Softkeys For:

Adventure Writer
Mychess II
Raster Blaster
Cranston Manor
Ghostbusters
E-Z Learner
American Challenge
Crime Wave

Core:
**The Games of 1986:
a review**

Feature:
**Taking the Wiz out of
Wizardry**

(Page 16)

Coping With COMPUTIST

Welcome to COMPUTIST, a publication devoted to the serious user of Apple][and Apple][compatible computers. Our magazine contains information you are not likely to find in any of the other major journals dedicated to the Apple market.

New readers are advised to read this page carefully to avoid frustration when attempting to follow a softkey or when entering the programs printed in this issue.

■ **What Is A Softkey Anyway?** Softkey is a term which we coined to describe a procedure that removes, or at least circumvents, any copy-protection on a particular disk. Once a softkey procedure has been performed, the resulting disk can usually be copied by the use of Apple's COPYA program (on the DOS 3.3 System Master Disk).

■ **Commands And Controls:** In any article appearing in COMPUTIST, commands which a reader is required to perform are set apart by being in boldface and indented:

PR#6

The RETURN key must be pressed at the end of every such command unless otherwise specified.

Control characters are specially boxed:

6⊡P

Press 6 . Next, place one finger on CTRL and press P . Remember to enter this command line by pressing RETURN .

■ **Requirements:** COMPUTIST programs and softkeys require one of the Apple][series of computers and a disk drive with DOS 3.3. These and other special needs are listed at the beginning of the article under "Requirements".

■ **Software Recommendations:**

1) Applesoft Program Editor such as Global Program Line Editor (GPLE).

2) Sector Editor such as DiskEdit (from the Book of Softkeys vol I) or ZAP from Bag of Tricks.

3) Disk Search Utility such as The Inspector, The CIA or The CORE Disk Searcher (from the Book of Softkeys vol II).

4) Assembler such as the S-C Assembler from S-C software or Merlin/Big Mac.

5) Bit Copy Program such as Copy][Plus, Locksmith or The Essential Data Duplicator

6) Text Editor (that produces normal sequential text files) such as Applewriter II, Magic Window II or Screenwriter II.

COPYA, FID and MUFFIN from the DOS 3.3 System Master Disk are also useful.

■ **Super IOB:** This powerful deprotection utility (COMPUTIST 32) and its various controllers are used in many softkeys. This utility is now available on each Super IOB Collection disk.

■ **RESET Into The Monitor:** Softkeys occasionally require the user to stop the execution of a copy-protected program and directly enter the Apple's system monitor. Check the following list to see what hardware you will need to obtain this ability.

Apple][Plus - Apple //e - Apple compatibles: 1) Place an Integer BASIC ROM card in one of the Apple slots. 2) Use a non-maskable interrupt (NMI) card such as Replay or Wildcard.

Apple][Plus - Apple compatibles: 1) Install an F8 ROM with a modified RESET vector on the computer's motherboard as detailed in the "Modified ROM's" article (COMPUTIST 6 or Book Of Softkeys III) or the "Dual ROM's" article (COMPUTIST 19).

Apple //e - Apple //c: Install a modified CD ROM on the computer's motherboard. Cutting Edge Ent. (Box 43234 Ren Cen Station-HC; Detroit, MI 48243) sells a hardware device that will give you this important ability but it will void an Apple //c warranty.

■ **Recommended Literature:** The Apple][Reference Manual and DOS 3.3 manual are musts for any serious Apple user. Other helpful books include: *Beneath Apple DOS*, Don Worth and Pieter Lechner, Quality Software; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley; and *What's Where In The Apple*, William Lubert, Micro Ink.

■ **Keying In Applesoft Programs:** BASIC programs are printed in COMPUTIST in a format that is designed to minimize errors for readers who key in these programs. If you type:

10HOME:REMCLEAR⎵SCREEN
The LIST will look like:

10 HOME : REM CLEAR SCREEN
because Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces usually don't pose a problem except in line numbers which contain REM or DATA commands. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be keyed in appear in COMPUTIST as delta characters (ᐃ). All other spaces are there for easier reading. NOTE: If you want your checksums (See "Computing Checksums" section) to match up, you must only key in (ᐃ) spaces after DATA statements.

■ **Keying In Hexdumps:** Machine language programs are printed in COMPUTIST as both source code and hexdumps. Hexdumps are the shortest and easiest format to type in. You must first enter the monitor:

CALL -151

Key in the hexdump exactly as it appears in the magazine, ignoring the four-digit checksum at the end of each line (a "$" and four digits). A beep means you have typed something that the monitor didn't understand and must, therefore, retype that line.

When finished, return to BASIC with:

E003G

BSAVE the program with the correct filename, address and length parameters given in the article.

■ **Keying In Source Code** The source code is printed to help explain a program's operation. To key it in, you will need the S-C Assembler.

Without this assembler, you will have to translate pieces of the source code into something *your* assembler will understand. A table of S-C Assembler directives appears in COMPUTIST 17.

■ **Computing Checksums** Checksums are four-digit hexadecimal numbers which tell if you keyed a program exactly as it appears in COMPUTIST. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both appeared in COMPUTIST 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in COMPUTIST 18. If the published checksums do not match those created by your computer, then you typed the program incorrectly. The line where the first checksum differs has an error.

■ **CHECKSOFT Instructions:**

LOAD *filename*
BRUNCHECKSOFT

Get the checksums with: & RETURN and correct the program where the checksums differ.

■ **CHECKBIN Instructions:**

CALL -151
BLOAD *program filename*

Install CHECKBIN at an out of the way place

BRUN CHECKBIN,A$6000

Get the checksums by typing the starting address, a period and ending address of the file followed by a ⊡Y RETURN .

xxx.xxx⊡Y

Correct the lines at which the checksums differ.

You asked for it...

A Disk–Magazine Combo Subscription!

We now have a combination library disk & first-class subscription rate to save you even more. With this new 'COMBO' subscription, you will receive each monthly issue AND its corresponding disk for as much as 43% off the individual rate.

COMBO subscriptions are mailed First Class. AND, you can UPGRADE to this new offer.

OLD NEWS:

COMPUTIST's annual subscription rate has

DROPPED.

Old Rates:
U.S. Domestic $40 per year
U.S. First Class $48 per year
Canada and Mexico $68 per year
All other foreign $120 per year

EXPIRED?

Keep your subscription ALIVE!

Check your mailing label to see if you need to renew your subscription. And if you think you might forget when the time arrives, renew right now. Just use this order blank. Remember, late renewals cannot be back dated.

if you're moving

Let us know at least 30 days in advance so that you won't miss a single issue. Just write your new address on the order form above, and your old address below (or include your present address label).

■ Issues missed due to non-receipt of Change-of-Address may be acquired at the regular back-issue rates. Please remember, the Post Office does not forward third class mail unless requested.
■ COMPUTIST is not responsible for replacing issues lost while forwarding order is in effect.

Name _____ ID#_____

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

The Book Of Softkeys
Volumes I and II

Legends tell of the days when the ancient back issues of Hardcore COMPUTIST were readily available to anyone who wished to purchase them. Those days may be long since past, but the information contained in these ancient documents has been diligently transcribed to the pages of a modern reference work: The Book Of Softkeys.

From deep within the COMPUTIST archives comes a collection of softkeys originally contained in issues 1 through 5 and 6 through 10 (B.O.S. Volumes I and II respectively). The two volumes also contain some of the more useful programs and tutorials presented in those early issues. The books make an economical alternative to those rare (and unavailable) back issues of Hardcore COMPUTIST.

Volume II: Issues 6-10 ($17.95)

contains softkeys for: Apple Cider Spider | Apple Logo | Arcade Machine | The Artist | Bank Street Writer | Cannonball Blitz | Canyon Climber | Caverns of Freitag | Crush, Crumble & Chomp | Data Factory 5.0 | DB Master | The Dic*tion*ary | Essential Data Duplicator I & III | Gold Rush | Krell Logo | Legacy of Llylgamyn | Mask Of The Sun | Minit Man | Mouskattack | Music Construction Set | Oil's Well | Pandora's Box | Robotron | Sammy Lightfoot | Screenwriter II v2.2 | Sensible Speller 4.0, 4.0c, 4.1c | the Spy Strikes Back | Time Zone v1.1 | Visible Computer: 6502 | Visidex | Visiterm | Zaxxon | Hayden Software | Sierra Online Software | **PLUS** the complete listing of the ultimate cracking program...Super IOB 1.5 | and more!

Volume I: Issues 1-5 ($12.95)

contains softkeys for: Akalabeth | Ampermagic | Apple Galaxian | Aztec | Bag of Tricks | Bill Budge's Trilogy | Buzzard Bait | Cannonball Blitz | Casino | Data Reporter | Deadline | Disk Organizer II | Egbert II Communications Disk | Hard Hat Mack | Home Accountant | Homeword | Lancaster | Magic Window II | Multi-disk Catalog | Multiplan | Pest Patrol | Prisoner II | Sammy Lightfoot | Screen Writer II | Sneakers | Spy's Demise | Starcross | Suspended | Ultima II | Visifile | Visiplot | Visitrend | Witness | Wizardry | Zork I | Zork II | Zork III | **PLUS** how-to articles and program listings of need-to-have programs used to make unprotected backups.

To Order: Send $17.95 + Shipping and Handling for Volume II and/or $12.95 + S&H for Volume I. Shipping and handling is $2.00 per book for US orders, $5.00 per book for foreign orders. U.S. funds drawn on U.S. banks only. Washington State orders add 7.8% sales tax. Send your orders to: **SoftKey Publishing, PO Box 110937-BK, Tacoma, WA 98411**
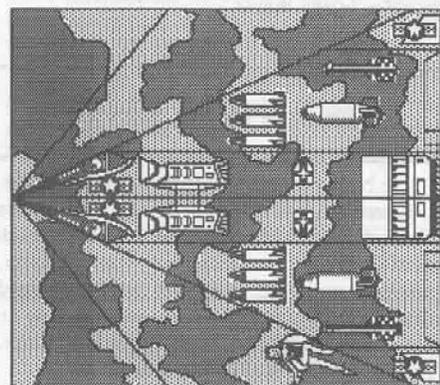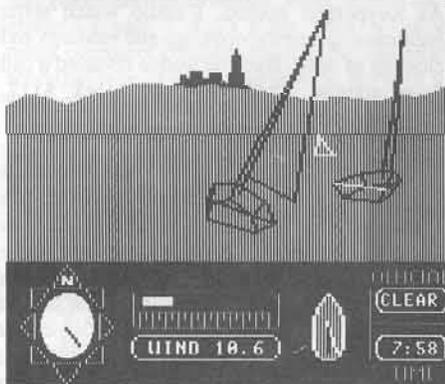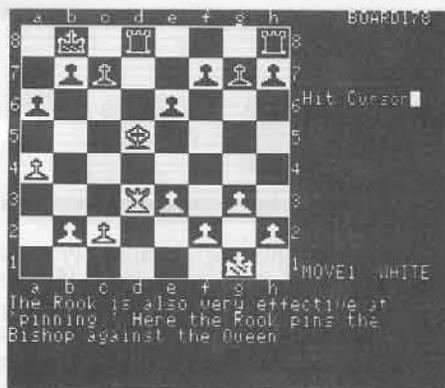
The Rook is also very effective at pinning. Here the Rook pins the Bishop against the Queen

This month's cover:
Graphics from Activision's "Little Computer People."

input

//c Drive Adjustment

I want to thank the person who sent in the letter concerning the adjustment of the internal disk drive on the Apple //c. I was having problems copying some disks that required decreasing the speed of the drive. Then I read the article in COMPUTIST No. 32. I adjusted the drive while using the "Check Drive Speed" with Echo Plus.

I would like to know if anybody has information on how to adjust the external drive made by Apple for the Apple //c. Could someone please give me information on this?

Jeffrey Whitman
Tewksbury, MA

Another Pirate Speaks

This concerns the letter from The Chemist concerning the Piracy Rebuttal Rebuttal. I understand completely there is no reason to spend $40.00 or more for a ware. There is no disk worth that much. I would have to get a bank loan to pay back all the wares I have accumulated throughout the years.

I have been a subscriber for almost 2 years and been very happy. I also have every issue. COMPUTIST is the best Apple magazine ever! The Chemist talks about paying 40 dollars for software, and he thinks that's bad. Did he ever think about this magazine, 40 dollars a year!

The good side of COMPUTIST is that it helps me learn more techniques that other people use on breaking wares, but then the $40.00 I don't like! I said at the beginning there is no disk worth 40 dollars, well there is no magazine worth more than 25 dollars a year!

P.S. Hardcore COMPUTIST is still the BEST computer magazine even if it costs an arm and a leg to get it.

The Hood has spoken!
Lacrosse, WI

Mr. Hood: We agree with you that 40 dollars a year is too much for COMPUTIST. That is why starting with COMPUTIST No. 36, we have lowered the subscription to 32 dollars per year.

Keyperfect Softkey

Like COMPUTIST, Nibble magazine has a checksum program to find errors in programs typed from the magazine. Unlike COMPUTIST, Nibble charges $29.95 for the program.

Not only that, but the program is protected, although the default setting for the Copy][Plus nibble copier copies it easily. The protection is still a nuisance however, because I like to have a number of utilities together on one disk (lately on hard drive) rather than having to boot a seperate disk for each operation desired.

It is fairly easy to remove the protection. The following procedure describes how to do it. You'll need a sector editor.

1) Copy the Keyperfect disk. COPYA will barf on track $22 but we're only interested in the first half of the disk anyway.

2) Search the disk for the string "AD 30 BF 8D 21 20 20 00 BF" I found the string at the beginning of track 07, sector 07.

3) Change the string to "AD 30 BF 8D 21 20 4C 13 20"

4) Write the sector back to the disk.

5) Use Copy][Plus utilities (or the //c utilities, or the abominable ProDOS FILER) to copy the file KEYPERF.SYSTEM to the disk of your choice.

For those of you who aren't familiar with it, the Copy][Plus disk is a bargain just for the utilites it contains, even if you never use the bit copy system.

A surprising little program called ",ProDOS Diagnostic Wedge" which appeared in the July 1986 issue of Call-A.P.P.L.E. made breaking Keyperfect much easier. The wedge displays on screen each call to the ProDOS Machine Language Interface and waits for a keypress. As Keyperfect booted, I could watch some legitimate-appearing openings and readings and closings of files. Then the wedge revealed a call from address $2009 to read block $117. Suspicion city.

That's all for now.

Ann Onymous
Santa BArbara, CA

Dearest Ann: Thanks for the softkey. Did you know that Keyperfect listings were used in the movie "The Terminator" as diagrams generated in the mind of the robot? You should check out this movie to see the Keyperfect listings and the Apple source code. In addition, I'm sure you will also enjoy Arnold's rippling muscles in the movie.

Senior PROM Service

I've been a regular subscriber since issue No. 4. I can't tell you how much I appreciate your publication. It's the only one I wait for each month. I, like so many of your readers, purchase software, then get really steamed about having to send the software company an additional $10.00 to $20.00 for back-up disks. Your magazine really fills a need.

I am writing this letter not only to compliment your magazine, but to let you know how happy I am with a product you advertise, and how helpful and supportive Cutting Edge Enterprises is to it's customers.

For those who are not familiar with the company, Cutting Edge Enterprises offers the Senior PROM, a firmware device that, along with many very useful functions, allows the Apple //e to reset into monitor at the flip of the switch.

I did experience some difficulties with my Apple //e when the Senior PROM was installed (I got garbage on my screen while using AppleWriter II and AppleWorks). I promptly contacted Clay Harrell of Cutting Edge Enterprises, explained the problem, and returned my Senior PROM for repair/replacement. After fully testing the Senior PROM he couldn't find fault with it. He asked for a detailed description of my computer setup, peripheral cards and switch settings, printer type, etc., and he asked for my AppleWriter II program. He wanted to duplicate my setup at his shop.

input

Once again, he couldn't find a problem in the Senior PROM and thought I may have a problem with my extended 80 column card. So, Clay Harrell sent an Apple //e extended 80 column card through the mail to me to see if that may cure the problem. Now if that isn't dealer support, I don't know what is. How many computer companies would be willing to send that type of expensive item in the mail to basically, a complete stranger?

As it turned out, when I had my Apple dealer install the Apple //e enhancement kit, he forgot to replace the Video ROM chip with the enhanced one. That was the problem!! The Senior PROM was fine all along, but nontheless, Clay Harrell was so willing to go well beyond any definition of good customer service to help solve my problem.

Cutting Edge Enterprises returned my AppleWriter program with my Senior PROM and I returned the 80 column card to Cutting Edge Enterprises along with my thanks to Mr. Harrell for all his efforts.

I do hope that you will take the time to publish this letter so that your readers can take advantage of the capabilities of the senior PROM and the **super** support Cutting Edge Enterprises offers it's customers.

Mike Luker
Mission Viejo, CA

Some Softkeys

Here are a few easy to do, step-by-step softkeys that could interest some people out there.

Softkey for: Transend II

Requirements:
COPYA
A blank disk
Transend II

1) Run COPYA

2) At the first prompt, press ⌂C to break the program.

```
CALL -151
B925:18 60
B988:18 60
9DBFG
```

5) Type "RUN" and copy the disk.

6) Boot a normal DOS disk.

7) Insert Transend II.

8) Type "MAXFILES1".

9) Type "BLOAD BUFFERED MODEM.WORK".

10) Now type "CALL -151" and type in the following changes.

```
8311:DE
9DBFG
```

11) Type "BSAVE BUFFERED MODEM.WORK,A$8200,L$18A0" (this is to save your changes)

You now have a COPYA version of Transend II!

Softkey for: Einstein Compiler

Requirements:
COPYA
A blank disk
A sector editor
Einstein Compiler

1) Copy your original with COPYA

2) Using a sector editor (EG. The Inspector) perform the following sector edits on your freshly made backup.

Track	Sector	Byte	From	To
$08	$04	$2A	$BD	$4C
$08	$04	$2B	$8E	$E2
$08	$04	$2C	$C0	$91

You should now have a COPYA version of Einstein compiler.

Jerome Thelia
Decorah, IA

Dam Busters and Star Trek

Now here is a couple of softkeys for Dam Busters and Star Trek (the Kobayashi alternative) ver. 1.1.

Dam Busters

Requirements:
Blank Disk
Dam Busters (Accolade)
COPYA
Sector editor

1) Copy original with COPYA or any fast copier. Ignore errors on tracks $21 and $22.

2) Use your sector editor and make these changes

Track	Sector	Byte	From	To
$13	$0B	$2E	$4C	$EA
$13	$0B	$2F	$00	$EA
$13	$0B	$30	$C6	$EA

That's it.

Star Trek ver. 1.1

Requirements:
Blank disk (double sided)
Star Trek ver. 1.1 (Simon & Schuster, Inc.)
COPYA
Sector editor

1) Copy disk with COPYA or any fast copier. Both sides.

2) Use your sector editor and make this change on side B.

Track	Sector	Byte	From	To
$12	$0E	$3D	$DB	$60

That's all for now. Your magazine is Great!

Steven Mood
Paris, IL

Small Softkey Blues

I would like to voice a complaint to the readers regarding those very, very short softkeys. In several issues, I have seen deprotection procedures for various software packages, which consisted solely of a sector edit list, some which only require that one byte be changed. Now I realize that not all of the readers who submit softkey procedures are machine language geniuses and hardware wizards who can recite the nuts and bolts of every copy protection scheme in existence. I certainly am not one, but I would appreciate a little insight as to what is going on.

My case and point, re: Mouse Desk softkey, COMPUTIST No. 32, page 9. I purchased Mouse Desk about five months ago to use in conjunction with my RamWorks card which is configured as a RAMdisk. However, since it would reload Mouse Desk off of the disk whenever I quit an application program, I wanted to move Mouse Desk onto the RAMdisk. It was then that I discovered the fact that it was protected. I have called VersionSoft and am waiting for a reply. Meanwhile, the Mouse Desk softkey came in with COMPUTIST No. 32, I felt an initial wave of relief. The softkey looked simple enough, change byte $53, track $0C, sector $04, from D0 to F0. To my dismay, D0 appears over 100 times on the disk, 6 times on track $0C and not at all in sector $04 of that track.

It became apparent that my version of Mouse Desk was different from the version used in the softkey. Thus it is possible for files to have

gotten shuffled around and program code relocated, but very unlikely that the copy protection scheme would be changed at all, so the softkey should still work, but the change would have to be made at a different location. If the softkey had contained more information, specifically, a search pattern consisting of two or three bytes before and after the critical byte, then we would know instantly (upon a disk search) and conclusively, whether or not the softkey will work for us. As of yet, I still have no clue as to how I can deprotect my copy of Mouse Desk.

Come on guys, give us softkey rookies a break and help us out a little bit, okay? Thanks.

Brian Chinn
Seattle, WA

Mr. Chinn: We couldn't agree more.

Time Zone Troubles

I saw in COMPUTIST No. 31, that you had a softkey for Time Zone ver. 1.1. The softkey defeats the nibble count alright, but apparently Mr. Pollack did not bother to play the game, because every time you tried to take an object, the program would just tell you the object was there and then print a description of the room. The program must somehow check to see if the routine was bypassed or if the routine was altered. The bytes that were to be edited on disk are at memory location $8D9 in memory, but I could find no reference to this location so I assume it uses nibble count checksums. Maybe someone could help with this problem. For those of you with Time Zone, here are a few APT's.

1) Don't enter the courthouse in Philadelphia.

2) Get the Boarding Pass for Columbus' ship for On Deck only.

3) Wait at the Russian Castle.

4) You cannot get the Emerald from the Buddha.

I have also discovered a way to cheat death on the Sierra On-Line adventures one through six. Anytime you die, press CONTROL-RESET and the program will print "Ouch!". Then go on playing the game. For example, in Time Zone, there is a buffalo herd stampeding toward you. After you are trampled, hit CONTROL-RESET and continue playing the game.

Finally, I have one question, that I hope you can answer. A program I have has every track listed as track zero in the sector address information, thus Super IOB cannot copy it.

Despite all my efforts at trying to ignore the track check, I have been unsuccessful. I would like to know if you or any of the readers can tell me how to tell the RWTS to ignore the track. Thanks for a great magazine!

Christopher Dean
Spring Hill, FL

Mr. Dean: Don't underestimate the power of Super IOB. The following Super IOB controller will copy a disk with every track marked as 0. It is a direct descendant of the controller for Rocky's Boots appearing in COMPUTIST No. 14. Note that during the copy procedure, the drive will recalibrate at the start of every read cycle. If the drive recalibrates during a read cycle, a bad copy will be produced.

```
1000 REM ALL TRACKS ARE 0
1010 TK = 0 : ST = 0 : LT = 34 : CD = WR : POKE
     48573 ,128
1020 GOSUB 490 : T1 = TK : TK = 0 : CD = 0 : GOSUB
     100 : GOSUB 80 : S = - 128 : GOSUB 130
1025 CD = RD : S = T1 * 2 : GOSUB 130 : S = 2
1030 TK = PH / 2 : GOSUB 430 : TK = 0 : GOSUB 100
     : ST = ST + 1 : IF ST < DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : GOSUB 130 : IF PH < LT * 2 THEN
     1030
1060 GOSUB 490 : TK = T1 : ST = 0
1070 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST
     < DOS THEN 1070
1080 ST = 0 : TK = TK + 1 : IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "DONE^ WITH^ COPY" : END
```

Coping with Copy Programs

I started out with a copy of Locksmith 5, which did a fair job at best. Looking for the ultimate copy program a dealer recommended Copy][Plus, which turned out to be the best so far. Then I came across COMPUTIST on a magazine stand, which is the only way to go. I then ordered all the back issues. However, some programs refer to back issues that are no longer available, especially volume 10 which I'm in need of. I received the Book of Softkeys volume I and had to wait a long time for the Book of Softkeys volume II to be compiled and finally received it.

Reading in COMPUTIST No. 8 about copy programs, I decided to get EDD III. The program is good except for the problem of entering the parameters manually. That's not much of a problem until you run across programs that are hard to copy. It has been always the case with me that it tells you to

recopy certain tracks if the program won't boot or run. Those tracks are the ones with a dozen parameters to enter manually over and over again just to find that the program still won't run!

I noticed in COMPUTIST No. 30 a note from a reader about an article in the January issue of inCider about a program called Echo Plus. There was an address and a price ($59.95). I ordered the program and in a short time I received a letter and my check back stating that I had to add shipping and handling ($5.00). So I sent it. Shortly thereafter, I received the program. Talk about a disappointment! Hoping it would copy the programs that the other copy programs won't this program won't either. In fact, I tried to copy some programs that the other copy programs did and Echo Plus still could not. However, it did copy non-protected programs and some simple copy protected programs. I read the book thoroughly and tried every combination possible and still poor results. They stated in the letter that if it doesn't meet with my expectations within two weeks to return it, so I did!.

George Cawthorne
Mt. Prospect, IL

Another Softkey

Here is a softkey I have recently completed.

Reader Rabbit

1) Load the COPYA program.

RUN COPYA

2) Break the program and delete line 70.

⌥C
70

3) Enter the monitor and change B942 from 38 to 18.

CALL -151
B942:18

4) Run COPYA and copy the disk.

⌥C
RUN

5) Format a disk. Use any name but HELLO as the boot file.

6) Transfer all files from your copy to the formatted disk (I use Copy][Plus 5.0).

7) Change the boot program to HELLO. (Again, I use Copy][Plus.)

All done!

John Cotter
Bay City, MI

readers' softkey & copy exchange

Lewis Clark's softkey for...

Cranston Manor

Sierra On-Line
36575 Mudge Ranch Road
Coarsegold, CA 93614

Requirements:
Super IOB v1.5
A blank disk
BOOT13 from DOS 3.3 System Master
DOS 3.2's RWTS in a binary file
Nibble copier and sector editor (optional)

Cranston Manor, published in 1981 by Sierra On-Line, is the third game of the "Hi-Res Adventure" series. The disk is protected by a nibble count, and the whole disk is readable under standard DOS 3.2 except track 0, sector $A and track $18. Track $18 is used by the nibble count routine and contains no useful data. If we circumvent the routine which checks track $18, we would have a working copy of Cranston Manor.

The nibble count routine of Cranston Manor starts at $9900 and is only executed once while booting up. It decodes itself before executing as shown here:

```
9900-   A0 00       LDY #$00
9902-   A9 55       LDA #$55
9904-   59 0E 99    EOR $990E,Y
9907-   99 0E 99    STA $990E,Y
990A-   88          DEY
990B-   D0 F5       BNE $9902
990D-   EA          NOP
```

After the program has checked the nibble count, it will store a $60 at $990D and jump to $9900 to re-encode the subroutine before it returns. Since this routine is only for the purpose of protection and is not needed for the program itself, we could put a $60 at $900D on the disk, and then it will not execute the checking routine.

Since this program has many shape tables, the program modified many subroutines inside of the DOS and RWTS. Therefore the deprotected disk will be in the DOS 3.2 format rather than DOS 3.3. For the first method I have given, you will need the DOS 3.2 RWTS in a binary file.

1) Install the Cranston Manor controller (notice the changes needed to make it work with DOS 3.3) into Super IOB 1.5 and RUN it. Answer "Yes" to the format option.

2) When Super IOB is finished with the disk, the nibble count routine will be removed.

To run this disk, you need to BRUN BOOT13 (from your DOS 3.3 System Master) and when the program asks for the slot number, put in the disk we copied by Super IOB and press RETURN. Then the program will boot.

The disk produced by this method is not friendly enough, since you always need to BRUN BOOT13 first, and then run the program (but you don't need a nibble copier either.) If you wish to prevent having to do that, or can't find a DOS 3.2 disk to get the RWTS from, you can use the following method.

1) Use a nibble copier to copy tracks $0-$22.

2) Set your sector editor so that it reads and writes DOS 3.2 sectors and change track $1F, sector $B, byte $0D from $EA to $60.

Now you have a working copy of Cranston Manor, and you may wish to store the original in a safe place.

controller

```
340 POKE 9984 + (A1 - T1 ) * DOS * 256 + ABS
    (FAST * (DOS - 1 ) - A2 ) * 256 + A3
    ,A4 : NEXT : RETURN
405 GOSUB 360
610 PR# 0 : IN# 0 : POKE 860 ,MB : POKE 861 ,LT
    : POKE 862 ,LS : POKE 900 ,DOS - 1
1000 REM CRANSTON MANOR CONTROLLER
1010 TK = 0 :LT = 35 :ST = 12 :LS = 12 :CD = WR
    : FAST = 1
1020 DOS = 13 :MB = 130 : POKE 775 ,96
1030 GOSUB 490 : GOSUB 610
1040 IF PEEK (TRK ) > 32 AND TK < 33 THEN T1 =
    TK :TK = 32 : GOSUB 310 :TK = T1
1050 GOSUB 490 : GOSUB 610
1060 TK = PEEK (TRK ) :ST = PEEK (SCT ) : IF TK
    < > LT THEN 1030
1070 HOME : PRINT "COPYDONE" : END
1080 DATA 1▲ CHANGES ,31 ,11 ,13 ,96
10010 PRINT CHR$ (4 ) ; "BLOAD▲
    RWTS.13,A$1900"
```

controller checksums

340	— $A295	1040	— $A11C
405	— $E7CD	1050	— $1511
610	— $5C32	1060	— $5D13
1000	— $0F36	1070	— $4641
1010	— $DEC0	1080	— $27E0
1020	— $B9DD	10010	— $790B
1030	— $709D		

Jean-Michel George's softkey for...

Ghostbusters and The Designers Pencil

Activision
P.O. Box 7287
Mountain View, CA 94039

Requirements:
COPYA
A sector editor
Two blank disk sides

The protection scheme is the same for the two programs. Only addresses, values, tracks and bytes change. On each disk, there are three protected tracks. The format of these tracks has been modified: the address field epilogue is changed. You can see it very well with a nibble editor. That's why COPYA can't read them. Normally, this epilogue is $DE AA EB; on these tracks, it is $DE xx EB (where $xx is a byte different from $AA). The byte $xx is different on each track:

Ghostbusters

Track	$xx
$01	$AB
$12	$BC
$13	$CD

Designer's Pencil

Track	$xx
$3	$8B
$4	$9C
$5	$AD

On Ghostbusters and Designer's Pencil, there is a routine that modifies RWTS to load those tracks and store the $xx bytes in memory at addresses $00, $01 and $02. Then there's a jump to a decode routine that needs the correct $xx bytes. So, we have to modify it: whatever it finds, the right values will be used anyway.

First, we have to modify RWTS to copy the whole disks with COPYA without checking the modified address field epilogue bytes.

1) Boot a DOS 3.3 disk. Then enter the monitor and make the following changes:

```
CALL-151
B99C:EA EA
3D0G
```

2) RUN COPYA and copy the Ghostbusters and Designer's Pencil disks. Now your copies are in DOS 3.3 format.

3) Boot your sector editor and make the following changes to Ghostbusters:

Modify track $00 sector $0A

Byte	From	To
$52	$A5	$A9
$53	$04	$BC
$59	$A5	$A9
$5A	$04	$CD
$AA	$A5	$A9
$AB	$04	$AB

Write the sector back to the disk and you're done with Ghostbusters.

4) Make the following changes to Designer's Pencil:

```
      Modify track $00 sector $0A

         Byte   From   To
         ----------------------
         $3D    $85    $EA
         $3E    $00    $EA
         $5E    $EE    $20
         $5F    $3E    $A2
```

While still working on track $0 sector $A, store the following bytes starting at byte $A2:

```
A9 8B 85 00 A9 9C 85 01 A9 AD 85 02 60
```

Write back the sector.

The routine you just put at byte $A2 of track $00 sector $0A of Designer's Pencil stores the right values in $00, $01 and $02 locations.

You now have COPYAable versions of Ghostbusters and Designer's Pencil!

This protection scheme is used in other Activision games such as Space Shuttle. So if you find some protected tracks like those, search for zero page STAs (STA $01) to locations $00-$02 ($85 00, $85 01 or $85 02) with a disk searcher and you might fall on their protection routine. Another hint: they use RWTS and an IOB table to load sectors, the same as in DOS 3.3. Good luck.

Keith Duff's softkey for...

E-Z Learner

Requirements:
Apple 48K][Plus or better
System Master disk
A blank disk

E-Z Learner is a very useful "flash card" program. With it, you can make up "flash card" like questions. After you have made up the questions and the answers, they will be reviewed by showing the question first. Then you are prompted for a keypress, after the keypress, you will be shown the answer. You tell the computer if you got the answer right or wrong. The wrong questions are reviewed over and over again until you get them correct.

The protection used in this program was, as far as I could tell, an altered version of DOS, deleting or changing some commands. To

deprotect it, follow the instructions below. You will then have a COPYAable version of E-Z Learner.

Step by Step

1) Boot your master disk.

PR #6

2) Remove your system master and format a blank disk.

INIT E-Z LEARNER

3) Boot the E-Z Learner disk and type ⌘C to break into Applesoft BASIC. This may take a few tries because pressing ⌘C too soon will reboot the disk.

4) Replace the E-Z Learner disk in the drive with the disk you formatted in step 2.

5) Save the program.

SAVE E-Z LEARNER

You may note that line 1 of this program is responsible for its ignoring of Applesoft commands and the Reset key. You may want to remove the offending code in it at your convience. The resulting line would read:

```
1 B$= CHR$(13) + CHR$(4) : ONERR GOTO 169
```

Larry Rando's softkey for...

The American Challenge

Mindscape, Inc.
3444 Dundee Rd.
Northbrook, IL 60062

Requirements:
Super IOB 1.5
A sector editor

The American Challenge is a sailing simulation which has smooth 3D animated graphics. The program can be played with one or two players. In 2 player mode you play over a modem and race against each other as in real life. You have a variety of different courses to choose from and your choice of what type of boat; motor or sail. A match competition pits the top eight sailors against each other to win a trip to Perth, Australia, where the next America's Cup race will be held in 1987.

The Protection

Examination of the disk reveals that the data and address epilogues were changed to FF FF. A copy could be made by disabling the RWTS' error checking and using COPYA, but a Super IOB controller programmed to read those particular markers is much more reliable (and faster).

After making the copy and trying it out, the disk drive rebooted after just a few seconds. I suspected there had to be a routine somewhere in the program that jumped to $C600 (the routine first executed when a disk is booted). So, I searched for this routine using the Copy II+ 5.5 search function to look for the bytes 4C 00 C6 (JMP $C600) on the disk. The bytes came up on track 0, sector $B, so I disassembled the sector to see if there was any thing that resembled protection code. Looking through the code that was executed before the $C600 on that sector I found the protection. It did a series of BNEs (Branch on result not zero) and BEQs (Branch on result zero) to a specific location in memory which happened to be one instruction above the JMP $C600. The instruction it went to (DEC $50) was where the program crashed and jumped to $C600. I replaced the BEQs and BNEs to this location with EAs (no operation) so that it wouldn't branch to the protection, therefore skipping it and continuing with the normal load. I booted up with my modifications and the program worked, but just to be on the safe side I searched for the instruction(s) that accessed the DEC $50 just in case it was somewhere else on the disk. Nothing turned up so track 0, sector $B was the only place containing the protection code.

The Procedure

1) Put the controller at the end of this article into Super IOB and copy the American Challenge disk.

2) Start up your favorite sector editor and make the following changes.

```
       Track $0, Sector $B:
       Byte      From      To
       ------------------------------
       $24       F0 67     EA EA
       $29       B0 62     EA EA
       $2F       D0 F1     EA EA
       $39       F0 52     EA EA
       $47       F0 44     EA EA
       $54       D0 37     EA EA
       $5D       D0 2E     EA EA
       $6C       F0 1F     EA EA
       $7B       D0 10     EA EA
```

3) Write the sector back out. Now you have a cracked copy of American Challenge. Smooth sailing!

controller

```
1000 REM AMERICAN CHALLENGE
1010 TK = 0 :LT = 35 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1020 GOSUB 170 : GOSUB 490 : GOSUB 610
1030 GOSUB 230 : GOSUB 490 : GOSUB 610 : IF
     PEEK (TRK) = LT THEN 1050
1040 TK = PEEK (TRK) :ST = PEEK (SCT) : GOTO
     1020
1050 HOME : PRINT "COPYDONE" : END
5000 DATA 255 ,255 ,255 ,255
```

controller checksums

1000	– $356B	1040	– $C10B
1010	– $2544	1050	– $52C7
1020	– $C219	5000	– $44E3
1030	– $F780		

Danny Pollak's softkey for...

Crime Wave

Requirements:
Apple][Plus or equivalent
Crime Wave
An INITialized disk

Crime Wave is a game in which you, the policeman, drive around in your blue police cruiser patrolling the city streets. Your objective is to catch the bank robbers in the act or soon after they rob a bank.

Crime Wave occupies tracks 0 through $11 with alternating address headers of D5 AA 96 and D4 AA 96. Both the address and data ending marks are changed from DE AA to DA AA. The following is how I went about unlocking Crime Wave with a boot code trace.

The first thing we will do is load Boot1 from track 0 sector 0 of the Crime Wave disk. Do this by entering the following:

```
CALL-151
9600<C600.C700M
96FA:98
9801:4C 59 FF
9600G
```

Boot1 has now been loaded into page 8 of memory. Move it to $9800 and modify it so that it loads Boot2 and then exits to the monitor.

```
9800<800.900M
9824:90
9827:98
9900:2C 81 C0 4C 59 FF
9600G
```

Boot2 has now been loaded at $900. Move it up to $9900 and modify it by entering the following:

```
9900<900.9FFM
9935:99
9938:99
9949:4C 00 60
6000:A9 2C 8D 10 01 A9 81 8D
6008:11 01 A9 C0 8D 12 01 A9
6010:4C 8D 13 01 A9 59 8D 14
6018:01 A9 FF 8D 15 01 4C 76 07
9600G
```

The game should finish loading and then exit to the monitor. If it doesn't exit to the monitor, hit Reset and it will. Type in the following:

```
2300<800.8FFM
2400<8000.BDFFM
```

Boot the initialized disk and type in the following:

```
CALL-151
800<2300.23FFM
7FD:20 00 23

2300:A9 00 8D F4 B7 A9 02 8D
2308:EC B7 20 E3 03 20 D9 03
2310:A2 00 A0 FF 84 3E C8 84
2318:3C 84 42 BD 3C 23 85 43
2320:E8 BD 3C 23 85 3D E8 BD
2328:3C 23 85 3F E8 20 2C FE
2330:E0 03 D0 DE A9 60 85 30
2338:20 05 BD 60 80 24 61

5C9E:DE
5D35:DE
5D91:DE
614E:EA EA
BSAVE CRIME WAVE,A$7FD,L$5A03
```

Now you have a totally unprotected version of Crime Wave. The high scores will be saved on track 2 sector 4.

Jim S. Hart's softkey for...

Encyclopaedia Britannica programs

Requirements:
Super IOB 1.5
A blank disk for each original
A fast DOS such as Diversi-DOS (optional)
A dislike for copy protection on eductional programs

Here is a short softkey to deprotect the eductional programs put out by Encyclopaedia Britannica. It has been successfully used on **Verbs #1-3**, **Pronouns**, and **Nouns** without any problem and should work on others in the series. For the curious among you, the disks have altered epilogues, a bad checksum here and there, and the nybble translate tables have been changed. The first POKE in line 1020 of the controller below takes care of the altered epilogues and checksums and the second POKE in line 1020 takes care of the changed translate table. The POKEs in line 1030 reset the values back to normal for writing to the copy disk.

Steps

1) INITialize the blank disk, preferably with a fast DOS.

2) Install the controller listed below into Super IOB and copy the disk. Do not format the target disk.

After the copy is made, boot it up. If you do not get a "FILE NOT FOUND" error you are done.

If you do get this error, look for a file with the name of BOOT or LOAD. This is the boot program name. Try again, INITializing the blank with the name of the boot program you found instead. Copy with Super IOB again and you will have a deprotected copy of your Encyclopaedia Britannica program. No more long waits and replacement fees for backups to arrive in the mail!

controller

```
1000 REM ENCYCLOPAEDIA BRITANNICA
1010 TK = 3 :LT = 35 :ST = 15 :LS = 15 :CD = WR
     :FAST = 1
1020 POKE 47426 ,24 : POKE 47786 ,0 : GOSUB
     490 : GOSUB 610
1030 POKE 47426 ,56 : POKE 47786 ,170 : GOSUB
     490 : GOSUB 610 : IF PEEK (TRK) = LT THEN
     1050
1040 TK = PEEK (TRK) :ST = PEEK (SCT) : GOTO
     1020
1050 HOME : PRINT "COPYᴬ DONE." : END
```

controller checksums

1000	– $356B	1030	– $FF50
1010	– $2445	1040	– $1103
1020	– $CBB6	1050	– $1BC5

The Disk Space You Didn't Know You Had

by Phil Goetz

Many people have a good deal of extra space which they don't even know about: the unused tracks on protected (or softkeyed) disks. In many cases, these tracks can be used for files while keeping the protected program on the other tracks. If you have private files (such as class grades) this is a simple way of hiding them. Using these extra tracks involves three steps: determining which tracks are in use, initializing those that aren't, and writing a VTOC and catalog directory to track $11.

This should only be done to nibble copies of protected disks. Never initialize tracks on an original. Also, I do not advise using this technique on anything but games. You may lose important data or programs if a business or utility program crashes due to this technique.

There are several methods of finding which tracks are in use. The easiest is to check a bit copier's parameter list for that program. The tracks to avoid are the ones listed in the parmlist. Another method would be to use the disk scan utilities on some copy programs such as Locksmith and Back-It-Up to determine where there is data, but nowadays this is highly unreliable. A third method is to inspect each of the tracks and half-tracks with a nibble editor, looking for obvious signs of data on those tracks.

Another method is to open the case of your drive and watch which tracks the program uses as it runs. For a description of how to do this, see "Getting On the Right Track" by Robert Linden in Hardcore COMPUTIST No. 5 (or Book of Softkeys I).

Note that the first method relies on the accuracy of the parameter list; the second and third methods will not point out "garbage tracks" which contain no data but are used to insure that the disk is protected (i.e. nibble count tracks); the fourth is reliable only if you are certain that you have watched every disk access that the program will ever do. I suggest using all four methods and playing it safe when in doubt.

If track $11 is in use, don't bother; using this technique would be more trouble than it's worth. If it isn't, initialize all of the unused tracks, using something like the INIT utility in Beneath Apple DOS or Bag of Tricks.

Next, copy track $11 from a normal, initialized disk with no programs on it to track $11 on the protected disk. Then use a sector editor such as The Inspector to read track $11 sector 0 from the protected disk. This is the VTOC, which tells which sectors are in use. The bit map for track 0 is at $38-39. For an explanation of the VTOC and the track bit maps, see pages 132-134 of the DOS Manual. After a careful reading of those pages, change the bit map to show track $11 and all tracks used by the protected program as being in use and the other tracks as not being in use. Write this back to track $11 sector 0. If you are using The Inspector, press M to see a disk map to make sure you have done it correctly. Super Copy and Copy II Plus can also show disk maps. None of them, however, show anything beyond track 34 ($22).

Now, when you boot the disk the protected program will run, but when you boot a normal disk you can store files on the protected disk and use it just like any other normal disk. I have done this to backup copies of Autobahn, Gorgon, and Snoggle without any adverse effects.

softkey for...

Adventure Writer

by Jim S. Hart

Codewriter Corporation

Requirements:
Adventure Writer original
A blank disk
A fast DOS (optional)
A file copy program
A dislike for copy protection

Adventure Writer is a program that allows you to compose your own adventure games (hence the name). I am an adventure game nut, with the Ultima series being my favorite, so when I bought Adventure Writer, I went about seeing what the program could do. I was not disappointed with it, except that it was copy protected. Since I dislike copy protection except for the fact that it was teaching me assembly language, I went about relieving that one problem. The cookbook instructions are at the end of the softkey, but if you are interested on how the protection was circumvented, read on...

The first thing I checked out was to see where data might be stored on the disk. Using EDD's "Analyze Original Disk" function showed that there was no data on track $23 and that whole tracks were probably being used. The next thing to check was that the disk's formatting was normal. A random sampling of tracks with Copy II Plus' nibble editor revealed that the disk was written in a normal format. Since the format was normal I took a chance that the disk could be CATALOGed. Sure enough, I was right.

I then used Locksmith's fast copy program to make a test copy to work with. All was fine until the copier reached track $04, sectors 0-1. It showed these sectors as not being normal while the rest of the disk was. Hmmm... could it be that these two sectors had something to do with the copy protection? Ignoring this for the moment, I tried booting up the copied disk. All seemed okay and it arrived at the main menu all right. Each menu item chosen worked fine with the exception of the "Run Adventure Writer" option.

After choosing this option, the disk drive came on, did some disk activity, and then printed garbage on the text screen and died. "Aha...", I thought, "those two sectors must have been important." I then followed the boot process in regard to what files were run in what order so that the guilty file could be found. The boot progam, "AW", BLOADs "AW.OBJ" and the displays the menu. If the "Run Adventure Writer" option is chosen, a CALL 24576 is executed, which does a MAXFILES 1 then BLOADs the binary files "DBASE", "TABLE", "EDITOR", and "INTERPRETER" (look at an ASCII dump of the file "AW.OBJ" to see how I found these file names). The next thing it does is call a subroutine at $4001. This is where the "DBASE" file was BLOADed in. At $4001 there is a JuMP to $454C. Success! Looking around $454C you notice references to memory locations $C0E8-$C0EF (direct use of the disk drive in slot 6). This must be where the protection is lurking around.

After some inspection, I determined that the protection code starts at address $457B and lingers on for about 30 bytes or so. Should I take out the call to the routine or satisfy it so that it thinks that everything was okay? The second option was the one I chose. At $4584, the code checks to see if what was read from those two odd sectors matches a table loaded at $4641. If the result checks out (result = zero) then everything is okay and the program proceeds. Well, I figured that the easiest way to fix this was to make sure that every byte matched up and gave a result of zero. I replaced the comparison instruction at $4584 (CMP $4641,X) with a LoaD Accumulator with zero (LDA #$00) instruction followed by a NOP. The code will now think that every comparison was equal and go about its merry way. The protection scheme is history!

Step by Step

1) INITialize the blank disk, preferably with a fast DOS. Use the file name "AW".

2) Copy the files from the original disk to the newly INITialized disk.

3) Load up DOS and get into BASIC.

4) Put in the copy disk and load in the file with the protection in it:

BLOAD DBASE, A$4000

5) Enter the monitor via CALL -151.

6) Replace the CMP instruction with LDA #$00 followed by a NOP.

4584:A9 00 EA

7) Save the file back to the disk:

BSAVE DBASE, A$4000, L$64A

That's it! You now have a deprotected copy of Adventure Writer. I hope the techniques I used can help out other folks who are trying to deprotect their originals also.

Softkey for

by Isaac de Pig

Datamost Software

Requirements:
48K Apple II and up
A blank disk
A disk with DOS (any DOS)

I have seen Mychess II appear on the most wanted list for several issues. I got a softkey for this software and I think it can help the people who have problem with Mychess II. Mychess is a chess game with 3-D action and it is very interesting.

The Protection

The software is two sided and both sides can be copied with COPYA. The back side is the data disk and is not protected. The protection on the program side causes the disk to reboot before the game starts. The booting of the front side is divided into two parts. First, all the data is read, a title page appears and the drive stops. After a key is pressed, the drive is turned, it recalibrates and moves track $10 to check the disk. If it is not the original disk, it reboots.

The protection scheme Mychess II uses is probably synchronized tracks. Unlike other softwares with the similiar protection, it can not be copied with bit copiers. This is because the track length of each track is shorter than usual and the tracks are synchronized from track 1. In addition, nearly all the data on the disk is encrypted and making it difficult to find the protection routine on the disk.

The Investigation

Since the data on the disk is encrypted, to deprotect the software, I used the boot code trace method. Following the normal procedure of boot code tracing, that is:

```
CALL-151
9600<C600.C6FFM
```

```
96F8:60 N 9600G
C0E8
```

We get the following code:

```
801-  LDY $0800  Load Y register with $01
804-  BEQ $081F  Not taken
806-  LDA $830,Y  Set sector to read
809-  STA $3D
80B-  DEC $800
80E-  INC $3F4  Set reset vector
811-  LDY #$02
813-  LDA $840,Y
816-  STA $3F2,Y
819-  DEY
81A-  BPL $813
81C-  JMP $C65C  Read a sector
81F-  LDA #$00
821-  STA $0D
823-  JMP $08FD  Run the code read in
```

Looking at this code, we find that after the data on track 0 is read, a JMP to $8FD is taken. So, we get rid of this JMP to $8FD and countiue the trace by:

```
96F8:A9 60 8D 23 08 4C 01 08
9600G
C0E8
```

We find the code at $8FD is as follows:

```
8FD-  BIT $C010
900-  JSR $FC58
903-  LDX #$00  Read data routine
      .  .  .
91E-  STA $C0E8  Turn drive off
921-  LDX #$00
      .  .  .
92B-  LDA $C000  Read keyboard
92E-  BMI $0938  if key press jump to $0938
      .  .  .
938-  BIT $C010  Clear keyboard
93B-  JMP $B200  Jump to $B200
```

These codes show that after all data is read and a key is pressed, it takes a JMP to $B200. So, we have to get the code at $B200 by the following method:

```
938:4C 59 FF
```
ⒼE
```
:0 96 N 8FDG
```

Examining the code at $B200 we find a data decoding routine at $B223 which ends at $B262. This routine EOR's each byte with its previous byte. So, changing one byte here would affect the whole page of data. The routine is listed as follows:

```
B223-  LDY #$FF
B225-  LDA $B0FF,Y  Load previous byte
B228-  EOR $B100,Y  EOR current byte
B22B-  STA $B100,Y  STA to current address
B22E-  DEY
B22F-  CPY #$FF
B231-  BNE $B225
```

To decode the data, we use:

B263:4C 59 FF N B223G

We now have the data decoded and we can start to find the protection routine. With the help of the modified F8 ROM from COMPUTIST, I searched memory for LDA $C08C,X (BD 8C C0) and found the disk checking routine at $182F. The routine is listed below:

```
182F-  LDX $05F8  Load X reg. with $60
1832-  STA $C089,X  Turn drive on
1835-  STA $00
1837-  INC $00  No of retry to read $D5
1839-  BEQ $186E  when $00 larger than FF
183B-  LDA $C08C,X  Read address mark
183E-  BPL $183B
1840-  CMP #$D5
1842-  BNE $1837  Branch not $D5
         .  .  .
1858-  STA $C08C,X  Read track and sect #
185B-  BPL $1858
185D-  ROL
185E-  STA $02
1860-  LDA $C08C,X
1863-  BPL $1860
1865-  AND $02
1867-  DEY
1868-  BPL $1858
186A-  CMP #$07  Check if sect 7 is read
186C-  BEQ $1872  when sector 7 is read
186E-  LDX #$FF
1870-  STX $01  Set error
1872-  LDA $00  Get the max and min #
1874-  CMP $03  of retry to read $D5
1876-  BCC $187A
1878-  STA $03
```

Mychess II

```
187A-  CMP $04
187C-  BCS $1880
187E-  STA $04
1880-  LDA $1FF1    Track counter
1883-  CMP #$11     Check # of tracks read
1885-  BNE $1817    Branch to track seeker
1887-  LDA $03      Compare the max and min
1889-  SEC          # of retry to read $D5
188A-  SBC $04      (the difference should
188C-  CMP #$20     be smaller than $20)
188E-  BCC $1894
1890-  LDX #$FF
1892-  STX $01      Set error
1894-  LDX $05F8    Load X reg. with $60
1897-  STA $C088,X  Turn drive off
189A-  BIT $01      Check if $01 is #$FF
189C-  BPL $18A1    Branch if it is not $FF
189E-  JMP $8BA9    Jump to reboot routine
18A1-  RTS
```

Examining the code at $1800, I found that if the check failed, a value $FF is stored in memory location $01. Therefore, we can get rid of the protection by replacing the STX $01 commands at $1870 and $1892 with NOP commands.

The Procedure

The main problem remaining is how to re-encrypt the changed data so that when it is decoded it will have our new NOPs. By studying the decode routine, I was able to come up with an encode routine. It is as follows:

```
800-  LDY #$00
802-  LDA $1800,Y
805-  EOR $17FF,Y   EOR pervious byte
808-  STA $1800,Y
80B-  INY
80C-  CPY #$00
80E-  BNE $802
810-  RTS
```

Now we are going to find where the data at $1800 is on the disk. I used a pattern of 5 bytes which is found at $1800 before decoding for searching. With the help of the Core disk scarcher, I found this pattern on track $03 sector $0F.

We we can now deprotect the disk by reading in the sector, decoding it, editing the data, recoding the data and writing it back to the disk. Since there are a lot of changes, it is difficult to use Super IOB. To make it easier, I wrote a program to do this job. Its disassembly is listed below:

```
800-  LDA #$00    Set IOB to read track 3
802-  STA $B7EB    sector F to $1800
805-  STA $B7F0
808-  LDA #$03    Track 3
80A-  STA $B7EC
80D-  LDA #$0F    Sector F
80F-  STA $B7ED
812-  LDA #$18    Read to $1800
814-  STA $B7F1
817-  LDA #$01    Set read mode
819-  STA $B7F4
81C-  LDX #$60
81E-  JSR $03E3    Set A and Y registers
821-  JSR $03D9    Call RWTS
824-  JSR $0844    Decode data
827-  LDA #$EA    Edit data
829-  STA $1870
82C-  STA $1871
82F-  STA $1892
832-  STA $1893
835-  JSR $0855    Code data
838-  LDA #$02    Set read mode
83A-  STA $B7F4
83D-  JSR $03E3    Set A and Y registers
840-  JSR $03D9    Call RWTS
843-  RTS
844-  LDY #$FF    Decode data routine
846-  LDA $17FF,Y
849-  EOR $1800,Y
84C-  STA $1800,Y
84F-  DEY
850-  CPY #$FF
852-  BNE $846
854-  RTS
855-  LDY #$00    Code data routine
857-  LDA $1800,Y
85A-  EOR $17FF,Y
85D-  STA $1800,Y
860-  INY
863-  BNE $857
865-  RTS
```

This program calls the RWTS to read track $03 sector $0F into memory page $18 then decodes the data and edits it. After that, it codes the data and calls the RWTS to write the sector back to the disk.

Step by Step

1) Enter the hexdump at the end of this article and save it with:

BSAVE MYCHESS.FIX, A$800, L$66

2) Use COPYA to copy both sides of the disk.

3) Load the program

BLOAD MYCHESS.FIX

4) Insert the backup disk to drive 1.

5) Run the program.

CALL 2048

6) Put the original in a safe place and enjoy the backup disk.

Mychess.fix

```
0800: A9 00 8D EB B7 8D F0 B7   $DA53
0808: A9 03 8D EC B7 A9 0F 8D   $2984
0810: ED B7 A9 18 8D F1 B7 A9   $FF91
0818: 01 8D F4 B7 A2 60 20 E3   $3186
0820: 03 20 D9 03 20 44 08 A9   $F401
0828: EA 8D 70 18 8D 71 18 8D   $6268
0830: 92 18 8D 93 18 20 55 08   $8C69
0838: A9 02 8D F4 B7 20 E3 03   $53BD
0840: 20 D9 03 60 A0 FF B9 FF   $4D23
0848: 17 59 00 18 99 00 18 88   $6F89

0850: C0 FF D0 F2 60 A0 00 B9   $9DA5
0858: 00 18 59 FF 17 99 00 18   $AA60
0860: C8 C0 00 D0 F2 60         $93D6
```

softkey for...

Raster

by Glenn Jones

Requirements:
48K APPLE][and up
A formatted disk with no HELLO program
RESET into monitor and save lower RAM
 or a RAM card

When I received my last issue of COMPUTIST I noticed that Raster Blaster was on the most wanted list. This prompted me to complete a softkey for this electronic pinball game that I have been working on/off for the last two years. If you just wish to make a BRUNable backup of this game then proceed to the cookbook part of the softkey at the end of this article.

Raster Blaster was the first game I purchased after getting my computer over 2 1/2 years ago. A few weeks later I went to play the game and *surprise*, it would not boot. Several tries later the disk finally proceeded to boot up properly. I think it's time to make a backup.

Not knowing much about copy protection, I tried first to make a copy with COPYA from the DOS 3.3 system master but all I got was an "UNABLE TO READ" error message. Turning next to TRAX (a disk inspection program from the Bag of Tricks diskette), I inspected the tracks on the disk. The message "UNABLE TO INTERPRET DATA" appeared on the screen track after track. This usually means that the disk uses some bizzare format like half tracks and the TRAX program cannot decipher this type of format. Based on this information I decided to try several of the bit-copiers that were available. Needless to say I still could not get a working backup using the supplied parameters and varying the speed of my disk drives.

The Discovery

Upon discovering COMPUTIST sometime later and following techniques suggested in it, I tried to boot code trace Raster Blaster. Having a limited knowledge of machine language at the time, I quickly got lost trying to trace through code conversions, indirect jumps through zero page locations that changed everytime you looked at them and direct use of the disk drive.

Finally, after long hours at my Apple, I gave up and only played Raster Blaster on an infrequent basis. If it was not for COMPUTIST No. 19 which contained two the articles "Double Your ROM Space" and "Towards a Better F8 ROM," I would still be using my original disk. After making the modifications outlined in the two articles I once more attempted to make a backup copy of Raster Blaster.

First of all, Raster Blaster is a single load program that does not access the disk for any nibble count checks or load any information after it has finished booting. To see just how much memory Raster Blaster needed to use, I first filled the memory with a constant value (I choose "11") from $800 to $BFFF. This is done by turning the computer on, pressing reset immediately, entering the monitor with CALL-151 and typing the following:

```
800:11
801<800.BFFFM
```

Next, I inserted the original disk and typed:

```
C600G
```

I choose to interrupt the program when it was asking for easy or hard play. Displaying memory showed consistent 11's after $A800 and references to page's 2 & 3 throughout memory. This indicated to me that the program lived from $200 to $A800. Since DOS occupies $9600 to $BFFF I could not just boot DOS and save the upper memory therefore, my original solution was to save the code in the following three pieces:

1) lower memory $0000 to $08FF

2) $0900 to $95FF

3) high memory $9600 to $A800

By using a DOS mover program to relocate DOS to my RAM card I could then BLOAD all the pieces back in at a higher memory location and have a small routine relocate the code to its original spot. At the time this worked fine but it resulted in file that was 176 sectors long and it required a RAM card for the DOS.

When I saw Raster Blaster on the COMPUTIST No. 32 most wanted list, I decided to go over my softkey and see if I could not improve on it. The following notes have lead to Raster Blaster being a BRUNable file of 129 sectors. After the program has fully loaded itself into memory using routines located in text page 1 ($0400 to $07FF), it proceeds to execute a fancy hi-res introduction title page. The game then draws (moves) the main game board and goes into a demonstration loop until a button on a joystick/paddle is pressed. At this point the game asks for ether easy or hard play. This is the point where I determined it best to interupt the game as it would make a good place to restart it.

After hitting CTRL-RESET, I pressed the "S" key to save the lower memory. Starting to browse at $2000 where the code is relocated to, I did not find any rational code until $2200. This code normally would be found in the input buffer which is a good hiding place for the code as it is wiped out when anything is typed at the keyboard. There is also similar code at $2300 as well. Any rational code on pages 4, 5, 6 and 7 deals with direct access of the disk drive and it is no longer needed. At this point I booted up a DOS disk with no HELLO program and saved pages 2 & 3.

BSAVE RB.LOWER,A$2200,L$1FF

I started to inspect the higher regions of memory after rebooting Raster Blaster and again resetting into the monitor at the same point. It appeared as if this code was responsible for copying bytes from high memory to high-res page 1 which would mean that this was the title page routine.

Before going further there were a few things to consider:

A) I wanted the whole program to load in using DOS at its normal location.

B) Pages 2 & 3 would have to be restored to their normal location as well.

After inspecting $800 - $9FF it appeared that this memory would not be needed as well.

Through subsequent trial and error, I decided that the last of the important information ended at $85FF so I BLOADed RB.LOWER at $8600. I also found it necessary to zero out page zero and to reset the stack pointer to $FF. The following is the relocation routine that I put at $8800.

```
8800 - LDY #$00     Initialize counter
8802 - LDA #$00     Store a "0"
8804 - STA $0000,Y  in the zero page
```

Blaster

```
8807 - LDA $8600,Y  Get a byte from page 86
880A - STA $0200,Y  and put it on page 2
880D - LDA $8700,Y  Get a byte from page 87
8810 - STA $0300,Y  and put it on page 3
8813 - INY          Increment counter
8814 - BNE $8802    Loop until 0
8816 - LDX #$FF     Set stack pointer
8818 - TXS          to $1FF
8819 - BIT $C057    Display the hi-res scrn
881C - BIT $C052    Entire scrn as graphics
881F - BIT $C050    Graphics on
8822 - JMP $6770    Jump to "Hard or Easy"
```

The jump instruction at the end came through searching for references to the game button ($C061). One reference at $6303 checks for a key or button press. The next reference is at $6760. Back-tracking from this point I found what appeared to be the start of the game at $672D. Locations are zeroed out and several subroutines are called, one of which waits for a button press before it returns to the main stream. This is the demonstration loop. The next bit of code stores a 0 or a 1 at location $672B for EASY or HARD mode. This is where the program has to be restarted from. The last thing to do to make the file BRUNable was to set a jump at $09FD to my relocation routine at $8800.

9FD:4C 00 88

Finally, one APT you now have available before you save the game to disk is to change location $680D which holds the number of balls you are allowed to have for each game.

**BSAVE RASTER BLASTER,
A$9FD,L$7E28**

Step by Step

Saving lower RAM: Follow procedure 1 if you have made the modifications from COMPUTIST No. 19. Follow procedure #2 if you have a RAM card.

Procedure 1

1) Switch to modified ROM while booting Raster Blaster.

2) Wait till the title page ends and press a button on joystick/paddle.

3) Hit CTRL-Reset.

4) Hit "S" so that your modified ROM will save of pages 0 to 8 into $2000 through $2800.

5) Insert a slave disk and boot it

C600G

6) Save this lower memory

BSAVE RB.LOWER,A$2200,L$1FF

7) Reboot Raster Blaster.

8) Wait till title page ends and press a button on joystick or paddle.

9) Hit CTRL-Reset.

10) Hit "M" so that your modified ROM will put you into the monitor.

11) Insert slave disk and type boot it

C600G

12) Enter the monitor

CALL -151

13) Re-load the lower portion of Raster Blaster

BLOAD RB.LOWER,A$8600

14) Key in the following hexdump:

```
8800: A0 00 A9 00 99 00 00 B9    $6B0C
8808: 00 86 99 00 02 B9 00 87    $4297
8810: 99 00 03 C8 D0 EC A2 FF    $47C6
8818: 9A 2C 57 C0 2C 52 C0 2C    $39F8
8820: 50 C0 4C 70 67             $F5F8
```

15) Modify the program to jump to our routine first

9FD:4C 00 88

16) Save the resulting code:

**BSAVE RASTER BLASTER,
A$9FD,L$7E28**

Procedure 2

1) Turn your computer on.

2) Hit CTRL-Reset.

3) Enter the monitor and prepare your RAM card.

**CALL -151
C081 N C081
D000<D000.FFFFM
FFFC:00 B0
C083 N C083**

4) Type in the following hexdump

**B000:A0 00 B9 00 02 99 00 22
B008:B9 00 03 99 00 23 C8 D0
B010:F1 4C 59 FF**

5) Insert Raster Blaster.

6) Boot the disk

C600G

7) Wait till the title page ends and press a button on the joystick or paddle.

8) Hit CTRL-Reset.

9) Insert your slave disk with no HELLO program and boot it

C600G

10) Save the lower portion of Raster Blaster

**BSAVE RB.LC.LOWER,
A$2200,L$1FF**

11) Enter the monitor and change your RAM card to just jump to the monitor on Reset.

**CALL -151
C081 N C081
FFFC:59 FC
C083 N C083**

12) Insert your Raster Blaster and boot it

C600G

13) Wait until the title page ends and press a button on the joystick or paddle.

14) Hit CTRL-Reset.

15) Insert your slave disk with no HELLO program and boot it

C600G

16) Enter the monitor and load the lower portion of Raster Blaster

**CALL -151
BLOAD RB.LC.LOWER,A$8600**

17) Type in the hexdump from step 14 of procedure 1.

18) Modify the program to jump to our routine first

9FD:4C 00 88

19) Save the resulting code

**BSAVE RASTER BLASTER,
A$9FD,L$7E28**

The Games of 1986

by Jeff Hurlburt

At the beginning of 1986 many software sages predicted the year would be a notably dry one for Apple gamers. With the //gs looming on the horizon, so the reasoning went, vendors would be reluctant to introduce products which might have compatibility problems; and other projects would go into a holding pattern as developers waited to incorporate expected new features. While it is clear that such factors have impacted Apple game production, nothing like the anticipated drought has materialized.

Of course, many products were already well along the pipeline, destined for release no matter what. Perhaps more important, the installed base of old II's is truly awesome, receiving additional support from Commodore's 64 and 128 models. (Several vendors routinely translate games designed for the C64 to an Apple II format, increasingly offering both versions in the same package.) The resulting market has attracted an ample supply of new products.

This month center stage belongs to new arcade, strategy, and simulation releases. Next time, Part II will focus upon the new adventures.

Since my goal is to help you squeeze the most bang from your gaming dollar, each review aims to provide the maximum information in the space available. Compatibility and peripherals requirements plus the number of players (if multi-player) for which the game is designed are shown 'up front'. In the descriptions I try to give you a 'feel' for the game, what it does, and how it works. Critiques and ratings wrap things up.

Comments and ratings reflect my evaluation based upon playing a game and, usually, watching as others play it. Each game is rated in five areas and "overall" on a ten-point scale: 10=Superior (rarely awarded; indicates a breakthrough of some kind), 09=Excellent, 07=Good, 05=Fair, 03=Poor, 01=Rotten.

Graphics (GRFX) is the first area rated. It concerns quality of artwork, clarity, impact, smoothness, speed, and realism. Good **Support Materials** (S.M.) include clear, thorough directions for play. In some cases attractiveness, tutorial value, or effectiveness in creating 'atmosphere' may be important. **Playability** (PLAY) relates to how much extraneous activity is required to play the game. Good command parsing (in an adventure), rapid "save" and "restore" functions, efficient menus, smooth controls, and readily available "Help" screens are features which enhance playability.

Difficulty (DIFF) is self-explanatory for single-player games. For others it relates to how much effort is required to achieve a decent level of play. Unlike other areas, the rating here simply reflects "level of". An "8" is "very tough", which may or may not equal "very good". High **Interest** (INTR) games are good at attracting and holding player attention. Typically, these are the adventures you can't wait to continue and the arcades you play, and replay, for hours at a time. The **Overall** (GAME) rating amounts to a summary of player reaction(s) during tryouts. For educationally-oriented games, a second summary rating (/ED)

of **educational value** is included.

It's a good bet that you will want to obtain several of the games showcased, so publisher data and price are shown towards the end of each review. (Occasionally a product is available ONLY from the publisher.) Prices, of course, are "list" and so represent the worst case layout. With a little careful shopping, discounts of around 40% are not difficult to find.

Fight Night

(Sports Simulation for one or two players)
Compatibility: 64K Apple II series
Requirements: One disk drive, joystick

OK, so the kid's quick and his legs won't turn to rubber after a few tough rounds. The question is, do you have the managing moxie to make him the next champ?! Anticipating your reply, Accolade is ready to give you a shot at the big one. Starting with 'building' one or more

challengers, *Fight Night* takes you through training, sparring matches, and finally the championship tournament.

Your road to the bigtime begins with building one or more boxers via the game's "Construction" utility. Here you select body parts to match desired attributes such as endurance and quickness. Additional "statistics" are then adjusted to fine-tune your boxer in the areas of hitting power to/resistance for blows to head and body. When everything looks just right the new fighter is saved to disk. (The same utility allows you to edit existing fighters and even define a boxer for computer control.)

Fight Night permits a total of eight boxing movements (four stick positions with button up or down) including blows to head and body, fakes, guards, and forward/backward positioning. "Training" drills you in these moves with a punching bag target with the emphasis on rhythm. In "Sparring" you practice moves and timing against another fighter. Once you feel ready for the 'real thing', matches are arranged in individual bouts with set, computer-controlled contenders like the "British Bulldog", or against selected boxers (perhaps some from another player's stable) in a small round robin tournament. During a bout (three three-minute rounds), the round, time remaining, and cumulative scores are displayed along with bar graph K.O. susceptibility indicators (so you know when to go in for the kill). If neither boxer is K.O.'d, the decision goes to the high scorer. The winner of a tournament becomes the "Champ".

Offering perhaps the finest, most comprehensive control of any one-on-one fighting simulation, *Fight Night* closes the loop by providing ample cues as well. Your opponent's movements actually do tell you something about what he is likely to do. Wide-ranging speed controls permit setting the action to a comfortable pace. The only notable weakness is that, for player vs. player matchups, one player must use keyboard controls instead of a second joystick. Clever and surprisingly addictive, *Fight Night* packs the entertainment punch you've been waiting for in a fight simulation.

Available from: Accolade, 20863 Stevens Creek Blvd., B-5/E, Cupertino, CA 95014. (408) 436-0900. $34.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
07	06	06	07	07	07

Chessmaster 2000

(Chess Player)
(for one or two players)
Compatibility: 64K Apple II series
Requirements: One disk drive; printer, SpeeDemon optional

Colossus Chess IV

(Chess Player)
(for one or two players)

Compatibility: 48K Apple II series
*Requirements:* One disk drive; 64K, printer, SpeeDemon optional

For reasons not altogether obvious, chess programs appear to arrive in flocks. Certainly there was nothing last year, whereas 1986 brings us three new packages, two of which qualify as serious challengers for the top spot in Apple chess.

Playing the two programs re-enforced the overall impression of quality and evoked just a touch of deja vu. *Chessmaster* looks very much like *Mychess II*, while *Colossus* bears a strong resemblance to *Sargon III* (which just happen to be the other two top notch Apple chess programs). Both programs offer such most-wanted features as solve-mate/problem modes, take back, variable speed replay, side switching, and board rotation. They permit program vs. itself play and will referee games for two human opponents. You can load, save, and delete games and obtain a move printout during or after a contest along with a board diagram. Neither, unfortunately, permits a graphics dump of the screen. (Nor does any other chess program I've seen; but a regrettable oversight even so.)

Colossus offers a black & white-only display in 2-D and, with 64K, very clear 3-D.
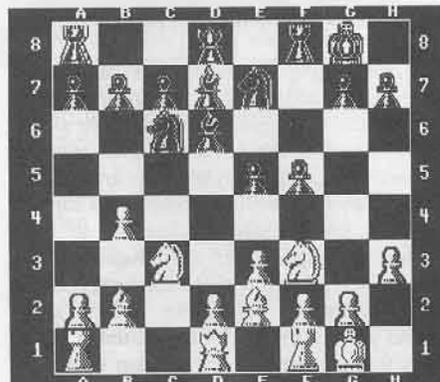
Chessmaster adds an option for eye-catching color (2-D or 3-D) and allows 90 degree rotation in case you want a side view.
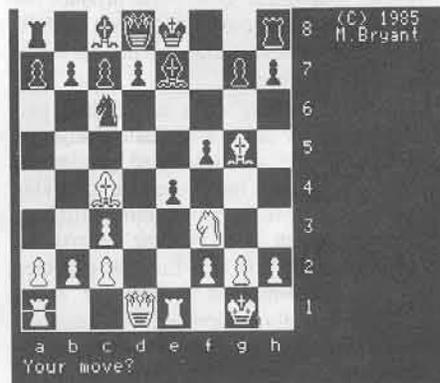
While both programs employ keyboard-only entry for moves and setting up positions, the *Colossus* on-screen cursor approach is more convenient and natural feeling than *Chessmaster*'s purely alpha-numeric system.

The expansion of available playing levels marks a major improvement over earlier designs. *Chessmaster* offers nineteen (five seconds through 37.5 minutes/move); *Colossus* allows direct entry of average move time (00:00 - 59:59). Both offer an "infinite" (no time-out) level for in-depth analysis. Only *Colossus* offers a chess clock (dual presettable), a serious tournament mode (with move and time controls), the "all moves" mode for "five minute chess", etc., and an "equality" mode which matches program move time to yours.

Colossus program specs and functions documentation is vastly superior to *Chessmaster*'s, which just barely describes key functions. On the other hand, *Chessmaster*'s well-designed on-screen menu makes it a much easier to use program. *Chessmaster* comes with a nicely done booklet including a brief tutorial, chess history, and a listing of the 100 classic games (1620-1986) on disk. The *Colossus* library includes 34 computer chess games (1978-1984) plus 19 solve-mate problems. Since *Colossus* files are unprotected, the diskette is a gold mine of ideas for those interested in developing their own chess programs.

Unfortunately, chess programs do not smirk, make snide comments, or look worried. They may, however, achieve a reasonable approximation via "technical information" supplied during a game's progress. Whereas *Chessmaster* displays only search depth, current move considered, and a hint; *Colossus* presents current best line, line currently being considered, assumed opponent move, depth, and evaluation scores. Thus, while well-matched in strength, *Colossus* is by far the more instructive, more entertaining opponent.

After playing several games against each program at two minutes/move I had about decided that *Chessmaster* is just slightly better than *Colossus*. But, when I gave some "expert" and "A" (USCF) rated friends a shot, *Colossus* seemed the tougher adversary. (After two straight losses, one flustered player couldn't leave the computer until he got his revenge.) As to which of the 'big four' is the best opponent, the newcomers appear to have a slight edge, based mainly upon endgame improvements. Otherwise, the human player's style is a major factor. *Chessmaster* has the largest opening 'book' and appears the better positional player; *Colossus* is the better tactician. All the programs are at their best under pressure; but a tendency to over-value material leaves them susceptible to involved and arcane gambit lines. Programs do not risk second-best moves in order to set traps, but then they seldom overlook anything simple and they never give up!

Even among today's more computer-wise players, the programs probably represent a solid "C", possibly "B" level challenge. Instructive, good fun, and consistently tough, *Chessmaster* and *Colossus* belong in the software library of any serious player of the royal game.

Chessmaster 2000 available from: Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403. (415) 571-7171. $39.95.

GRFX	S.M.	PLAY	DIFF	INTR	GAME
Chessmaster:					
08	06	07	5-9	07	07
Colossus:					
07	07	07	5-9	08	08

Colonial Conquest
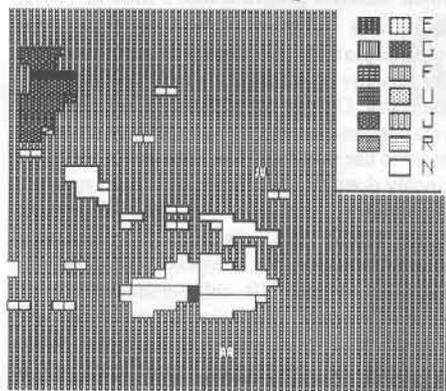
(Geopolitical Conflict Simulation)
(for one to six players)
Compatibility: 64K Apple II series
Requirements: one disk drive; joystick optional

Since the advent of personal computer gaming there have been numerous attempts to capture the excitement, complexities, and back-stabbing intensity of war strategy/diplomacy boardgames. While a few offerings, *Dark Forest* and *Lordlings of Yore* for instance, have proved very entertaining, no computer game has seriously challenged the board game favorites "Risk" and "Diplomacy". Now SSI, long respected for its battlefield simulations, has taken up the gauntlet. *Colonial Conquest* is a multi-player struggle for world domination set in the days when every major power lived by the motto: "Grom or be grommed"!

A game of *Colonial Conquest* always involves six players, one for each major colonial power: England, France, Germany, Russia, U.S.A., and Japan. Any or all positions may be human or computer directed. Your goal is to grom up on enough minor and poorly managed major powers to achieve the preset victory score. Resources include armed forces, treasury, bases, and income producing holdings. The game employs six hi-res maps to cover the world and takes care of all record keeping (except, of course, secret agreements).

Only a few minutes into a game I gradually began to recall why it is that I and other war strategy/diplomacy devotees conduct our geopolitical conflicts via board games. Naturally, fortifications, naval forces, provisions for espionage and subversion, and other such embellishments are welcome. Unfortunately, a nearly unreadable display renders the game, for practical purposes, unplayable. Ownership of an area is signified via color-keyed crosshatch patterns which blur into each other obscuring borders and obliterating naval base/fortification indicators.

Hopefully SSI will recall *Colonial Conquest* perhaps with the aim of issuing an upgraded display version on the //gs.

GRFX	S.M.	PLAY	DIFF	INTR	GAME
04	06	03	06	05	03

Gulf Strike
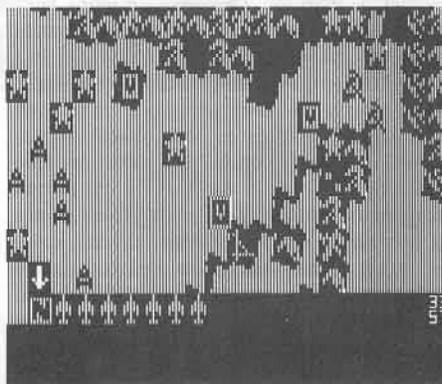
(War Strategy Simulation)
(for one or two players)
Compatibility: 48K Apple II series, Commodore 64/128
Requirements: One disk drive; joystick optional

Demonstrating that wargames can indeed make strange bedfellows, *Gulf Strike* matches Iran and the U.S. (plus assorted minor allies) against Iraq and the U.S.S.R. (plus assorted puppets). Evidently, as 1988 rolls around, the Iranians (annoyed with on-going Soviet support for Iraq) significantly escalate aid to their freedom-fighting brethren in Afghanistan. This doesn't sit at all well with the Russians (i.e. the freedom-fightees), who have already spent several years screwing up their attempt not to botch things worse in Afghanistan than you-know-who did in Vietnam. When the Russkies enter the war on the side of Iraq this naturally provokes the U.S. and concerned Persian Gulf states to intervene in favor of their old pal, Iran. Thus does Avalon Hill contrive to produce the dream match of the decade: a U.S. vs. U.S.S.R. mini-war where neither side can do much real damage to the other.

Gulf Strike action embraces most of Iran, part of Iraq, and other lands immediately adjacent to the Persian Gulf. As a translated boardgame, the computer version incorporates a scrollable map with seventeen terrain features inflicting varying movement costs. Among the ground, air, and sea units you will find the expected detail and differentiation options for which Avalon Hill is justly reknowned. For example, thirteen ground types include airborne, armor, artillary, marine and infantry units in sizes ranging from battalion through division. Each has specific movement, attack, and defense capabilities influenced largely by your choice of formation. Air and naval units are similarly diverse, with nearly thirty aircraft types in various configurations and naval units ranging

from sub groups through carriers. Unless the two-player version is selected, you automatically control the U.S./Iran position and all twenty-one victory point squares spread across the map. During the standard 25-turn game, your mission is to prevent the U.S.S.R./Iraq side from ever holding nine of the squares.

All of this looked pretty juicy; but, perhaps an hour into a game it became clear that something was missing (something, I mean, besides a skilled player). Sure enough, a thorough check of the box confimed my worst fears: Avalon Hill had, indeed, failed to include the other computer! This is an unfortunate oversight considering that only a small part of the map is displayed at any time and that only star and hammer & sickle identifiers appear on the main map to locate units which may be stacked eighteen (18) deep! Lacking the missing computer, you must point to a map square to obtain a "unit symbols display" in the "stack window" and then select a symbol to obtain unit status.

If theatre commanders had to use *Gulf Strike*'s format to fight their wars, the battlefields of the world would be littered with plowshares. Imagine, an attractive, well-documented, fully functioning wargame **That nobody can play!** Right-On, A.H.!

GRFX	S.M.	PLAY	DIFF	INTR	GAME
07	07	02	10	04	NA

Little Computer People Discovery Kit

(Simulation)
Compatibility: 64K Apple II series
Requirements: One disk drive, joystick

Probably, like any clear-thinking computer user, you tend to view the occasional glitch as having a 'perfectly reasonable explanation', (power surges, component failure, even human error). Such, in fact was the predisposition of Activision's Glitch Research Team; that is, until their amazing discovery that little human-like people actually live in computers! We now know that these Little Computer People ("LCP's") produce otherwise inexplicable

glitches largely by accident, though sometimes out of boredom or in a bid for attention.

The above, at least, is the gist of Activision's press release. Well, I'm not just anybody's fool; nor am I the empty-headed, gullible tuft hunter some may imagine. Immediately upon receipt of my LCP research package I booted the "House on a Disk" and waited to see what might transpire. Imagine, if you will, my astonishment when an honest-to-goodness LCP showed up and shyly began to inspect the offered residence! Before long I was filling the water tank and leaving groceries and dog food at the front door. An LCP, it seems, almost invariably owns an LCD. (Indeed, Activision has requested notification if an LCP fails to be accompanied by his dog.)

As suggested in kit materials, LCP's seem to appreciate music and so record albums make great mood boosters. (For a contented LCP, it's important to respond to mood changes.) Even better, my LCP especially enjoys games, enthusiastically playing "Card War", anagrams, and draw poker. Verbal communications are still at the rudimentary stage. However if I type "Please light a fire" or make a similarly simple request, he will often comply.

Each research kit comes with full documentation, a deed (for the house), and suggestions for record keeping. Needless to say, I am thoroughly convinced of the significance of LCP's and the validity of the "House on a Disk" method. (Seeing, after all, is believing!) *Available from:* Activision, P.O. Box 7286, Mountain View, CA 94039. (415) 960-0410. $39.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
08	08	07	04	07	08

Ogre
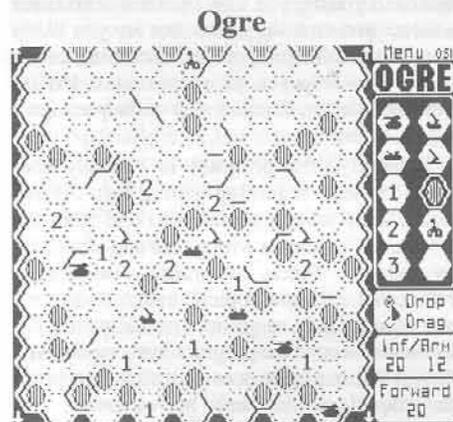
(War Strategy Simulation)
Compatibility: 64K Apple II series
Requirements: One disk drive; joystick or mouse optional

By the mid-21st century effective countermeasures have rendered tactical nukes virtually useless. Taken along with the development of bi-phase carbide ("BPC") armor and atomic engines the inevitable result is that tanks once again rule the battlefield. To be sure, there are tanks and then there are

TANKS. The former, truly awesome by 20th century standards, include the massive "heavy tank", long-reach "missile tank", and fast-moving ground effect vehicle. There is really only one TANK: the cybernetic attack machine. Up to 50 meters in length and wrapped in three meters of BPC plate, once launched these unmanned juggernauts seek only to destroy; and in the 21st century the TANK is called an "OGRE."

The *Ogre* battlefield is a 15x22 scrollable grid of which about half is visible at any time. Rubble and craters restrict or block movement, thus giving each field a unique character. Each contest involves three major phases: loading-in or design of a battlefield, defensive player placement of units and command post, entry of the Ogre and subsequent battle.

Like many wargames, Origin System's *Ogre* recognizes varying levels of victory. In the single-player version your primary concern is to prevent the destruction of your command post (normally placed somewhere near the top of the battlefield, farthest from Ogre's entry point). Only if a sizable chunk of your forces are intact and the Ogre does not escape can you claim complete victory. (A nice distinction, since the Ogre is much more likely to "escape" all over your position, reducing it to rubble.) In a two-player game the Ogre player must destroy everything to claim an unblemished win.

Of course, as defender you are not exactly without resources. Along with the tanks already mentioned, you may deploy long-range (but non-movable) howitzers and BPC armor-suited infantry units. In fact, though "standard game" battlefields, infantry, and armor units are defined for scenarios involving the Mark III and even more powerful Mark V Ogres, the program itself imposes no restrictions upon numbers or placement of any unit or upon battlefield characteristics. Add the option to set the "skill level" of the Ogre you face (in the one-player game), and the tactical possibilities are endless.

The game is supplied with manuals, reference card, and a plastic clip-on radiation exposure badge which, Origin claims, "really works". (I decided not to test it.) Setup is a snap, especially if you start with one of the pre-designed battlefields; and explicit pull-down menus guide you through each phase. With smooth, clear graphics, sound effects, and convenient, uncomplicated controls, "Ogre" amounts to the niftiest tactical wargaming kit ever.
Available from: Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403. (415) 571-7171. $34.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
08	08	07	7-8	08	08

Road Rally U.S.A.

(Educational Simulation)
Compatibility: 64K Apple II series
Requirements: One disk drive

According to the popular wisdom, if you

want to "Discover America", then the way to do it is to hop in your car and go. Evidently, some people at Looking Glass Software agree, even going the extra mile to add a touch of competition pizazz. In *Road Rally U.S.A.* you race the clock as maps and clues guide your travels.

Your objective in this geography learning game is to visit all designated sites before the twelve-minute timer on your dashboard reads "00:00". After you select one of seven major regions (Northwest, South Central, etc.) and difficulty level, a closeup "section map" appears showing your auto, highways, filling stations, city markers, and such location clues as barns, oil wells, and factories. At the lowest difficulty level you will be directed to your first target city by name; otherwise, you get a clue (e.g. "Look for the cowboys here" = Dallas).

From here on the key to success is to locate yourself on the appropriate region map card (only the cards show route numbers and place names) and to make use of the information supplied on the back of the card to interpret destination clues. Once you have figured out where to go and the best route to take, you drive to the target where you will be given the next destination. An occasional car-crunching hazard (falling rocks, etc.) and running out of gas can cost you an auto (you start with three); so you need to pay attention when your radio flashes a hazard warning and keep an eye on your gas gauge. When you reach your last designated target the game is over and a score is awarded based mainly upon time remaining, accuracy of site identification, and cars left. (High scores are not saved.)

Road Rally is a thoroughly enjoyable way to get a feel for U.S. geography besides learning some useful facts. Displayed and on-card maps are accurate and highways approximate actual routes. Well done sound effects and attractive, smooth graphics round out a fine educational experience.
Available from: Bantam Software, 666 Fifth Avenue, New York, NY 10103. (212) 765-6500. $39.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME/ED
08	08	08	05	06	07/07

Law of the West

(Conflict Resolution Simulation)

Compatibility: 64K Apple II series
Requirements: One disk drive, joystick

Well, pardner, there's not much left of the Old West, just a few dusty saloons, boring TV commercials, and (thank goodness) some computer games. In Accolade's *Law of the West* you are the sheriff of Gold Gulch, a rough little place straight out of "Gunsmoke". Here you encounter the expected mix of peaceful townsfolk, drifters, crooks, bullies, and desperados. Some will test your gunhand; some, your nose for crime; and a few will try your romantic expertise. Each, if only subtly, challenges your authority and ability to maintain order.

Law of the West operates very much like the training villages employed by some law enforcement agencies. You find yourself standing near a saloon, bank, railroad station, etc. and someone walks out into the street and begins the encounter. You might get "Howdy Sheriff, I hear you're pretty fast with a gun" from a desperado or "Hi Sheriff, I know a secret; but I won't tell" from the town brat. The program then offers several response options generally ranging in character from wimp to hard nosed. Each conversation normally involves three or four such exchanges, unless someone goes for a gun. In fact, once you get the hang of aiming your '45, you can guarantee that YOU won't ever be shot simply by blowing away everyone you meet. (Except that this approach pretty well ruins your "innocent citizens killed" and "romance" scores.)

Your principal aims in each encounter vary according to the context. For instance, you would like to become friends with the pretty schoolmarm, avoid a shootout with a gunfighter, encourage the town doctor to stop drinking (otherwise he may not be available

when you're wounded), etc. An overriding concern is to prevent crime; so you may have to dig to obtain warning of an impending bank robbery. Each such alert switches you to the scene and you have a chance to chalk up a few "crime prevention" points. If you are not killed first, after fifteen or so encounters the game ends and your performance is scored.

Law of the West is attractive, fun to play, and surprisingly challenging. Evidently, a fair amount of thought went into the conversation sequences (making the game an excellent model for real world human relations training packages). Except that high scores are not saved, the main weakness is the obvious limit on replays (there are only about a dozen characters) once a player becomes adept in the role of sheriff. For best value, you may wish to reserve the game for parties; it's sure to be a hit.

Available from: Accolade, 20863 Stevens Creek Blvd., B-5/E, Cupertino, CA 95014. (408) 436-0900. $34.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
07	06	07	06	07	07

Paul Whitehead Teaches Chess
including
Coffeehouse Chess Monster

(Chess Tutorial/Player)

Compatibility: 64K Apple II series
Requirements: One disk drive; printer optional

Before computers, would-be wood pushers used to learn how to play chess via instruction at the hands of a chess player and/or by studying 'how to' books. Hence the proliferation of such titles as "Learn Chess Fast", "They Laughed When I Sat Down to Play", etc. The advantages of the guru-plus-books approach are obvious, since learning the game involves both learning how the pieces move and a chance to maneuver them against an active opponent. Fortunately

for those long on desire but short on gurus, Paul Whitehead and his pal, the "Coffeehouse Chess Monster" are ready to lead you into the wonderful world of forks, pins, and open files.

Paul Whitehead Teaches Chess (PWTC) is a collection of discussions accompanied by examples and pictorial illustrations. Where an example involves a move sequence, you are able to step through the moves, obtaining additional commentary where warranted. A key feature of the experience is the ready accessibility of a chess playing program (actually a pretty good one) named the "Chess Monster". Here you may experiment with the tutorial position from either side and then return to the tutorial.
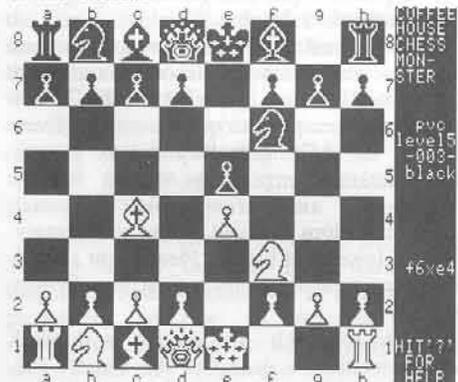
Each tutorial unit is an instructional jewel, but much of this effort would be wasted were it not for careful organization. PWTC makes sure the units and sub-units are arranged so that, as you move through the program, everything makes sense. Beginning with "Main Topics" ("Rules", "Tactics for Each Piece", "Opening Principals", and "The Endgame") all you need do is select topics and sub-topics according to your own interests. You may or may not choose to immerse yourself in, for example, knight tactics. If you do, you won't suddenly encounter a discussion on bishop pins; nor are you likely to miss something important concerning knight tactics. To help you plan your quest PWTC includes a large, detailed map of the package's topics "tree."

Evidently, the program is virtually fool proof, at least regarding getting lost. At one point, after several switches to and from the "Monster", I took a wrong turn and wound up 'elsewhere'. Since all branches eventually return to the "Main Topics" menu, I was soon unlost and ready to go on. (Actually, it's fun to wander around the program just to see what's there.) The hardcopy documentation, except for the map, is a weakness; but otherwise, the package is first rate. Offering a very flexible, interesting learning environment, Paul Whitehead and the "Coffee House Chess Monster" have teamed up to produce an in-depth, action-packed route to chess mastery.
Available from: Enlightenment, 1240 Sanchez, San Francisco, CA 94114. (415) 641-9131. $49.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME/ED
07	06	07	05	07	06/08
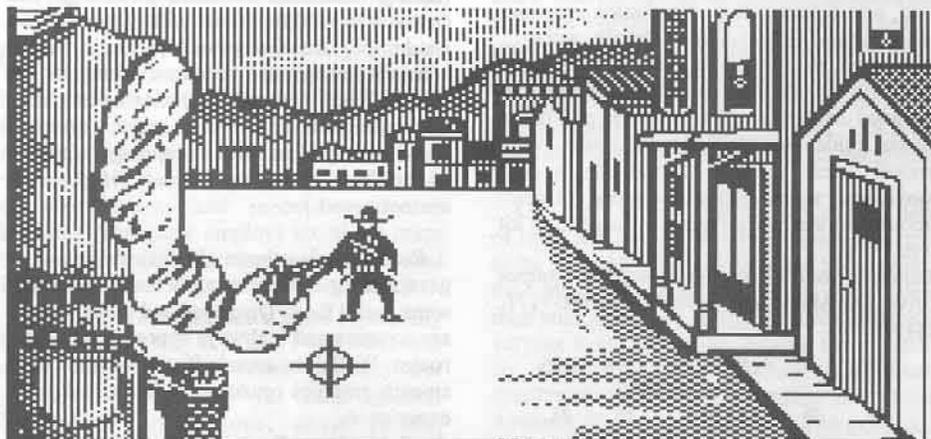
WELL, THIS FIGHT'S COME LOOKIN' FOR YOU.
I'D REALLY RATHER NOT.
NO, PLEASE, NOT THAT!
GOOD! I NEED THE PRACTICE.
ARE YOU SURE YOU WANT TO DO THIS?

Super Boulder Dash
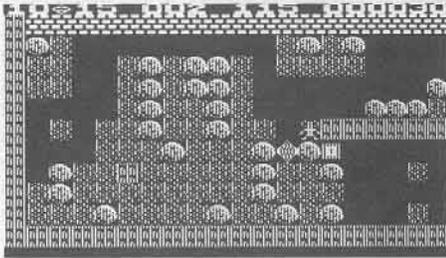
(Arcade)
Compatibility: 64K Apple II series
Requirements: One disk drive, joystick

Once again as the spelunking Rockford you enter the marvelous sub-surface world of precariously balanced boulders, sparkling diamonds, and explosive fireflies. Your primary objective in each of *Boulderdash II*'s sixteen new caves is to reach the exit without being crushed, blown up, or trapped, until finally you reach the surface. If you are unacquainted with magic butterflies and other denizens of the depths, then you will want to start with the original game. The Electronic Arts *Super Boulder Dash* package offers both versions.

To get through a cave you tunnel through the earth, racing against time to collect diamonds which vanish when touched, adding to your score. (Each 500 points gets you an extra life.) When you have the required number of diamonds, the screen flashes once to indicate the exit has opened. Leaving a cave adds points for any time remaining, resets the timer, and viola!, you're in the next cave.

Sounds easy, doesn't it? Well, there are good reasons why every quick-sticking kid on the block isn't a *Boulder Dash* master. First, the earth is full of boulders strung out and stacked in various, often hair-trigger, patterns. Single boulders can be pushed along a tunnel, and some clever undermining can drop whole columns; but a falling boulder or diamond squashes Rockford like a toad. Even more troublesome are the pyrotechnic insects which seem especially attracted to tunnels. Touch any of these and you can scratch one life. They are, however, liable to ignition via dropped boulders and very handy for blowing holes in otherwise impassable walls.

Each cave offers one or more puzzles of varying complexity. Starting off, these are simple and limited in scope (e.g. how to get through a wall to collect some needed diamonds). Later caves produce more involved challenges, such as what to do about a tunnel-filling amoeba. In fact, the principal attraction of *Boulderdash II* is its somewhat more complex, more entertaining brain teasers. Finally, you do not have forever to figure out what's going on. Running out of time, like getting mashed, forces a fresh start in a cave.

Though the continued failure to maintain an on-disk high scores record is disappointing, *Boulder Dash* remains an excellent entertainment value with wide appeal. Time pressure is not severe so success depends as much upon a feeling for the game's physics and

puzzle busting as upon arcade skills. The game is fun, however, largely because the mechanics are fun. Featuring responsive controls, good sound, and superb graphics, *Boulderdash II* elevates an arcade favorite to the status of a computer gaming classic.
Available from: Electronic Arts, 2755 Campus Drive, San Mateo, CA 94403. (415) 571-7171. $39.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
08	07	07	08	07	08

Mind Pursuit

(Knowledge game)
(for one to four players or teams)
Compatibility: 64K Apple II series
Requirements: One disk drive; SpeeDemon optional

When Trivial Pursuit hit the board game market like a bombshell, computer game designers were quick to see their own medium's potential. What, after all, could be more made-for-computer than a data base with a few bells and whistles? The resulting packages, while not altogether unsuccessful, have demonstrated that a worthwhile trivia game is not such a trivial design task. Practically everyone underestimates human memory; and publishers have been shocked by the rapidity with which several hundred juicy questions become 'old favorites' and, thus, obsolete. A good many otherwise decent knowledge games have proved to be remarkably short-lived entertainment values.

Mind Pursuit is DataSoft's new entry in the trivia sweepstakes. Modestly labeled "the ultimate test of information and knowledge", the game features a large question base covering five areas: science and nature, history and geography, TV and films, sports and games, and culture. Questions occupy two diskette sides with at least one supplemental question diskette available from the publisher.

Recognizing that flexibility and variety in format play a major role in otherwise dry question-answer contests, DataSoft offers occasional sound and picture clues plus several game arrangement options. You can play the mainly-text version to a preset point or time limit; or you may choose to race along a Chutes and Ladders-type gameboard; both variations can be played with or without a question timer.

Since each question's category is randomly selected (you press a key to stop the roll of an on-screen die), *Mind Pursuit* mitigates the impact of chance by offering questions in three flavors. Fill-in-the-blankers score 100 points,

multiple choice types are worth 50; and you make just 25 points on true/false questions. Except on the gameboard, wrong answers subtract; so the correct strategy might just balance an over-confident opponent's edge in knowledge. On the gameboard, question 'flavor' availability is limited by point value of accessible squares. Here strategy involves such choices as selecting a 50-pointer to traverse two 25-point squares or taking a shortcut consisting mainly of 100-point squares. Finally, a sixth "Grab Bag" category awards an accurate response with a shot at another question on the same turn.

As claimed, *Mind Pursuit* evidences significantly reduced occurance of question repeats vis-a-vis similar products. Game speed is adequate, though still perhaps half what it should be; and there is no provision for saving and displaying winning scores. Still, of the several trivia/knowledge buffs who tried the game all proclaimed it the best, most enjoyable computer contest of its kind. Smooth running, attractive, and challenging, *Mind Pursuit* is a solid investment in parlor gaming fun.
Available from: DataSoft, 19808 Nordhoff Place, Chatsworth, CA 91311. (818) 886-5922. $29.95 (Supplemental Diskette: $14.95)

GRFX	S.M.	PLAY	DIFF	INTR	GAME
07	06	06	06	07	07

The Eidolon
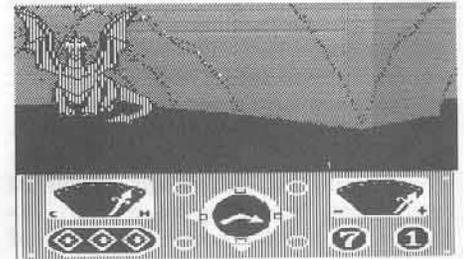
(Maze Arcade-Adventure)
Compatibility: 64K Apple II series
Requirements: One disk drive, joystick optional

There's no telling how many daring explorers have wandered through Argon House in the roughly ninety years it's been deserted. Certainly, finding the hidden basement was a piece of dumb luck. Or, maybe it was fate. Perhaps YOU were meant to unravel the enigma of the beautifully crafted device (a time machine?) resting in the center of what appears to be a laboratory. A yellowed journal (the records of old Josef Argon himself!) only confirms these speculations. This machine, you read, is named *The Eidolon*; but where it goes not even its eccentric creator was prepared to guess!

The word "Eidolon" equates with "phantom", "ignormus", "figment", etc. -- that which is but really isn't or, at least, shouldn't be. All aptly characterize the strange grey labyrinth to which *The Eidolon* transports its passenger. Here your craft moves about freely, though some care must be exercised to avoid power-draining contact with the place's

odd inhabitants. Fortunately, the controls are easily mastered and responsive. Better yet, the maze's physics supply energy and some valuable new capabilities via variously colored "fireballs" and jewels which *The Eidolon* can capture as well as launch.

Red, the journal informs you, repels most creatures, sometimes transforming one into a jewel; yellow recharges your power reserves; green changes one creature into some other; and blue seems to back up time a few ticks. As for the jewels, they are clearly the keys to mastering the dragon guardians of the portals between levels. (Argon theorized there are seven such portals, though the journal ends with only expressions of optimism that a breakthrough was near.) Impressive; but you soon come to realize that this catalogue is but the beginnings of an involved set of bizaare relationships.

As in Activision's *Hacker*, Epyx's *The Eidolon* leaves much about how things operate to be discovered by the player-- a nice ploy when it works, and in this game it works very well. At the start you know which instruments indicate proximity to a portal, heading, active fireballs and jewels, elapsed time, and power level. You are also aware that if power reaches zero or time runs out, poof!, you wind up back in the lab. The journal neglects to mention that if your return is timer-prompted you must reenter at the beginning level; whereas a no-power return allows reentry at the level vacated. (Valuable information, since there is no game save option.)

Every once in a while you come across a game so refreshingly different, yet so wierdly alien that doubts may arise as to its terrestrial origins. *The Bilestoad* and *Ankh* are examples; an elite group now joined by *The Eidolon*. Despite temporal constraints, pacing is liesurely, so there is plenty of time to explore a very interesting, increasingly convoluted environment. Fascinating in concept, *The Eidolon* will transport you to many hours of mind-blowing fun.

Available from: EPYX, 1043 Kiel Court, Sunnyvale, CA 94089. (408) 745-0700. $39.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
08	07	07	07	09	08

The Great International Paper Airplane Construction Kit
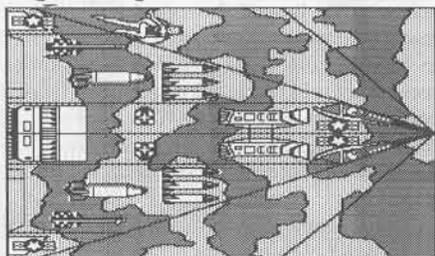
(Design/Construction Activity)
Compatibility: 64K Apple II series
Requirements: One disk drive, dot-matrix printer; second drive, hard disk optional

Beyond question, the design, production, and flight of paper airplanes ranks high among man's more rewarding liesure time pursuits. Here creativity and craftsmanship meet their ultimate challenge and any test launch may signal dramatic breakthroughs. (Will the Zarf B's radical new wing configuration give it the lift and stability to surpass the Zarf A's hallflight record? How will it handle air conditioner vent turbulence?, etc.). Recognizing the obvious potential for computer-aided design and manufacturing in such a high-tech field, Simon & Schuster has introduced a CAD/CAM package guarenteed to revolutionize your paper airplane development capabilities.

The Great International Paper Airplane Construction Kit ("GIPACK") consists of an 80-page large-format manual, reference card, and diskette. After describing program operation and offering a few helpful hints, the bulk of the manual is devoted to the thirteen planes available on the diskette. There is a photograph of each plane, a short blurb describing awards won and flight characteristics, and how-to diagrams for cutting and folding. Later on, each of the diskette's hi-res templates is reproduced, both with and without markings. The remaining pages depict available markings (e.g. wingflaps, logos, numbers, etc.).

Since the program's only functions are to access and printout plane templates (standard hi-res picture files), you will need to call upon your favorite graphics utility to modify existing designs, create new ones, and to lift and add markings. (The manual recommends "Mouse Paint".)

GIPACK is a clever idea which just happens to produce paper airplanes that actually perform. The archetypal executive status symbol, yet suitable for children of all ages, it will produce many afternoon's worth of fodder for the active hand and mind.

Available from: Simon & Schuster, 1230 Avenue of the Americas, New York, NY 10020. (212) 333-2882. $29.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME
07	08	06	04	06	07

War or Peace? You Decide!
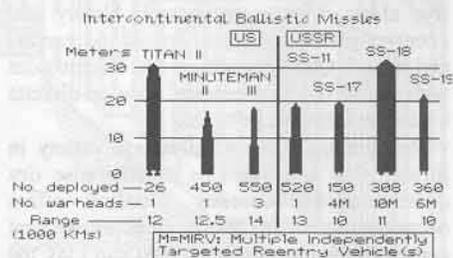
(Crisis Management Simulation)
(for one or more players)
Compatibility: 48K Apple II series
Requirements: One disk drive; second drive optional

Suppose, as the American president, you receive information that the U.S.S.R. may be preparing a 'first strike'; or perhaps you are the President's opposite number in Russia, and it is the U.S.A. which may be preparing a little nuclear surprise. What do you do? Bright Ideas offers these and four other scenarios in its foreign policy studies / crises management package: *War or Peace? You Decide!* (As might be expected in an educational package subtitled "Decision-Making in a Nuclear Age", if you decide upon war there is a very good chance you blew it!) Your objective, basically, is to get your side (and, hopefully, the world) through a crisis in one piece.

War or Peace proved a real surprise; not at all the set of boringly easy exercises I had feared. The manual, which includes a well-designed "Teacher" section, begins with a very readable, lucid, and informative discussion of current major power nuclear force and foreign policy issues. Though rather lengthy, it turns out to be just the right preparation for some very tough decision making.

Except for a few essentially decorative hi-res displays, the game employs a pure text format. Each crisis scenario is preceded by force status listings following which you may choose the role of U.S. President or Chairman of the U.S.S.R. Situation bulletins, intelligence reports, etc. then begin to outline the problem. You receive advice from crisis team members (e.g. Secretaries of State and Defense, C.I.A. Director, Joint Chiefs) and select from among several options. As program documentation observes, you can't be certain that any one source is entirely accurate, the options may not be all you might wish, your orders will not always be carried out exactly as intended; and, of course, you don't have all day! (A countdown timer allows three or four minutes, after which the the computer makes the decision for you.) The effects of your choice will usually lead to another round of reports, perhaps a communication from the other side, advice, and some new options. Normally, (that is, if you and your opposite number can avoid blowing-up each other first) a scenario will involve several such sequences before the crisis is resolved.

The only flaw, surprising in an educational package of this quality, is a bit of sloppiness in diskette handling prompts and more flipping and switching than seems justified on a two-drive system. Still, I found *War or Peace?* a remarkably absorbing, entertaining venture; as did others who gave it a try. The scenarios are well-planned and realistic, with just enough built-in randomness so that one does well not to take the other side's responses for granted in any replays. (Even so, invading Cuba without my direct order was a bit much.) An excellent learning experience which just happens to be the perfect ice-breaker for your next get-together, *War or Peace?* is one decision you won't regret.

Available from: Bright Ideas, 52 Exeter, Portland, MN 04102-2839. $49.95

GRFX	S.M.	PLAY	DIFF	INTR	GAME/ED
04	08	07	07	08	07/08

taking the wiz out of...

Wizardry

by Rob Hall

Requirements:
A Wizardry program
A good sector editor
A scenario disk with six characters
A blank disk to copy your scenario disk

Most veteran wizardry players know that, in "Proving Grounds", if your Bishop successfully "Identifies item number "9", he'll receive 100,000,000 experience points. Also, if he successfully "Identifies "S", the character below the bishop will receive 100,000,000 experience points. And by "Identifing "J", he gives the character below him 100,000,000 gold pieces. This technique allows you to create almost unbeatable characters in "Proving Grounds" and when you transfer your characters to "Knight of Diamonds", they keep their experience and powers. But just when you think you've got all the answers Sir Tech throws you a curve.

Yes, this technique doesn't work with "Legacy of Llylgamyn" and you can't just transfer your characters from a previous scenario, you must perform a "rite of passage". This "rite of passage" creates new characters who are descendants of your characters, but have minimal experience and power. Are all those months, even years, of wandering through dungeons and tunnels, "lost forever"?? Not if you grab your handy sector editor and read on.

The Scenario Disk

In all the Wizardry games, the scenario disk is the key to your adventure. On this disk is stored all the information about the maze, the monsters and your characters. Your character information is usually located on track $1A, sector $01 through track $1B, sector $0F (track $1D for "Proving Grounds"). This information is $CF bytes long and is formatted as follows:

```
bytes   information
-------------------------------------
00      Number of letters in name
01-0F   Name (15 letters maximum)
10      Number of letters in password
11-1F   Password (15 letters maximum)
20      ($01) character is "on expedition"
        ($00) character is available
22      Race; see Table I
```

24	Class; see Table II
26-27	Age; see Table III
28	Life; ($00) character is OK,
	($05) indicates character is dead,
	($07) indicates character is lost
2A	Alignment; see Table IV
2C-2F	Characteristics:
	($52 4A 52 4A is perfect)
34-38	Gold; see Table V
3A	# of pieces of equipment (8 max)
3C-7B	Equipment information; see Table VI
7C-80	Experience (like gold; see Table V)
82	Last level of character
84	Current level of character
86-87	# of hit points character has
88-89	max # of hit points you can possess
8A-91	Type of spells; see Table VII
92	# of level 1 mage spells can cast
94	2
96	3
98	4
9A	5
9C	6
9E	7
A0	# of level 1 priest spells can cast
A2	2
A4	3
A6	4
A8	5
AA	6
AC	7
AE-AF	Last armor class
B0-B1	Current armor class
B6-BF	Measures effect of weapons & items
CE-CF	Indicates honors

Now that you know how your character is formatted, it's time to turn your wimp into a wizard.

Using the Sector Editor

Before creating your new character, you should make backup copy of your scenario disk. The scenario disk is not copy protected so you can use COPYA or any fast copy program to duplicate it. To get the most out of this procedure, I recommend that your scenario disk contains at least six characters you would like to strengthen.

For a start, read track $1A or $1B ($1D for proving grounds) to do some investigation work. The location of your character within a sector or track will vary depending on the sequence that your characters were created on the disk. However, it is easy to find the beginning of your character's information by looking for the flashing ASCII representation

of your character's name. This will be the first line of information, bytes $01 to $0F (remember that byte $00 is the number of letters in your character's name).

If you're lucky, your character's information will start at the start of a sector. However, it is more likely that your character's information will start in the middle of a sector. This is because each character's information block is only $CF bytes long and new characters are added immediately after old characters. If your character does not start at byte 0 of a sector, count bytes starting with the byte to the left of your character's name (starting at 0) and use the chart to find the feature you want changed.

Table I

01	Human
02	Elf
03	Dwarf
04	Gnome
05	Hobbit

Table II

00	Fighter
01	Mage
02	Priest
03	Thief
04	Bishop
05	Samurai
06	Lord
07	Ninja

Unfortunately, the character's information is usually not ordered in sequential sectors (starting with sector $1 and finishing in sector $2). It is usually in reverse order (starting with sector $F and finishing in sector $E) or may jump from the beginning of a track to the end. Careful attention to the length of each character's information block and a little trial and error can solve this problem.

The Little Fix

Along with the literature Sir Tech provides with its program, a note warns of tampering with the characters: "While it may seem appealing to use these products (cheat programs), we urge you not to succumb to the temptation. It took more than four years of careful adjustment to properly balance Wizardry." Well, it took me all of last summer

Table III

Age is calculated using bytes $26 and $27 under the following formula:
   [($26) / $34] + [($27) X 5]
= years of character (decimal)
   Therefore, if $26 and $27 are $D0 and $04 respectively; your character's age is 24,

---

to get through "Proving Grounds" even using the bishop trick. If these games are really balanced, those dungeon monsters sure weigh a lot. In fact, I was quite surprised at the number of items I never came across when I played Proving Grounds and Knight of Diamonds the normal and laborious way. Wouldn't it be nice to use a "great mage wand" for a change?

Anyway, for those afraid to upset the proper balance of a Wizardry game, may I suggest the "little fix":

● change the gold bytes ($34-38) to "00 00 00 01 00" ● change the experience bytes ($7C-80) to "00 00 00 01 00" ● take your character to the "Adventurer's Inn" and have him "rest" there several times.

This method will allow your character to progress normally, learning and acquiring normal spells and powers. It will also give him enough hit points to survive most levels. In addition, you now have the money to buy the items you want from Boltac's Trading Post.

Table IV

01   Good
02   Neutral
03   Evil

Table V

Gold is calculated by bytes $34 to $38 using the following formula:

byte $34, 1st nybble: 16's place
              2nd nybble: 1's place
byte $35, 1st nybble: 4,096's place
              2nd nybble: 256's place
byte $36, 1st nybble: 160,000's place
              2nd nybble: 10,000's place
byte $37, 1st nybble: 40,960,000's place
              2nd nybble: 2,560,000's place
byte $38, 1st nybble: 1,600,000,000's place
              2nd nybble: 100,000,000's place
Experience is calculated in the same way by bytes $7C to $80.

The Super Lord

And for those who have no mercy for monsters and the creatures of the dungeon, I suggest the creation of a "super lord":
● change the class byte ($24) to "06" ● change age bytes ($26-27) to "00 04" ● insure life byte ($28) is "00" ● insure alignment byte ($2A) is "01" ● change the characteristic bytes ($2C-2F) to "52 4A 52 4A"   - change the gold bytes ($34-38) to "00 00 00 00 01" ● change equipment possession byte ($3A) to "08" ● equip the character with eight items as explained in Table VI

Table VI

Each piece of equipment is represented by an eight byte string. The string contains the following:

```
byte 1:  status of possession ($00 = unequiped, $01 = equiped)
byte 2:  not used
byte 3:  not used
byte 4:  not used
byte 5:  status of equipment ($00 = unidentified, $01 = identified)
byte 6:  not used
byte 7 and 8:  actual equipment
```

When giving a character a piece of equipment; insure all "not used" bytes are $00, set byte #1 to $00 (unequiped), byte #5 to $01 (identified), and bytes #7-#8 as follows:

Proving Grounds Of The Mad Overlord

7/8 Byte	Equipment	7/8 Byte	Equipment	7/8 Byte	Equipment
0100	Long Sword	0200	Short Sword	0300	Anointed Mace
0400	Anointed Flail	0500	Staff	0600	Dagger
0700	Small Shield	0800	Large Shield	0900	Robes
0A00	Leather Armor	0B00	Chain Mail	0C00	Breast Plate
0D00	Plate Mail	0E00	Helm	0F00	Dios Potion
1000	Latumofis Pot	1100	Long Sword + 1	1200	Short Sword + 1
1300	Mace + 1	1400	Staff of Mogref	1500	Scroll/Katino
1600	Leather + 1	1700	Chain Mail + 1	1800	Plate Mail + 1
1900	Shield + 1	1A00	Breast Plate + 1	1B00	Scroll/Badios
1C00	Scroll/Halito	1D00	Long Sword - 1	1E00	Short Sword - 1
1F00	Mace - 1	2000	Staff + 2	2100	Dragon Slayer
2200	Helm + 1	2300	Leather - 1	2400	Chain - 1
2500	Breast Plate - 1	2600	Shield - 1	2700	Jeweled Amulet
2800	Scroll/Badios	2900	Potion of Sopic	2A00	Long Sword + 2
2B00	Short Sword + 2	2C00	Mace + 2	2D00	Scroll/Lomilwa
2E00	Scroll/Dilto	2F00	Copper Gloves	3000	Leather + 2
3100	Chain + 2	3200	Plate Mail + 2	3300	Shield + 2
3400	Helm + 2 (evil)	3500	Potion of Dial	3600	Ring of Porfic
3700	Were Slayer	3800	Mage Masher	3900	Mace Pro Poison
3A00	Staff/Montino	3B00	Blade Cusinart'	3C00	Amulet/Manifo
3D00	Rod of Flame	3E00	Evil Chain + 2	3F00	Neut P-Mail + 2
4000	Evil Shield + 3	4100	Amulet/Makanito	4200	Diadem of Malor
4300	Scroll/Badial	4400	Short Sword - 2	4500	Dagger + 2
4600	Mace -2	4700	Staff - 2	4800	Dagger of Speed
4900	Cursed Robe	4A00	Leather - 2	4B00	Chain - 2
4C00	Breast Plate - 2	4D00	Shield - 2	4E00	Cursed Helmet
4F00	Breast Plate + 2	5000	Silver Gloves	5100	Evil Sword + 3
5200	Evil Ssword + 3	5300	Thieves Dagger	5400	Breast Plate +3
5500	Lords Garb	5600	Murasama Blade	5700	Shuriken

● change the experience bytes ($7C-80) to "00 00 00 00 01" ● change level bytes ($82) and ($84) to "60" ● change status bytes ($86-87) to "00 03" ● change hits bytes ($88-89) to "00 03" ● change spell type bytes ($8A-91) to "FE FF FF FF FF FF 07" ● change spell bytes ($92-AC) to "09 00 09 00 09 00 09 00, etc." ● equip your lord in "camp" or at "Gigamesh's Tavern"   This will give you a "super being" which will defeat all your foes, if you play your "spells" right!

Other Uses

There are quite a few other uses for your sector editor. For instance, Sir Tech will charge you $15.00 to delete a password you have forgotten. Now all you have to do is to insert $00 from bytes $10 to $1F to delete your password or better yet, just look at these bytes to see your password.

If you have a power failure or you accidentally press reset while you're playing Wizardry, your character's record on the disk will be logged as "on expedition". Normally, you would not be able to use your character until you recovered it using the utility recover program. Now you may recover your character on your own by placing a "$00" in byte $20.

If you own Legacy of Llylgamyn and have always had less then twenty characters on your scenerio disk you will be in for a surprise. As you examine the character information sectors, you are likely to find many ghosts of past

7/8 Byte	Equipment	7/8 Byte	Equipment	7/8 Byte	Equipment
5800	Chain Pro Fire	5900	Evil Plate + 3	5A00	Shield + 3
5B00	Ring of Healing	5C00	Ring Pro Undead	5D00	Deadly Ring
5E00	Werdna's Amulet	5F00	Statuette/Bear	6000	Statuette/Frog
6100	Bronze Key	6200	Silver Key	6300	Gold Key
6400	Blue Ribbon				

Knight of Diamonds

Same as "proving grounds" with the following exceptions:

7/8 Byte	Equipment	7/8 Byte	Equipment	7/8 Byte	Equipment
5E00	Rod of Raising	5F00	Amulet of Cover	6000	Robe + 3
6100	Winter Mittens	6200	Nklc. Pro. Magic	6300	Staff of Light
6400	Long Sword + 5	6500	Swd./Swinging	6600	Priest Puncher
6700	Priest's Mace	6800	ShSwd./Swinging	6900	Ring Pro Fire
6A00	Cursed + 1 Plate	6B00	Plate + 5	6C00	Staff of Curing
6D00	Ring of Regen	6E00	Metamorph Ring	6F00	Stone Stone
7000	Dreamer's Stone	7100	Damien Stone	7200	Great Mage Wand
7300	Coin of Power	7400	Stone of Youth	7500	Mind Stone
7600	Stone of Piety	7700	Blarney Stone	7800	Amulet of Skill
7900	Amulet of Skill	7A00	Great Mage Wand	7B00	Coin of Power
7C00	Staff of Gnilda	7D00	Hrathnir	7E00	Kod's Helmet
7F00	Kod's Shield	8000	Kod's Gauntlets	8100	Kod's Armor

Legacy Of Llylgamyn

7/8 Byte	Equipment	7/8 Byte	Equipment	7/8 Byte	Equipment
0104	Haubeck	0204	Breast Plate	0304	Plate Armor
0404	Sallet	0504	Potion of Dios	0604	Latumofis Oil
0704	Short Sword + 1	0804	Broadsword + 1	0904	Mace + 1
0A04	Battle Axe + 1	0B04	Nunchaka	0C04	Dagger + 1
0D04	Katino Scroll	0E04	Cuirass + 1	0F04	Hauberk + 1
1004	Breastplate + 1	1104	Plate Armor + 1	1204	Heater + 1
1304	Bascinet	1404	Iron Gloves	1504	Badios Scroll
1604	Halito Potion	1704	Short Sword - 1	1804	Broadsword -1
1904	Mace - 1	1A04	Dagger - 1	1B04	Battle Axe - 1
1C04	Margauz's Flail	1D04	Bag of Gems	1E04	Wizard's Staff
1F04	Flametougue	2004	Round Shield - 1	2104	Cuirass - 1
2204	Hauberk - 1	2304	Breastplate - 1	2404	Plate Armor - 1
2504	Sallet - 1	2604	Sopic Philtre	2704	Gold Ring
2804	Salamander Ring	2904	Serpent's Tooth	2A04	Short Sword + 2
2B04	Broadsword + 2	2C04	Battle Axe + 2	2D04	Ivory Dagger
2E04	Ebony Dagger	2F04	Amber Dagger	3004	Mace + 2
3104	Mithril Gloves	3204	Dialko Amulet	3304	Cuirass + 2
3404	Heater + 2	3504	Displacer Robes	3604	Hauberk + 2
3704	Breastplate + 2	3804	Plate Armor + 2	3904	Armet
3A04	Wargan robes	3B04	Giant's Club	3C04	Blade Cuisinart
3D04	Shepherd Crook	3E04	Unholy Axe	3F04	Rod of Death
4004	Gem of Exorcism	4104	Bag of Emeralds	4204	Bag of Garnets
4304	Blue Pearl	4404	Ruby Slippers	4504	Necrology Rod
4604	Book of Life	4704	Book of Death	4804	Dragon's Tooth
4904	Trollkin Ring	4A04	Rabbit's Foot	4B04	Thief's Pick
4C04	Book of Demons	4D04	Butterfly Knife	4E04	Gold Tiara
4F04	Mantis Gloves	FF03	Mage's Robe	FE03	Heater Shield
FD03	Round Shield	FC03	Flail	FB03	Dagger
FA03	Battle Axe	F903	Hand Axe	F803	Staff
F703	Mace	F603	Broadsword	F503	Short Sword
F403	Butterfly Knife	F303	Orb of Mhuuzfis	F203	Gold Medallion
F103	Rod of Fire	F003	Holy Water	EF03	Amulet of Air
EE03	Staff of Earth	ED03	Ship in Bottle	EC03	Crystal of Good
EB03	Crystal of Evil	EA03	Neutral Crystal	E903	Orb of Earithin
E803	Broken Item				

Once your characters have the proper equipment, they can "equip" themselves in "camp" or at "Gigamesh's Tavern."

explorers. No one is sure where they came from! These characters are usually "lost" (byte $28 is "$07") and may have "$00" in byte $00 (number of letters in name). To resurrect these explorers, place a "$00" in byte $28 and place a hex number corresponding to the number of letters in the explorer's name in byte $00. Be careful with these characters. I wouldn't trust a character named "6502."

Table VII

The type of spells a character can cast is determined by bytes $8A to $90.

```
            bytes: $8A 8B 8C 8D 8E 8F 90
       all spells: FE FF FF FF FF FF 07
  all mage spells: FE FF 3F 00 00 00 00
 all priest spells: 00 00 C0 FF FF FF 07
```

Each bit in these bytes represents a type of spell. If the bit is on ("1"), that spell is known by the character. Don't forget that in order for a character to cast a spell, he must have "spells left" in that level. The "spells left" are indicated in bytes $92-$AD.

Mage Spells

	byte 8A	byte 8B	byte 8C
bit 0:	not used	0:MOLITO	0:ZILWAN
	1:HALITO	1:MORLIS	1:MASOPIC
	2:MOGREF	2:DALTO	2:HAMAN
	3:KATINO	3:LAHALITO	3:MALOR
	4:DUMAPIC	4:MAMORLIS	4:MAHAMAN
	5:DILTO	5:MAKANITO	5:TILTOWAIT
	6:SOPIC	6:MADALTO	6:pst spell
	7:MAHALITO	7:LAKANITO	7:pst spell

Priest Spells

	byte 8C	byte 8D	byte 8E
bit 0:	not used	0:BADIOS	0:ZILWAN
	1:mage spell	1:MILWA	1:DIALKO
	2:mage spell	2:PORFIC	2:LATUMAPIC
	3:mage spell	3:MATU	3:BAMATU
	4:mage spell	4:CALFO	4:DIAL
	5:mage spell	5:MANIFO	5:BADIAL
	6:KALKI	6:MONTINO	6:LATUMOFIS
	7:DIOS	7:LOMILWA	7:MAPORFIC

	byte 8F	byte 90
bit 0:	DIALMA	0:MABADI
	1:BADIALMA	1:LOKTOFEIT
	2:LITOKAN	2:MALIKTO
	3:KANDI	3:KADORTO
	4:DI	4:not used
	5:BADI	5:not used
	6:LORTO	6:not used
	7:MADI	7:not used

A character with all "spells left" is indicated on the status screen as:

```
MAGE   9/9/9/9/9/9/9
PRIEST 9/9/9/9/9/9/9
```

and we find bytes $92 to $AC are:

```
byte:
 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
value:
 09 00 09 00 09 00 09 00 09 00 09 00 09 00
byte:
 A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC
value:
 09 00 09 00 09 00 09 00 09 00 09 00 09
```

by Jerry Lewis

Requirements:
Deprotected Newsroom (COMPUTIST No. 23)
  or a bit copy of Newsroom
A sector editor with search functions
Blank disks

Print Shop and Newsroom are two programs that are currently enjoying a high level of popularity and the use of either program can be nipped in the bud when your printer or printer card is not supported. After creating a page of output with Newsroom, I discovered that the list of printer interface cards in the setup menu did not include my card. So I did the natural thing; I tried all the cards that were listed.

Several hours later it became obvious that I had two options if I wanted to use Newsroom; buy a new printer card or patch Newsroom with the proper driver routine for my card. Having no use for two printer cards, I opted to patch Newsroom. The information presented here and a little knowledge of assembly language will allow you to patch a print driver into Newsroom, but because of individual print card variables, this is not a cookbook operation.

Some of the following is a little technical, but do not despair. To conserve space, only the results of many hours in front of the monitor and study of program listings is presented. The number of my dead ends and failed attempts is not tallied, but even bad results from trial and error provided information. Fortunately you can not break anything with programming mistakes.

Getting Started

Searching or altering Newsroom requires access to the disk and we are fortunate, as a softkey for Newsroom was previously published in COMPUTIST No. 23. *Do not* use the original disk, but work with only an unprotected backup disk. Making alterations to originals may result in a disk suitable only for reformatting.

My printer card is the predecessor to the Fingerprint II reviewed in COMPUTIST No. 28. The original Fingerprint printer card is seldom found in program setup menus, and with what I learned from studying Newsroom and Print Shop, I have patched several other programs with a print driver. The patch presented here for the original Fingerprint card will work, but is presented primarily as an example.

Driver Characteristics

A print card driver is a short routine that bridges the gap between the graphic program and the printer card. Each card manufacturer and programmer has its own way of doing things, and as a result, the driver routine must be tailored to satisfy the requirements of both the card and the program. There are few absolutes, and each program to be patched has to be studied to determine how to meet both objectives. Even with all the different printer cards, the basic requirements of a driver routine are similar, its the methods of meeting those requirements that differ.

There are three things that all print drivers need to do: (1) turn on the card, (2) send the data, and (3) turn off the card. A driver routine is this simple, but keep in mind there are a couple problems that have to be handled along the way. The first thing to remember is that the printer is a mechanical device and is S-L-O-W compared to the speed of your Apple. The results of this speed disparity is that if you sent the data to the printer with the speed of your Apple, the printed output would contain only a small part of your data.

To ensure that all the data sent is printed, the printer tells the card if it is busy (printing) or not, and by sending data only when the printer is not busy, all the data will be printed. The "busy signal" may be monitored by the ROM on the printer card or the driver may have to do the checking. The determination of printer status usually requires reading a slot memory location until the value returned indicates the printer is ready to receive data and may involve detecting the change in a single bit.

There is another consideration that has to be taken into account by the program, and that is how to handle slot access. Most Apple users install their printer interface card in slot one, but since this may not always be the case, the program must allow for use of a different slot. Slot access not only applies to the slot Input-Output (I/O) addresses, but may include addressing of onboard ROM. As we shall see, various programs use different methods of handling slot access.

Printer Card ROM

There are two memory areas available for use by an onboard printer card ROM: 1) the dedicated 256 bytes occupying $Cx00 to $CxFF (x = slot), or 2) the shared 2K memory area $C800 to $CFFF. If the printer card has onboard ROM using the $Cx00 area, the driver routine will have to know the slot being used to access the ROM. Those cards with ROM that use the 2K memory area must have sole use of that memory area; therefore, the driver routine must first access memory location $CFFF to turn off all slot ROMs before turning on the printer card.

Some printer cards have specific requirements. The cards with ROM may require slot information in a certain register upon entry into the ROM or the card will only accept values of data with the high bit set (values $80 or greater). About the only thing you can count on from program to program is that the data to be printed will arrive at the driver in the accumulator.

Getting the Driver

Before I could patch Newsroom, I needed the driver for my printer card. The documentation that came with my card did not contain a sample driver routine, so I sent a letter to the card manufacturer requesting the information. While waiting for a reply, I figured that since I was already using Print Shop, it was a possible source for my driver routine.

I used CIA's disk search utility, The Tracer, and searched an unprotected copy of Print Shop to locate the driver routines. I took advantage of the fact that many of the "intelligent"

interface cards use the $C800 memory area and searched the Print Shop disk for a reference to $CFFF (this soft switch turns off all peripheral card ROMs using the $C800 area). I located the driver routines and after a bit of study, identified the specific one for the Fingerprint card. I have since found that locating one driver is usually enough to locate all the drivers, as they are normally grouped together.

Using the driver from another program requires that you determine how that program uses the driver. Once you know how the driver routine works in the source program, you then need to determine how the destination program handles data and slot access. With this information, you can then make any needed changes so the driver will operate after being patched into the destination program.

Newsroom Drivers

The handling of the print driver routines in Newsroom is straight forward and it is not difficult to patch with another driver. Newsroom is a file based program and CATALOGing Newsroom resulted in early success. The file DRIVERS.0 looked promising and initial examination with a sector editor showed all the interface card names available in the menu.

I used CIA's Tricky Dick, as it will disassemble the sector being examined, and studied DRIVERS.0. Looking for common factors as a clue to the requirements Newsroom needed in a driver routine, I found the following:
1) The driver for each interface card was located right after the interface card name.
2) A value of $00 separates each interface card name and each driver routine.
3) A hex byte having the value equal to the length of the interface card name, including spaces, precedes the interface card name.
4) The interface card name is stored with the high bit clear (values less than $80).
5) There is a "mystery" byte with the value of $00 or $80 as the first byte of each driver routine.
6) All the drivers are written for slot 1 using absolute addressing, even the slot I/O addresses. That is, I/O is accessed with instructions like STA $C091 instead of STA $C081,X (where the X register would contain the slot number times $10).
7) All driver routines are exited via an RTS instruction.
8) The A, X, and Y registers leave with the same values they had when the driver was called.
9) The "normal" load address is $0000 (strange!) and the length is $055F.

Setup Storage

The next piece of information I needed was how the drivers were accessed. I made another backup copy of Newsroom, and then during the setup menu, told Newsroom the printer was in slot 1 on one copy and selected slot 2 on the other copy. Comparison of the two disks with a disk compare utility identified the sectors

which were not identical and those sectors were then examined to isolate what was different. Repeating the process, one time selecting two different printer cards and another time selecting two different printers, showed me where the configuration information was stored.

The result was that all the configuration information was stored in the file NEWSPRINT.0 which loads at $9000. An examination of the loaded file showed the interface card name begins at $903B, the printer name begins at $906B, and the driver routine is patched in at $9714. The selected slot information is stored starting at $905B. The slot information is stored in the sequence of and in the formats of: $0x, $Cx, and $x0 (x representing the slot number). The driver can load and use any of these slot values if required for operation.

Upon examination of the changed sectors, with a chosen slot other than slot one, all slot ROM access in the driver had been changed from slot one to the selected slot. The file NEWSPRINT.0 did not contain any routines that would make the changes, so back to the catalog. The file NEWSETUP.0 sounded like what I was looking for and does in fact contain the routine. NEWSETUP.0 takes the slot, card and printer selections and patches the memory image of the file NEWSPRINT.0. The driver slot addresses are changed to the selected slot and the driver is patched at $9714 before saving the new setup back to disk as NEWSPRINT.0.

While working with the files NEWSETUP.0 and NEWSPRINT.0 I noticed that the file NEWSPRINT.0 length was not constant. The length of NEWSPRINT.0 is based on the length of the driver added to the end of the file. Jumping ahead a bit, when I finished patching in my driver, I determined that there is a total of $74 bytes available for the driver. The driver will work if the last byte is not beyond $9787 when NEWSPRINT.0 is loaded into memory.

What to Patch?

I now had enough information to start patching Newsroom with my driver routine. There were two options: (1) patch the file NEWSPRINT.0 directly or (2) patch the file DRIVERS.0. Since a patch to NEWSPRINT.0 would be overwritten and lost if the setup was ever changed, I decided to patch my driver routine into the file DRIVERS.0. An additional benefit to patching DRIVERS.0 was that a change in printer slot would be taken care of during configuration.

Deciding to patch the file DRIVERS.0 still leaves two options, (1) replace one of the existing printer card selections or (2) add my card on to the end of the file. Either option will work. To get things going quickly I replaced a card with enough space available for my driver while I studied the other option. The driver I replaced was longer than mine so I filled the space remaining with $EAs (NOP) up to the $00 marker prior to the next driver name.

The first problem with lengthening the file is that the additional length could overwrite program data or be overwritten during program operation. Determining the location of DRIVERS.0 during setup requires running Newsroom and resetting into the monitor during the setup menu. Before booting Newsroom, memory must be filled with a known value in order to tell what memory is occupied by the program.

For example, with the Newsroom disk in the disk drive:

```
CALL-151
FF59G
0300:22 N 0301<0300.BFFEM
6⊡P
```

Select the setup menu
Reset into the monitor.

A search of memory showed that the file DRIVERS.0 was loaded at $5FC3. The fill
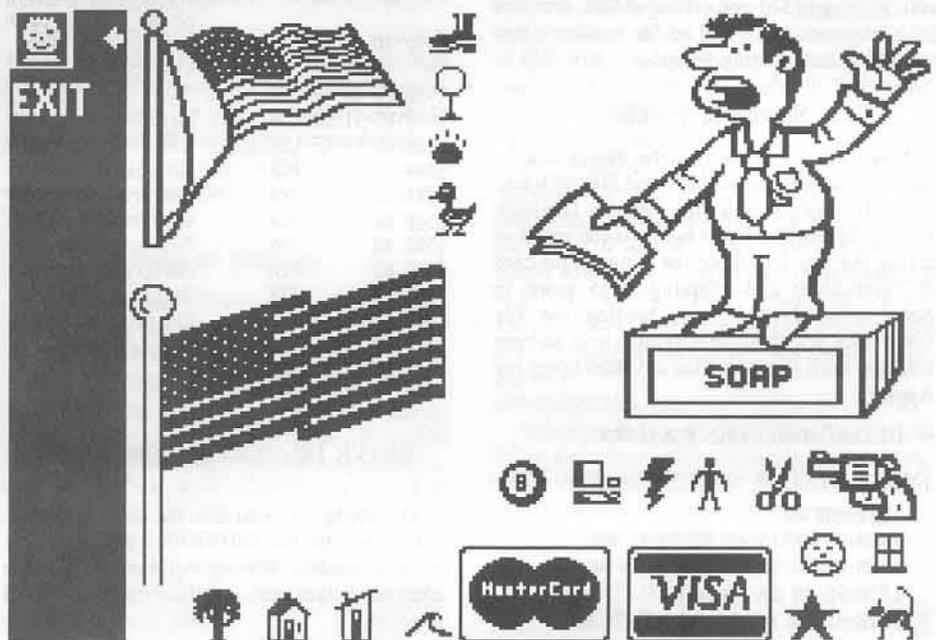
value ($22) started at $6523 and extended to $7700, thus there was plenty of memory area available for lengthening DRIVERS.0. I could be reasonably assured that a longer DRIVERS.0 would not be overwritten or overwrite any other part of the program.

The second hazard to adding my driver on to the file DRIVERS.0 was disk space. Even though Newsroom is file based, some file based programs will still use sector access. If the file DRIVERS.0 could be lengthened without using an additional sector, I would avoid the problem of possibly overwriting program data on the disk. The track/sector list (track $1E, sector $0F) for DRIVERS.0 shows the last sector to be track $1E and sector $09. The last sector appeared to be full, but by adding four bytes (two for the load address and two for the file length) to the length $055F, the actual end of the file DRIVERS.0 is calculated to be byte $63 of the last sector. The three $00 bytes at the end, starting at $61 and ending at $63, are used by Newsroom as an end of file marker when the file is read during setup.

Sewing a Patch

Now that we know that the driver can be installed, all we have to do is load DRIVERS.0, make the changes and then save the file back to disk. The driver listed below is the result of taking the one found for the Fingerprint card in Print Shop and adapting it to work in Newsroom. By the way, loading the file DRIVERS.0 without specifying a load address will not work (loading files at $0000 upset my Apple).

BLOAD DRIVERS.0,A$1000

Type in the driver for the Fingerprint card:

```
155E:0B 46
1560:69 6E 67 65 72 70 72 69
1568:6E 74 00 80 48 8D 79 06
1570:98 48 8A 48 AD FF CF AC
1578:5D 90 8D 90 C0 AD 79 06
```

```
1580:20 00 C8 8D 91 C0 8D FF
1588:CF 68 AA 68 A8 68 60 00
1590:00 00
```

Check your typing against the following:

155E:0B	???	Length of card name
155F through 1569		ASCII for "Fingerprint"
156A:00	BRK	Data separation mark
156B:80	???	Mystery byte
156C:48	PHA	Save data byte on stk
156D:8D 79 06	STA $0679	Temp. data storage
1570:98	TYA	Y reg. to A reg.
1571:48	PHA	Store Y on stack
1572:8A	TXA	X reg. to A reg.
1573:48	PHA	Store X on stack
1574:AD FF CF	LDA $CFFF	Turn off card ROMs
1577:AC 5D 90	LDY $905D	Card requires Y have slot (form $s0) upon entry for use in ROM
157A:8D 90 C0	STA $C090	Slot 1 switch to turn on the printer card
157D:AD 79 06	LDA $0679	restore from temp. storage
1580:20 00 C8	JSR $C800	Enter card ROM
1583:8D 91 C0	LDA $C091	Turn off printer card
1586:8D FF CF	LDA $CFFF	Turn off all card ROM
1589:68	PLA	Get X back from stack
158A:AA	TAX	Restore X register
158B:68	PLA	Get Y back from stack
158C:A8	TAY	Restore Y register
158D:68	PLA	Restore accumulator
158E:60	RTS	Return to caller
158F:00	BRK	End of file markers
1590:00	BRK	
1591:00	BRK	

BSAVE DRIVERS.0,A$1000,L$0592

The change in load address and the change in length of the file DRIVERS.0 does not seem to be a problem. Newsroom worked just fine after setup was run and the Fingerprint card selected.

Still a Mystery

Now that I had Newsroom patched, I continued my investigation of the driver mystery byte. The first thing I did was change it from $80 to $00 in the driver on one of the backup disks and then setup both disks with Fingerprint. Comparison of the two disks showed that the byte is stored in the file NEWSPRINT.0 and is loaded at $9064. Search of the disk for references to $9064 resulted only in locating the instruction that stored it at $9064 during setup.

My first thought was that it was used to set or clear the high bit (bit 7) in the data, but there was no difference when I printed a page with the byte set to $80 or $00. Looking at the drivers again, I noted that the driver for the Apple Hi-Speed Serial card set the data high bit with a ORA #$80 instruction, indicating that the driver has to set or clear data bits. I am still not sure if the byte is used or not; there are just too many ways to reference an address.

Newsroom Works, But...

Now that I could get output from Newsroom, I still was not satisfied because a printed page was not centered but was offset to the left. Locating and changing the printer control codes was a possible fix for my problem. I loaded NEWSPRINT.0, since it was the focal point of all printer output, and identified five printer command string locations, two of which were immediately discarded as they contained no data.

I determined that the printer command sequences are set up such that the value of the first byte equals the number of bytes in the actual command sequence. Each sequence is accessed by storing the starting address of the data string in the LDA instruction at $9106. The first byte of the sequence is loaded and put in the X register, to be used as a counter, while the load address at $9106 is incremented to get the actual commands and send them to the printer.

Comparison of the command sequences to those in my Okidata Microline 92 printer manual told me the command functions of each string. The following table shows where the setup of the load address starts, starting address of each command string and the purpose of each data string for my printer.

Setup	Storage	Purpose
$91E3	$902B	Not used
$9134	$908B	Initialize printer
$9124	$909B	Enter graphics mode
$9154	$90AB	Exit graphics mode
$914E	$90BB	Not used

Problem Source

The control codes for the initialization of my printer held the cause of my offset problem. They placed my printer in the 12 characters per inch (cpi) print mode. During graphic printing 12 cpi is desirable, as the resolution, both vertical and horizontal, is then equal. What this means is that at 12 cpi a printed box will be

square but at 10 cpi (normal text mode) the box would be rectangular. The problem was that the printer started printing at its normal location, but at 12 cpi the print head did not go as far across the page.

Thinking that all I had to do was to add the control codes to set the left margin, I determined the necessary control codes and adjustment of the first byte (length) in the sequence to reflect the additional bytes. I worked with the file NEWSPRINT.0 so I could restore it back to the original form by redoing the setup menu. After trying to correct my problem at both the initialization and graphics mode data strings, I finally figured out that the left margin set for my printer worked only in the text mode.

Command Access

Unwilling to give up, I altered NEWSPRINT.0 by inserting an ASCII value for a different number in each printer command string. To ensure that the number was printed, I had make sure that it was sent to the printer when it was in the text mode. I then printed a page and each number was printed out when the command string was sent to the printer. The results of my investigation is as follows:

$902B: This command storage location is not used by my printer, so I created a command with a length of $01 and entered $31 ("1") as the command. The command was accessed only once and was the first command sent. This location is used when Newsroom is configured for an Apple //c, and is zeroed out by NEWSETUP.0 for all other configurations.

$908B: I increased the command length by one and added a $32 ("2") after the command. The command was sent prior to each vertical segment of the output, that is, prior to the banner, and before each of the three horizontal panel pairs.

$909B: Since this is the command location for my printer to enter the graphics mode, I increased the length by one and entered $33 ("3"), followed by the enter graphics mode command. By printing the number followed by the enter graphics command, the printer would still be in the text mode when the number was printed. While printing the page, the command was sent prior to each line.

$90AB: The command to exit the graphics mode is stored here, so I added a $34 ("4") after the command for my printer to enter the text mode. I found that the command was sent after each line of printing, but was sent after the end of line carriage return, thus causing the "4" to be printed to the far left of the page.

$90BB: Since my printer does not use this location, I entered a length of $01 followed by $35 ("5"). This command is sent after each vertical segment.

The Fix

Since my printer exited the graphics mode to the text mode after every line, I could correct my problem by moving the print head to the right with spaces prior to the start of printing. I entered a new string length at $909B and a new command string of several ASCII spaces ($20) followed by the enter graphics command.
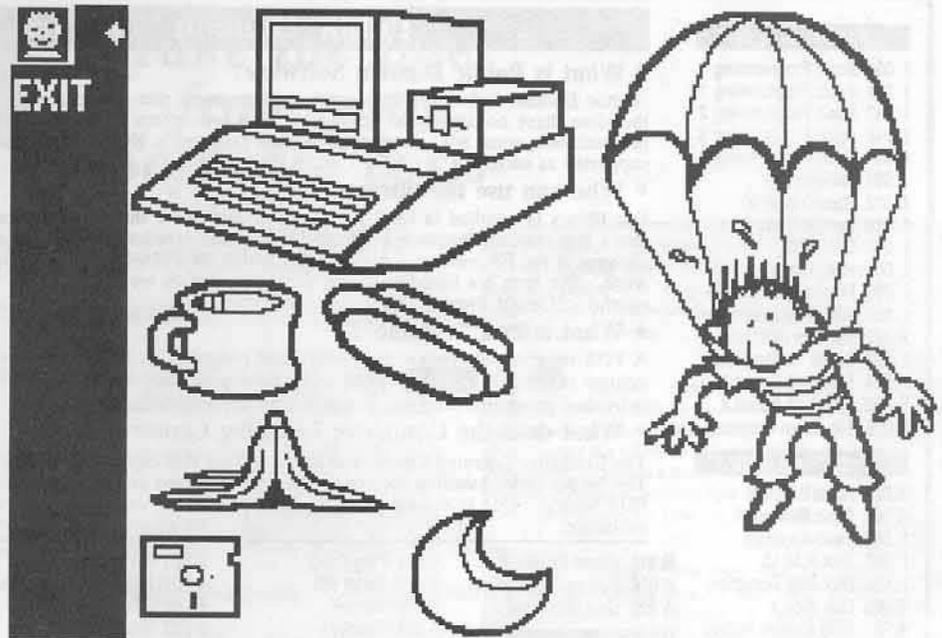
After experimenting with the number of spaces to center the output on the page, I made the change permanent by editing my printer data in the file PRINTERS.0.

Track $03, Sector $03, byte $EB:
Change to 09 20 20 20 20 20 20 20 20 03

This method of printing spaces should work with any printer exhibiting a similar problem. Prior to altering any of the command strings, print a page of output with numbers inserted into the command string as previously discussed. Not all printers respond to commands in the same way as mine does. Also, there are six different routines available in Newsroom that handle the data output, and the one used is not the same for all printers. When I inserted numbers in a friend's printer commands (Epson MX-100) and printed the page, some numbers were printed at the extreme right or left, and one was even repeated about a quarter of the way across the page.

Some commands at $90AB are a carriage return. The addition of spaces after the carriage return command should move the print head to the right prior to the printing of data. Adjustment of the command string length is required, but keep in mind that the maximum length of the command string, including the length byte, is $10.

Printer Storage

I discovered the following printer information while solving my printer output problem. The file PRINTERS.0 is loaded at $458E and is immediately followed by the file DRIVERS.0. To test memory usage and to see if the loading addresses of files changed at all, I loaded PRINTERS.0 and saved it with a longer length, then checked memory again during the setup menu. The file DRIVERS.0 still loaded at its original address of $5FC3 and overwrote the added length of PRINTERS.0. The address for DRIVERS.0 appears to be fixed by the program and is always at $5FC3.

Installation of a new printer in Newsroom is therefore best done by replacing one of the existing printers in the file PRINTERS.0. The printer data block in the file NEWSPRINT.0 is loaded starting at $906B and is identical to the $6B bytes for each printer entry in the file PRINTERS.0. The printer data is allocated as follows:

$00-$1F: The printer name entry, with the high bit cleared.

$20-$2F: Command sequence loaded at $908B, normally the printer initialization command.

$30-$3F: Command sequence loaded at $909B, normally the enter graphics mode command.

$40-$4F: Command sequence loaded at $90AB, normally a carriage return or the exit graphics command.

$50-$5F: Command sequence loaded at $90BB, normally the exit graphics command or is unused.

$60: Contains values of $80 or $00. Used by setup to determine if user is to be asked if a line feed is required after each line. A value of $80 will result in the option being offered and a "yes" response results in $FF being stored at $90CB.

$61-$64: Unknown.

$65: Loaded at $90D0 and used by NEWSPRINT.0 at $9542 as the key to the data and address table at $94B0. This byte will have the values of $00, $05, $06, $10, $0F and $0A which will result in use of data output routines at $935E, $9397, $9400, $9468, $93C6 and $93D6 respectively. The routine selected will affect when and what commands are sent to the printer.

$66-$6B: Unknown.

Although this isn't everything there is to know about Newsroom, it may assist those who need to install a printer in Newsroom or want to experiment with the data output. If any of the readers determine the use of the unknowns, maybe they will share the information with the rest of us.

Apple Tutor P01

I	004	Basic Programming
I	036	Basic Programming 1
I	047	Basic Programming 2
I	048	Basic Programming 3
I	044	Basic Programming 4
A	004	Basic-Integer
B	022	Basic-Integer.X
I	019	Conventions
I	016	CPU 6502
I	002	Hello Sample
I	030	Micro 6502 Simulate
I	051	Mini Assembler Tutor
A	022	Random Drill Tutor
I	007	Sweet 16 Disassemb
I	004	Sweet 16 Speed ?
B	002	Sweet 16 Speed.X
I	026	Top Down Programg

Apple Tutor P02

A	021	Apple II Demo
A	004	Basic Exercise
I	043	Basic-Applesoft
I	093	Disk Aide.13
I	038	Disk Aide Document
B	003	Disk Aide.X
A	028	DOS System Instruct
A	013	Pilot Version I
I	054	Program Devel. pkg
A	014	Sup. Mini Assem.
B	006	Supplement.X
I	003	Text File Read
I	003	Text File Write
I	009	Text Hello
T	009	Text How To
T	010	Text Intro
T	006	Text Peek Poke Call
T	010	Text Programming
T	007	Text Redbook
T	011	Text Software
A	002	Buzz

Art & Graphic P11

A	006	Art Align
A	002	Art Bars
A	003	Art Circle
A	002	Art Demo
A	003	Art Diverging Circles
A	003	Art Diverging Oct
A	003	Art Double Cross
A	006	Art Dougs Theme
A	003	Art Drifting Circle
A	003	Art Drifting Octagon
A	004	Art Elephant
A	003	Art Figure 8
A	005	Art Horizon
A	007	Art IBM
A	003	Art Octal 8
A	002	Art Oneliner 7
A	007	Art Super Kalied
A	003	Art Twist II
A	015	Art Xmas Card
A	014	Auto Space War II
		and 30 other programs

Astronomy P14

A	022	Astronomy Exposures
I	028	Astronomy Tutor
A	006	Calendar Moon Age
A	011	Ephemeris
A	060	Great Circle Bearings
A	014	Jupiter Moons
B	002	Jupiter.X
B	018	Jupiter.Y
A	018	Moon Finder
A	025	Oscar
A	011	Oscar Orbit Periods
A	031	Oscar Locator
T	003	Oscar Locator File
A	009	Oscar Loc File Init
A	019	Oscar Loc Instruct
A	024	Planet Locator
B	002	Planet Locator.X

B	010	Planet Locator.Y
A	018	Sunrise And Sunset
A	023	Ursa Star Time

Business & Finance P21

A	004	Annuity
A	004	Average Growth Rate
T	001	Basenamefile
A	046	CAC Record Sys ◄
A	009	Check Book Balancer
A	008	Check Writer
A	004	Depreciation Amount
A	003	Depreciation Rate
A	010	Depreciation Sched
A	003	Depreciation Straight
A	004	Disc. Commer Paper
A	024	House Sales
A	005	Income Taxes
A	012	Interest Earned
A	003	Interest Rate
A	038	Inventory Home II
A	004	Invest. For Withdraw
A	004	Investmt Future Val
A	004	Investment Initial
A	013	Lease Computation
A	031	Life Mgmt & Financ
A	012	Life Mgmt Txt Org
A	005	Loan Balance
A	007	Loan Interest Rate
A	005	Loan Last Payment
A	004	Loan Principal
A	005	Loan Regular Paymt
A	004	Loan Term
A	004	Mortgage Computer
A	007	Mortgage Table
A	063	Payroll
A	004	Regular Deposits II
A	003	Regular Withdrawals
A	004	Salvage Value
A	009	Savings Growth
A	011	Survey Data Reduct
A	003	System Reliability
A	004	Treasury Bill Valuat

Game P36

A	027	Blackjack Strategy
A	012	Combat
A	017	Craps BW
A	035	Cribbage I BW
A	028	Football Predictions
A	011	Fox And Hounds
A	026	French Military Game
A	020	Gold Mine
A	022	Golf II
A	016	Hi Q
A	027	Hockey I
A	021	Horse Race III
A	031	Kingdom
A	008	Literature Quiz
A	026	Marooned In Space

A	004	Ping Pong
A	010	Robot BW
A	016	Survive
A	018	Twonky I
A	018	Word Maze Maker
A	003	Football Predict.note

Game P50

A	017	Battle
A	015	Century 51
B	002	Century 51.X
A	020	Checkers II
A	021	Commodity Market
A	016	Craps I
A	013	Cryptograms
A	017	Frustration
A	027	Geography II
A	015	Hangman II
A	022	Lunar Landing BW
A	061	Market Crash
B	005	Market Crash.X
A	009	Name the States
A	018	Puzzle Generator
A	007	Solitaire Checker Puz
A	010	Spell the States
A	012	Subscan for Two
B	002	Subscan.X
A	031	World War

Math & Statistics P59

A	008	Anglo To Metric I
A	011	Anglo To Metric II
A	003	Arcsin Arccos
A	016	Calculator
A	007	Calculus I
A	013	Calculus II
A	007	Cash Register
A	005	Circle Area Circum
A	009	Compound Interest
A	010	Critical Path Analysis
A	006	Curve Fit I
A	013	Curve Fit II
A	013	Curve Fit III
A	003	Derivative of Eqn
A	010	Differentl Eqn Solver
A	015	Equation Solver
A	005	Equations I
A	014	Equations II
A	014	Factor
A	003	Fibonacci Numbers
A	012	Foot Candle Analysis
A	010	Fourier
A	011	Fourier Transform
A	007	Gaussn Quadrature II
A	010	Math Drill I
A	011	Math Drill II
A	005	Matrix Inversion II
A	009	Matrix Operation
A	016	Mean Vari Devia II
A	007	Mode Finder

A	003	N Factorial
A	007	Plot Consec Reactns
A	006	Plot Functions of X
A	003	Prog Chart ?
A	011	Robot Motor Design
A	005	Sine Function
A	010	Statistics I
A	014	Statistics II
A	014	T Test Stdn Deviation
A	004	Time Speed Dist Ex
A	006	Time Speed Dist Fuel
A	014	Triangle Solver
A	011	Unpaired Gp Compar

Math & Statistics P61

A	003	Angle Conversion
A	012	Anglo To Metric III
A	003	Area of Polygon
A	008	Binomial Distributn
A	004	Blackbody
A	003	Chi Square Distributn
A	004	Chi Square Test
A	004	Coordinate Convers
A	008	Coordinate Plot
A	003	Curvilinear Interpolat
A	002	Derivative
A	016	Dfit
A	011	Diff Eqn Solver Demo
A	004	Exponential Regress
A	004	F Distribution
A	004	Gauss Quadrature I
A	003	Geometric Mean
A	004	Geometr Regression
A	004	Greatest Com Denom
A	005	Histogram
A	003	Linear Interpolation
A	007	Linear Programming
A	004	Linear Regression
A	005	Mann Whitney U Test
A	022	Math Drill III
A	019	Math Multiply Drill
A	019	Math ◄
A	013	Matrices
A	004	Matrix Inversion I
A	004	Matrix Multiplication
A	004	Matrix Opern Simple
A	004	Mean Vari Devia I
A	007	Mult Linear Regressn
A	004	Normal Distribution
A	006	Nth Order Regression
A	004	Nmbr Combinations
A	002	Parabola Plot
A	003	Permut Combination I
A	007	Permut Combinatn II
A	003	Poisson Distribution
A	006	Polar Equation Plot
A	011	Polyfit
A	013	Polynomial Regressn
A	006	Power Curve Fit ©

A	003	Prime Factors I
A	006	Prime Factors II
A	003	Quadratic Formula
A	006	Quadratic Surface
A	015	Right Triangle Solver
A	016	Root Finder
A	005	Roots of Poly Half
A	005	Roots of Polynomials
A	004	Simpson's Rule
A	004	Simultaneous Eqns
A	004	T Distribution
A	005	T Distribution Test
A	003	Trapezoidal Rule
A	003	Trig Polynomial
A	004	Vector Analysis
A	003	Vector Operations

Utility P76

A	003	Base Convers Chart I
A	003	Base Conver Chart II
A	006	Base Convert **
A	005	Base Convert Beymer
A	012	Base Convert Ellmers
A	005	Base Convert Jenkins
A	029	Base Conv Massimo
I	043	Basic-Applesoft
A	007	Binary To FP
A	053	Calendar And Posters
A	002	Capture A Program
A	005	Catalog Printer
A	003	Clear Memory
A	004	Copy Text File I
A	004	Copy Text File II
A	004	Copy Text File III
A	004	Ctrl Char Catalog
A	003	Ctrl Char Reveal
A	003	Dump ASCII Memory
A	002	Erase Hires Screen
A	011	Format #
A	003	Free Sectors Aldrich
A	004	Free Sectors Brown
B	002	Free Sectors Brown.X
A	019	Illegal Commands
A	005	List Page Diaz
A	004	List Page Wysocki
B	002	List Page Wysocki.X
B	002	List Page.X
A	007	Mach To Pokes Conv
A	002	Musical Keys
B	002	Musical Keys.X
A	013	Phone Dialer
A	003	Pointers
A	002	PR# 6 On Reset
A	003	Random Sentence
B	002	Remark Stripper
A	003	Rem Stripper Doc
T	001	Ren/Mrg
A	006	Renumber
A	004	Renumber Merge

B	010	Renumber Merge.X
A	003	Reverse Print
A	018	Sort Catalog
A	003	Sort Catalog ◄
B	002	Sort Catalog.X
A	015	SSM AIO Board
A	004	String Comparator
A	004	Str In A Str Search
A	004	Text File Edit
A	003	Text File Peek
A	003	Text File Read
A	003	Text Fl Read & Prnt I
A	005	Text Read & Prnt II
A	005	Text File Write
A	004	Token Address Tbl I
A	005	Utility Statements

Utility P86

B	002	EP1.3
B	002	EP2.3
B	002	EP3.3
B	003	EP4.3
A	002	Epson Cat Dbl Strike
A	008	Epson Driver Instr
B	002	Epson Driver.
A	002	Epson Greeting
A	004	Epson Labels I
A	005	Epson Labels II
A	006	Epson Labels III
A	002	Epson Lower Case
A	002	MX/FX-80 List Capt
A	008	MX/FX-80 List Format
A	005	MX/FX-80 List Instr.
A	003	Epson MX80 Catalog
A	005	Epson MX80 Demo I
A	007	Epson MX80 Demo II
A	024	MX80 Label Maker I
A	052	MX80 Label Maker II
A	053	MX80 Label Maker III
B	034	MX80 Label.PIC
A	003	MX80 Letterhead
A	015	MX80 Prog. List
A	023	MX80 Calendar
A	033	Epson MX80 Setup I
A	033	Epson MX80 Setup II
A	005	Epson Util 3.3 Demo
A	023	Epson Util 3.3 Info
A	022	Epson Word Game
A	003	Ep.EXEC.Maker A3.3
I	003	Ep.EXEC.Maker I3.3
B	005	Text.Demo
B	004	Text.MXtest
T	002	XEpson.V3.3

Modem P105 (Part 1)

B	005	ABBS1.OBJ0
B	002	ABBS3.OBJ0
T	009	BBS
T	004	Bulletin
T	248	Messages
T	017	Club Store
T	019	Meetings
T	002	Startup
T	002	Users
B	018	Users.Obj
A	078	WAPABBS

Modem P106 (Part 2)

T	047	ABBS1
T	013	ABBS3
A	003	Build Users.Obj
T	035	Instructions
T	062	Remember II
A	004	Run This First
A	005	Run This Second
T	120	SysOp Instructions
T	002	To SysOp
T	002	Upload2
T	008	Upload1
T	073	Rem II Instructions
T	002	Opinion
T	103	WAPABBS Note

• What is Public Domain Software?

Public Domain Software (PDS) consists of programs that are donated to the public, and therefore, have no copyrights attached. They are written by a variety of people, some professionals, some not - in most cases each program is NOT commercial quality and is not supported as such.

• Who can use the library?

Our library is supplied in DOS 3.3 16 sector format for the Apple II computers. Please note that a few machine language programs will funtion erratically on the IIe and IIc because of changes in the F8 monitor. A title listed with a ◄ will tell you that the program still needs work. We have not tested all of the programs, nor do we have a list of what works. So, be careful - "Caveat Emptor."

• What is in a "volume"?

A PDS volume will <u>contain</u> many individual programs on a certain subject. The name of the volume is listed in reverse print. The catalog of titles under each volume name list the individual programs, language it was written in, and the sectors used.

• What does the Computer Learning Center do?

The Computer Learning Center provides a service that copies and DISTRIBUTES software. The $4 per order handling fee covers the costs incurred in copying, packaging and mailing the PDS library. Due to the nature of the library, volumes can not be returned for refund or exchange.

Description of Available Back Issues.

CORE 3 ....Games:
Constructing Your Own Joystick | Compiling Games | GAME REVIEWS: Over 30 of the latest and best | Pick Of The Pack: All-time TOP 20 games | Destructive Forces | EAMON | Graphics Magician and GraFORTH | Dragon Dungeon | ....

CORE 2 ....Utilites:
Dynamic Menu | High Res: Scroll Demo | GOTO Label: Replace | Line Find | Quick Copy: Copy | ..

CORE 1 ....Graphics:
Memory Map | Text Graphics: Marquee | Boxes | Jagged Scroller | Low Res: Color Character Chart | High Res: Screen Cruncher | The UFO Factory | Color | Vector Graphics:Shimmering Shapes | A Shape Table Mini-Editor | Block Graphics: Arcade Quality Graphics for BASIC Programmers | Animation | ....

Hardcore Computing 3
HyperDOS Creator | Menu Hello | Zyphyr Wars | Vector Graphics | Review of Bit Copiers | Boot Code Tracing | Softkey IOB | Interview with 'Mike' Markkula | ....

Looking for the
Best Deal in Town?

How about ALL of our Super IOB controllers, (through 1986) in ONE package!

This package contains:

► **TWO DISKS** (supplied in DOS 3.3). Each containing at least 60 Super IOB Controllers including the standard, swap, newswap and fast controllers. In addition, each disk has the Csaver program from COMPUTIST No. 13; version 1.5 of Super IOB; and a Menu Hello Program that lists the available controllers and, when you select one, automatically installs it in Super IOB and RUNs the resulting program.*

► A reprint of **Disk Inspection and the Use of Super IOB**, from COMPUTIST No. 17. This article explains how to write your own Super IOB controllers.

► **COMPUTIST No. 32**, which contains an extensive article detailing the hows and whys of Super IOB v1.5 and at least 5 articles using the new Super IOB program.

● Several of the controllers deprotect the software completely with no further steps. This means that some programs are only minutes away from deprotection (with virtually no typing).

● The issue of COMPUTIST in which each controller appeared is indicated in case further steps are required to deprotect a particular program.**

Disk 1	Disk 2
Volume 1 of the Super IOB collection covers all the controllers appearing in COMPUTIST No. 9 through No. 26. In addition, the newswap and fast controllers from COMPUTIST No. 32 are included. The following 60 controllers are on volume 1:	Volume 2 of the Super IOB collection covers all the controllers appearing in COMPUTIST No. 27 through No. 38. The following 65 controllers are on volume 2:
Advanced Blackjack, Alphabet Zoo, Arcade Machine, Archon II, Archon, Artsci Software, Bank Street Writer, Barrons SAT, Beyond Castle Wolfenstein, BSW //c Loader, Castle Wolfenstein, Computer Preparation: SAT, Dazzle Draw, DB Master 4 Plus, Death in the Carribcan, Dino Eggs, DLM Software, Electronic Arts, F-15 Strike Eagle, Fast Controller, Fathoms 40, Financial Cookbook, Gessler Software, Grandma's House, The Heist, In Search of the Most Amazing Thing, Instant Recall, Kidwriter, Lions Share, Lode Runner, Mastertype, Match Maker, Miner 2049er, Minit Man, Mufplot, Newsroom, Newswap controller, Penguin Software, Print Shop Graphic Library, Print Shop, Rendezvous with Rama, Rockys' Boots, Sargon III, Sea Dragon, Shiela, Skyfox, Snooper Troops, Standard controller, Stoneware Software, Summer Games, Super Controller, Super Zaxxon, Swap Controller, TAC, Ultima III, Word Challenge, Xyphus, Zaxxon	Alice in Wonderland, Alphabetic Keyboarding, Alternate Reality, Autoduel, Checkers, Chipwits, Color Me, Conan.data, Conan.prog, CopyDOS, Crisis Mountain, Disk Director, Dragonworld, Early Games, Easy as ABC, F-15 Strike Eagle, Fantavision, Fast controller, Fishies, Flight Simulator, Halley Project, Hartley Software (a), Hartley Software (b), Jenny of the Prarie, Jingle Disk, Kidwriter, Kracking Vol II, Lode Runner, LOGO II (a), LOGO II (b), Masquerade, Mastering the SAT, Microtype: The Wonderful World of Paws, Microzines 1, Microzines 2-5, Miner 2049er, Mist & View to a Kill, Murder on the Zinderneuf, Music Construction Set, Newswap controller, Olympic Decathlon, Other Side, Phi Beta Filer, Pitstop II, Print Shop Companion, RDOS, Robot War, Spy vs Spy, Standard controller, Sundog V2, Swap controller, Sword of Kadash, Synergistic Software, Tawala's last Redoubt, Terripin Logo, Threshold, Time is Money, Time Zone, Tink! Tonk!, Troll's Tale, Ultima IV, Wilderness, Word Attack & Classmate, World's Greatest Baseball, World's Greatest Football

To Order: Send $9.95 for each volume or $19.95 for a complete package that includes: both disks, a reprint of "Disk Inspection and the use of Super IOB" and COMPUTIST No. 32. U.S. funds drawn on U.S. banks. Foreign orders (other than Canada or Mexico) add 20% shipping. Washington state residents add 7.8% sales tax. Mail orders to: Super IOB Collection   PO Box 110846-BK   Tacoma, WA 98411

*Requires at least 64K of memory.

**Although some controllers will completely deprotect the program they were designed for, some will not and therefore require their corresponding issue of COMPUTIST to complete the deprotection procedure.