 STEP3:B$=
        MID$(IN$,I):GOSUB320:IF
        I<25 THEN GOSUB380:A(I
        /3)=A
PK 570  NEXT:IF A<>CK THEN GOSU
        B1060:PRINT"{BLK}{RVS}
        {SPACE}ERROR: REENTER L
        INE {4}":F=1:GOTO440
HJ 580  GOSUB1080:B=BS+AD-SA:FO
        R I=0 TO 7:POKE B+I,A(I
        ):NEXT
QQ 590  AD=AD+8:IF AD>EA THEN C
        LOSE3:PRINT"{DOWN}{BLU}
        ** END OF ENTRY **{BLK}
        {2 DOWN}":GOTO700
GQ 600  F=0:GOTO440
QA 610  PRINT"{CLR}{DOWN}{RVS}
        {SPACE}DISPLAY DATA ":G
        OSUB400:IF IN$=N$ THEN2
        20
RJ 620  PRINT"{DOWN}{BLU}PRESS:
        {RVS}SPACE{OFF} TO PAU
        SE, {RVS}RETURN{OFF} TO
        BREAK{4}{DOWN}"
KS 630  GOSUB360:B=BS+AD-SA:FOR
        I=BTO B+7:A=PEEK(I):GOS
        UB350:GOSUB380:PRINT S$
        ;
CC 640  NEXT:PRINT"{RVS}";:A=CK
        :GOSUB350:PRINT
KH 650  F=1:AD=AD+8:IF AD>EA TH
        ENPRINT"{DOWN}{BLU}** E
        ND OF DATA **":GOTO220
KC 660  GFT A$:IF A$=R$ THEN GO
        SUB1080:GOTO220
EQ 670  IF A$=S$ THEN F=F+1:GOS
        UB1080
AD 680  ONFGOTO630,660,630
CM 690  PRINT"{DOWN}{RVS} LOAD
        {SPACE}DATA ":OP=1:GOTO
        710
PC 700  PRINT"{DOWN}{RVS} SAVE
```

```
        {SPACE}FILE ":OP=0
RX 710  IN$=N$:INPUT"{DOWN}FILE
        NAME{4}";IN$:IF IN$=N$
        {SPACE}THEN220
PR 720  F=0:PRINT"{DOWN}{BLK}
        {RVS}T{OFF}APE OR {RVS}
        D{OFF}ISK: {4}";
FP 730  GET A$:IF A$="T"THEN PR
        INT"T{DOWN}":GOTO880
HQ 740  IF A$<>"D"THEN730
HH 750  PRINT"D{DOWN}":OPEN15,8
        ,15,"I0:":B=EA-SA:IN$="
        0:"+IN$:IF OP THEN810
SQ 760  OPEN 1,8,8,IN$+",P,W":G
        OSUB860:IF A THEN220
FJ 770  AH=INT(SA/256):AL=SA-(A
        H*256):PRINT#1,CHR$(AL)
        ;CHR$(AH);
PE 780  FOR I=0 TO B:PRINT#1,CH
        R$(PEEK(BS+I));:IF ST T
        HEN800
FC 790  NEXT:CLOSE1:CLOSE15:GOT
        O940
GS 800  GOSUB1060:PRINT"{DOWN}
        {BLK}ERROR DURING SAVE:
        {4}":GOSUB860:GOTO220
MA 810  OPEN 1,8,8,IN$+",P,R":G
        OSUB860:IF A THEN220
GE 820  GET#1,A$,B$:AD=ASC(A$+Z
        $)+256*ASC(B$+Z$):IF AD
        <>SA THEN F=1:GOTO850
RX 830  FOR I=0 TO B:GET#1,A$:P
        OKE BS+I,ASC(A$+Z$):IF(
        I<>B)AND ST THEN F=2:AD
        =I:I=B
FA 840  NEXT:IF ST<>64 THEN F=3
FQ 850  CLOSE1:CLOSE15:ON ABS(F
        >0)+1 GOTO960,970
SA 860  INPUT#15,A,A$:IF A THEN
        CLOSE1:CLOSE15:GOSUB10
        60:PRINT"{RVS}ERROR: "A
        $
GQ 870  RETURN
EJ 880  POKE183,PEEK(FA+2):POKE
        187,PEEK(FA+3):POKE188,
        PEEK(FA+4):IFOP=0THEN92
        0
HJ 890  SYS 63466:IF(PEEK(783)A
        ND1)THEN GOSUB1060:PRIN
        T"{DOWN}{RVS} FILE NOT
        {SPACE}FOUND ":GOTO690
CS 900  AD=PEEK(829)+256*PEEK(8
        30):IF AD<>SA THEN F=1:
        GOTO970
SC 910  A=PEEK(831)+256*PEEK(83
        2)-1:F=F-2*(A<EA)-3*(A>
        EA):AD=A-AD:GOTO930
KM 920  A=SA:B=EA+1:GOSUB1010:P
        OKE780,3:SYS 63338
JF 930  A=BS:B=BS+(EA-SA)+1:GOS
        UB1010:ON OP GOTO950:SY
        S 63591
AE 940  GOSUB1080:PRINT"{BLU}**
        SAVE COMPLETED **":GOT
        O220
XP 950  POKE147,0:SYS 63562:IF
        {SPACE}ST>0 THEN970
FR 960  GOSUB1080:PRINT"{BLU}**
        LOAD COMPLETED **":GOT
        O220
DP 970  GOSUB1060:PRINT"{BLK}
        {RVS}ERROR DURING LOAD:
        {DOWN}{4}":ON F GOSUB98
        0,990,1000:GOTO220
PP 980  PRINT"INCORRECT STARTIN
        G ADDRESS (";:GOSUB360:
        PRINT")":RETURN
GR 990  PRINT"LOAD ENDED AT ";:
        AD=SA+AD:GOSUB360:PRINT
        D$:RETURN
FD 1000 PRINT"TRUNCATED AT END
        ING ADDRESS":RETURN
```

```
RX 1010 AH=INT(A/256):AL=A-(AH
        *256):POKE193,AL:POKE1
        94,AH
FF 1020 AH=INT(B/256):AL=B-(AH
        *256):POKE174,AL:POKE1
        75,AH:RETURN
FX 1030 IF AD<SA OR AD>EA THEN
        1050
HA 1040 IF(AD>511 AND AD<40960
        )OR(AD>49151 AND AD<53
        248)THEN GOSUB1080:F=0
        :RETURN
HC 1050 GOSUB1060:PRINT"{RVS}
        {SPACE}INVALID ADDRESS
        {DOWN}{BLK}":F=1:RETU
        RN
AR 1060 POKE SD+5,31:POKE SD+6
        ,208:POKE SD,240:POKE
        {SPACE}SD+1,4:POKE SD+
        4,33
DX 1070 FOR S=1 TO 100:NEXT:GO
        TO1090
PF 1080 POKE SD+5,8:POKE SD+6,
        240:POKE SD,0:POKE SD+
        1,90:POKE SD+4,17
AC 1090 FOR S=1 TO 100:NEXT:PO
        KE SD+4,0:POKE SD,0:PO
        KE SD+1,0:RETURN
```

©

MLX Machine Language Entry Program For Apple

Tim Victor, Editorial Programmer

To make it easier to enter machine language programs into your computer without typos, COMPUTE! is introducing its "MLX" entry program for the Apple II series. It's our best MLX yet. It runs on the II, II+, IIe, and IIc, and with either DOS 3.3 or ProDOS.

A machine language (ML) program is usually listed as a long series of numbers. It's hard to keep your place and even harder to avoid making mistakes as you type in the listing, since an incorrect line looks almost identical to a correct one. To make error-free entry easier, COMPUTE! generally lists ML programs for Commodore and Atari computers in a format designed to be typed in with a utility called "MLX." The MLX program uses a checksum system to catch typing errors almost as soon as they happen.

Apple MLX checks your typing on a line-by-line basis. It won't let you enter invalid characters or let you continue if there's a mistake in a line. It won't even let you enter a line or digit out of sequence. Best of all, you don't have to know anything about machine language to enter ML programs with MLX. Apple MLX makes typing ML programs almost foolproof.

Using Apple MLX
Type in and save some copies of Apple MLX on disk (you'll want to use MLX to enter future ML programs in COMPUTE!). It doesn't matter whether you type it in on a disk formatted for DOS 3.3 or ProDOS. Programs entered with Apple MLX, however, must be saved to a disk formatted with the same operating system as Apple MLX itself.

If you have an Apple IIe or IIc, make sure that the key marked CAPS LOCK is in the down position. Type RUN. You'll be asked for the starting and ending addresses of the ML program. These values vary for each program, so they're given at the beginning of the ML program listing and in the program's accompanying article. Find them and type them in.

The next thing you'll see is a menu asking you to select a function. The first is (E)NTER DATA. If you're just starting to type in a program, pick this. Press the E key, and the program asks for the address where you want to begin entering data. Type the first number in the

first line of the program listing if you're just starting, or the line number where you left off if you've already typed in part of a program. Hit the RETURN key and begin entering the data.

Once you're in Enter mode, Apple MLX prints the address for each program line for you. You then type in all nine numbers on that line, beginning with the first two-digit number after the colon (:). Each line represents eight bytes and a checksum. When you enter a line and hit RETURN, Apple MLX recalculates the checksum from the eight bytes and the address. If you enter more or less than nine numbers, or the checksum doesn't exactly match, Apple MLX erases the line you just entered and prompts you again for the same line.

Invalid Characters Banned
Apple MLX is fairly flexible about how you type in the numbers. You can put extra spaces between numbers or leave the spaces out entirely, compressing a line into 18 keypresses. Be careful not to put a space between two digits in the middle of a number. Apple MLX will read two single-digit numbers instead of one two-digit number (F 6 means F and 6, not F6).

You can't enter an invalid character with Apple MLX. Only the numerals 0–9 and the letters A–F can be typed in. If you press any other key (with some exceptions noted below), nothing happens. This safeguards against entering extraneous characters. Even better, Apple MLX checks for transposed characters. If you're supposed to type in A0 and instead enter 0A, Apple MLX will catch your mistake.

Apple MLX also checks to make sure you're typing in the right line. The address (the number to the left of the colon) is part of the checksum recalculation. If you accidentally skip a line and try to enter incorrect values, Apple MLX won't let you continue. Just make sure you enter the correct starting address; if you don't, you won't be able to enter any of the following lines. Apple MLX will stop you.

Editing Features
Apple MLX also includes some editing features. The left- and right-arrow keys allow you to back up and go forward on the line that you are entering, so you can retype data. Pressing the CON-

TROL (CTRL) and D keys at the same time (*delete*) removes the character under the cursor, shortening the line by one character. Pressing CTRL-I (*insert*) puts a space under the cursor and shifts the rest of the line to the right, making the line one character longer. If the cursor is at the right end of the line, neither CTRL-D nor CTRL-I has any effect.

When you've entered the entire listing (up to the ending address that you specified earlier), Apple MLX automatically leaves Enter mode and redisplays the functions menu. If you want to leave Enter mode before then, press the RETURN key when Apple MLX prompts you with a new line address. (For instance, you may want to leave Enter mode to enter a program listing in more than one sitting; see below.)

Display Data
The second menu choice, (D)ISPLAY DATA, examines memory and shows the contents in the same format as the program listing. You can use it to check your work or to see how far you've gotten. When you press D, Apple MLX asks you for a starting address. Type in the address of the first line you want to see and hit RETURN. Apple MLX displays program lines until you press any key or until it reaches the end of the program.

Save And Load
Two more menu selections let you save programs on disk and load them back into the computer. These are (S)AVE FILE and (L)OAD FILE. When you press S or L, Apple MLX asks you for the filename. The first time you save an ML program, the name you assign will be the program's filename on the disk. If you press L and specify a filename that doesn't exist on the disk, you'll see a disk error message.

If you're not sure why a disk error has occurred, check the drive. Make sure there's a formatted disk in the drive and that it was formatted by the same operating system you're using for Apple MLX (ProDOS or DOS 3.3). If you're trying to save a file and see an error message, the disk might be full. Either save the file on another disk or quit Apple MLX (by pressing the Q key), delete an old file or two, then run Apple MLX again. Your typing should still be safe in memory.

Apple MLX: Machine Language Entry Program

For instructions on entering this program, please refer to "COMPUTE!'s Guide to Typing In Programs" elsewhere in this issue.

```
35 100 N = 9: HOME : NORMAL : PR
       INT CHR$ (17);"APPLE MLX
       V1.1": POKE 34,2: ONERR G
       OTO 610
CC 110 VTAB 1: HTAB 20: PRINT "S
       TART ADDRESS";: GOSUB 530
       : IF A = 0 THEN PRINT CHR
       $ (7): GOTO 110
8C 120 S = A
E3 130 VTAB 2: HTAB 20: PRINT "E
       ND ADDRESS  ";: GOSUB 530
       : IF S > = A OR A = 0 THE
       N PRINT CHR$ (7): GOTO 13
       0
28 140 E = A
B5 150 PRINT : PRINT "CHOOSE:(E) 
       NTER DATA";: HTAB 22: PRI
       NT "(D)ISPLAY DATA": HTAB
       8: PRINT "(L)OAD FILE  (
       S)AVE FILE  (Q)UIT": PRIN
       T
AE 160 GET A$: FOR I = 1 TO 5: I
       F A$ < > MID$ ("EDLSQ",I,
       1) THEN NEXT : GOTO 160
93 170 ON I GOTO 270,220,180,200
       : POKE 34,0: END
AF 180 INPUT "FILENAME: ";A$: IF
       A$ < > "" THEN PRINT CHR
       $ (4);"BLOAD";A$;",A";S
A1 190 GOTO 150
6D 200 INPUT "FILENAME: ";A$: IF
       A$ < > "" THEN PRINT CHR
       $ (4);"BSAVE";A$;",A";S;"
       ,L";E - S
92 210 GOTO 150
C2 220 GOSUB 590: IF B = 0 THEN
       150
9E 230 FOR B = B TO E STEP 8:L =
       4:A = B: GOSUB 580: PRIN
       T A$;": ";:L = 2
B5 240 FOR F = 0 TO 7:V(F + 1) =
       PEEK (B + F): NEXT : GOS
       UB 560:V(9) = C
F2 250 FOR F = 1 TO N:A = V(F):
       GOSUB 580: PRINT A$" ";:
       NEXT : PRINT : IF PEEK (4
       9152) < 128 THEN NEXT
94 260 POKE 49168,0: GOTO 150
CC 270 GOSUB 590: IF B = 0 THEN
       150
48 280 FOR B = B TO E STEP 8
A6 290 HTAB 1:A = B:L = 4: GOSUB
       580: PRINT A$;": ";: CAL
       L 64668:A$ = "":P = 0: GO
       SUB 330: IF L = 0 THEN 15
       0
F9 300 GOSUB 470: IF F < > N THE
       N PRINT CHR$ (7);: GOTO 2
       90
27 310 IF N = 9 THEN GOSUB 560:
       IF C < > V(9) THEN PRINT
       CHR$ (7);: GOTO 290
72 320 FOR F = 1 TO 8: POKE B +
       F - 1,V(F): NEXT : PRINT
       : NEXT : GOTO 150
BE 330 IF LEN (A$) = 33 THEN A$
       = O$:P = O: PRINT CHR$ (7
       );
22 340 L = LEN (A$):O$ = A$:O =
       P:L$ = "": IF P > 0 THEN
       L$ = LEFT$ (A$,P)
E0 350 R$ = "": IF P < L - 1 THE
       N R$ = RIGHT$ (A$,L - P -
       1)
55 360 HTAB 7: PRINT L$;: FLASH
       : IF P < L THEN PRINT MID
       $ (A$,P + 1,1);: NORMAL :
       PRINT R$;
7B 370 PRINT " ";: NORMAL
E6 380 K = PEEK (49152): IF K <
       128 THEN 380
C1 390 POKE 49152,0:K = K - 128
5B 400 IF K = 13 THEN HTAB 7: PR
       INT A$;" ";: RETURN
A7 410 IF K = 32 OR K > 47 AND K
       < 58 OR K > 64 AND K < 7
       1 THEN A$ = L$ + CHR$ (K)
       + R$:P = P + 1: GOTO 330
C7 420 I = FRE (0): IF K = 4 THE
       N A$ = L$ + R$
5F 430 IF K = 9 THEN A$ = L$ + "
       " + MID$ (A$,P + 1,1) +
       R$
0A 440 IF K = 8 THEN P = P - (P
       > 0)
93 450 IF K = 21 THEN P = P + (P
       < L)
9D 460 GOTO 330
37 470 F = 1:D = 0: FOR P = 1 TO
       LEN (A$):C$ = MID$ (A$,P
       ,1): IF F > N AND C$ < >
       " " THEN RETURN
BB 480 IF C$ < > " " THEN GOSUB
       520:V(F) = J + 16 * (D =
       1) * V(F):D = D + 1
5F 490 IF D > 0 AND C$ = " " OR
       D = 2 THEN D = 0:F = F +
       1
0B 500 NEXT : IF D = 0 THEN F =
       F - 1
1E 510 RETURN
B5 520 J = ASC (C$):J = J - 48 -
       7 * (J > 64): RETURN
AB 530 A = 0: INPUT A$:A$ = LEFT
       $ (A$,4): IF LEN (A$) = 0
       THEN RETURN
6F 540 FOR P = 1 TO LEN (A$):C$
       = MID$ (A$,P,1): IF C$ <
       "0" OR C$ > "9" AND C$ <
       "A" OR C$ > "Z" THEN A =
       0: RETURN
2D 550 GOSUB 520:A = A * 16 + J:
       NEXT : RETURN
2B 560 C = INT (B / 256):C = B -
       254 * C - 255 * (C > 127
       ):C = C - 255 * (C > 255)
2B 570 FOR F = 1 TO 8:C = C * 2
       - 255 * (C > 127) + V(F):
       C = C - 255 * (C > 255):
       NEXT : RETURN
DA 580 I = FRE (0):A$ = "": FOR
       I = 1 TO L:T = INT (A / 1
       6):A$ = MID$ ("0123456789
       ABCDEF",A - 16 * T + 1,1)
       + A$:A = T: NEXT : RETUR
       N
1F 590 PRINT "FROM ADDRESS ";: G
       OSUB 530: IF S > A OR E <
       A OR A = 0 THEN B = 0: R
       ETURN
0D 600 B = S + 8 * INT ((A - S)
       / 8): RETURN
B6 610 PRINT "DISK ERROR": GOTO
       150                      ©
```

www.commodore.ca

Classified

Advertisers Index

www.commodore.ca

COMPUTE!'s
FREE Reader Information Service

Use these cards to request FREE information about the products advertised in this issue. Clearly print or type your full name and address. Only one card should be used per person. Circle the numbers that correspond to the key number appearing in the advertisers index.

Send in the card and the advertisers will receive your inquiry. Although every effort is made to insure that only advertisers wishing to provide product information have reader service numbers, COMPUTE! cannot be responsible if advertisers do not provide literature to readers.

Please use these cards *only* for subscribing or for requesting product information. Editorial and customer service inquiries should be addressed to: COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Check the expiration date on the card to insure proper handling.

Use these cards and this address only for COMPUTE!'s Reader Information Service. Do not send with payment in any form.

COMPUTE!

101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151
152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168
169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185
186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236
237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253

Circle 101 for a one year new U.S. subscription to COMPUTE!: you will be billed for $24.

Please let us know. Do you own:

	plan to buy:
□ Apple _____ 270	□ 271
□ Atari _____ 272	□ 273
□ Commodore _____ 274	□ 275
□ IBM _____ 276	□ 277
□ TI-99/4A _____ 278	□ 279
□ Other _____ 280 (specify model)	□ 281

Please print or type name and address.
Limit one card per person.

Name

Address

City

State/Province _____ Zip

Country

Phone

Please Include ZIP Code

Expiration Date 4/30/88   GA0388

SUBSCRIBE TO COMPUTE!

For Fastest Service, Call Our **Toll-Free** US Order Line **800-727-6937**

□ $24.00 One Year US Subscription
□ $45.00 Two Year US Subscription

Name

Address

City _____ State _____ Zip

□ Payment Enclosed   □ Bill me
Charge my: □ VISA   □ MasterCard   □ American Express
Account No. _____ Expires ___ / ___

Your subscription will begin with the next available issue. Please allow 4-6 weeks for delivery of first issue. Subscription price subject to change at any time. Outside the U.S.A. please add $6 per year for postage.

J3C33F

Place
Stamp
Here

COMPUTE! Reader Service
P.O. Box 2141
Radnor, PA 19089

||||| |||

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS   PERMIT NO. 7478   DES MOINES, IA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPUTE!

P.O. Box 10955
Des Moines, IA 50347-0955

www.commodore.ca

KONAMI HITS ARE HEADING HOME.

With our new software, your favorite arcade games come alive on your home computer.

If you own an IBM, Amiga or Commodore computer—Watch out! Because 4 of Konami's most awesome adventures are about to invade.

You'll confront the missions you've dreamed of in Rush 'N Attack, Contra, Jackal and Boot Camp. And you'll find yourself face to face against unequaled challenges, with incredible graphics, awesome music, and sound effects that'll blow you away.

So check out Konami's newest action wherever great software is sold. And be prepared for war.

KONAMI®