

For The Serious User Of Apple][Computers

Hardcore

COMPUTIST

Issue No. 23 \$3.75

Softkeys For:

Choplifter
Mufplot
Flashcalc
Karateka
Newsroom
E-Z Draw

Using A.R.D. to
Softkey Mars Cars

How to be
the Writemaster

Core's
Wheel of Money



Hardcore COMPUTIST
PO Box 110846-T
Tacoma, WA 98411

BULK RATE
U.S. Postage
PAID
Tacoma, WA
Permit No. 269

Apple II, II+, IIe, Franklin users:

Do you have problems
backing-up your
copy-protected programs?
Do you lack parameters for
your copy programs?
Are you looking for programs
that you can AFFORD?
Are you hesitating to
upgrade your equipment due to
expensive prices
quoted in other ads?

**It's simple now.
Just drop us a line.**

Send \$1.00 U.S. funds to:

Reliant
P.O. Box 33610
Sheungwan, Hong Kong

IMPORTANT: We have over 1000 PC name-brand programs and various hardware offers. Programs @ U.S. \$8.00/PC includes the disk and registered air-mail handling.



THE FRANKLIN USERS GROUP



Supporting Users of

II, II+, FRANKLINS, AND ALL COMPATIBLES

We support all members with a monthly newsletter the 'Clone Ace,' a growing public domain library, and personal attention.

MEMBERSHIP IS \$20.00 A YEAR.

We offer hardware and software at great savings.

Checkmate Multiview 80/160

80 to 160 column display and up to 48 lines.

Compatible with II+'s and all compatibles.

Supports all major operating systems including PCP I.

!!! RUNS APPLEWORKS ON 64K CLONES AND II+'S !!!

\$299

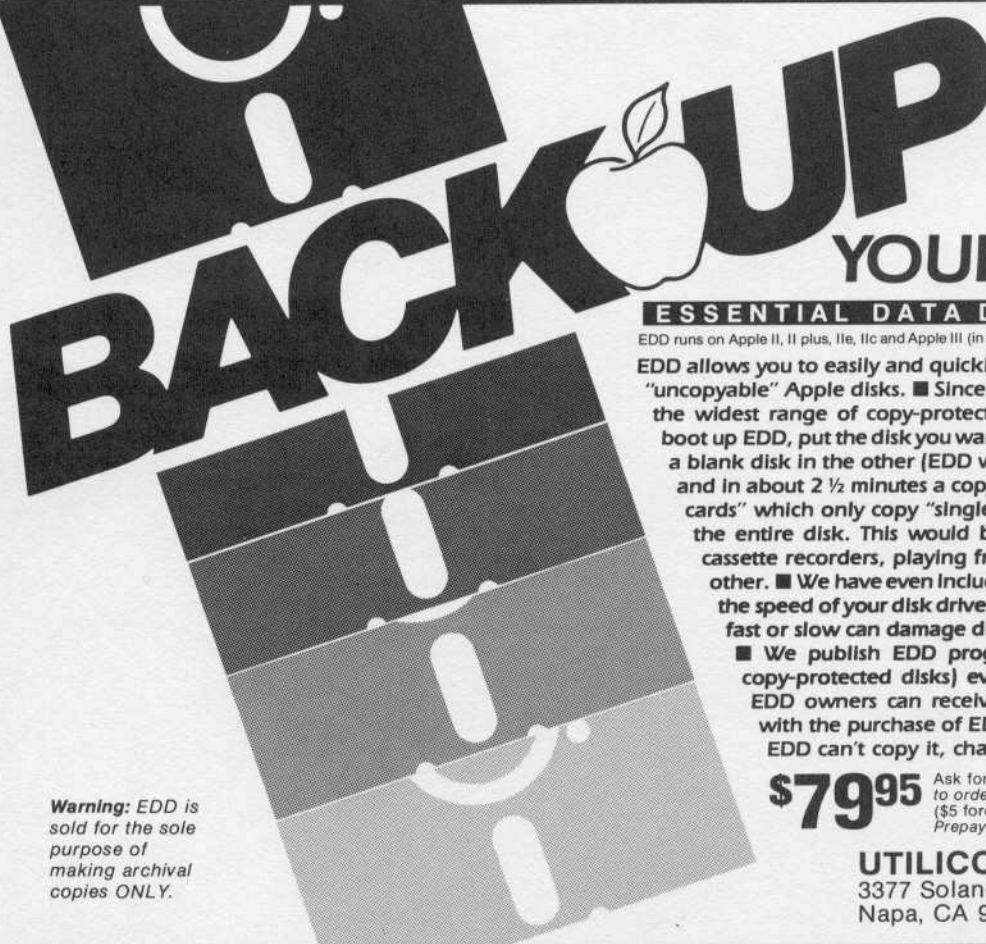
Axlon 320 RAM Disk

Supports all major operating systems including PCP I

\$480

THE FRANKLIN USERS GROUP

East Wind Rd., Apt. L
Tecumseh, MO 65790



NOW
AVAILABLE
AT YOUR LOCAL
COMPUTER STORE

YOUR DISKS

ESSENTIAL DATA DUPLICATOR III™

EDD runs on Apple II, II plus, IIe, IIc and Apple III (in emulation mode) using one or two disk drives.

EDD allows you to easily and quickly make back up copies of your "uncopyable" Apple disks. ■ Since EDD has been preset to copy the widest range of copy-protections possible, you just simply boot up EDD, put the disk you want to copy in one disk drive and a blank disk in the other (EDD will work using one drive also) and in about 2 ½ minutes a copy is made. ■ Unlike the "copy-cards" which only copy "single load" programs, EDD copies the entire disk. This would be similar to hooking up two cassette recorders, playing from one, and recording to the other. ■ We have even included an option so you can check the speed of your disk drives because drive speeds running fast or slow can damage disks and cause other problems.

■ We publish EDD program lists (information about copy-protected disks) every couple of months, which EDD owners can receive. The current list is included with the purchase of EDD. ■ The bottom line is this; if EDD can't copy it, chances are nothing will.

\$79⁹⁵

Ask for EDD at your local computer store, or, to order direct; send \$79.95 plus \$2 shipping (\$5 foreign). Mastercard/Visa accepted. Prepayment required.

UTILICO MICROWARE

3377 Solano Ave., Suite #352
Napa, CA 94558 (707) 257-2420

Warning: EDD is sold for the sole purpose of making archival copies ONLY.

5 FREE DISKS

With every set of 20 disks you order.
 You can get sets for as low as \$17.25.
 That's a total of 25 disks for as little as...

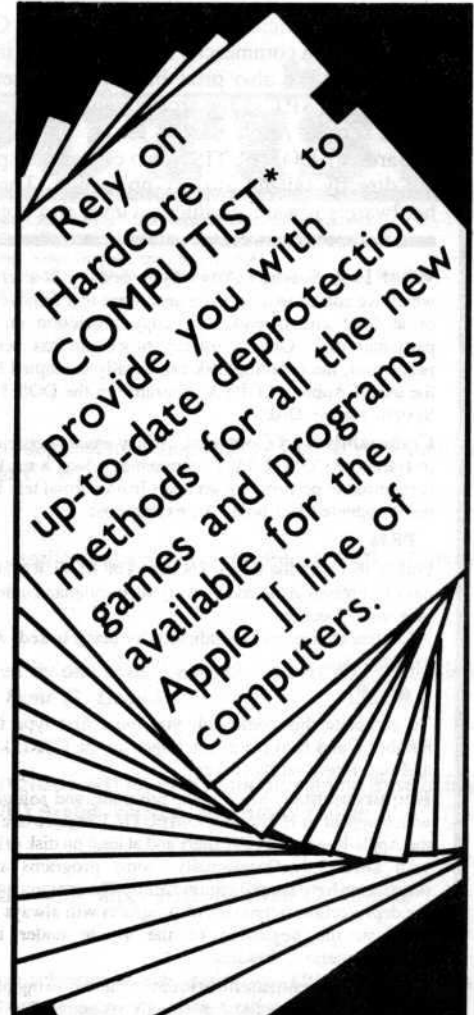
69¢ a disk

These are SS/DD Namebrand* 5¼" floppies
 100% guaranteed and include:
 reinforced hubs and write-protect tabs

Send me _____ sets without sleeves at \$17.25 per set
 Send me _____ sets, with Tyvek sleeves at \$18.50 per set.
 Add \$3 shipping and handling for the first set, \$1 for each additional set. Foreign orders add 20%. U.S. funds drawn on U.S. bank. Washington orders add 7.8% sales tax. Send your orders to:
 Softkey Publishing, PO Box 110816, Tacoma, WA 98411

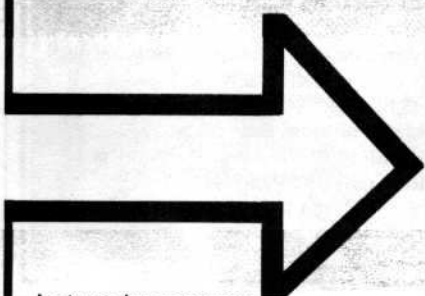
Name _____
 Address _____
 City _____ State _____ Zip _____
 Country _____ Phone _____
 Signature _____
 Visa/MC _____ Exp _____

* BASF or Verbatim.



Rely on
 Hardcore
 COMPUTIST* to
 provide you with
 up-to-date deprotection
 methods for all the new
 games and programs
 available for the
 Apple II line of
 computers.

*The
 magazine for
 the serious user of
 Apple II computers.



Moving Soon?

Let us know your
NEW ADDRESS at
**least 30 days in
 advance** so you
 won't miss a single
 issue of Hardcore
 COMPUTIST (issues
 missed due to non-
 receipt of change of
 address may be
 obtained at the
 current back-issue
 rate).

Fill out your new address on
 the form provided, paste
 your present address label in
 the space below and send it
 to:
 Hardcore COMPUTIST,
 Subscription Dept., PO Box
 110846-T, Tacoma, WA 98411.

My new address is:
 Address _____
 City _____ State _____ Zip _____
 Phone _____

Paste present address label here or fill in old address:
 Name _____
 Address _____
 City _____ State _____ Zip _____
 Country _____
 Phone _____

Subscription rates (for six issues):
 Please check one

<input type="checkbox"/> US	\$20
<input type="checkbox"/> US 1st Class	\$24
<input type="checkbox"/> Canada	\$34
<input type="checkbox"/> Mexico	\$39
<input type="checkbox"/> Foreign	\$60
<input type="checkbox"/> SAMPLE - US	\$4.75
<input type="checkbox"/> SAMPLE - Foreign	\$5.75

() Yes! Start my subscription now.

Name _____
 Address _____
 City _____ State _____ Zip _____
 Country _____
 VISA/MC _____ Exp _____
 Signature _____

Send check or money order (US funds drawn on US bank) to: Hardcore COMPUTIST, PO Box 110846-T, Tacoma, WA 98411.

Many of the articles published in Hardcore COMPUTIST detail the removal of copy protection schemes from commercial disks or contain information on copy protection and backup methods in general. We also print bit copy parameters, tips for adventure games, advanced playing techniques (APT's) for arcade game fanatics and any other information which may be of use to the serious Apple user.

Hardcore COMPUTIST also contains a special CORE section which focuses on information not directly related to copy protection. Topics may include, but are not limited to: tutorials, hardware/software product reviews and application and utility programs.

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 Hardcore COMPUTIST, commands which a reader is required to perform are set apart from normal text by being indented and bold. An example is:

PR#6

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

Control characters are indicated by being boxed. An example is:

6

To complete this command, you must first type the number 6 and then place one finger on the CTRL key and one finger on the P key.

Requirements: Most of the programs and softkeys which appear in Hardcore COMPUTIST require one of the Apple II series of computers and at least on disk drive with DOS 3.3. Occasionally, some programs and procedures have special requirements. The prerequisites for deprotection techniques or programs will always be listed at the beginning of the article under the "Requirements:" heading.

Software Recommendations: The following programs (or similar ones) are strongly recommended for readers who wish to obtain the most benefit from our articles:

- 1) **Applesoft Program Editor** such as Global Program Line Editor (GPLE).
- 2) **Sector Editor** such as DiskEdit, ZAP from Bag of Tricks or Tricky Dick from The CIA.
- 3) **Disk Search Utility** such as The Inspector, The Tracer from The CIA or The CORE Disk Searcher.
- 4) **Assembler** such as the S-C Assembler or Merlin/Big Mac.
- 5) **Bit Copy Program** such as Copy II Plus, Locksmith or The Essential Data Duplicator
- 6) **Text Editor** capable of producing normal sequential text files such as Appewriter II, Magic Window II or Screenwriter II.

You will also find COPYA, FID and MUFFIN from the DOS 3.3 System Master Disk useful.

Super IOB: This program has most recently appeared in Hardcore COMPUTIST No. 22. Several softkey procedures will make use of a Super IOB controller, a small program that must be keyed into the middle of Super IOB. The controller changes Super IOB so that it can copy different disks. To get the latest version of this program, you may order Hardcore COMPUTIST No. 22 as a back issue or order Program Library Disk No. 22.

RESET Into The Monitor: Many softkey procedures require that the user be able to enter the Apple's system monitor during the execution of a copy protected program. Check the following list to see what hardware you will need to obtain this ability.

Apple II Plus - Apple IIe - 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 II 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 of Hardcore COMPUTIST No. 6 or the "Dual ROM's" article in Hardcore COMPUTIST No. 19.

Apple IIe - Apple IIc: Install a modified CD ROM on the computer's motherboard. Don Lancaster's company (Synergetics; 746 First Street; Box 809-HC; Thatcher, AZ 85552; free voice Helpline 602-428-4073) sells the instructions necessary to make this modification. Making this modification to an Apple IIc will void its warranty but the increased ability to remove copy protection may justify it.

Recommended Literature: The Apple II Reference Manual and DOS 3.3 manual are musts for any serious Apple user. Other helpful books include: *Beneath Apple DOS*, Don Worth and Peter Lechner, Quality Software, \$19.95; *Assembly Language For The Applesoft Programmer*, Roy Meyers and C.W. Finley, Addison Wesley, \$16.95; and *What's Where In The Apple*, William Lubert, Micro Ink., \$24.95.

Keying In Applesoft Programs: BASIC programs are printed in Hardcore COMPUTIST in a format that is designed to minimize errors for readers who key in these programs. To understand this format, you must first understand the formatted LIST feature of Applesoft.

An illustration - If you strike these keys:

10 HOME:REMCLEAR SCREEN

a program will be stored in the computer's memory. Strangely, this program will *not* have a LIST that is exactly as you typed it. Instead, the LIST will look like this:

10 HOME ; REM CLEAR SCREEN

Programs don't usually LIST the same as they were keyed in 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 command words. The space inserted after these command words can be misleading. For example, if you want a program to have a list like this:

10 DATA 67,45,54,52

you would have to omit the space directly after the DATA command word. If you were to key in the space directly after the DATA command word, the LIST of the program would look like this:

10 DATA 67,45,54,52

This LIST is different from the LIST you wanted. The number of spaces you key after DATA and REM command words is very important.

All of this brings us to the Hardcore COMPUTIST LISTING format. In a BASIC LISTING, there are two types of spaces; spaces that don't matter whether they are keyed or not and spaces that must be keyed. Spaces that must be keyed in are printed as delta characters (Δ). All other spaces in a Hardcore COMPUTIST BASIC listing are put there for easier reading and it doesn't matter whether you type them or not.

There is one exception: If you want your checksums (See "Computing Checksums" section) to match up, you *must not* key in any spaces after a DATA command word unless they are marked by delta characters.

Keying In Hexdumps: Machine language programs are printed in Hardcore COMPUTIST as both source code and hexdumps. Only one of these formats need be keyed in to get a machine language program. Hexdumps are the shortest and easiest format to type in.

To key in hexdumps, you must first enter the monitor:

CALL -151

Now 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). If you hear a beep,

you will know that you have typed something incorrectly and must retype that line.

When finished, return to BASIC with a:

E003G

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

Keying In Source Code The source code portion of a machine language program is provided only to better explain the program's operation. If you wish to key it in, you will need an assembler. The S-C Assembler is used to generate all source code printed in Hardcore COMPUTIST. 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 just for this purpose was printed in Hardcore COMPUTIST No. 17. To translate source code, you will need to understand the directives of your assembler and convert the directives used in the source code listing to similar directives used by your assembler.

Computing Checksums Checksums are four digit hexadecimal numbers which verify whether or not you keyed a program exactly as it was printed in Hardcore 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 programs appeared in Hardcore COMPUTIST No. 1 and The Best of Hardcore Computing. An update to CHECKSOFT appeared in Hardcore COMPUTIST No. 18. If the checksums these programs create on your computer match the checksums accompanying the program in the magazine, then you keyed in the program correctly. If not, the program is incorrect at the line where the first checksum differs.

1) To compute CHECKSOFT checksums:

**LOAD filename
BRUNCHECKSOFT**

Get the checksums with

&

And correct the program where the checksums differ.

2) To compute CHECKBIN checksums:

**CALL -151
BLOAD 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 **QY**.

xxx.xxx QY

And correct the lines at which the checksums differ.

How-To's Of Hardcore

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

Our editorial policy is that we do NOT condone software piracy, but we do believe that honest users are entitled to backup commercial disks they have purchased. In addition to the security of a backup disk, the removal of copy protection gives the user the option of modifying application programs to meet his or her needs.

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.

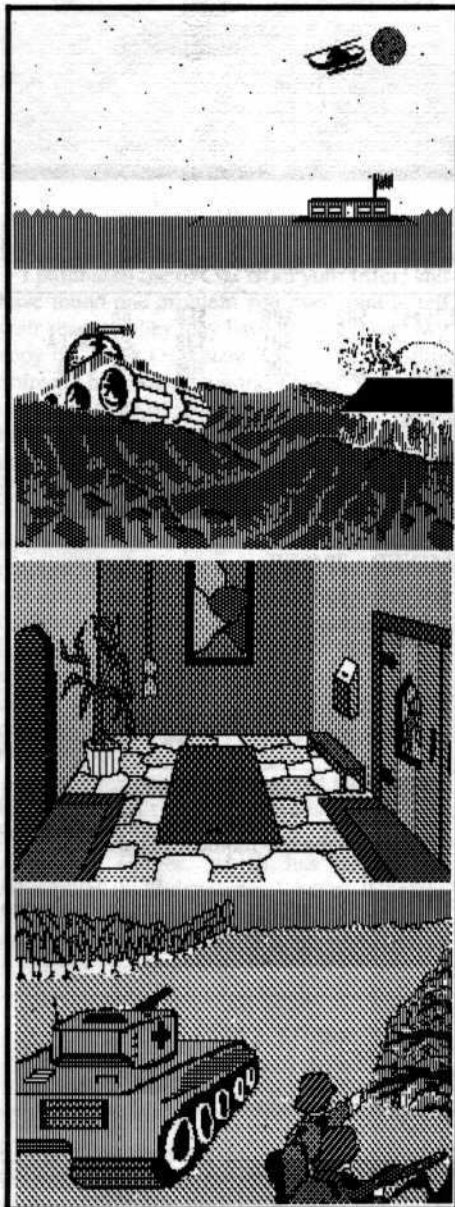
COMPUTIST

Publisher/Editor: Charles R. Haight Managing Editor: Ray Darrah Technical Editor: Robert Knowles

Production & Graphics: Taryn Bates, Bev Haight Circulation: Michelle Frank, Debbie Koval

Advertising: (206) 474-5750 Printing: Valco Graphics Inc., Seattle, WA

Hardcore COMPUTIST is published monthly, except December, by SoftKey Publishing, 5233 S. Washington, Tacoma, WA 98409
Phone: (206) 474-5750



This month's cover: Graphics from *DEATH IN THE CARIBBEAN*, a hi-res adventure game from Micro Lab.

Address all advertising inquiries to Hardcore COMPUTIST, Advertising Department, PO Box 110816, Tacoma, WA 98411. Mail manuscripts or requests for Writer's Guides to Hardcore COMPUTIST, PO Box 110846-K, Tacoma, WA 98411.

Return postage must accompany all manuscripts, drawings, photos, disks, or tapes if they are to be returned. Unsolicited manuscripts will be returned only if adequate return postage is included.

Entire contents copyright 1985 by SoftKey Publishing. All rights reserved. Copying done for other than personal or internal reference (without express written permission from the publisher) is prohibited.

The editorial staff assumes no liability or responsibility for the products advertised in the magazine. Any opinions expressed by the authors are not necessarily those of Hardcore COMPUTIST magazine or SoftKey Publishing.

Apple usually refers to the Apple II or II Plus Computer, and is a trademark of Apple Computers, Inc.

SUBSCRIPTIONS: Rates (for 6 issues): U.S. \$20, U.S. 1st Class \$24, Canada \$34, Mexico \$39, Foreign \$60. Direct inquiries to: Hardcore COMPUTIST, Subscription Department, PO Box 110846-T, Tacoma, WA 98411. Please include address label with correspondence.

DOMESTIC DEALER RATES: Call (206) 474-5750 for more information.

Change Of Address: Please allow 4 weeks for change of address to take effect. Use postal form 3576 to supply your new and former address using your most recent magazine label. Issues missed due to non-receipt of change of address may be acquired at the regular back issue rate.

softkeys

10 Choplifter

Now you can shorten the boot up time and have a deprotected Choplifter as well as APT control keys that let you really take command. *by Todd C. Wilson*

13 Mufplot

Ham radio operators, try the Super IOB's Swap Controller and a modified Mufplot RWTS to softkey this program. *by Edward E. Harte, M.D.*

14 Flashcalc

Believe it or not, Flashcalc is effectively protected by one byte. Learn about ProDOS protection schemes by softkeying this business utility. *by Doni G. Grande*

16 Karateka

With smooth graphics, great sound effects, and realistic action, how could this game miss being equally well protected? Kick your way to a COPY-able version. *by Danny Pollak*

18 Newsroom

This handy graphic package is quickly normalized with Super IOB. *by Doni G. Grande*

20 E-Z Draw

Some older programs have sophisticated protection schemes worth exploring. This 1980 graphics system is made COPY-able in our Antique Softkey section. *by Nick Galbreath*

features

21 Using A.R.D. to Softkey Mars Cars

This fun little maze-type game reboots everytime Reset is pressed. This and others that do the same thing can be softkeyed by using the Anti-Reboot Disk. *by Chris Chenault*

22 How to be the Writemaster

Our hardware special feature shows you how to take command of the write-protect status of your disk by adding your own switch. *by William Wingfield Jr.*

core

24 Wheel of Money

The Wheel of Fortune TV game show comes to your computer screen. *by Jiro Feingold and Ray Darrah.*

departments

4 Input

6 Most Wanted List

6 Bugs

7 Readers' Softkey & Copy Exchange

Softkeys for: Spectrum Holobyte's *Gato* by *Clay Harrell*, Microlab's *Dino Eggs* by *Jay O'Donnell*, Electronic Art's *Pinball Construction Set* by *Steve Hite*, Avalon Hill's *TAC* by *Ken Black*, Broderbund's *The Print Shop: Graphics Library* by *Greg Schroeder*, Micro Lab's *Death In The Caribbean* by *Nick Galbreath*.

input

Please address letters to:

Hardcore COMPUTIST
Editorial Department
PO Box 110846-K
Tacoma, WA 98411

Include your name, address and phone.

Correspondence appearing in the INPUT section may be edited for clarity and space requirements. In addition, because of the great number of letters that we receive and the small size of our staff, a response to each letter is not guaranteed.

Unlocking Lode Runner's 150 Levels

I have found a way to unlock Lode Runner's 150 levels. You should start by write protecting the original disk, just to be safe. Then get out your best Bit copier and copy tracks 3-B. Then get out a sector editor and select the DOS 3.3 PATCHED option. Copy sectors C, 0-C, 5. Now you can look at and edit any levels you can't get passed.

Paul Kippes
Colby, KS

Hitchhiker's Adventure Tips

Look behind the bar.

The dog appears to be hungry.

Have a drink or three.

What can you block the robot panel with? (Hint. It's not yours.)

A cleaning robot would probably miss some things if there was some way to have the mail in the air while the babel fish is in the air too.

You'll need the atomic vector plotter.

Get some "Common Sense".

The long, dangly bit could do with a spot of

tea, or at least something that tastes almost, but not quite entirely unlike tea.

The guards can't kill you if their rifles are blasted.

The seat cushion feels lumpy.

Steer your boat at something dangerous.

Don't be afraid to say your name to the Beast.

Cover up! The Beast can't stop you from carving your name if it can't see you.

Who am I? <RETURN>

It is essential to give back a towel one has borrowed if that person's planet is about to blow up.

Have a talk with Prosser before going to the pub with Arthur.

Rob

A Bunch of APT's From France

First I want to tell you that I really like your magazine. It brings lots of interesting information. So keep up this good work.

As you can see, your magazine is also famous in France. I have, at the moment, an Apple IIe, two drives, a modem and a mouse. I'm 17 years old and like games very much but most of the time I'm not patient enough to play till I end the game. Here are a few APT'S I found.

The APT'S are of course only possible on unlocked softwares (readable with Inspector or Disk fixer). It is also possible that my patches won't work because of a difference of version. Anyway, I believe they will work on many versions.

CONAN:

Trait	Track	Sector	Byte	New Value
Men	\$05	\$09	\$F5	EA EA EA
Axes	\$05	\$0B	\$4E	EA EA EA

KARATEKA:

Trait	Track	Sector	Byte	New Value
# of >	\$09	\$08	\$53	85 B1 A9 1A 8D B0
	\$0C	\$08	\$6A	nn < 1B
	\$0F	\$08	\$5A	nn < 1B
	\$12	\$08	\$4D	nn < 1B
	\$09	\$07	\$9B	EA EA EA EA
	\$0C	\$07	\$C6	EA EA EA EA
	\$0F	\$07	\$50	EA EA EA EA
	\$0F	\$07	\$47	nn

MINER II:

Trait	Track	Sector	Byte	New Value
Lives	\$02	\$06	\$E0	nn
	\$02	\$01	\$38	EA EA EA
Level	\$02	\$06	\$D6	nn < 10

CONGO BONGO:

Trait	Track	Sector	Byte	New Value
Lives	\$05	\$03	\$4E	nn

MINER 2049ER:

Trait	Track	Sector	Byte	New Value
Lives	\$02	\$07	\$58	nn

THE HEIST:

Trait	Track	Sector	Byte	New Value
Lives	\$02	\$07	\$92	nn

That's it for the moment. Thanks.

Remi Grandjean
France

WANTED: Silky Sylc expose'

Is there anyone out there who can do an expose on SYLC files. Microsoft uses them in their spreadsheet MULTIPLAN. Unfortunately very few of any other programs use this file format. While MULTIPLAN is friendly, fast and can perform complicated logic calculations, it won't graph like the Visicalc/Visiplot combination. Microsoft indicated some time ago that they wanted to have a program to graph for the Apple II but they only wrote it for the Mac. Is there an easy way to convert SYLC files to a DIF format, or a program that can read SYLC files and plot directly?

Robert Classon
Mendon, MA

The Disk Organizer Meets The 65C02

Yours is easily the best magazine on my shelf, and I'm going to continue my subscription as long as I keep my computer, or until you run

input

out of colors for your cover, whichever comes first.

I purchased the 65C02 from your offer, and have found one problem you may want to tell your readers, they may have had the same: My copy of "Disk Organizer 2.0" from Sensible Software has ceased to work. I have "backups" of it, and none of them worked. This is easily the best directory editor I have, (it's also the only one I have. But anyway.) and I wondered if any of the other readers had experienced this.

As for your most wanted list, just about every single one of those programs I've seen around, cracked by somebody or another, and the only one that I haven't (actually, I have, but the cracked version didn't work) is Flight Simulator II. It will copy with any nibble copier that supports manual speed changing (I used EDD, copied according to the listed parameters), but copying tracks 0-22 manually can be a pain in the ASCII.

Lastly, for you adventurers out there, if the hints here aren't enough, there is an excellent book written by Kim Schuette, The Book of Adventure Games, which has helped me through all the adventures I have. It comes complete with maps and answers to problems, and short reviews. AT \$19.99, it's a bit (byte>) expensive, but worth it.

Thanks a lot, and keep up the good work.

Mike Whistler
Del Mar, CA

Mr Whistler: Shortly after installing a 65C02 in our computers, we noticed the same problem. It occurs because the folks at Sensible Software thought they would be tricky and take advantage of the 6502's bug mentioned on page 20 of Hardcore COMPUTIST No. 18.

To cure the problem search your disk for the sequence "A9 80 8D FF 02 A9 0A 8D 00 02" and replace the last byte with \$03.

Print Shop Printer Problems

A word of praise and then a couple of questions. Your stuff is great, exactly the reason I'm a confirmed Hardcore subscriber!

Question #1: More on DOS 3.3 will be greatly appreciated but please, what mechanism causes the stuff on the disk to go where it's supposed to? Is it equivalent to entering Monitor and typing it in?

Question #2: I agree with you that PRINT SHOP is great but can you tell me how to get it to do its work without an extra line feed after

each line of graphics?

My printer card is a "Parallel Printer Card" by MicroSystems Engineering of Kokomo, Indiana and it is not listed on PRINT SHOP's menu. I've tried the Menu software for all of the printer cards they list, but none of them will let me avoid throwing the hard switch in my Epson printer and that's not easily done. Can you advise me of a more convenient approach? I'll be grateful.

Ray Werner
Blue Ash, OH

Mr. Werner: A routine in DOS 3.3 called the file manager is responsible for putting things on a disk where they are supposed to go. I suggest you read a book from Quality Software (21601 Marrilla Street, Chatsworth, CA 91311) entitled "Beneath Apple DOS."

Secondly, several people have written to us about incompatibility problems with their printer and "The Print Shop." We at Hardcore know of no way to alter "The Print Shop" so that it will work with other printers. But the idea would make a heck of an article.

A Different Wizardry

I just got Issue #20 and the Softkey for SARGON III worked great. My Wizardry disk was dated 22-Jan-82. The correct values for the boot side are shown below. Sector edit track 22 sector 04 (the routine is the same as in #20 only the checksums are different).

```
A0 00 AE 28 8B BD 29 8B
85 0D 91 02 E8 C8 BD 29
8B 91 02 85 0E E8 8E 28
8B 60 00 07 12 B8 07 0F
12 17 09 0F 12 B7 07 12
12 12 21 0E 12 B6 07 03
12 15 09
```

To get these values I performed the boot code trace as in HC #20, page 29. However, please note the address values in step #6 should be B6 A6 & B6 AE, not B9 A6 & B9 AE. Also before step #7 (6827G) I had to store a 00 in memory location B6B6 (i.e. B6B6:00). For my disk the correct values were stored between memory locations B6B7-->B6CE.

I repeated the above procedure a second time and got a somewhat different set of checksums. However, these also seem to work. Apparently the checksums may change slightly due to small variations in disk speed. Although I really don't know the exact reason for these variations.

Second set of checksums effective on boot side:

```
0C 12 B6 07 18 12 18 09
0D 12 B6 07 11 12 12 21 10
12 B9 07 0A 12 12 09
```

Also I have an Apple //e so to reset I had to hit control-reset twice really fast. Oddly enough, sometimes I got only the first \$10 values and not the \$18 I have shown above, I am not sure why but since your article showed \$18 checksum values I assumed these to be correct.

To copy side 2, i.e. the scenario master side I put a jump to B69E in track 13 sector E bytes 08 09 0A, i.e. a 4C 9E B6 [replacing BD 88 C0] I simply repeated steps 1 through 7 in the article as I previously described, put the scenario master in when prompted, and hit control-reset twice when the screen prompted insert master/duplicate. Listing from B6B6 revealed the following new bytes:

```
1D 12 4C 09 21 12 EC 07
```

So for a side 2 backup I did the following track 13 sector 00 starting byte C0 as before the routine is as published in HC #20.

```
AE DC 20 BD DD 20 85 0D
8D 00 20 E8 BD DD 20 8D
01 20 85 0E E8 8E DC 20
ED E8 C0 60 00 1D 12 4C
09 21 12 EC 07
```

Both sides are COPYAable and both seem to work. Many thanks to HCC & TACO Van/Eperen for the ground work. I still spent several hours on this and I hope it will be helpful to their readers.

J. C.
Arlington, VA

Assorted Adventure Tips

ZORK II

WELL, that riddle is pretty tuff, isn't it?

The Dragon just needs a bump on the head to get him to follow you, and why not cool him down while you're at it?

Mats can be useful under the door, and if you can't get the door open, why not think of it as a letter, open it.

Boy, that Lizard has a sweet tooth!

Oddly angled rooms seem to remind me of "take me out to the ballpark".

input

MINDWHEEL

That woman's riddle makes me crow!
The audience wants a rose, give it to them.
Choose the disc, and use it wisely, it can cut,
and kill!
The toad has wisdom, and a big mouth!
That door puzzle could take me for eternity!
You have a GREAT publication! Keep up the
good work!

Ben Youngdahl

Notes on Sensible Speller and COMPRESS

I recently got to look at the ProDOS Sensible Speller. I was able to use the information in Hardcore COMPUTIST Issue No. 16 (Deprotecting Sensible Speller for ProDOS). I found that the Checksum routine occurs in several places on the diskette. There are four places that the routine can be found. One of the places is actually split across two sectors and can not be found by looking for the first part of the checksum routine. I just happened to find it by accident, while looking for just the last part of the routine. I was using Copy II Plus as my main copy program. The following is a track and sector list of where I found the first three routines:

track	sector	byte
\$0A	\$C	\$67
\$18	\$6	\$68
\$1E	\$4	\$68

The last one is split across two tracks starting at:

track	sector	byte
\$0F	\$E	\$\$F

and continuing to:

track	sector	byte
\$0F	\$D	\$00

I have found it useful in some cases to change the third and/or second byte of a multibyte instruction to a NOP (\$EA) instruction. This helps in disassembly later.

On another note, I have some software from a company called COMPRESS. It is a tutorial series on chemistry. Besides using strange address and data headers, which are different for some tracks, they also employ an encrypted directory and DOS commands. It has been resistance to the usual bit copiers, Copy II Plus, EDD, and LockSmith. From what I have seen so far, the main copy protection is in the first three tracks. The DOS is almost normal, except

for the encrypted DOS commands and the copy protection. Can anyone out there give me any hints as to where to go from here?

George S. Mochizuki
Madera, CA

Backing up Gutenberg

While I can not help your readers to break GUTENBURG, the makers of Copy II Plus (ver 5.0) can. This letter is written on a copy of the GUTENBURG master disk created by Copy II Plus using the built in parameters on that disk. The only difference I can find between the master and the copy is that the Serial number has been lost.

C. G. Ivey, Jr.
Salt Lake City, UT

bugs

HC #21:

DB Master 4 Plus Article:

The Sector edit step should say 24 EA's instead of 23 EA's

HC #22:

The Macro Editor:

On page 27, a line of source code reading "92C3: 02 93 .DA M.V" is missing.

Super IOB v1.5:

Lines 340 and 600-620 of the Super IOB checksums should be shaded.

Most Wanted List

Need help backing-up a particularly stubborn program?

Send us the name of the program and its manufacturer and we'll add it to our Most Wanted List, a column (updated each issue) which helps to keep Hardcore COMPUTIST readers informed of the programs for which softkeys are MOST needed. Send your requests to:

Hardcore COMPUTIST
Wanted List
PO Box 110846-K
Tacoma, WA 98411

If you know how to deprotect unlock, or modify any of the programs below, let us know. You'll be helping your fellow Hardcore COMPUTIST readers and earning MONEY at the same time. Send the information to us in article form on a DOS 3.3 diskette.

Mouse Calc Apple Computer
Apple Business Graphics Apple Computer
Flight Simulator II Sub Logic
Factory Sunburst Communicating
Jane Arktronics
Bookends Sensible Software
Visiblend Microlab
Sundog FTL Games
Sundog, V2 FTL Games
Lifesaver Microlab
Catalyst Quark, Inc.
Gutenberg
Jr. & Sr. Micromation LTD
Prime Plotter Primesoft Corp.
Zardax Computer Solutions
The Handlers Silicon Valley Systems
Milliken Math Series (NEW) Milliken Publishing
The Apple's Core: Parts 1-3 The Professor
King's Quest Sierra On-line
Hayes Terminal Program Hayes
Fun Bunch Unicorn

readers' softkey & copy exchange

Clay Harrell softkeys..

Gato

Spectrum HoloByte Inc.
1050 Walnut, Suite 325
Boulder, CO 80302

Requirements:

Apple //e or //c with 128K (program requirement)
At least one disk drive
COPYA
A sector editor
A blank disk
Gato disk

Gato is an excellent WWII submarine program! It plays very nicely, and even has a built-in fake "spreadsheet" screen so that it looks like you're busy with financials -in case you're using Gato at work! The protection used on Gato is also very good, and it could involve a lot of time to deprotect because of the subtle protection that is deep in the program.

The Gato DOS lives in the RAM card around \$D000, but the protection code lives at \$B65A. You may run COPYA and copy the original disk, and it will almost run...

The routine at \$B65A does a nibble read on track \$11, looking for a byte sequence of \$AA AB. If the sequence is not found, it loads the accumulator with #\$00 and pushes it on the stack. Then the program locks up. But if the correct bytes are found, it loads the accumulator with #\$01 and pushes that on the stack. Then the program continues happily along.

So we must find this routine on the disk, and change it so that no matter what, it pushes #\$01 on the stack instead of \$00. Using a disk search utility, I found the offending code on track \$15, sector \$C, byte \$8C.

To deprotect Gato, do the following:

- 1) Copy the original Gato disk to a blank disk using COPYA with no modifications.
- 2) RUN your favorite sector editor and make the following byte changes to your COPYA copy of Gato:

Track	Sector	Byte	From	To
\$15	\$C	\$8C	\$AD	\$A9
\$15	\$C	\$8D	\$E9	\$01
\$15	\$C	\$8E	\$C0	\$D0
\$15	\$C	\$8F	\$A9	\$67

- 3) Write the sector back out to your COPYA copy of Gato.

This code will load the accumulator with #\$01 and then branch to the end of the routine. This is the best (and safest way) to disable the protection. And you're all done!

Jay O'Donnell softkeys...



MicroLab, Inc.
2699 Skokie Valley Road
Highland Park, IL 69935

Requirements:

48K Apple][
Blank disk
Dino Eggs Disk
Super IOB v1.5

Dino Eggs was reviewed nicely by Ray Darrah in Hardcore COMPUTIST no. 10, so I won't bother with describing the game here. Inspection of the address and data epilogues revealed that they had been altered. Copying the disk with Super IOB was no problem from there, but the disk would not run. There were several locations in the DOS that checked for these markers. You could attempt to search for and change all these places, but in the case of this game, two changes will do. Both tell the checking routines (one in DOS, one in a normally empty location in DOS) to ignore the epilogue and other errors.

Install the controller here into Super IOB v1.5 and copy the Dino Eggs disk. It uses the Fast routine in version 1.5. You now have a COPYA-able version of Dino Eggs.

controller

```
1000 REM DINO EGGS CONTROLLER
1010 TK=0 :LT=35 :ST=15 :LS=15 :CD=WR :FAST
      =1
1020 GOSUB 490 : RESTORE : GOSUB 170 : GOSUB 610
      : GOSUB 310
1030 GOSUB 490 : GOSUB 230 : GOSUB 610 : IF PEEK
      (TRK) = LT THEN 1050
1040 TK=PEEK (TRK) : ST=PEEK (SCT) : GOTO 1020
1050 HOME : PRINT "COPYDONE" : END
5000 DATA 158 ,231 ,213 ,170
5010 DATA 2*CHANGES
5020 DATA 0 ,9 ,66 ,24
5030 DATA 0 ,11 ,117 ,24
```

controller checksums

1000 - \$356B	1050 - \$E9F8
1010 - \$2544	5000 - \$D707
1020 - \$B9B0	5010 - \$3BF4
1030 - \$E89B	5020 - \$2253
1040 - \$FECE	5030 - \$B300

Steve Hite's softkey to...

Pinball Construction Set

Electronic Arts
2755 Campus Dr.
San Mateo, CA 94403
\$39.95

Requirements:

Apple][Plus or //e
Copy program such as Locksmith 5.0
Disk sector editing program

Pinball Construction Set is a fascinating program that allows you to construct and play your own pinball machines. You control sound, scoring, colors, gravity, bounce, and all other aspects of the game. The complexity of your control is such that the game has even been referred to as a low level programming language. Unfortunately, backing up this program can be quite a problem. Although parameters are available for Nibbles Away II, Locksmith 5.0, Essential Data Duplicator, and other copy programs, several different protection schemes have been employed during different releases of this game. I found that none of the parameters would give me a working copy. The technique described here not only gave me a working copy, but resulted in a copy that is COPYA-able.

- 1) Make a copy of the entire disk. Locksmith 5.0 Fastcopy works just fine. You can use Nibbles Away II, Essential Data Duplicator or Copy II Plus, employing just the normal default parameters, if you wish. The important thing is to ignore any errors you get on track 6. COPYA will not work. It dies due to an I/O error on that track.

2) Using a disk sector editing program, read track 3, sector 6 on your copy. I used ZAP from Bag of Tricks, but any editor with similar capabilities will do. Look for the pattern A0 01 20 CF 17 20 D9 1E. This pattern occurs only once in this sector, and is found in the last half. If you are using ZAP, the offset is +92. This code disassembles as follows:

```
LDY #01
JSR $17CF
JSR $1ED9
```

- 3) Change the \$A0 to \$60 (an RTS in machine language) and write the modified sector back to the disk.

That's it. You now have a COPYA version of Pinball Construction Set.

readers' softkey & copy exchange

Ken Black's softkey for...

T A C Tactical Armored Command

Avalon Hill Microcomputer Games
4517 Harford Rd.
Baltimore, MD 21214

Requirements:

48K Apple II
Super IOB v1.5
TAC disk
An initialized DOS 3.3 disk

TAC (Tactical Armoured Command) is one of those irritating programs that is hard to get a good bit copy back-up. I found out it's easier to deprotect (softkey) it than to make a bit-copied backup. This procedure is easily and quickly done with Super IOB's NewSwap controller.

The procedure is pretty straightforward but will require some changes to make it work properly. Not only were epilogues and sync bytes on the disk changed, but the Applesoft programs on it were written with altered DOS commands. Rather than change the programs, it is easier (and less time consuming) to copy the altered commands to DOS 3.3. This can be done by copying track 1, sector 7 to the new disk.

This is how to do it:

- 1) Turn on your computer and press Reset.
- 2) Enter the monitor and move the boot code into RAM.

CALL -151
9600<C600.C6FFM

- 3) Alter the re-located boot code to jump into the monitor after the RWTS is loaded in.

96F8:A9 00 8D 4A 08 4C 01 08

- 4) Boot up the TAC disk partially.

9600G

- 5) You should be in the monitor with the disk still spinning. Move the TAC RWTS to a safe location.

6000<B800.BFFFM

- 6) Boot normal DOS 3.3 with a small or no HELLO program and BSAVE the TAC RWTS.

6[P]
BSAVE RWTS.TAC,A\$6000,L\$800

- 7) If you haven't already done so, INIT a blank disk with DOS 3.3.

INIT HELLO

- 8) Install the controller following this softkey into Super IOB v1.5 and copy TAC onto your freshly INITIALized disk (don't format).

- 9) Have fun with your backup copy.

If you **really** wanted to, you could use a totally normal DOS or even a high-speed DOS and change the commands "FIXDOSFL", "GETIT", and "KEPIT" found in the Applesoft programs to MAXFILES, BLOAD, and BSAVE, respectively. In this case, do NOT copy TAC's track 1, sector 7 to the new disk.

controller

```
1000 REM TAC CONTROLLER FOR S-IOB V1.5
1005 TK=1:ST=7:CD=WR:GOSUB 490:GOSUB 360
1006 GOSUB 430:GOSUB 100:GOSUB 490:GOSUB 360
:GOSUB 100
1010 TK=3:LT=35:ST=15:LS=15:CD=WR:FAST
=1
1020 GOSUB 360:GOSUB 490:GOSUB 610
1030 GOSUB 360:GOSUB 490:GOSUB 610:IF PEEK
(TRK)=LT THEN 1050
1040 TK=PEEK(TRK):ST=PEEK(SCT):GOTO 1020
1050 HOME:PRINT "DONE":END
10010 PRINT CHR$(4)"BLOAD RWTS.TAC,A$1900"
```

controller checksums

1000	- \$356B	1030	- \$0A0B
1005	- \$8D68	1040	- \$3EA4
1006	- \$0680	1050	- \$0FA5
1010	- \$D5D2	10010	- \$2BFA
1020	- \$389F		

Greg Schroeder's softkey for...

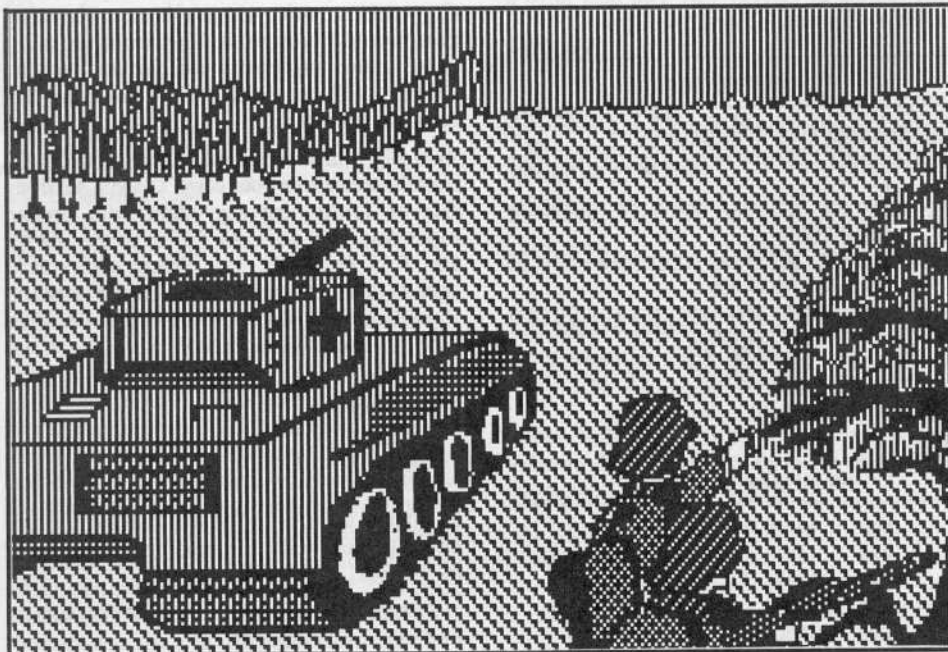
The Print Shop: Graphics Library

Broderbund Software
17 Paul Drive
San Rafael, CA 94903
\$49.95

Requirements:

Apple II
Super IOB v1.2

After reading the Tutorial For Disk Inspection in Hardcore 16, I decided to attempt to write a controller for the Graphics Library disk for Print Shop. The disk is in standard format for tracks \$1-\$22 and can be copied normally. On track 0 sectors \$0-\$F the data field epilog bytes have been changed to DE A9 instead of the normal DE AA. I attempted to use line 1024 of the super controller (in the tutorial) to normalize this track but my controller didn't work, so I used line 1028 to ignore the epilogs instead. This is the routine that was dropped when Super IOB was updated in Hardcore 14. I find this routine very useful and am glad it was included in the Super Controller. I was even more impressed with it when I used it on side two of the Graphics Library disk and copied it, even though each side used a different protection method. I hope this softkey is some help to you.



readers' softkey & copy exchange

Controller

```

1000 REM GRAPHICS LIBRARY CONTROLLER BY
      GREG SCHROEDER.
1010 TK = 0 : ST = 0 : LT = 35 : CD = WR
1020 T1 = TK : GOSUB 490
1028 POKE 47405, 24 : POKE 47406, 96 : POKE 47497
      , 24 : POKE 47498, 96
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST <
      DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 490 : TK = T1 : ST = 0
1069 POKE 47405, 208 : POKE 47406, 19 : POKE
      47497, 208 : POKE 47498, 183
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
    
```

Controller Checksums

1000 - \$356B	1060 - \$93DC
1010 - \$3266	1069 - \$735F
1020 - \$C11A	1070 - \$7B5A
1028 - \$4D7F	1080 - \$F3CF
1030 - \$5B7E	1090 - \$7A41
1040 - \$C1A5	1100 - \$7692
1050 - \$8FF0	

Nick Galbreath's softkey for...

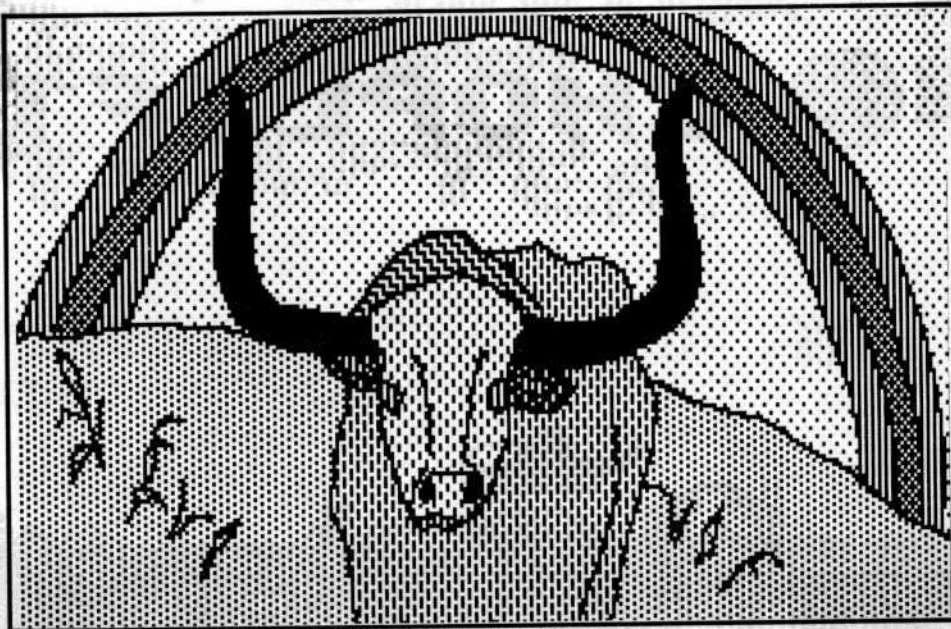
Death In The Caribbean


Micro Lab
2699 Skokie Valley Rd.
Highland Park, IL 60035
\$35.00

Requirements:

Apple II
Super IOB v1.2
Two blank disk sides

Death in the Caribbean is a beautiful graphic adventure game; however, like many adventure games (or just in general) it is protected. Its protection involves a drastic changing of header and epilog bytes on tracks \$1-\$22 and reset plugs. Amazingly enough, Micro-Fun forgot to plug Control-C! So the first part of our softkey would be to get the RWTS routine by breaking into BASIC. Do so by following these instructions:



1) Boot up your copy while holding down .

2) When you break into BASIC, enter the monitor by typing

CALL-151

3) Move the RWTS routine into "safe" memory by typing

6800<B800.BFFF

4) Boot up a slave disk.

5) Save the RWTS routine with

BSAVE RWTS.DIC, A\$6800, L\$0800

Run Super IOB with the swap controller listed at the end of this article installed. Copy both sides of Death in the Caribbean. The controller copies track 0 as is, then beginning with track 1 starts swapping the RWTS in and out until done. During the copy procedure, Super IOB changes track 0, sectors 2 and 3 as in the following table:

Track	Sector	Byte	From	To
\$00	\$02	\$E7	\$96	\$D5
\$00	\$02	\$F1	\$D3	\$AA
\$00	\$02	\$FC	\$E5	\$AD
\$00	\$03	\$35	\$96	\$DE
\$00	\$03	\$3F	\$FF	\$AA
\$00	\$03	\$55	\$D3	\$D5
\$00	\$03	\$5F	\$96	\$AA
\$00	\$03	\$6A	\$F2	\$96
\$00	\$03	\$91	\$D3	\$DE
\$00	\$03	\$9B	\$B2	\$AA

Death in the Caribbean is finally (at last!) cracked. Enjoy!

controller

```

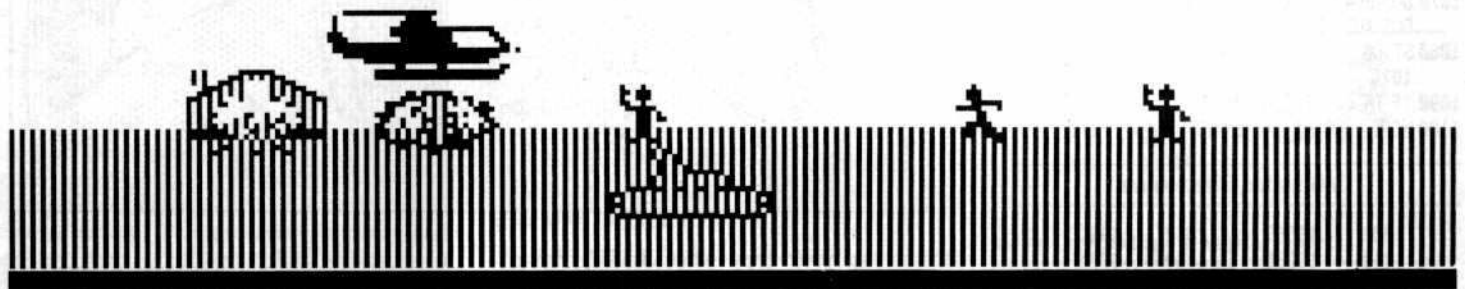
1000 REM DEATH IN THE CARIBBEAN
1010 TK = 0 : ST = 0 : LT = 35 : CD = WR
1020 T1 = TK : GOSUB 490 : IF TK > 0 THEN GOSUB 360
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST <
      DOS THEN 1030
1040 IF BF THEN RESTORE : GOSUB 310 : GOTO 1060
1050 ST = 0 : TK = TK + 1 : ON (TK = 1) GOSUB 360 :
      IF TK < LT THEN 1030
1060 GOSUB 490 : TK = T1 : ST = 0 : GOSUB 360
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 "COPYDONE!" : END
5000 DATA 10*CHANGES, 0, 2, 231, 213
5001 DATA 0, 2, 241, 170
5002 DATA 0, 2, 252, 173
5003 DATA 0, 3, 53, 222
5004 DATA 0, 3, 63, 170
5005 DATA 0, 3, 85, 213
5006 DATA 0, 3, 95, 170
5007 DATA 0, 3, 106, 150
5008 DATA 0, 3, 145, 222
5009 DATA 0, 3, 155, 170
10010 IF PEEK (6400) <> 162 THEN PRINT CHR$ (4
      ) "BLOOD" RWTS.DIC, A$1900"
    
```

Controller Checksums

1000 - \$356B	5000 - \$9B4F
1010 - \$3266	5001 - \$2285
1020 - \$2060	5002 - \$1A1B
1030 - \$3661	5003 - \$1F5A
1040 - \$5472	5004 - \$2770



CHOPLIFTER



by Todd C. Wilson

Choplifter
Broderbund Software
17 Paul Drive
San Rafael, CA 94903
\$34.95

Requirements:

48K Apple II
DOS 3.3
1 Blank disk
Choplifter disk

Flying low along the ground to avoid the satellites, I spotted a group of people. Landing quickly, I helped them inside. Rumbling over the plains came a squadron of tanks. Suddenly a jet fired its missiles at me. Time to go. Jerking up on the control stick, I sent the chopper skyward. Just missed me! Whoops! Satellites are coming in, too low to dodge. Firing at one, I backed up and crashed into the other. Blast! The chopper lost control and spun down to crash into the desert. Ten people lost. Next chopper, please. I am, of course, playing Choplifter. The only difference is that the one I'm playing is deprotected.

According to the electronic grapevine,

Choplifter is so heavily protected that even a copy card has a hard time copying it. EDD III can copy the entire disk, but the booting time is twice as long and has a tendency to fail.

Being a software "purist", I tend to shy away from copy-cards, due to the fact that I am too cheap to buy one and that none of them work on an Apple IIc. And for a matter of fact, neither does Choplifter. Snarl, snarl!

But that is all in the past. Choplifter is about to join the ranks of deprotected software. If you wish to skip the main bulk of this article and get right to the softkey, be my guest. You can always read this later.

How Broderbund Protected Choplifter

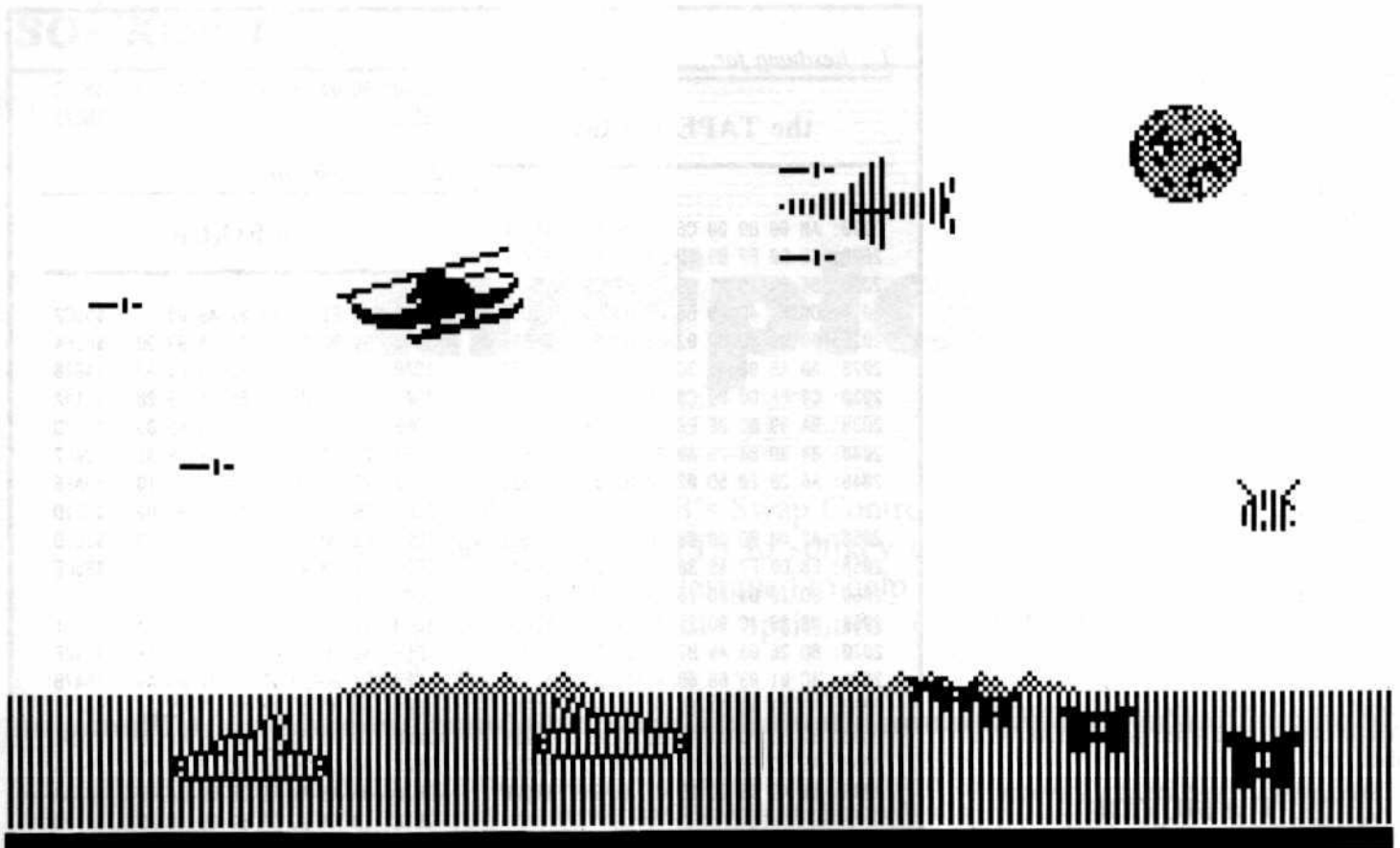
When the disk controller card loads in and runs track \$00, sector \$00, the code copies itself from \$800 to \$200 - which is also the location of the keyboard buffer! It then messes around with the data in \$800, and loads the second sector into \$300 and transfers execution to \$301. \$301 to \$328 builds a RTS table and some code in \$100 to \$1FF (the stack), and then goes to \$131 through a jump via RTS' code. \$131 to \$1FF does many, many things. First it clears text page 1 & 2, and copies ROM Basic into the language card, if one is present. Then the BRK & reset vectors are set to the disk controller card. \$188 to \$18F is a very good piece of misdirection. It seems to call a routine at \$190 and then exit via a RTS to \$405. But it DOESN'T! That code is just there to fool you.

At \$190 it loads in 4 pages of code into \$400 to \$7FF - text page #1. This is Standard Operating Procedure. This is done so that, if you stop the program, the screen and the code will be erased by whatever you type. The code is read in from the disk by the common 4 x 4 format (See: Quality Softwares beneath APPLE DOS, pages 3-13 to 3-21 on Data Field Encoding). Not only is this fast as heck, many other DOSs use this - Atari, Commodore, etc.

Transfer is then given to \$4A3 from \$1FB to \$1FF, again by a RTS. This code starts out by copying \$500 to \$5FF into \$200 to \$2FF (notice the keyboard buffer use again) and \$600 to \$6EF into \$300 to \$3EF. It then resets the character hooks (\$36,\$37) and keyboard hooks (\$38,\$39) to \$326, which zaps the memory and reboots if anything is printed or the keyboard is read. Then it appears to clear out \$C00 to \$C000, but it doesn't - it clears out \$C00 to \$C7F, then \$D80, \$E81, \$F82, ect., on up to \$C000. It seems Broderbund made a mistake. The opcode at \$4F2 should be a BNE instead of a BPL.

In any case, control is transferred by some nifty code to \$440. At \$440, the hi-res screens are cleared out, and the title page is loaded up, plus some code at \$6000 to \$63FF. By using an in-line DECREMENT memory opcode, control is given to \$400, not \$700 as it appears. \$400 gobbles up the stack, and loads some more code into \$500. By this time, \$200 and \$201 hold the value of \$6300, which is the next piece of code to be run. Onward!

The code at \$6300 could have been left out,



and I would have been happy. The only reason for this code is to check for an Apple][,][+, or //e Monitor ROM.

This is what got me mad in the first place! This code is responsible for Choplifter not booting on a //c. The //c's ROM doesn't match with any of the "known" ROMs. Result: Reboot and try again.

When the above routine exits, the new code at \$500 is run. This code is the best piece of protection ever concocted! By using the fast 4 x 4 format, the code makes some in-line mods and THEN runs them. The result is quite different than that expected. Control is given to \$4000, which does 3 instructions, then goes to \$566.

At last! This is the code that loads in the main program. The program is loaded into \$800-\$1FFF, and then into \$6000-\$BFFF. Control is given to \$6BF, which loads in the necessary zero-page values, then checks the game for errors. If the checksums match,

control is given to \$800 - and Choplifter is run! (Whew!)

The Softkey

The softkey is the result of 8 hours worth of work; writing what is known to the trade as a tapeworm. A tapeworm is simply a program that can insert itself into another program, monitor and/or stop it when it gets so far. This is exactly what the softkey does. It boots the disk, then loads in the second sector, copies it to \$3000, modifies it, runs it, etc. When the tapeworm gets so far, it replicates itself, finishes its work, copies the zero page to \$1F00, and then -- dies. Its work is done, and thus it commits suicide so that we may gain from it.

The tapeworm was built by moving the disk controller card into \$B600-\$B6FF, and then modifying it to enter the Monitor instead of \$801. Then, by using some code, a printer, lots and lots of paper, logic, more code, and just plain luck, the boot code was traced. By

copying parts of the code and feeding it to the tapeworm, it grew large enough to handle everything the protection threw at it.

When the Monitor bell rings and the drive stops, we have in memory a copy of Choplifter (\$800 to \$1EFF and \$6000 to \$BFFF) and the zero-page values at \$1F00 to \$1FFF. Here's how to get all of Choplifter:

1) Boot a DOS 3.3 disk, INIT a slave disk with a blank HELLO file, and delete it.

**INIT HELLO
DELETE HELLO**

2) Type in the TAPEWORM, FIXUP, and MOVER hexdumps listed (listings #1, #2, and #3). Save as follows:

**BSAVE TAPEWORM,A\$2000,LS149
BSAVE FIXUP,A\$1D91,LSA9
BSAVE MOVER,A\$2000,LS4C**

3) Load the tapeworm program

BLOAD TAPEWORM

4) Insert your Choplifter disk into drive 1, slot 6 and start the tapeworm on it's journey through Choplifter's intestines.

CALL 8192

5) When the Monitor beeps (if the cursor looks funny, ignore it), check and make sure that no errors have occurred:

2000<4000.5FFFV

You should get no coding listed. If you do, or the Monitor will not accept your input, reboot the slave disk and go back to step 3.

Choplifter A.P.T. Control Keys

- [A]** -change the X-axis of the joystick
- [V]** -change the Y-axis of the joystick
- [R]** -reset the game to demo mode
- [U]** -toggle unlimited choppers option
- [1]** -plus 1 number: change playing level (0 to 3)
- [2]** -plus 2 numbers: change the number of seats in your chopper to the 2-digit number (00 to 64. 64 and up will give you the maximum amount of seats)

Sometimes, for unknown reasons, the tapeworm will not work. This is shown by garbage on the hi-res screen. If this happens, go back to step 3.

6) If everything seems fine, enter the following:

```
800:4C 00 20
0DAB:20 91 1D
2000<800.8FFM
2100<6000.BFFFM
```

This step patches the program so the Mover and APT programs will work and save page 8 from the reboot. The last command moves the code from \$6000 to \$C000 down to \$2100.

7) Insert your slave disk, and enter:

```
C600G
CALL-151
800<2000.20FFM
3D0G
BLOAD FIXUP,A$1D91
BLOAD MOVER,A$2000
BSAVE CHOPLIFTER,A$800,L$7900
```

That's it! Choplifter is now a BRUNable file. Just to make sure, the file length of the file should be 123 (that's the 3-digit number just left of the filename when you CATALOG a disk).

The MOVER

The MOVER program sets the reset vector to the Monitor, and then moves the code from \$2100 to \$80FF into \$6000 to \$BFFF, but it does it backwards. Starting with the code at \$8000, it moves it to \$BF00, then \$7F00 to \$BE00, etc. It then fixes the program so that an 800G from the Monitor will restart the game.

The FIXUP

The FIXUP program takes care of the one thing that Choplifter cannot do: redraw the screen. First it wipes the top 2 lines of hi-res page 1 and page 2 to black, and then redraws the ground at the bottom. Then it copies the saved zero page from \$1F00, and calls \$9021, which redraws the scoring block. The last line exits to \$12C1, which displays hi-res page 1.

And Some A.P.T.s

The patch at \$DAB calls the APT package. Pressing **U** will toggle the Unlimited Choppers option. It will also reset you to the first chopper. Pressing **P** and then two (2) numbers will change the amount of seats in your chopper. **P 01** will make a very hard game, while **P 66** will make a very easy game.

Pressing **R** will restart the program, by re-entering the demo mode.

Also, by pressing **U** and then a number from 0 to 3, you select the level you wish to play on. This is not part of the package, but was left in by the original author. This is useable from either the demo or during play.

1. hexdump for...

the TAPEWORM

```
2000: A0 00 B9 00 C6 99 00 B6 $E230
2008: C8 D0 F7 B9 1D 20 99 F8 $BA7F
2010: B6 B9 1D 21 99 F8 B7 C8 $B0DA
2018: D0 F1 4C 00 B6 A2 00 BD $7D3F
2020: 00 08 9D 00 02 E8 D0 F7 $EB85
2028: A0 AB 98 85 3C 4A 05 3C $5D52
2030: C9 FF D0 09 C0 D5 F0 05 $E5F0
2038: 8A 99 00 08 E8 C8 D0 EA $9FFB
2040: 84 3D 84 26 A9 03 85 27 $16C8
2048: A6 2B 20 5D 02 20 D1 02 $4039
```

```
2050: A2 00 BD 00 03 9D 00 30 $8561
2058: E8 D0 F7 A9 30 8D 0A 03 $EA01
2060: 8D 10 03 8D 16 03 8D 1C $9C55
2068: 03 A9 4C 8D 25 03 A9 56 $EC08
2070: 8D 26 03 A9 B7 8D 27 03 $20E6
2078: 4C 01 03 68 68 A9 4C 8D $1E1B
2080: FB 01 A9 6A 8D FC 01 A9 $53F2
2088: B7 8D FD 01 4C 31 01 A9 $C0B9
2090: AE 8D FB 01 A9 65 8D FC $5584
2098: 01 A9 05 8D FD 01 AE 65 $482F
```

```
20A0: 05 9A 68 68 A9 4C 8D 21 $7759
20A8: 04 A9 96 8D 22 04 A9 B7 $328F
20B0: 8D 23 04 A9 60 8D ED 04 $17B6
20B8: 4C A3 04 20 93 FE 20 89 $E64B
20C0: FE A2 00 8A 48 CA D0 FC $8BC4
20C8: A2 FF 9A A6 2B A9 FF 20 $2D88
20D0: 37 05 C9 DD D0 F7 20 39 $F463
20D8: 05 C9 F5 D0 F5 20 39 05 $3E93
20E0: C9 D5 D0 F5 20 30 05 8D $770B
20E8: CF B7 20 30 05 8D D0 B7 $8939
```

```
20F0: 20 30 05 8D FF FF C9 EA $8A56
20F8: D0 EA A2 43 BD E3 B7 9D $5F57
2100: BC 05 CA 10 F7 4C 73 05 $2037
2108: A9 FF 20 F6 06 C9 D5 D0 $69EF
2110: F7 20 F8 06 C9 FF D0 F6 $17D6
2118: 20 F8 06 C9 D5 D0 F5 20 $83A8
2120: EF 06 8D E3 05 20 EF 06 $EAD7
2128: 8D E4 05 20 EF 06 8D FF $AA45
2130: FF C9 EA D0 EA AD 82 C0 $8E26
2138: BD 88 C0 A2 00 BD 00 00 $4772
```

```
2140: 9D 00 1F CA D0 F7 4C 59 $5CFC
2148: FF $BBA3
```

2. hexdump for...

the FIXUP

```
1D91: 4C E1 1D A0 27 A9 00 $9DC2
1D98: 99 00 20 99 00 24 99 00 $F2E4
1DA0: 40 99 00 44 88 10 F1 A2 $4B28
1DA8: 19 BD 02 1C 85 26 85 28 $314A
1DB0: BD C2 1C 85 27 49 60 85 $D73D
1DB8: 29 A0 27 A9 55 49 7F 91 $90E7
1DC0: 26 91 28 88 10 F7 CA 10 $6A96
1DC8: E0 A2 00 BD 00 1F 95 00 $CE1D
1DD0: E8 D0 F8 A9 FF 85 79 20 $188D
1DD8: 21 90 A9 00 85 79 4C C1 $85CE
```

```
1DE0: 12 2C 10 C0 C9 92 D0 03 $0B31
1DE8: 4C 00 08 C9 95 D0 11 AD $290F
1DF0: B7 0B 49 40 8D B7 0B A9 $547B
1DF8: 00 8D 1C 73 68 68 18 60 $847A
1E00: C9 90 F0 01 60 98 48 AD $34D9
1E08: 00 C0 10 FB 2C 10 C0 29 $9464
1E10: 0F A8 A9 F6 18 69 0A 88 $8333
1E18: 10 FA 8D D1 60 68 A8 AD $432D
1E20: 00 C0 10 FB 2C 10 C0 29 $5350
1E28: 0F 18 6D D1 60 8D D1 60 $CA09
```

```
1E30: 8D 7C 61 8D A8 61 68 68 $4A9B
1E38: 18 60 $88A7
```

3. hexdump for...

the MOVER

```
2000: AD 00 C0 2C 10 C0 C9 9B $7FB8
2008: D0 03 4C 59 FF A9 59 8D $3004
2010: F2 03 A9 FF 8D F3 03 49 $C07C
2018: A5 8D F4 03 A0 00 84 00 $D4C4
2020: 84 02 A9 80 85 01 A9 BF $9E49
2028: 85 03 A2 60 B1 00 91 02 $749C
2030: C8 D0 F9 C6 01 C6 03 CA $F58A
2038: D0 F2 A9 20 8D 00 08 A9 $1087
2040: 94 8D 01 08 A9 1D 8D 02 $66EF
2048: 08 4C 00 08 $8AF9
```



mufplot

Just use the Super IOB's Swap Controller and a modified Mufplot RWTS to softkey this useful program. Mufplot is designed to help ham radio operators to predict optimum operating frequencies.

by Edward E. Harte, M.D.

Base(2) Systems

Requirements:

- Mufplot disk
- Super IOB v1.2
- A way into the Monitor
- A Blank Disk

The program Mufplot by Base (2) Systems is designed to aid ham radio operators in predicting optimum operating frequencies. It is a very useful program, the only drawback was the copy protection. The first copy of the program I used self-destructed after six months use, forcing another cash expenditure to replace it. This motivated me to de-protect the program.

Upon examining the disk I found the following protection schemes.

One, the address and data epilogs have been changed to FF FF. The program used normal DOS address and data headers, but with an interesting twist. Prior to the normal DOS header of D5 AA 96 is a pseudo header of D5 AA BF. This header apparently would mislead a nibble copy program, such as Copy II Plus.

The second protection scheme became apparent when I changed a normal DOS to account for the changed address and data epilogs. Even though the address and data marks were ok, cataloging the disk resulted in an I/O error. It turned out that a checksum error

was occurring.

As noted in the CIA files, a likely cause for checksum errors is a change in the byte translation tables. I booted up Mufplot and entered the monitor via my Wildcard II. A peek at the byte translation tables revealed the following changes. Memory location \$BA29 had AA instead of 96 and \$BAAA had 00 instead of AA. A quick insertion of the byte table changes, along with the above epilogue changes allowed inspection of the normal catalog. Needless to say, I felt like the explorer who found a treasure.

The technique to copy the disk presented here does not require utilization of the above information, but I included it to hopefully expand others' knowledge in cracking disk protection. To break the protection we will utilize the Swap Controller of Super IOB. This will allow us to use the modified Mufplot RWTS to read the disk and write to another disk with a normal RWTS.

The Technique

- 1) Boot a normal DOS disk.
- 2) Initialize a destination disk.
- 3) Boot the Mufplot disk.
- 4) At the menu stage, enter the monitor by whatever means.
- 5) Move the modified RWTS to a safe location.
1900<B800.BFFFM
- 6) Boot up a slave disk.
- 7) Save the RWTS to your Super IOB disk.

BSAVE RWTS.MUFPLOT, A\$1900, L\$800

8) Install the controller into Super IOB and run it. Do not use the format option.

9) Enjoy!

controller

```

1000 REM MUFPLOT CONTROLLER
1010 TK = 3 : ST = 0 : LT = 35 : CD = WR
1020 T1 = TK : GOSUB 490 : GOSUB 360 : ONERR GOTO
550
1030 GOSUB 430 : GOSUB 100 : ST = ST + 1 : IF ST <
DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0 : TK = TK + 1 : IF TK < LT THEN 1030
1060 GOSUB 490 : TK = T1 : ST = 0 : GOSUB 360
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 "COPY^ OK, ^ NO^ DOS^ ON^ COPY"
: END
10000 REM CONFIGURATION TIME
10010 PRINT CHR$( 4 ) ; "BLOAD^ RWTS.MUFPLOT,
A$1900"
    
```

controller checksums

2077	- \$0255	1070	- \$AF1B
1010	- \$EB5E	1080	- \$B299
1020	- \$2AD8	1090	- \$F6B4
1030	- \$3CD9	1100	- \$E2A5
1040	- \$E49C	10000	- \$8EF6
1050	- \$3CF3	10010	- \$8344
1060	- \$A71E		

Flashcalc

Flashcalc is one of the first products from Paladin (aka Visicorp) since the Software Arts - Visicorp lawsuit. This is not Visicalc, but it does have most all of the features plus full support for most 80 column cards. It is written by the same folks who wrote Magic Office System (distributed by Artsci, Inc.). Flashcalc is a ProDOS based program, and can serve as an introduction to ProDOS copy protection.

by Doni G. Grande

Paladin (Visicorp-Paladin)
2895 Zanker Rd.
San Jose, CA 95134
\$99.00

Requirements:

FlashCalc
COPYA
Any sector editor

Like many ProDOS programs, Flashcalc allows you to place it onto a hard disk. This means that Flashcalc must use ProDOS for most of its disk access, and probably checks the disk only when it is loading to insure an original disk is in the drive. Also, the protection scheme is probably either a nibble-counting type or a signature checking type. This is also supported by the fact that the original is COPYAable, but that the copy will not run.

To try to find the protection code, you must have a place to start looking. There are two possibilities: (1) ProDOS itself has been changed or (2) the startup code does the protection check. There is a simple way to check for #1:

1) Boot a standard ProDOS disk and get into Basic.

2) CATALOG the protected disk. You will probably be able to do this if a mostly standard ProDOS has been used.

3) Type

PREFIX,D1

to set the ProDOS prefix to that of the protected disk.

4) Now run the 1st "xxx.SYSTEM" file in the catalog by typing

-xxx.SYSTEM

If the program starts executing using this procedure, then you have a standard ProDOS and the program must be doing its own copy protection dirty work. The secret here is that ProDOS always runs the first file with the name "xxx.SYSTEM" when a disk is booted. By following the above steps, you have booted the original disk using the version of ProDOS in memory!

If you have come this far, then the protection code is buried in the program itself somewhere. So now you must try to catch the program executing the copy protection. When a COPYA Flashcalc is booted, ProDOS gets loaded and then it almost immediately re-boots the disk. This means that the protection scheme is very early in the boot process. The first place to look is in the "xxx.SYSTEM" file mentioned above (FC.SYSTEM on the Flashcalc disk). This can

be done by placing the protected disk in the drive and typing

BLOAD FC.SYSTEM,A\$2000,TSYS

Note that xxx.SYSTEM files always load in memory at location \$2000, and they are of type SYS. Now, lets look and see what Flashcalc does upon booting.

Enter the monitor and type 2000L. This will produce the following:

```
2000- A9 00 LDA #$00
2002- A2 15 LDX #$15
2004- 9D 59 BF STA $BF59,X
2007- CA DEX
2008- 10 FA BPL $2004
200A- A9 CF LDA #$CF
200C- 8D 58 BF STA $BF58
200F- A9 01 LDA #$01
2011- 8D 6F BF STA $BF6F
2014- AD 30 BF LDA $BF30
2017- 8D 26 20 STA $2026
201A- 20 00 BF JSR $BF00
201D- 80 ???
201E- 25 20 AND $20
2020- D0 F2 BNE $2014
2022- 4C 00 22 JMP $2200
2025- 03 ???
2026- 00 BRK
2027- 00 22
2028- 07 00
```

This code references locations \$BF58-\$BF6F in locations \$2000-\$2011. A close look at Quality Software's *Beneath Apple ProDOS* (BAP) will show you that this is where ProDOS keeps its map of free and used memory. ProDOS keeps this map as a bit-map. A one

bit means that that page of memory is in use, and a zero bit means that it is available. The following table shows the relationship between memory locations and the bit map.

Memory Page	\$BFxx location																	
	Lo	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Lo
0	58																	59
1	5A																	5B
2	5C																	5D
3	5E																	5F
4	60																	61
5	62																	63
6	64																	65
7	66																	67
8	68																	69
9	6A																	6B
A	6C																	6D
B	6E																	6F

The low page numbers are arranged in the order that ProDOS examines the bits within each byte of the bit-map. For example, at location \$200A, a \$CF is stored in location \$BF58. Examining the table above shows that \$BF58 controls memory from \$0000-\$0700. A \$CF would mean:

```
Page # 00 01 02 03 04 05 06 07
$CF = 1 1 0 0 1 1 1 1
```

Therefore pages 2 and 3 are available, and the rest are used. Continuing in this manner shows that Flashcalc is marking memory as available in the following chunks:

```
$800-$B7FF - Lines 2000-2008
$200-$3FF - Lines 200A-200C
$B800-$BEFF - Lines 200F-2011
```

After all this memory management, the data from location \$BF30 is stored at \$2026. Again, looking at BAP shows that \$BF30 is the last slot and drive accessed in the form DSSS0000, where D=0 is drive 1, D=1 is drive two, and SSS is 1-7 to indicate the slot number. This is beginning to look interesting. The only reason to get a slot and drive number is if a disk access is about to be done!

Location \$201A bears out this suspicion: a JSR to \$BF00, the entry point of ProDOS's machine language interface (MLI). A full description of the MLI can be found in BAP, so only the pertinent information will be given here. Immediately after a JSR to the MLI, ProDOS expects to find two pieces of data: (1) A single byte indicating what function the MLI is to perform, and (2) a two byte address in lo, hi form pointing to where the MLI can find the rest of the parameters for this call.

In this case, the first byte at \$201D indicates a call to the READ_BLOCK function and the next two bytes show that the address of the parameters is \$2025. The parameter block and

interpretation for the READ_BLOCK command is:

location	value	meaning
\$2025	\$03	Three parameters follow
\$2026	\$xx	Slot and drive parameter
\$2027	\$2200	Address to load data to
\$2029	\$0007	Block number to read

Note that the byte at \$2026 was filled with the correct value by the code at \$2014-\$2017. This call means that the MLI will load block 7 from the disk to the address \$2200. If this call is successful, the zero flag will be set and the branch at \$2020 will not occur. Then, the JMP \$2200 will jump into the code just read!

To find out what happens at \$2200, re-route the jump at \$2022 to jump to the monitor by typing:

```
2023:59 FF
```

Then start the program with

```
2000G
```

The disk should come on and then you will get the monitor prompt back.

Now, list the code at \$2200 by typing

```
2200L
```

The first few instructions will be:

```
2200- A0 00 LDY #00
2202- B9 00 22 LDA $2200,Y
2205- 99 00 40 STA $4000,Y
2208- C8 INY
2209- D0 F7 BNE $2202
220B- 4C 0E 40 JMP $400E
220E- 20 75 40 JSR $4075
```

This code moves memory from \$2200 to \$4000 and then jumps to this new code at \$400E. To see what is moved, type

```
220C:59 FF
2200G
```

You will quickly find yourself back in the monitor, but the code will have been moved. List the code at \$400E. The first thing you see is a JSR \$4075, so without looking any further, type

```
4075L
```

It begins with:

```
4075- 20 91 40 JSR $4091
4078- 90 01 BCC $407B
407A- 60 RTS
407B- AD 30 BF LDA $BF30
407E- 29 70 AND #$70
4080- 4A LSR
4081- 4A LSR
4082- 4A LSR
4083- 4A LSR
4084- 09 C0 ORA #$C0
4086- 85 37 STA $37
4088- A9 00 LDA #$00
408A- 85 36 STA $36
408C- A9 A0 LDA #$A0
408E- 6C 36 00 JMP ($0036)
```

First, look at the code at \$407B-\$408E. Remember that \$BF30 contains the drive and

slot number in the form DSSS0000. With this in mind, and assuming that you boot from drive 1 in slot 6 so that this should be 01010000, follow the code. You will find that, by the time you reach \$408A, locations \$36-\$37 will contain 00C6, which will cause \$408E to jump to C600. This reboots the disk! We have found part of the protection code!

Now, look back at \$4075. It contains a JSR \$4091, then if the carry flag is clear, it executes the rebooting code. This must mean that the carry flag is set if all is okay, and cleared otherwise. One more excursion is in order. List the code at \$4091-\$40DC. I'll leave it up to you to do a full deciphering of this code. However, I will tell you that it looks for the following bit pattern on the disk:

```
10101010 11101011 00...000 11101100
$AA $EB $00 $EC
```

The special thing to note here is the string of zero bits between \$EB and \$EC. Normally, ProDOS cannot read or write a string of zeros reliably. In fact, if ProDOS read the above, it would see just \$AA EB EC, and completely ignore the zeros! Apparently, the protectors of Flashcalc have written the string of zeros using a special write routine. The software just looks for the sequence \$AA EB 00 EC. This will not be found on a disk copied by COPYA and some bit copiers. This type of protection I call signature analysis, because the protection code looks for a certain signature on the disk.

At the end of the protection code, you can see the following sequence (without the comments, of course):

```
40D1- CMP #SEC Is byte a SEC?
40D3- BNE $40D8 Go if not
40D5- SEC Signal everything OK
40D6- BCS $40D9 jump always
40D8- CLC Signal bad disk
40D9- LDA $C088,X Turn off the drive
40DC- RTS Return to caller
```

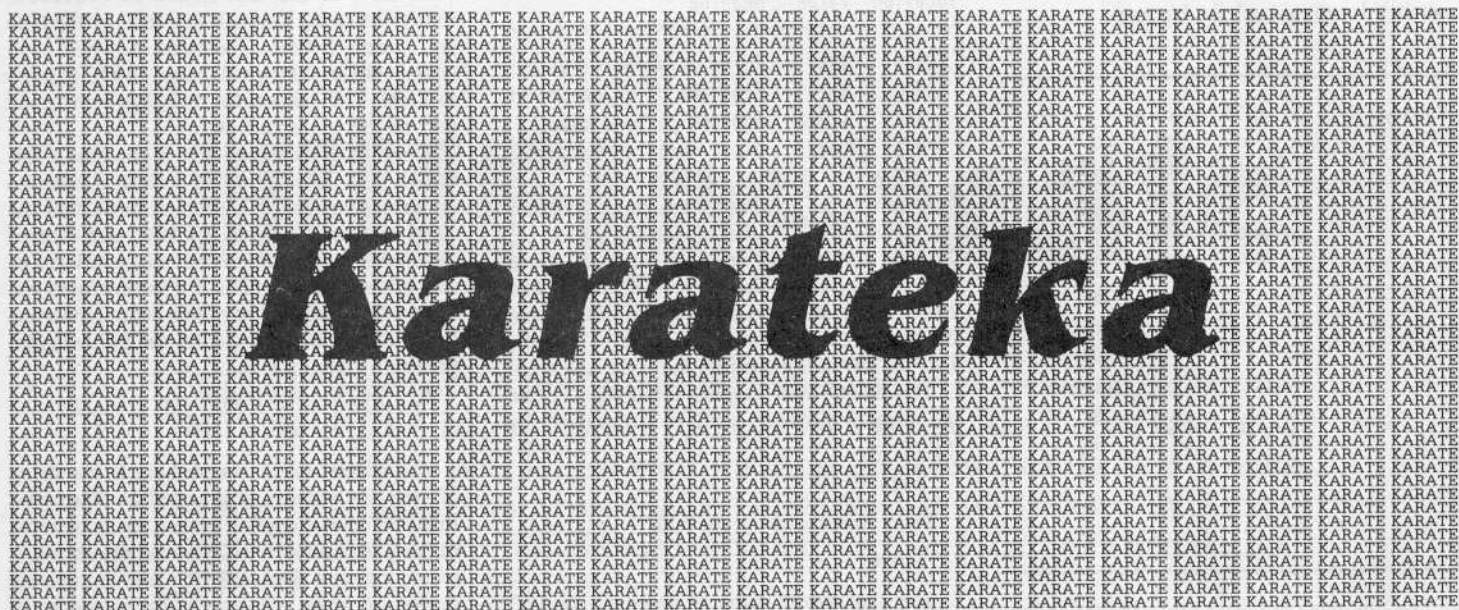
Since we want to set the carry flag no matter what, the only change would be to change location \$40D8 from a CLC (\$18) to a SEC (\$38). This would cause the calling routine at \$4075 to think that everything was just fine. Believe it or not, Flashcalc is effectively protected by just one byte!

The procedure to make a copyable version of Flashcalc is as follows: 1) COPYA the original Flashcalc disk. 2) Use a sector editor to make the following change. Note that ProDOS uses a different notation for disk access. I have given the ProDOS block number, the DOS 3.3 sector number, and the actual Physical sector number for the patch. Most Zap programs use the DOS 3.3 sector notation, so if you don't know which one to use, DOS 3.3 is your best guess.

```
ProDOS Block: Block 7
DOS 3.3 Sector: Track 0, Sector 1
Physical Sector: Track 0, Sector D
```

```
Change byte $D8 from $18 to $38
```

softkey to...



by Danny Pollak

Broderbund Software
17 Paul Dr.
San Rafael, CA 94903

Requirements:

Apple II Plus with 64K
Karateka
One blank disk

Karateka is a recent release from Broderbund Software which has set the trend for upcoming games. With smooth graphics, great sound effects, and a realistic action, how could this game miss? Broderbund didn't overlook a single minor detail. In his review of Karateka in issue no. 19, Jeff Hurlburt closed with this statement: "Simply put, Karateka is a breakthrough. Art merges with technology to produce a game almost as much fun to watch as to play."

Their Protection

An examination of the Karateka disk shows that the game resides from Track \$00 to Track \$20. Track \$01 contains a loader which reads in another loader from Track \$02.5. Then, depending on the value in the accumulator, the second loader loads in a portion of the game. With this in mind, I set out to deprotect Karateka.

The Softkey

1) The first thing we need to do is to initialize the blank disk so that it will BLOAD the program when booted. Boot an initialized disk

(preferably with a fast DOS). Then insert the blank disk and type in the following:

```
CALL -151
9E42:34
INIT KARATEKA
DELETE KARATEKA
```

2) The next thing we need to do is move the disk controller ROM down to RAM where we will modify it so that it will exit into the Monitor upon loading Boot1.

```
9600<C600.C700M
96FA:98
9801:4C 59 FF
```

3) Execute the modified controller by entering

```
9600G
```

4) Boot1 has now been loaded into memory at \$800. Move it to \$9800 and modify it so that it will also exit into the Monitor.

```
9800<800.900M
9805:98
9839:93
9301:4C 59 FF
```

5) Execute the controller at \$9600 again.

```
9600G
```

6) The code at \$9800 has now loaded another section of code at \$300. This code moves a portion of itself to \$1000, jumps to \$1000, and then reads data from the disk and stores it from \$200 to \$7FF in memory. At \$3BE there is a jump to \$300. We will move this code at \$300 to \$9300 and modify it so that instead of jumping to \$300, it will move the code from \$200-\$7FF to \$2100-\$26FF and then exit to the Monitor.

```
9300<300.400M
9305:93
93BE:4C 00 60
```

```
6000:A2 00 BD 00 02 9D 00 21
:E8 D0 F7 EE 04 60 EE 07
:60 AD 04 60 C9 08 90 EA
:4C 59 FF
```

7) Execute the disk controller for the last time.

```
9600G
```

8) The code that was read in at \$300 is responsible for reading data from Track \$01 and storing it from \$4000-\$41FF in memory. It then jumps to the subroutine at \$4000 which reads data from Track 2.5 and stores it from \$5000-\$57FF. The routine at \$5000 loads a certain portion of the game depending on the value in the accumulator.

We can alter the code at \$300 so that it will exit into the Monitor instead of jumping to \$4000. Do this by entering:

```
390:4C 59 FF
300G
```

9) An examination of the code at \$4000 shows a jump to \$5000 at \$401E. Alter this code to exit into the Monitor and then execute it.

```
401E:4C 59 FF
4000G
```

10) Examining the code at \$5000 shows that depending on the value of the accumulator, execution will either fall through to the jump to \$5300 or branch past the jump statement and continue executing. Before the jump to \$300 back at step 6, the accumulator was loaded with \$80. This value would cause the execution to fall through to the jump to \$5300. The code at \$5300 loads the main portion of the game and then jumps to \$200. We will alter the code so that after it loads the game, it will exit to the Monitor.

```
5320:4C 59 FF
5300G
```

11) The main program is now in memory located at \$800-\$FFF and \$6000-\$9FFF. We will compact the code and then boot the initialized disk.

```
2700<800.FFFM
2F00<6F00.9FFF
6F00:00
6F01<6F00.70FFM
6☐
```

12) Enter the following hexdumps. The routine at \$2000 is the memory move which will put the program back in its proper locations. The routines at \$6F00-\$70FF will be moved to \$F600-\$F7FF on the language card. The routine at \$6F00 loads the picture of the castle. The routine at \$6F20 loads the files for each section of the game. The routine at \$7000 swaps the area occupied by DOS with all of Bank 2 on the language card and part of Bank 1. This routine is used when DOS is needed to load a program.

```
2000: 2C 81 C0 2C 81 C0 A2 00 $76EB
2008: BD 00 F8 9D 00 F8 E8 D0 $C920
2010: F7 EE 0A 20 EE 0D 20 AD $349E
2018: 0D 20 C9 00 00 EA 2C 83 $396E
2020: C0 2C 83 C0 BD 00 6F 9D $C0FB
2028: 00 F6 E8 D0 F7 EE 26 20 $CC0E
2030: EE 29 20 AD 29 20 C9 F8 $7110
2038: 90 EA 20 F0 9F 20 00 F7 $6D36
2040: A2 00 BD 00 21 9D 00 02 $CF4F
2048: E8 D0 F7 EE 44 20 EE 47 $8AC2
2050: 20 AD 47 20 C9 10 90 EA $447D
2058: BD 00 2E 9D 00 6E EB D0 $3F90
2060: F7 EE 5A 20 EE 5D 20 AD $CAF6
2068: 5D 20 C9 A0 90 EA A9 00 $CA52
2070: AA 95 00 E8 D0 EB A9 01 $5ABC
2078: 85 86 85 46 2C 50 C0 2C $4162
2080: 57 C0 2C 52 C0 2C 55 C0 $5E79
2088: 4C 0C 02 $C167
```

```
6F00: 20 00 F7 A2 00 BD 17 F6 $1D91
6F08: 9D 75 AA E8 E0 07 D0 F5 $4FE1
6F10: 20 72 F6 20 00 F7 60 D0 $B983
6F18: C9 C3 D4 D5 D2 C5 00 00 $1FEF
6F20: 18 69 B0 8D 6D F6 20 00 $BD8D
6F28: F7 A2 00 BD 69 F6 9D 75 $BA16
6F30: AA E8 E0 08 D0 F5 20 72 $983F
6F38: F6 A9 C1 8D 7A AA 20 72 $4AAA
6F40: F6 20 00 F7 A9 40 8D 52 $A094
6F48: F6 A9 A0 8D 55 F6 A2 00 $1494
```

```
6F50: BD 00 40 9D 00 A0 E8 D0 $77CB
6F58: F7 EE 52 F6 EE 55 F6 AD $97E0
6F60: 55 F6 C9 C0 90 EA 4C 2C $393C
6F68: 02 CC CF C1 C4 A0 A0 A0 $6259
6F70: A0 A0 AD 78 04 8D 00 F0 $9198
6F78: AD 7E 04 8D 01 F0 AD F8 $B0F8
6F80: 04 8D 02 F0 AD FB 04 8D $EEBA
6F88: 03 F0 AD 78 05 8D 04 F0 $DEF3
6F90: AD F8 05 8D 05 F0 AD F8 $C7D5
6F98: 06 8D 06 F0 AD EC B7 0A $BF96
```

```
6FA0: 8D 7E 04 20 93 03 20 5D $5EF8
6FA8: A3 20 93 03 AD 00 F0 8D $4000
6FB0: 78 04 AD 01 F0 8D 7E 04 $D733
6FB8: AD 02 F0 8D F8 04 AD 03 $A3C3
6FC0: F0 8D FB 04 AD 04 F0 8D $925B
6FC8: 78 05 AD 05 F0 8D F8 05 $7E96
6FD0: AD 06 F0 8D F8 06 60 $1D97
```

```
7000: 2C 83 C0 2C 83 C0 A9 A0 $50B9
7008: 85 FD A9 D0 85 FF A0 00 $8A45
7010: 84 FC 84 FB B1 FC A8 B1 $71AF
7018: FE 91 FC 68 91 FE C8 D0 $758B
7020: F3 66 FD E6 FF A5 FF C9 $50F1
7028: F0 90 E9 2C 8B C0 2C 8B $1E9E
7030: C0 A9 96 85 FD A9 D0 85 $233D
7038: FF B1 FC 48 B1 FE 91 FC $C567
7040: 68 91 FE C8 D0 F8 E6 FD $B050
7048: E6 FF A5 FF C9 DA 90 E9 $6C2F
7050: 60 $97DC
```

13) The next thing that must be done is to alter the Karateka code so that it will jump to our routines when it goes to load a portion of the program.

```
210E:4C 20 F6
214B:20 00 F6
2299:F1
229C:F1
```

14) That's it for now. Now all that is left to do is to save the file and we will have the main program stored as a single file on disk.

```
BSAVE KARATEKA,AS2000,LS5100
```

15) We're not done yet. Now we must capture the picture and the other portions of the game which are loaded in during play. We will let the loader from Karateka do this for us. For the loader to work, the disk read head must be located at Track \$01. We must use DOS to cause the read head to seek to Track \$01 and alter our main program so that after it has moved itself to its original location, it will exit to the Monitor.

```
300:20 E3 03 4C D9 03
B7EC:01 00
B7F4:00
300G
2088:4C 59 FF
2299:4F
229C:4F
2000G
```

16) We will now use the loader which is located at \$300 to load each section of the game. After we load each section, we will use our swap routine at \$F700 to move DOS back into place and then save the files to our initialized disk. At this point it would help to have two disk drives, but it is not mandatory. If you only have one drive, then you must insert your initialized disk, save the files for that section, seek the read head back to Track \$01 and then put the Karateka disk back into the disk drive. If you have two disk drives, then you need only to insert your initialized disk into drive 2 and save the files. The read head on drive 1 will still be at Track \$01. Insert the Karateka disk into drive 1 and type in the following:

```
390:4C 59 FF
300G
401E:4C 59 FF
4000G
☐
:00
5000G
```

```
4000<A000.BFFFM
```

```
C083
F700G
A851G
```

17) The files are now in memory. If you have two disk drives, then complete this section and skip step 18. If you have only one disk drive, then go to step 18. Insert the initialized disk into drive two and enter the following:

```
BSAVE LOAD0,AS4000,LS2000,D2
BSAVE LOAD0A,AS1000,LS1000
9EE0G
F700G
C0EA
```

18) Insert the initialized disk into drive one and enter the following:

```
BSAVE LOAD0,AS4000,LS2000
BSAVE LOAD0A,AS1000,LS1000
6000:A9 B7 A0 E8 4C B5 B7
B7EC:01 00
B7F4:00
6000G
9EE0G
F700G
```

Reinsert the Karateka disk into drive one.

19) There are five different loads during the game and also the picture. We have just saved the first load to disk under the names LOAD0 and LOAD0A. To get the other four sections loaded and saved to disk, you need to loop back up to step 16. There are two things that will be done differently. After you type ☐, you will enter the number of the section (0-4) that you want to load. You already loaded section 0 so the next number will be 1. The only other change that must be made is when you save the files. Replace the 0 with whatever number section that you loaded. For instance, you would use BSAVE LOAD4,AS4000,LS2000 and BSAVE LOAD4A,AS1000,LS1000 if you had just loaded in section four. Loop back to section 16 and follow the instructions doing the changes described above. After you have done this for all five sections, go to step 20.

20) The only thing left to do is to capture the picture file on disk. Do this by entering the following:

```
3A5G
2000<A000.BFFFM
C083
F700G
A851G
BSAVE PICTURE,AS2000,LS2000
```

A Final Word

Now you have a completely COPYA-able version of Karateka. Maybe someone out there can come up with some APTs which will stop you from being hit or continue the game if you get killed. There are many jump tables throughout the main code which could help in your search for a certain routine. Good Luck!

softkey to...

NEWSROOM

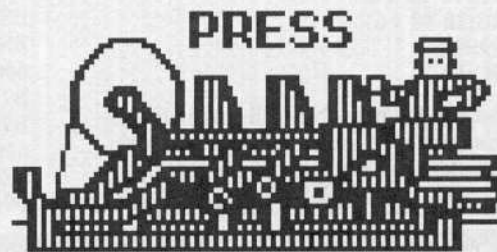
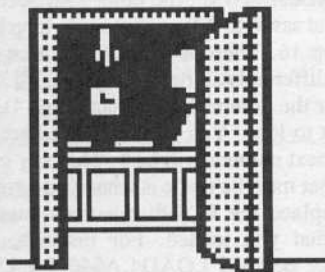
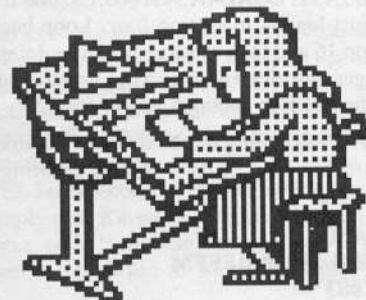
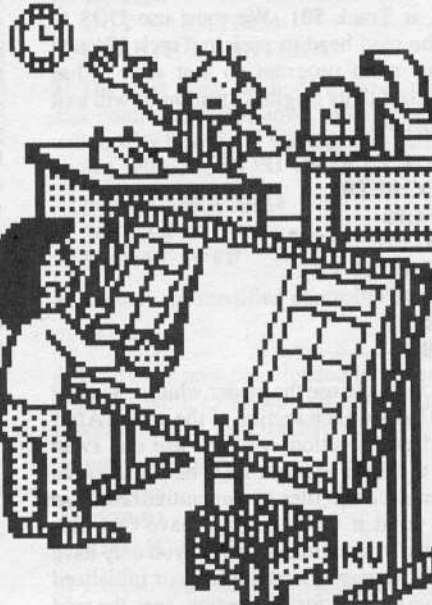


PHOTO LAB



WIRE SERVICE



BANNER

LAYOUT

by Doni G. Grande

Springboard Software Inc.
7807 CreekrIDGE Circle
Minneapolis, MN 55435
\$49.95

Requirements:

Any Apple with at least 64K of memory (required by Newsroom)
Super IOB v1.5
A sector editor with search functions

The Newsroom is a terrific program for producing newsletters. Unfortunately, the program disk must be left in the disk drive at all times, and it cannot be easily copied. This is a great deterrent for schools, as a spilled Coke quickly does the disk in! The only alternative is to deprotect Newsroom.

Delving into the Newsroom

The Newsroom can be partially deprotected by use of a Super IOB controller. However, the final copy will not run because of four reasons.

1. The Newsroom switches between normal and modified DOS to access different disks. Patches to switch DOS's were entered at \$BA69 and \$BA7F. Since we want Newsroom to only access normal DOS disks (including itself), these patches must be circumvented by placing an RTS as the first byte in the routines.

2. The Newsroom uses drive three to load parts of the program. As we all know, there is no drive three, and DOS 3.3 loudly signals this with an error: Upon closer examination, I discovered that Newsroom sets up a RAM disk as drive three on a 128k Apple and copies most

Newsroom includes a large file of "clip art" and other graphic utilities to help you produce newsletters. Deprotect this handy graphic package by using the Super IOB v1.5.



of it's programs to this RAM disk upon bootup. That is why program transfer is so quick using a 128k Apple. On a JI Plus, the programs are loaded from the disk drive when needed. Drive three is used by the Newsroom DOS to designate the RAM drive.

3. Newsroom checks a disk before accessing it to be sure that the desired disk is in the drive. There are three possible outcomes to this check:

a. Program Disk: verified by presence of modified address epilog marks and the presence of SSI NR on track \$11 sector \$00.

b. Clipart Disk: verified by presence of normal address epilog marks and SSI CLIP on track \$22 sector \$00.

c. Storage Disk: verified by presence of normal address epilog marks but not Clipart disk.

Notice that the presence of the modified address epilog marks (\$ADBB instead of \$DEAA) is checked, but then a double check is made in all the above cases. Unfortunately, the program checks for these modified epilog marks first, and then does the other checks. This means that to defeat the protection scheme, the routine which checks for the modified epilog marks must be changed to always return the correct value corresponding to the disk which is supposed to be in the disk drive. The second check will then determine which disk is actually in the drive. Very close (and long) examination of Newsroom's code revealed that location \$3AD contains the expected value returned by the protection code AND-ed with \$80. \$3AD contains \$80 if altered epilog marks (program disk) are expected and \$82 if normal epilog marks are expected. The protection code must return the value at \$3AD AND \$7F and the carry flag clear whenever it is called.

4. On the latest version of Newsroom (the one that allows you to convert hi-res pictures in the photo program) there is a nibble count during boot up.

Step By Step

As usual, we can now use our findings to deprotect the disk. Following is a step by step procedure which you can follow to copy the Newsroom.

1) Install the Super IOB controller at the end of this article into your copy of Super IOB v1.5 and RUN it. Note that if you have the later version of Newsroom (the one that allows hi-res picture conversions in the photo section), you will have to change the 15 in line 1100 to a 20.

2) Search the disk for the following bytes and change to the indicated sequences. Note that they will be found in more than one location on the disk, so be sure to search the entire disk!

Search for: AE E9 B7 BD 8E C0 BD 8C C0 A0
Change to: AD AD 03 29 7F 18 60 EA EA EA

Search for: 2C 50 C0 A9 01 85 FB 20 24 1F
Change to: 2C 50 C0 A9 00 38 B0 09 EA EA

Note that the second sequence will only be found if you have the later version of Newsroom as it is the nibble count routine.

About The Controller

The Newsroom controller normalizes the Newsroom DOS by performing sector edits which simulate the routine to change to normal DOS. If you have the later version of Newsroom, there are more changes that have to be done to Newsroom so that it will read normal disks. That is why you must make the 15 in line 1100 a 20.

In addition to normalizing the Newsroom DOS, the controller also disables the DOS switching routines by placing an RTS as thier first instruction.

The two POKES of line 1010 set DOS to the super extended retry mode. I found that because of the strange sector marks being used by

Newsroom, the sectors are harder than normal to read with normal DOS. Therefore, some sectors will speed by and some sectors will take a while to read. Hopefully the super extended retry mode will make DOS patient enough to read every sector of your Newsroom original.

Finally, if your clipart disks aren't COPYAble, you will have to deprotect them by using the Super IOB controller and removing the "GOSUB 1100" in line 1020.

Many Thanks to Glen Tatum

controller

```

1000 REM NEWSROOM CONTROLLER
1010 TK=0:LT=35:ST=15:LS=15:CD=WR:FAST
    =1:POKE 48573,0:POKE 48588,208
1020 RESTORE:GOSUB 170:GOSUB 490:GOSUB 610
    :IF TK=0 THEN GOSUB 1100
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
1100 FOR A=1 TO 15:READ A1,A2:POKE A1,A2:
    NEXT:RETURN
5000 DATA 173,187,237,187
5010 DATA 13470,222,13475,170
5020 DATA 13109,222,13119,170,13201,222
5030 DATA 13204,234,13205,189,13206,140,
    13207,192,13208,16,13209,251,13210,
    201,13211,170
5040 DATA 12905,96,12927,96
5050 DATA 12469,222,12467,170,12462,222
5060 DATA 12221,48,11985,40
    
```

controller checksums

1000	- \$356B	5000	- \$1542
1010	- \$1EB4	5010	- \$A6B7
1020	- \$5698	5020	- \$314B
1030	- \$6301	5030	- \$51B5
1040	- \$F4AD	5040	- \$DC3E
1050	- \$9EF2	5050	- \$AA65
1100	- \$FCB7	5060	- \$B72F

E-Z Draw

This 1980 graphics system has some useful functions which more recent programs still lack and it has some sophisticated protection schemes, too.

by Nick Galbreath

Sirius Software, Inc.
10398 Rockingham Dr., Ste. 12
Sacramento, CA 95827

Requirements:

Any of the Apple][line
COPYA or an equivalent sector copier
A blank disk

E-Z Draw, though very old (published in 1980), is still a complete graphics package and has some very useful functions which the more modern programs lack. It also has a very sophisticated protection scheme which, though not hard to copy, appears to be very hard to crack. If you are only interested in the cracking go to the end of this article.

The Scheme

As mentioned in my Controller Writer article (HC #16), E-Z Draw uses eight different address headers per track and the epilog bytes change nearly every sector. This changes every track so there is really no pattern. It is F8 ROM protected by loading its own monitor routine into the RAMcard and its RWTS routine is nonstandard so using Demuffin or a Swap controller produces only "I/O ERROR."

This discouraged me, until I finally did get into the monitor and found how E-Z Draw reads in sectors. It ANDs every byte with \$D5 (the normal byte for an address header). If the result is \$D5 then it has found the first byte in the

address header. It also does a similar routine for the last byte in the address header (\$96).

The Crack

This is very complicated, since the protection was very complicated, too. If you're a novice in the field of cracking and machine code, I suggest going to the cookbook instructions at the end.

First off, enter the monitor and disable the checksums:

B92D: 18 60
B989: 18 60

We need some space in DOS so we can add some extra code to read in the E-Z Draw sector. A good place is the \$B600 area since it is only used when putting on DOS after initialization. To free it enter:

B74A: 60
A56B: 60

The second change prevents a Hi-file from being saved during initialization; this just saves time. And now, our patch:


B600:29 D5 C9 D5 60 29 96 C6
B608:96 D0 03 4C 6D B9 4C 54
B610:B9

Check your typing against the following:

B600- 29 D5	AND #\$D5
B602- C9 D5	CMP #\$D5
B604- 60	RTS
B605- 29 96	AND #\$96
B607- C9 96	CMP #\$96
B609- D0 03	BNE \$B60E
B60B- 4C 6D B9	JMP \$B96D
B60E- 4C 54 B9	JMP \$B954

To enable this patch enter:


B954: 20 00 B6 EA
B969: 4C 05 B6 EA

Go back to basic with  and run COPYA. When done copying, you will have cracked it! It will boot up just fine, however I suggest adding a new DOS since one can then RESET out of the program normally.

Also if one needs more space on the diskette, UNLOCK and DELETE FPBASIC and APPLESOFT for 57 sectors. To get 50 more sectors, run a program that rebuilds the VTOC (like FIXCAT or Disk Recovery) or copy all the files (yuk!) onto a new disk.

Recap

Boot up a DOS 3.3 System Master diskette.

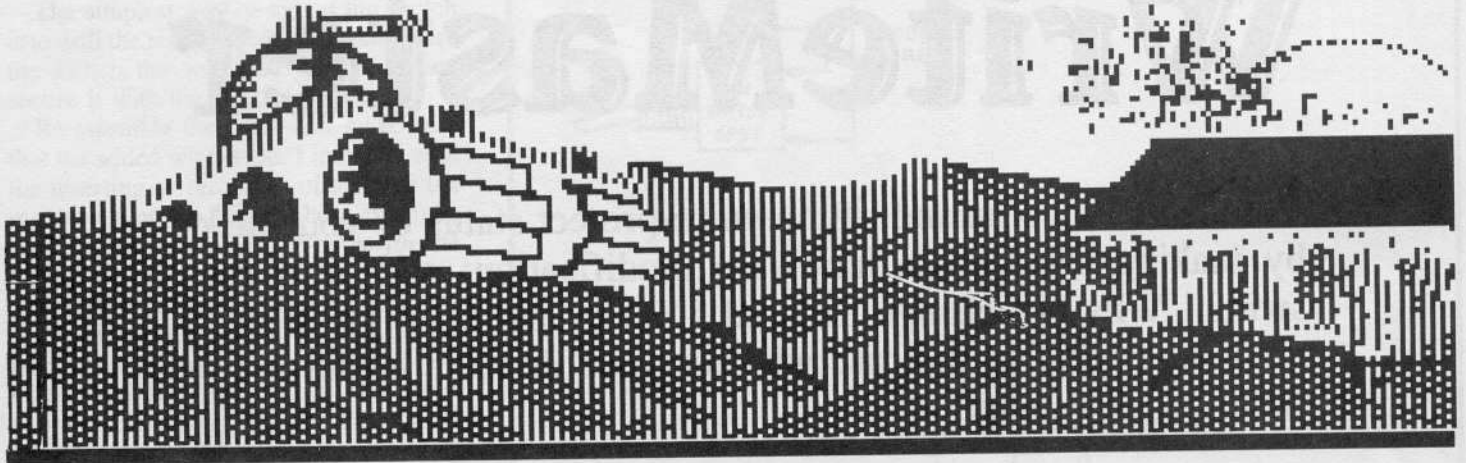
CALL-151
B92D: 18 60
B989: 18 60
B74A: 60
A56B: 60
B600: 29 D5 C9 D5 60 29 96 C9
B608: 96 D0 03 4C 6D B9 4C 54
B610: B9
B954: 20 00 B6 EA
B969: 4C 05 B6 EA


Run COPYA and copy the E-Z Draw master disk.

BRUN MASTER CREATE

And add a new DOS with the hello file named "SIRIUS SOFTWARE (C) 1980". Copy complete!

MARS CARS



by Chris Chenault

Mars Cars
Datamost, Inc.
8943 Fulbright Ave.
Chatsworth, CA 91311

Requirements:

Apple II
48K slave disk
A blank disk
A sector editor
Mars Cars disk

Your first question after reading the title must be "What in the world is an ARD?" Well, the ARD (Anti-Reboot Disk) is a disk that's had its boot sector modified so that it fails to boot the rest of the DOS, but jumps to the Monitor instead. Some disks reboot after Reset is pressed. The ARD prevents this. Most of the memory is left alone except for \$800-8FF, \$300-355, and the Monitor's favorite hangouts in Zero Page. In this way, the RWTS could be captured for use in Super IOB. It can also be used to capture a single load program if it does not need those locations I mentioned. As an example, we will use this to crack Mars Cars from Datamost.

Preparing the ARD

- 1) Initialize a blank (or unwanted) disk.
- 2) Run your sector editor and then make the following changes to track 0, sector 0:

Byte	From	To
01-	A5	AD
02-	29	E8
03-	C9	C0
04-	09	4C
05-	D0	59
06-	18	FF

- 3) Write it back to the disk and you now have an official Anti-Reboot Disk!

How it Works

When a disk first boots up, the ROM on the controller card reads track 0, sector 0 into \$800-8FF then it jumps to \$801, where it normally continues the boot process. That is what we're interested in. In step 2 we changed the code stored in the first sector so that it turns off the drive and jumps into the Monitor for you. Here it is:

```
801- AD E8 C0 LDA $C0E8
804- 4C 59 FF JMP $FF59
```

Complicated, isn't it? \$C0E8 is the soft switch that turns off the drive, and \$FF59 is, of course, the entry point into the Monitor. From there, the rest of the sector is ignored. Once it reaches the Monitor, the computer thinks everything is normal except for the lack of DOS.

Using It

Using the ARD is easy. After the game is loaded and running, remove the program disk and replace it with the ARD. Hit Reset and let the disk do its magic. After the drive stops you should be in the Monitor and free to do what you wish. From here you can move the game or RWTS out of the way and boot normal DOS from there.

Deprotecting Mars Cars

Mars Cars is a fun (but lame) little maze-type game. The main difference between this and other maze games is that the game's walls are also edible. If you're in trouble, you can eat a hole through a wall and get away.

To deprotect Mars Cars we require a Mars Cars disk, the ARD, and a blank slave disk with no HELLO program.

- 1) Boot the Mars Cars disk. Press a key to get past the title picture and the irritating tank noise. You should now be at the menu.
- 2) Remove the game disk and insert the ARD.
- 3) Press Reset and let the ARD do its thing.
- 4) When you are in the Monitor, move the program out of the way with

2000<8000.BFFF

- 5) Boot your empty slave disk and enter the Monitor.

**6 P
CALL-151**

- 6) Type in the hexdump with this article. All it does is move the game out of its hiding place and jump into it. Save the program like this:

BSAVE MARS CARS,A\$800,L\$7800

That's all you do. Play the game by typing "BRUN MARS CARS".

hexdump for...

ARD: Mars Cars

```
0800: A2 00 BD 00 20 9D 00 80 $B8D5
0808: E8 D0 F7 EE 04 08 EE 07 $0D9C
0810: 08 AD 04 08 C9 5F D0 E8 $4348
0818: 4C 00 70 $B059
```

How to be the WriteMaster

You can take command of the write-protect status of your disks by making this simple hardware modification: add your own switch! Why not change that red LED to amber or green or to a tri-state LED to indicate whether your drive is in normal or write-protect status?

by William Wingfield Jr.

Requirements:

One disk drive
Soldering Iron (optional)
Drill and bit
Hardcore COMPUTIST No. 22

Note: The procedure described in this article requires the dismantling of the disk drive which voids its warranty. Hardcore COMPUTIST will not be held responsible for any damages incurred while following this procedure.

When a person gets heavily into the deprotection scene, he often finds himself going through write-protect tabs at a substantially increased rate. In addition, he finds himself often notching disks to use the elusive backsides.

All this led me to make my first modification to my Apple: the adding of my own switch which controlled the write-protected status. It was something that I thought should have been standard equipment. In fact, some Commodore drives come with this feature.

Over the past four years, I've had my write master switch configured several different ways depending on how I was using the drive at the time.

If your drive is under warranty you may not wish to perform the modification described below. However, instead of

drilling a hole in the front of your drive, you can connect a switch with enough wire to let it hang out the back of the drive. This way you can remove it if you have to return the drive for any reason and it'll look like new. I had my switch this way until the day my warranty expired at which time I mounted the switch on the front of the drive.

What's Already There

If you've ever had the cover off your drive to adjust the speed, or out of curiosity, then you've probably looked at the write-protect switch. It's a normally closed (NC) micro-switch with a lever that senses the lack of a notch in the disk that's in the drive. If there is no notch, then the switch is opened (or turned OFF) and all write attempts are subdued. Similarly, if a notch is found then it rests in its normally closed (or turned ON) position which allows writing to occur.

Possible Modifications

The simplest modification is to wire a SPST switch parallel with the one in the drive (fig. 1). If the new switch is in the OFF position, then the drive will operate as it always has. If the new switch is in the ON position, then the computer believes there is a notch in the disk and will allow writing to any disk. This lets you write to the back of a disk without a notch (and to

the front with a write-protect tab too, so be careful).

You can disconnect the switch in the drive completely and connect the wires to a SPST switch and have complete control of the R/W operations. ON writes anytime and OFF never writes but I've found it more convenient to have a position that lets the drive switch read the disk.

If we use a SPDT switch with a center-off position, we can do whatever we want (fig. 2). In one ON position, the disk will be checked for a notch by the drive switch. In the other position, we can write anytime. Finally, in the center on position, nothing can be written at all - ultimate control of the write protect status!

Installation

Begin by removing the four screws from the bottom of the drive that secure the cover and remove it. Next, remove the four screws from the sides of the face plate, open the drive door and let it slide off the tracks as you move the face plate forward.

A small plastic ring secures the LED in its mount. Remove it as illustrated on page 23 by slipping a small knife or screwdriver under it and prying the ring straight back until it pops off.

Next, gently pull the LED out of the mount and the face plate will be free of the drive. Refer to fig. 3 as a guide to the mounting hole. The face plate on Apple

disk drives has a thin aluminum sheet on the front so a small pilot hole (1/16" - 1/8") should be drilled as a guide first. This will keep the drill bit from biting into the aluminum.

Put the face plate on something soft so it won't be marred and drill from the back. Access to a drill press is an advantage but not a necessity.

The simplest way to mount the switch is to drill the recommended size hole, wire the switch the way you want and then secure it with the retaining nut.

Re-assemble the drive and make sure that the added wire doesn't interfere with the inserting or removing of disks or the movement of the head. There is ample room in the bottom of the drive for the wire.

A more professional look can be achieved by drilling a hole slightly smaller than recommended and threading the switch into the plastic. Screw the switch in until the shaft is flush with the front of the faceplate. The retaining nut isn't needed. Wire the switch and re-assemble the drive.

I've found that having the switch operate vertically is the most convenient. The up position, toward the disk, is the position that checks the disk. The down position, toward the red LED, is the BE CAREFUL position, it can always write.

What About the LED

If you've grown tired of always seeing a red LED this is a good time to change it to amber or green. I used a 3PDT switch and a tri-state LED to give a visual indication of the mode the drive is in. Wire the extra 2 poles as a polarity reverser to the LED wires and the tri-state. Red signals that the write protect is bypassed; green that the disk is checked for protection. If your 3PDT has a center off position, then the LED light will not light at all in the center position.

Advantages

Now you don't need a separate game-save disk for your adventures. Just use the back of the disk for saving. The same idea works well for your spread sheet program, data base or word processor. Of course it is a simple way to quickly double the storage capacity of your disks.

Finally, I would like to say that making this modification is easier than it sounds. Also, I have never had a problem using both sides of my disks.

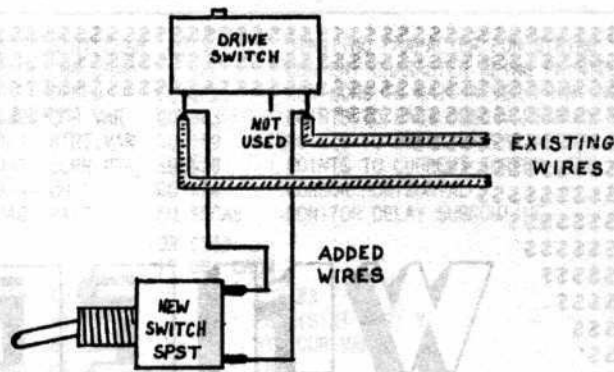


Fig. 1

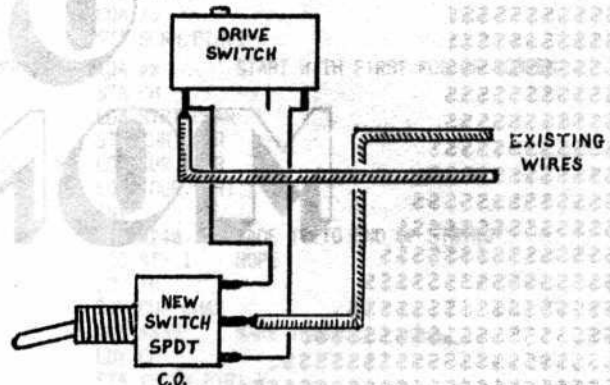


Fig. 2

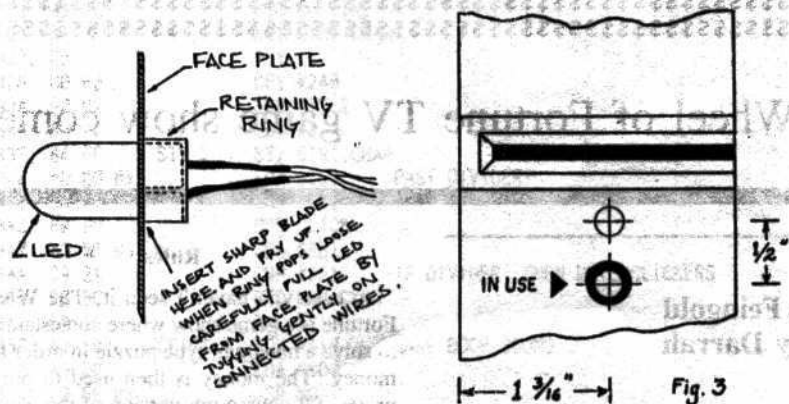
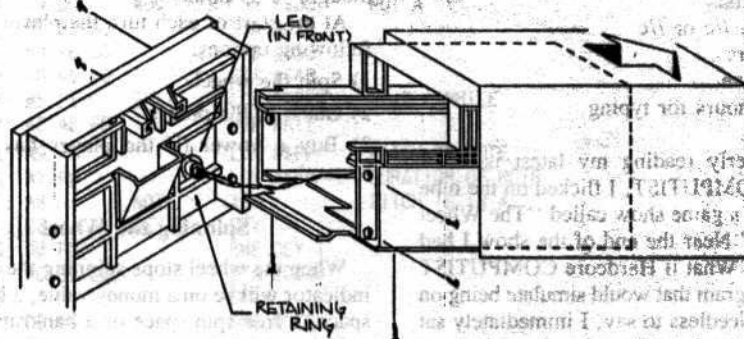
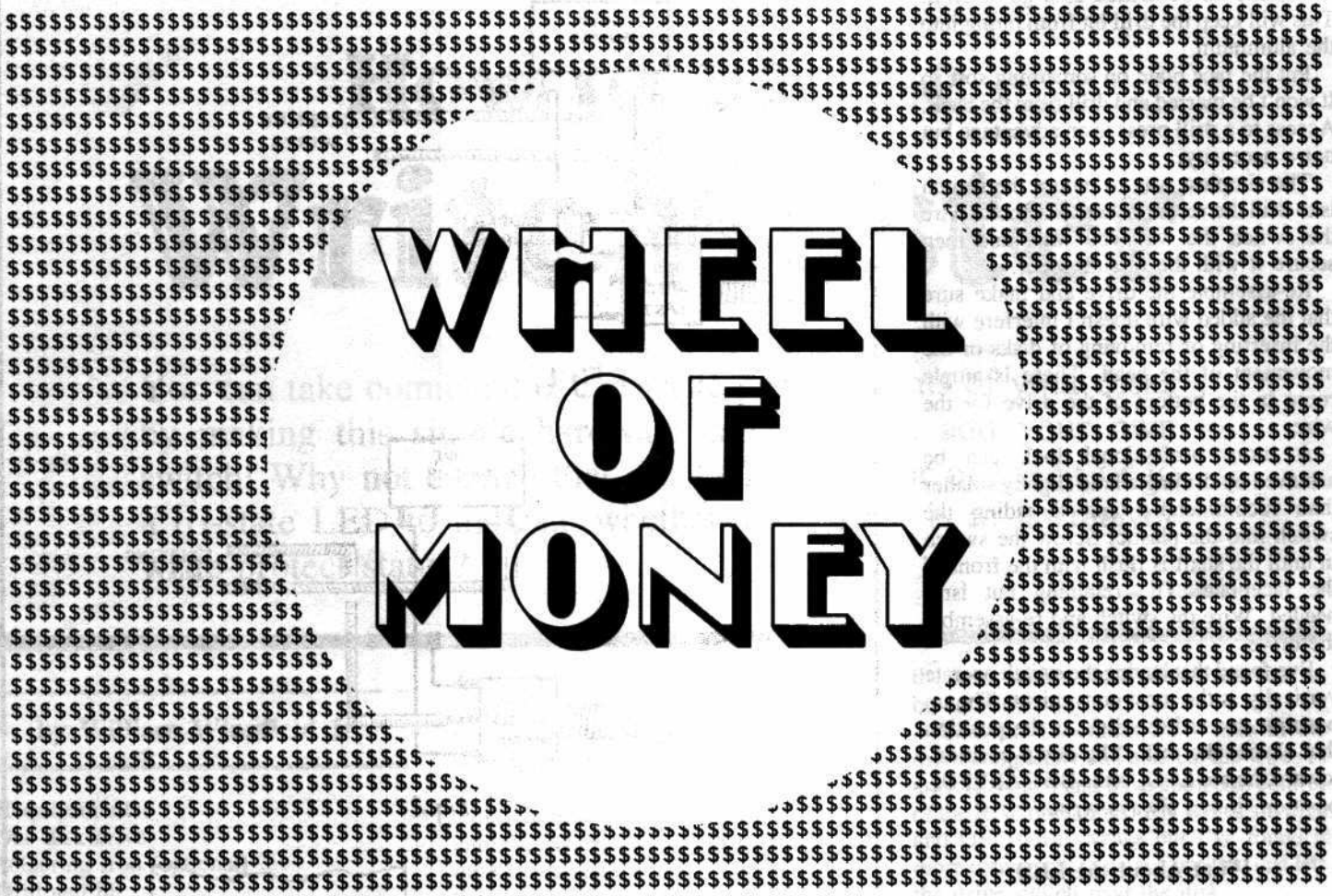


Fig. 3





WHEEL OF MONEY

The Wheel of Fortune TV game show comes to your computer screen.

by Jiro Feingold and Ray Darrah

Requirements:
An Apple II, //e or //c
2 or 3 players
One disk drive
A few free hours for typing

After eagerly reading my latest issue of Hardcore COMPUTIST, I flicked on the tube and watched a game show called "The Wheel of Fortune." Near the end of the show I had a revelation: What if Hardcore COMPUTIST printed a program that would simulate being on this show? Needless to say, I immediately sat down at my Apple keyboard and cranked out the following program which does just that.

Rules

In case you haven't seen it, The Wheel of Fortune is a game show where contestants have to solve a hangman-type puzzle in order to win money. The money is then used to purchase prizes. Of course my version of the game has no prizes and the money therefore can be thought of as points.

At the start of each turn the player has the following options:

- 1) Spin the wheel
- 2) Guess at the puzzle
- 3) Buy a vowel (if the player has enough money)

Spinning the Wheel

When the wheel stops spinning the player's indicator will be on a money value, a lose-turn space, a free-spin space or a bankrupt space.

If the wheel stops on a money space then the player must guess a consonant he thinks is a

letter of the puzzle. If he guesses correctly, then those letters are revealed in the puzzle, the player receives the amount of money indicated on the wheel multiplied by the the number of occurrences of the consonant he guessed in the puzzle and the player gets another turn. If the player guesses a consonant that is not in the puzzle, then play continues with the next player.

If the wheel stops in a free-spin space then that player is awarded a free spin and the wheel automatically spins again. Players can have more than one free spin.

If the wheel stops in the lose-turn space then play continues with the next player, unless the player who spun the wheel has a 'free spin' (indicated by an asterisk above his name). In which case the player then can use his free spin to spin the wheel again.

If the wheel stops in a bankrupt space then that player loses all money earned during this puzzle and play continues with the next player (unless of course the player has a free spin).

If the player elects to guess at the puzzle then the player must type what he thinks the puzzle is. If the player guesses the puzzle correctly, then he is awarded all the money earned during this puzzle and play continues with the next player and a new puzzle. If the player guesses incorrectly, play continues with the next player. When guessing the puzzle, spelling and punctuation are extremely important.

Buying Vowels

If the player wishes, he may unveil specified vowels in the puzzle. Each occurrence of the vowel will cost the player \$150. If the player guesses a vowel that is not in the puzzle, then play continues with the next player, unless the first player has a free spin! The letter "Y" is not recognized by "The CORE Wheel of Money" as a vowel.

Winning the Game

At the end of four puzzles the player who has acquired the most money by completing puzzles is declared the winner.

Typing It In

First type in the first BASIC program listed at the end of this article and save it with:

SAVE CORE WHEEL OF MONEY

Next type in the first hexdump and save it with:

BSAVE OBJ.CWOM,A\$300,L\$79

To keep you from accidentally seeing the answers to the puzzles, the words for The CORE Wheel of Money are provided as an encoded hexdump. To get the words, type in the second hexdump and save it with:

BSAVE CWOM.WORDS.BIN,A\$2000,L\$847

To decode the words and put them in a text file, a BASIC program is provided. Type in the second BASIC program at the end of this article and save it with:

SAVE CONVERT WORDS

To perform the conversion, type:

RUN CONVERT WORDS

To Start up The CORE Wheel of Money, type:

RUN THE CORE WHEEL OF MONEY

All of the puzzles used in The CORE Wheel of Money are divided into seven categories: thing, things, person, people, title, phrase and place. Note that the title category can mean more than just a title of a book or movie.

A modification you might try is to convert the game to hi-res and have a shape table that looks somewhat like Vanna White reveal the letters.

Now if I could only play the game as well as Merv Griffin.

Source Code for...

Core's Wheel Of Money

```

00FF- STRT.CHAR .EQ $FF  STARTING CHARACTER FOR STRING
00FE- CUR.CHAR .EQ $FE  CURRENT CHARACTER BEING PRINTED IS STRING
00F1- SPEED .EQ $F1  HOW LONG TO WAIT AFTER PRINTING LINE
0083- CUR.VAR .EQ $83  POINTS TO LAST USED VARIABLES VALUE
0069- STRT.VAR .EQ $69  POINTS TO START OF VARIABLES
0028- SCR.N.PTR .EQ $28  POINTS TO CURRENT SCREEN LINE
0024- CH .EQ $24  CURSOR HORIZONTAL
FCAB- WAIT .EQ $FCAB  MONITOR DELAY SUBROUTINE
      .OR $300
      .TF OBJ.CWOM

0300: A0 03          LDY #3
0302: B1 69          LDA (STRT.VAR),Y
0304: 85 83          STA CUR.VAR
0306: C8             INY
0307: B1 69          LDA (STRT.VAR),Y
0309: 85 84          STA CUR.VAR+1
030B: A9 A8          LDA #$A8      SET SCREEN LINE TO $A8
030D: 85 28          STA SCR.N.PTR
030F: A9 05          LDA #5
0311: 85 29          STA SCR.N.PTR+1
0313: A9 00          PRNTSTR LDA #0      START WITH FIRST POS OF SCREEN
0315: 85 24          STA CH
0317: A5 FF          LDA STRT.CHAR
0319: 85 FE          STA CUR.CHAR
031B: A4 FE          DO.LINE LDY CUR.CHAR
031D: B1 83          LDA (CUR.VAR),Y
031F: C8             INY
0320: C0 F0          CPY #240     MADE IT TO END OF STRING?
0322: 90 02          BCC STY.1   NOPE
0324: A0 00          LDY #0
0326: 84 FE          STY.1 STY CUR.CHAR
0328: 29 3F          AND #$3F     MAKE INVERSE
032A: A4 24          LDY CH
032C: 91 28          STA (SCR.N.PTR),Y
032E: C8             INY
032F: 84 24          STY CH
0331: C0 28          CPY #40
0333: 90 E6          BCC DO.LINE
0335: A4 FF          LDY STRT.CHAR
0337: C8             INY
0338: C0 F0          CPY #240
033A: 90 02          BCC STY.2
033C: A0 00          LDY #0
033E: 84 FF          STY.2 STY STRT.CHAR
0340: AD B0 05        LDA $5B0     PAST DIVIDER?
0343: C9 21          CMP #$21
0345: F0 09          BEQ CLICK
0347: AD B1 05        LDA $5B1
034A: C9 21          CMP #$21     IF DIVIDER, THEN MOVE CLICKERS
034C: F0 0C          BEQ MOV.CLKR
034E: D0 15          BNE EXIT
0350: A2 10          CLICK LDX #$10 MAKE BOOP SOUND
0352: A9 D0          LDA #$D0
0354: 20 6E 03        JSR NOTE
0357: A9 A1          LDA #$A1     REPLACE WITH "!"
0359: 2C             .HS 2C      SKIP NEXT TWO BYTES
035A: A9 DC          MOV.CLKR LDA #$DC PUT A "\"
035C: 8D 31 06        STA $631
035F: 8D 3B 06        STA $63B
0362: 8D 45 06        STA $645
0365: A5 F1          EXIT LDA SPEED WAIT A WHILE
0367: 4C A8 FC          JMP WAIT
036A: A9 FF          LDA #$FF     PITCH OF NOTE
036C: A2 FF          DURATION OF NOTE
036E: A8          NOTE TAY     PITCH IS IN A
036F: 88          DEY.1 DEY
0370: D0 FD          BNE DEY.1
0372: 8D 30 C0        STA $C030
0375: CA          DEX
0376: D0 F6          BNE NOTE
0378: 60          RTS

```

.....end of source code

BASIC listing for Core's Wheel of Money

```
10 REM \
20 REM / \
30 REM !
40 REM CORE WHEEL OF MONEY
50 REM BY JIRO FEINGOLD
60 REM AND RAY DARRAH
70 REM !
80 REM / \
90 REM / \
100 GOTO 880
110 POKE 34, 23 : HOME : POKE 34, 0 : POKE 35, 11
    : HOME : FT = W
120 TY = INT ( RND ( W ) * 7 ) : PZ = PZ + W : IF PZ
    > NW THEN 700
130 TEXT : VTAB W : HTAB W : PRINT "PUZZLE # "
    PZ;
140 W0$ = GS ( RND ( 1 ) * 20 + W, TY ) : SX = 20 - LEN
    ( W0$ ) / 2
150 RESTORE : FOR X = 0 TO TY : READ A$ : NEXT : HTAB
    20 - LEN ( A$ ) / 2
160 PRINT A$ : VTAB 4 : IF SX = 0 THEN SX = W
170 HTAB W : PRINT SPC ( 40 ) : HTAB SX : VTAB 4 : VC
    = 0 : LL = 0 : FOR X = W TO LEN ( W0$ )
180 A$ = MID$ ( W0$, X, W ) : IF A$ < "A" OR A$ > "Z"
    THEN PRINT A$ : GOTO 200
190 PRINT "- " : LL = LL + W
200 VC = VC + ( A$ = "A" OR A$ = "E" OR A$ = "I" OR
    A$ = "O" OR A$ = "U" ) : NEXT
210 IF PM ( CP, 2 ) = 0 OR FT = W THEN FT = 0 : GOTO 270
220 HTAB W : VTAB 8 : PRINT SPC ( 39 ) : HTAB W :
    PRINT "USE A FREE SPIN ( Y/N ) ^ " ;
230 GET A$ : IF A$ <> "Y" AND A$ <> "N" THEN 230
240 HTAB W : PRINT SPC ( 39 ) : IF A$ = "N" THEN 270
250 PM ( CP, 2 ) = PM ( CP, 2 ) - W : HTAB CP * 10 : VTAB
    14
260 PRINT CHR$ ( 32 + 10 * ( PM ( CP, 2 ) > 0 ) ) : GOTO
    350
270 GOSUB 760 : CP = CP + PS : IF CP > PL THEN CP = W
280 VTAB 8 : HTAB W : PRINT SPC ( 39 ) : INVERSE :
    GOSUB 760
290 IF LL = < VC THEN VTAB 6 : HTAB 1 : PRINT "ONLY ^
    VOWELS ^ LEFT" : GOTO 520
300 VTAB 8 : HTAB W : PRINT "S)PIN, ^G)UESS" ;
310 IF PM ( CP, 0 ) > 149 THEN PRINT " ^A)R ^B)UY ^
    VOWEL" ;
320 WAIT - 16384, 128 : GET A$ : FOR X = W TO 3
330 IF A$ <> MID$ ( K$, X, W ) THEN NEXT : GOTO 320
340 HTAB W : PRINT SPC ( 40 ) : VTAB 8 : ON X GOTO
    350, 520, 580
350 GOSUB 790 : Y = 10 * ( 1 + INT ( ( PEEK ( 255 ) -
    3 ) / 10 ) ) + 10 * ( CP - W ) + 5
360 Y = ABS ( Y - 240 * ( Y > 240 ) ) : X = VAL ( MID$
    ( W$, Y, 5 ) ) : IF X = 0 THEN 610
370 VTAB 8 : HTAB W : PRINT "LETTER ^ TO ^ TRY => " ;
380 GET A$ : IF A$ = "E" OR A$ = "I" OR A$ = "O" OR
    A$ = "U" OR A$ < "B" OR A$ > "Z" THEN 380
390 PRINT A$ : IF PEEK ( 1942 + ASC ( A$ ) ) = 160
    THEN 420
400 VTAB 22 : HTAB 15 : INVERSE : PRINT "USED" :
    POKE DR, 0 : POKE PI, 0 : CALL PI - W
410 FOR X = W TO 750 : NEXT : NORMAL : VTAB 22 : HTAB
    15 : PRINT "USED" : GOTO 210
420 LC = 0 : FOR Y = W TO LEN ( W0$ )
430 IF MID$ ( W0$, Y, W ) = A$ THEN HTAB SX - W + Y
    : VTAB 4 : PRINT AS : LC = LC + W
440 NEXT : VTAB 24 : HTAB ASC ( A$ ) - 57 : PRINT
    A$ : IF LC <> 0 THEN 470
450 HTAB W : VTAB 8 : PRINT "THERE ARE ^ NO ^ AS
    ^ S ^ IN ^ THE ^ PUZZLE"
460 POKE DR, 128 : FOR X = 100 TO 200 STEP 25 : POKE
```

```
PI, X : CALL PI - W : NEXT : GOTO 410
470 PM ( CP, 0 ) = PM ( CP, 0 ) + X * LC : LL = LL - LC
480 POKE DR, 64 : IF PM ( CP, 0 ) < 0 THEN PM ( CP, 0
    ) = 0
490 VC = VC - LC * ( A$ = "A" OR A$ = "E" OR A$ = "I"
    OR A$ = "O" OR A$ = "U" )
500 IF LL = 0 THEN 540
510 FOR X = 250 TO 50 STEP - 25 : POKE PI, X : CALL
    PI - W : NEXT : GOTO 280
520 POKE 34, 4 : POKE 35, 11 : HTAB W : PRINT
    "PUZZLE ^ GUESS : " : INPUT " " : A$
530 HOME : TEXT : IF A$ <> W0$ THEN POKE DR, 0 :
    POKE PI, 192 : CALL PI - W : POKE PI, 255 :
    CALL PI - W : GOTO 210
540 PM ( CP, W ) = PM ( CP, W ) + PM ( CP, 0 ) : FT = CP
    : FOR CP = W TO PL STEP PS
550 PM ( CP, 0 ) = 0 : GOSUB 760 : NEXT : CP = FT : VTAB
    4
560 HTAB SX : PRINT W0$ : POKE DR, 20 : FOR X = 1
    TO 200
570 POKE PI, RND ( W ) * 255 : CALL PI - W : NEXT
    : GOTO 110
580 HTAB W : PRINT "VOWEL ^ TO ^ BUY => " ;
590 GET A$ : IF A$ <> "A" AND A$ <> "E" AND A$ <>
    > "I" AND A$ <> "O" AND A$ <> "U" THEN 590
600 X = - 150 : GOTO 390
610 A$ = MID$ ( W$, Y - 3, W ) : IF A$ = "L" THEN 650
620 IF A$ = "F" THEN 670
630 POKE DR, 40 : FOR X = 20 TO 200 STEP 2 : POKE
    PI, X
640 CALL PI - W : NEXT : PM ( CP, 0 ) = 0 : GOSUB 760
    : GOTO 210
650 POKE DR, 200 : FOR X = W TO 5 : POKE PI, 100 +
    150 * ( X / 2 = INT ( X / 2 ) )
660 CALL PI - W : NEXT : GOTO 210
670 POKE DR, 20 : FOR X = 1 TO 50 : POKE PI, 128 +
    127 * SIN ( X / 10 ) : CALL PI - W : NEXT
680 PM ( CP, 2 ) = PM ( CP, 2 ) + W : VTAB 14 : HTAB CP
    * 10 : PRINT " ^ " : GOTO 350
690 REM END OF GAME
700 FOR X = W TO PL STEP PS : IF PM ( X, W ) > PM ( FT
    , W ) THEN FT = X
710 NEXT : TEXT : HOME : FOR X = 1 TO 20 : PRINT
    PNS ( FT ) CHR$ ( 10 ) : NEXT
720 VTAB 22 : HTAB W : PRINT "HAS ^ WON ^ THE ^
    GAME ^ WITH ^ $ " PM ( FT, W )
730 VTAB 24 : PRINT "CARE ^ FOR ^ ANOTHER ^ MATCH ^
    ( Y/N ) ^ " ;
740 PZ = 0 : GET A$ : ON ( A$ = "Y" ) GOTO 1010 : GOTO
    940
750 REM PRINT PLAYERS NAME
760 VTAB 16 : HTAB PM ( CP, 3 ) : PRINT PNS ( CP ) :
    HTAB ( CP - W ) * 10 + 8
770 PRINT "$ " PM ( CP, 0 ) : NORMAL : PRINT " ^ ^ ^ ^
    " : PRINT : HTAB ( CP - W ) * 10 + 8 : PRINT "$ "
    PM ( CP, W ) : RETURN
780 REM SPIN WHEEL
790 SPEED = 255 : FOR X = W TO RND ( W ) * 125 + 25
    : CALL 768 : NEXT : B = 100
800 FOR X = 255 TO 150 STEP - 1 : SPEED = X : CALL
    768 : NEXT
810 FOR X = 150 TO 0 STEP - 75 : SPEED = X : CALL 768
    : NEXT : SPEED = 255 : RETURN
820 REM DO PRETTY COLORS
830 GR : HOME : PRINT : PRINT SPC ( 20 - LEN ( A$ )
    ) / 2 : A$
840 X = 0 : FOR Y = 0 TO 39 : COLOR = RND ( W ) * 255
    : HLIN X, 39 AT Y : HLIN X, 39 AT 39 - Y
850 VLIN X, 39 AT Y : VLIN X, 39 AT 39 - Y : IF PEEK
    ( 49152 ) < 128 THEN NEXT : GOTO 840
860 RETURN
```

```
870 REM INITIALIZE VARIABLES
880 W$ = " " : W = 1 : X = 0 : Y = 0 : K$ = "SGB" : PI =
    875 : DR = 877
890 DIM GS ( 20, 6 ), PNs ( 3 ), PM ( 3, 3 )
900 PRINT CHR$ ( 4 ) "READ ^ CNOM. WORDS" : FOR Y =
    0 TO 6 : FOR X = W TO 20
910 INPUT GS ( X, Y ) : NEXT : NEXT : PRINT CHR$ ( 4 )
    "CLOSE"
920 IF PEEK ( 768 ) + PEEK ( 769 ) <> 163 THEN PRINT
    CHR$ ( 4 ) "BLOOD ^ OBJ. CNOM, AS300"
930 REM TITLE PAGE
940 A$ = "THE ^ CORE ^ WHEEL ^ OF ^ MONEY" : GOSUB
    830 : GET A$ : IF A$ = CHR$ ( 27 ) THEN TEXT
    : HOME : END
950 A$ = "2 ^ OR ^ 3 ^ PLAYERS ? " : GOSUB 830 : GET A$
    : IF A$ < "2" OR A$ > "3" THEN 950
960 TEXT : HOME : PS = W : PL = VAL ( A$ ) : PRINT SPC (
    8 ) "THE ^ CORE ^ WHEEL ^ OF ^ MONEY"
970 IF PL = 2 THEN PS = 2 : PL = 3
980 VTAB 8 : FOR X = W TO PL STEP PS : PRINT : PRINT
990 INPUT "PLAYERS ^ NAME => " : PNS ( X ) : PNS ( X ) =
    LEFT$ ( PNS ( X ), 8 ) : IF PNS ( X ) = " " THEN 990
1000 NEXT
1010 W$ = " " : Y = W : FOR X = 0 TO 23
1020 IF X / 4 = INT ( X / 4 ) THEN W$ = W$ + "I" + MID$
    ( "FREE ^ SPIN LOSE ^ TURN ^ BANKRUPT", Y, 9 )
    : Y = Y + 9 : X = X + W : IF Y > 27 THEN Y = W
1030 RE = 50 * ( INT ( RND ( W ) * 18 + . 5 ) + 2 ) : W$
    = W$ + "! ^ $ " + STR$ ( RE )
1040 W$ = W$ + LEFT$ ( " ^ ^ ^ ^ ", 6 - LEN ( STR$ ( RE
    ) ) ) : NEXT
1050 HOME : VTAB 22 : PRINT " ^ ^ ^ ^ ^ USED ^
    LETTERS ^ ^ ^ ^ " ;
1060 VTAB 13 : HTAB 10 : FOR X = W TO 3 : PRINT " ! "
    SPC ( 9 ) : NEXT : VTAB 14 : HTAB 9
1070 FOR X = W TO 3 : PRINT " / ^ " CHR$ ( 92 ) SPC (
    7 ) ;
1080 FOR Y = 0 TO 2 : PM ( X, Y ) = 0 : NEXT : NEXT :
    VTAB 15 : HTAB 8 : FOR X = W TO 3
1090 PRINT " / ^ ^ ^ ^ " CHR$ ( 92 ) SPC ( 5 ) : NEXT :
    FOR CP = W TO PL STEP PS
1100 PM ( CP, 3 ) = 11 + ( CP - 1 ) * 10 - LEN ( PNS ( CP
    ) ) / 2 : GOSUB 760 : NEXT
1110 POKE 255, 0 : CALL 768 : CP = PL : NW = 4 : GOTO
    110
1120 DATA THINGS, THING, PERSON
1130 DATA PEOPLE, TITLE, PHRASE, PLACE
```

Convert Words BASIC Listing

```
10 HGR : POKE - 16302, 0 : NMS$ = "CNOM. WORDS" : FOR
    A = 768 TO 798 : READ PK : POKE A, PK : NEXT
20 PRINT CHR$ ( 4 ) "BLOOD NMS" : BIN, AS2000 : PK
    = 0 : FOR A = 8192 TO 10310 : PK = PK + PEEK ( A
    ) : NEXT
30 IF PK <> 202623 THEN TEXT : HOME : PRINT "WORD ^
    LIST ^ INCORRECT" : STOP
40 PRINT CHR$ ( 4 ) "NOMONC10" : PRINT CHR$ ( 4 )
    "OPEN" NMS
50 PRINT CHR$ ( 4 ) "DELETE" NMS : PRINT CHR$ ( 4 )
    "OPEN" NMS : PRINT CHR$ ( 4 ) "WRITE" NMS
    : CALL 768
60 PRINT CHR$ ( 4 ) "CLOSE" : TEXT : HOME : PRINT
    CHR$ ( 4 ) "CATALOG"
70 DATA 160, 0, 185, 0, 32, 73, 170, 32, 237, 253,
    200, 192, 71, 240, 8
80 DATA 152, 208, 240, 238, 4, 3, 208, 235, 173,
    4, 3, 201, 40, 144, 241, 96
.....end of BASIC listings
```

hexdump for...

OBJ.CWOM

0300: A0 03 B1 69 85 83 C8 B1 \$2C4B
0308: 69 85 84 A9 A8 85 28 A9 \$AAB6
0310: 05 85 29 A9 00 85 24 A5 \$5808
0318: FF 85 FE A4 FE B1 83 C8 \$D498
0320: C0 F0 90 02 A0 00 84 FE \$BA1E
0328: 29 3F A4 24 91 28 C8 84 \$3937
0330: 24 C0 28 90 E6 A4 FF C8 \$493E
0338: C0 F0 90 02 A0 00 84 FF \$B6D8
0340: AD B0 05 C9 21 F0 09 AD \$9E7E
0348: B1 05 C9 21 F0 0C D0 15 \$31CE

0350: A2 10 A9 D0 20 6E 03 A9 \$C06F
0358: A1 2C A9 DC 8D 31 06 8D \$CBFC
0360: 3B 06 8D 45 06 A5 F1 4C \$DF92
0368: A8 FC A9 FF A2 FF A8 88 \$B20F
0370: D0 FD 8D 30 C0 CA D0 F6 \$69F8
0378: 60 \$4AD9

hexdump for...

CWOM.WORDS.BIN

2000: 68 65 73 79 0A 6B 64 6E \$7B4B
2008: 0A 6D 63 78 66 79 27 66 \$65D6
2010: 63 6D 62 7E 0A 6B 64 6E \$427A
2018: 0A 6E 6B 78 61 27 62 6B \$7301
2020: 67 67 6F 78 0A 6B 64 6E \$B4EE
2028: 0A 64 6B 63 66 27 7A 6F \$4526
2030: 64 0A 6B 64 6E 0A 7A 6F \$71B2
2038: 64 69 63 66 27 62 65 7E \$CE6E
2040: 0A 6B 64 6E 0A 69 65 66 \$0CAE
2048: 6E 27 79 62 63 78 7E 0A \$C2BD

2050: 6B 64 6E 0A 7A 6B 64 7E \$5AF1
2058: 79 27 6B 7A 7A 66 6F 79 \$9BB3
2060: 0A 6B 64 6E 0A 65 78 6B \$C066
2068: 64 6D 6F 79 27 6E 65 6D \$01C8
2070: 79 0A 6B 64 6E 0A 69 6B \$B07C
2078: 7E 79 27 67 6F 64 0A 6B \$450E
2080: 64 6E 0A 7D 65 67 6F 64 \$39D8
2088: 27 6D 65 65 6E 0A 6B 64 \$6EEE
2090: 6E 0A 68 6B 6E 27 78 6B \$03E2
2098: 63 66 78 65 6B 6E 79 27 \$B7D6

20A0: 67 6B 6D 6B 70 63 64 6F \$FE04
20A8: 79 27 64 65 7C 6F 66 79 \$8D61
20B0: 27 7E 78 6F 6F 79 27 69 \$5B64
20B8: 65 67 7A 7F 7E 6F 78 79 \$F5A0
20C0: 27 66 63 6D 62 7E 0A 68 \$9896
20C8: 7F 66 68 79 27 63 64 79 \$E7BA
20D0: 7E 78 7F 69 7E 63 65 64 \$AA19
20D8: 79 27 6B 7F 7E 65 67 65 \$15E8
20E0: 68 63 66 6F 79 27 6D 66 \$EBD8
20E8: 6B 79 79 6F 79 27 79 6F \$125F

20F0: 6B 7E 0A 68 6F 66 7E 79 \$5BBF
20F8: 27 68 78 63 6F 6C 69 6B \$CDB0

2100: 79 6F 27 7D 6B 66 66 6F \$29BA
2108: 7E 27 6C 66 65 7A 7A 73 \$7727
2110: 0A 6E 63 79 61 27 7E 6F \$B5D2
2118: 66 6F 7C 63 79 63 65 64 \$ABC3
2120: 27 7A 63 69 7E 7F 78 6F \$6B42
2128: 27 7F 64 63 7C 6F 78 79 \$420A
2130: 6F 27 7A 6F 64 69 63 66 \$EC55
2138: 27 68 78 63 69 61 27 7E \$6EBA

2140: 73 7A 6F 7D 78 63 7E 6F \$C60D
2148: 78 27 66 63 6D 62 7E 0A \$D88C
2150: 68 7F 66 68 27 79 7A 6B \$F6DF
2158: 69 6F 0A 79 62 7F 7E 7E \$EEBD
2160: 66 6F 27 7A 66 6B 79 7E \$D57F
2168: 63 69 27 69 6B 78 7A 6F \$369B
2170: 7E 27 69 62 7F 78 69 62 \$C3DB
2178: 27 68 7F 63 66 6E 63 64 \$4ECA
2180: 6D 27 6F 66 6F 7C 6B 7E \$F401
2188: 65 78 27 6E 63 79 62 7D \$E92B

2190: 6B 79 62 6F 78 27 6C 65 \$ADE6
2198: 78 6F 79 7E 27 78 6F 69 \$57F7
21A0: 65 78 6E 27 67 65 7F 64 \$56AB
21A8: 7E 6B 63 64 27 78 65 64 \$94D7
21B0: 6B 66 6E 0A 78 6F 6B 6D \$63EE
21B8: 6B 64 27 6D 6F 65 78 6D \$E993
21C0: 6F 0A 68 7F 79 62 27 60 \$70F3
21C8: 6B 64 6F 0A 6C 65 64 6E \$D581
21D0: 6B 27 7D 6B 66 7E 6F 78 \$3459
21D8: 0A 69 78 65 64 61 63 7E \$AB4B

21E0: 6F 27 67 6B 6E 65 64 64 \$2317
21E8: 6B 27 63 79 79 6B 69 0A \$CC0E
21F0: 6B 79 63 67 65 7C 27 60 \$B7B5
21F8: 65 62 64 64 73 0A 69 6B \$436B
2200: 78 79 65 64 27 6D 6F 65 \$5482
2208: 78 6D 6F 0A 7D 6B 79 62 \$9952
2210: 63 64 6D 7E 65 64 27 69 \$B4C7
2218: 62 6B 78 66 6F 79 0A 6E \$3789
2220: 63 69 61 6F 64 79 27 79 \$25FD
2228: 62 6F 78 66 65 69 61 0A \$70E8

2230: 62 65 66 67 6F 79 27 6F \$D4FA
2238: 6E 6D 6B 78 0A 6B 66 66 \$C8E3
2240: 6F 64 0A 7A 65 6F 27 6B \$598D
2248: 6D 6B 7E 62 6B 0A 69 62 \$D654
2250: 78 63 79 7E 63 6F 27 69 \$3FFE
2258: 62 6B 78 66 6F 79 0A 6E \$4CE0
2260: 6B 78 7D 63 64 27 78 63 \$EA0D
2268: 69 62 6B 78 6E 0A 64 63 \$EB82
2270: 72 65 64 27 69 62 6F 7C \$53D0
2278: 73 0A 69 62 6B 79 6F 27 \$CC93

2280: 6E 7F 79 7E 63 64 0A 62 \$945C
2288: 65 6C 6C 67 6B 64 27 79 \$0E3C
2290: 7E 6F 7C 6F 64 0A 79 7A \$3333
2298: 63 6F 66 68 6F 78 6D 27 \$E09A
22A0: 79 7E 6F 7A 62 6F 64 0A \$CB66
22A8: 61 63 64 6D 27 7D 6B 66 \$E6B7
22B0: 7E 6F 78 0A 67 65 64 6E \$C8AD
22B8: 6B 66 6F 27 6E 63 6B 64 \$8AC5
22C0: 6B 0A 78 65 79 79 27 63 \$1D06
22C8: 64 6E 63 6B 64 79 27 69 \$0842

22D0: 65 7D 68 65 73 79 27 6C \$1FE7
22D8: 63 78 6F 6F 64 27 67 \$C1F5

22E0: 65 78 6B 66 0A 67 6B 60 \$5D56
22E8: 65 78 63 7E 73 27 7E 62 \$7B20
22F0: 6F 0A 68 6F 6B 7E 66 6F \$B220
22F8: 79 27 6E 6F 67 65 69 78 \$6F9D
2300: 6B 7E 79 27 78 6F 7A 7F \$7095
2308: 68 66 63 69 6B 64 79 27 \$7A80
2310: 69 65 67 67 7F 64 63 79 \$28ED
2318: 7E 79 27 6B 67 6F 78 63 \$BDF4

2320: 69 6B 64 79 27 64 7F 69 \$286E
2328: 66 6F 6B 78 0A 7A 62 73 \$6A42
2330: 79 63 69 63 79 7E 79 27 \$F3AC
2338: 78 7F 79 79 63 6B 64 79 \$DE1E
2340: 27 62 6B 78 6E 69 65 78 \$1C1A
2348: 6F 0A 69 65 67 7A 7F 7E \$ACC5
2350: 63 79 7E 79 27 7D 62 63 \$BA4A
2358: 7E 6F 0A 69 65 66 66 6B \$7270
2360: 78 0A 7D 65 78 61 6F 78 \$FB5F
2368: 79 27 61 66 6F 7A 7E 65 \$6FFD

2370: 67 6B 64 63 6B 69 79 27 \$7DD3
2378: 69 65 64 79 7E 78 7F 69 \$CC1D
2380: 7E 63 65 64 0A 7D 65 78 \$F210
2388: 61 6F 78 79 27 68 7F 78 \$5C22
2390: 7E 0A 78 6F 73 64 65 66 \$3F1D
2398: 6E 79 0A 0C 0A 6E 65 66 \$E712
23A0: 66 73 0A 7A 6B 78 7E 65 \$75E7
23A8: 64 27 6F 6D 73 7A 7E 63 \$B740
23B0: 6B 64 79 27 68 66 65 64 \$8F01
23B8: 6E 6F 79 27 7E 62 6F 0A \$BC19

23C0: 6D 78 6B 7E 6F 6C 7F 66 \$B797
23C8: 0A 6E 6F 6B 6E 27 7E 6F \$C22E
23D0: 6F 64 6B 6D 6F 78 79 27 \$414F
23D8: 6B 64 63 67 6B 66 0A 62 \$3645
23E0: 65 7F 79 6F 27 7D 6B 78 \$D959
23E8: 0A 65 6C 0A 7E 62 6F 0A \$E9B3
23F0: 7D 65 78 66 6E 79 27 78 \$9778
23F8: 65 69 61 0A 6B 78 65 7F \$0270
2400: 64 6E 0A 7E 62 6F 0A 69 \$1B31
2408: 66 65 69 61 27 6D 65 64 \$14F2

2410: 6F 0A 7D 63 7E 62 0A 7E \$2CEA
2418: 62 6F 0A 7D 63 64 6E 27 \$5161
2420: 7E 62 6F 0A 64 6F 7D 0A \$A88B
2428: 73 65 78 61 0A 7E 63 67 \$3021
2430: 6F 79 27 7E 62 6F 0A 7D \$8D4B
2438: 6B 66 66 0A 79 7E 78 6F \$8E87
2440: 6F 7E 0A 60 65 7F 78 64 \$1BAD
2448: 6B 66 27 78 6B 63 6E 6F \$47F3
2450: 78 79 0A 65 6C 0A 7E 62 \$F9CB
2458: 6F 0A 66 65 79 7E 0A 6B \$5A32

2460: 78 61 27 7A 78 6F 79 63 \$D08B
2468: 6E 6F 64 7E 0A 65 6C 0A \$3A70
2470: 7E 62 6F 0A 7F 64 63 7E \$B810
2478: 6F 6E 0A 79 7E 6B 7E 6F \$75BA
2480: 79 27 68 6B 69 61 0A 7E \$25B9
2488: 65 0A 7E 62 6F 0A 6C 7F \$4029
2490: 7E 7F 78 6F 27 79 6B 64 \$7C35
2498: 63 7E 6B 7E 63 65 64 0A \$4C03
24A0: 6F 64 6D 63 64 6F 6F 78 \$5ECE
24A8: 27 79 6F 69 78 6F 7E 6B \$4DD4

24B0: 78 73 0A 65 6C 0A 79 7E \$C240
24B8: 6B 7E 6F 27 6E 63 78 6F \$0462

continued: hexdump for...

CWOM.WORDS.BIN

24C0: 69 7E 65 78 27 78 6F 7E \$68A0
24C8: 7F 78 64 0A 65 6C 0A 7E \$52A6
24D0: 62 6F 0A 60 6F 6E 63 27 \$CC1F
24D8: 7D 62 6F 6F 66 0A 65 6C \$9FCD
24E0: 0A 6C 65 78 7E 7F 64 6F \$29DC
24E8: 27 7A 78 63 64 69 6F 0A \$2DE9
24F0: 65 6C 0A 7D 6B 66 6F 79 \$22BB
24F8: 27 68 6F 7C 6F 78 66 73 \$87FD
2500: 0A 62 63 66 66 79 0A 69 \$35FC
2508: 65 7A 27 6E 7F 64 6D 6F \$D0D5
2510: 65 64 79 0A 6B 64 6E 0A \$FE96
2518: 6E 78 6B 6D 65 64 79 27 \$0C56
2520: 7E 62 6F 0A 66 6B 79 7E \$9EB0
2528: 0A 79 7E 6B 78 6C 63 6D \$4C92
2530: 62 7E 6F 78 27 78 6B 67 \$3323
2538: 68 65 27 7E 6F 69 62 64 \$9E88
2540: 63 69 6B 66 0A 6F 6E 63 \$3B01
2548: 7E 65 78 27 63 7E 0D 79 \$4A01
2550: 0A 78 6B 63 64 63 64 6D \$6F2A
2558: 0A 69 6B 7E 79 0A 6B 64 \$291D
2560: 6E 0A 6E 65 6D 79 27 7E \$88A7
2568: 65 7E 6B 66 66 73 0A 6B \$6321
2570: 7D 6F 79 65 67 6F 27 67 \$835A
2578: 65 64 61 6F 73 0A 65 64 \$2954
2580: 0A 67 73 0A 68 6B 69 61 \$D2E9
2588: 27 73 65 7F 0A 69 6B 64 \$18BE
2590: 0D 7E 0A 7E 6F 6B 69 62 \$4F1F
2598: 0A 6B 64 0A 65 66 6E 0A \$A84C
25A0: 6E 65 6D 0A 64 6F 7D 0A \$0945
25A8: 7E 78 63 69 61 79 27 7E \$D601
25B0: 62 6F 0A 69 6B 7E 0D 79 \$CA2D
25B8: 0A 67 6F 65 7D 27 7D 62 \$2F71
25C0: 6F 64 0A 62 6F 66 66 0A \$318C
25C8: 6C 78 6F 6F 70 6F 79 0A \$A86D
25D0: 65 7C 6F 78 27 7F 7A 0A \$BE46
25D8: 7E 62 6F 0A 69 78 6F 6F \$94AC
25E0: 61 0A 7D 63 7E 62 65 7F \$F33B
25E8: 7E 0A 6B 0A 7A 6B 6E 6E \$9092
25F0: 66 6F 27 6B 79 0A 79 7E \$004F
25F8: 7F 68 68 65 78 64 0A 6B \$ACC5
2600: 79 0A 6B 0A 67 7F 66 6F \$1702
2608: 27 66 65 79 63 64 6D 0A \$821A
2610: 73 65 7F 78 0A 67 6B 78 \$885E
2618: 68 66 6F 79 27 7A 6B 79 \$A161
2620: 79 63 64 6D 0A 7D 63 7E \$CAB1
2628: 62 0A 6C 66 73 63 64 6D \$047F
2630: 0A 69 65 66 65 78 79 27 \$ED54
2638: 68 6F 63 64 6D 0A 65 64 \$8D4F
2640: 0A 7E 62 6F 0A 68 6B 66 \$9635
2648: 66 27 79 62 65 7D 63 64 \$CC6C
2650: 6D 0A 73 65 7F 78 0A 7E \$9479
2658: 78 7F 6F 0A 69 65 66 65 \$D557
2660: 78 79 27 6E 78 63 7C 63 \$832B
2668: 64 6D 0A 67 6F 0A 7F 7A \$36AC
2670: 0A 7E 62 6F 0A 7D 6B 66 \$8886

2678: 66 27 6C 6B 79 7E 6F 78 \$7E47
2680: 0A 7E 62 6B 64 0A 6B 0A \$AE6C
2688: 79 7A 6F 6F 6E 63 64 6D \$8D45
2690: 0A 68 7F 66 66 6F 7E 27 \$3579
2698: 6B 0A 7A 6B 63 64 0A 63 \$FDC2
26A0: 64 0A 7E 62 6F 0A 64 6F \$1C73
26A8: 69 61 27 78 7F 64 64 63 \$F2CD
26B0: 64 6D 0A 6B 7D 6B 73 0A \$8DDA
26B8: 6C 78 65 67 0A 7E 62 6F \$4BCE
26C0: 0A 6C 6B 69 7E 79 27 6C \$332C
26C8: 63 66 7E 62 73 0A 78 63 \$2D90
26D0: 69 62 27 7E 7D 65 0A 69 \$5573
26D8: 6B 64 79 0A 79 62 65 78 \$18A5
26E0: 7E 0A 65 6C 0A 6B 0A 79 \$0F62
26E8: 63 72 0A 7A 6B 69 61 27 \$58F0
26F0: 63 0A 7D 65 7F 66 6E 64 \$D811
26F8: 0D 7E 0A 7E 65 7F 69 62 \$1E8A
2700: 0A 63 7E 0A 7D 63 7E 62 \$1DFB
2708: 0A 6B 0A 7E 6F 64 0A 6C \$3C6E
2710: 65 65 7E 0A 7A 65 66 6F \$0671
2718: 27 7E 6B 61 6F 0A 6B 0A \$A51A
2720: 66 65 64 6D 0A 7D 6B 66 \$3D9C
2728: 61 0A 65 6C 6C 0A 6B 0A \$CEE8
2730: 79 62 65 78 7E 0A 6E 65 \$8FD0
2738: 69 61 27 7A 6B 66 67 0A \$D769
2740: 79 7A 78 63 64 6D 79 27 \$36C6
2748: 7E 62 6F 0A 7F 64 63 7E \$7446
2750: 6F 6E 0A 79 7E 6B 7E 6F \$790C
2758: 79 0A 65 6C 0A 6B 67 6F \$372C
2760: 78 63 69 6B 27 67 6B 78 \$425C
2768: 7C 63 64 0A 6D 6B 78 6E \$DCA3
2770: 6F 64 79 27 62 65 7F 79 \$DEA9
2778: 7E 65 64 0A 6B 79 7E 78 \$BF0D
2780: 65 6E 65 67 6F 27 64 6F \$A58E
2788: 7D 0A 73 65 78 61 0A 69 \$E093
2790: 63 7E 73 27 7A 63 61 6F \$90D4
2798: 0D 79 0A 7A 6F 6B 61 27 \$8384
27A0: 7E 62 6F 0A 64 65 78 7E \$0E50
27A8: 62 0A 7A 65 66 6F 27 79 \$45A7
27B0: 63 66 63 69 65 64 0A 7C \$2525
27B8: 6B 66 66 6F 73 27 64 63 \$89F7
27C0: 6B 6D 6B 78 6B 0A 6C 6B \$C6D4
27C8: 66 66 79 27 6B 64 7E 6B \$6941
27D0: 78 69 7E 63 69 6B 27 62 \$F2E4
27D8: 6F 6B 7C 6F 64 27 6B 7E \$9BC4
27E0: 66 6B 64 7E 63 79 27 7E \$DECF
27E8: 6B 69 65 67 6B 0A 7D 6B \$577A
27F0: 79 62 63 64 6D 7E 65 64 \$9E46
27F8: 27 68 78 65 6B 6E 7D 6B \$FE27
2800: 73 27 6F 64 6D 66 6B 64 \$BED5
2808: 6E 27 7E 62 6F 0A 79 7D \$4D14
2810: 63 79 79 0A 6B 66 7A 79 \$8863
2818: 27 6D 78 6B 64 6E 0A 69 \$83B2
2820: 6B 64 73 65 64 27 7E 62 \$CAC6
2828: 6F 0A 6F 7B 7F 6B 7E 65 \$373A
2830: 78 27 79 7E 65 64 6F 62 \$23B4
2838: 6F 64 6D 6F 27 7E 62 6F \$BE16
2840: 0A 6B 66 6B 67 65 27 \$0C11

checksums for...

Wheel of Money BASIC

10 - \$BADD 580 - \$32CF
20 - \$9B13 590 - \$3E37
30 - \$4D3B 600 - \$FA71
40 - \$AD92 610 - \$4A97
50 - \$C899 620 - \$9BB9
60 - \$FF65 630 - \$C6C7
70 - \$A3BF 640 - \$1FBF
80 - \$A900 650 - \$C6FA
90 - \$924D 660 - \$A034
100 - \$7F08 670 - \$6760
110 - \$FCF9 680 - \$8563
120 - \$1A24 690 - \$8B81
130 - \$AD42 700 - \$67EF
140 - \$979B 710 - \$9145
150 - \$AEDD 720 - \$2CE8
160 - \$1348 730 - \$0F15
170 - \$923F 740 - \$D382
180 - \$2A6C 750 - \$1DEB
190 - \$FEE0 760 - \$505E
200 - \$3006 770 - \$A575
210 - \$FE6A 780 - \$0C1F
220 - \$3B4A 790 - \$694A
230 - \$3E70 800 - \$6309
240 - \$9A42 810 - \$D3E9
250 - \$8944 820 - \$80E5
260 - \$D05D 830 - \$7060
270 - \$67E0 840 - \$6A07
280 - \$50D5 850 - \$5A12
290 - \$0B21 860 - \$4C0B
300 - \$2533 870 - \$FAEF
310 - \$A796 880 - \$4C53
320 - \$3434 890 - \$3A48
330 - \$0599 900 - \$4607
340 - \$B3A1 910 - \$9919
350 - \$5833 920 - \$07FD
360 - \$B85E 930 - \$0C2E
370 - \$C7FB 940 - \$0ECF
380 - \$557D 950 - \$57B9
390 - \$B401 960 - \$C07C
400 - \$1CEE 970 - \$A1BA
410 - \$A0BE 980 - \$E47B
420 - \$BAAE 990 - \$36BE
430 - \$F767 1000 - \$D374
440 - \$0F11 1010 - \$65BB
450 - \$0ED0 1020 - \$EE82
460 - \$AC86 1030 - \$C8BF
470 - \$7295 1040 - \$88DD
480 - \$C36F 1050 - \$EF98
490 - \$EB69 1060 - \$A650
500 - \$89F2 1070 - \$A4F1
510 - \$6330 1080 - \$185C
520 - \$309B 1090 - \$0476
530 - \$8A7E 1100 - \$CA02
540 - \$2F3F 1110 - \$0DE0
550 - \$344E 1120 - \$6FBB
560 - \$EA97 1130 - \$E62E
570 - \$2795

checksums for...

Convert Words

10 - \$9FA2 50 - \$6803
20 - \$B3BE 60 - \$CBC4
30 - \$4969 70 - \$4468
40 - \$9625 80 - \$14BF

BACKUP PROTECTED SOFTWARE with COPY II PLUS™ ver. 5

From the team who first brought you COPY II PLUS in 1981 comes a completely updated disk backup utility for your Apple // computer. New features include:

- **Fully automatic bit copy***. All parameters are stored on disk. Simply type in the name of the program you wish to backup, and COPY II PLUS does the rest!
- **New utilities** including Alphabetize Catalog, Fast 2-pass Disk Copy on a //c or //e, and an all-new Sector Editor.
- **Supplied on a standard DOS** diskette. Runs on the Apple //, Apple //+, Apple //e, Apple //c. Requires 64K and one or two disk drives.

Increase the power of your Apple //...

Use COPY II PLUS™ 5.0

Call M-F 8-5:30 (W. Coast time) with your ☎ : 503/244-5782.
Or send a check (add \$3 s/h, \$8 overseas) to



\$39.95

**CENTRAL POINT
Software, Inc.**

9700 SW Capitol Hwy. #100
Portland, OR 97219

* We update Copy II Plus regularly to handle new protections; you as a registered owner may update at any time for 1/2 price! (To update, just send original disk and \$20.)

This product is provided for the purpose of enabling you to make archival copies only.

WHO LIKES THE CIA ?



HERE ARE JUST A
FEW OF THE MANY

"an essential part of the Apple-user's repertoire" - APPLE USER

"a valuable buy ... manual is practically worth having on its own" - WASHINGTON APPLE PI*

"the folks at Golden Delicious should be commended ... worth waiting for" - HARDCORE COMPUTIST

"multifaceted" - NIBBLE

"the most comprehensive disk accessor I have ever come across" - A.B., VERNON, CANADA

"its ability to unlock other programs will greatly help me" - DR. B.P., SAN FRANCISCO, CALIFORNIA

"an excellent set of programs ... just great - and good value too" - E.A.S., MILTON KEYNES, ENGLAND

"very, very educational ... great manual ... it is FANTASTIC!!!!!!" - J.C., TUCSON, ARIZONA

"a very enlightening piece of software/book ... top of my list for good buys" - H.S., BLAINE, MINNESOTA

"I like yours the BEST" - R.R., CHICAGO

Why all the excitement about the CIA (confidential information advisors)? Probably because it is the ONLY set of utilities (5 in all) which enable even a beginner to investigate, edit, locate, list, trace, rescue, translate, patch, repair, verify, examine, protect, unprotect, analyse, encrypt, and decrypt programs on normal AND protected disks. You also get the "CIA Files", a 65000+ word book which contains detailed instructions for using the C.I.A. plus easy-to-follow, hand-holding tutorials about patching, repair, formatting, encoding, protection, and numerous other disk topics. You'll find plenty of material here which has never appeared in print. PROGRAMS NOT COPY PROTECTED

To put the 5 C.I.A. utilities, plus book, on the trail of your Apple II+, IIe, & IIC disks, send \$95.00 by check or money order to:

GOLDEN DELICIOUS SOFTWARE LTD.
350 Fifth Avenue, Suite 3308, Dept H, New York, New York 10118

DEALERS

Overstocked on computer magazines
that seem to have
MORE ADS than articles?

Put Hardcore COMPUTIST
on your shelves, instead!

Here's how:

Send in the coupon below and we'll put your dealer info in the mail. DON'T WAIT! Apple computer users everywhere salute you!

Yes! Please send me information on how I can obtain copies of Hardcore COMPUTIST at substantial savings.

Dealer Name _____

Address _____

City _____

State _____

Zip _____

Send to: Hardcore COMPUTIST, PO Box 110816,
Tacoma, WA 98411 or call (206) 474-5750.



DISC COMMANDER

COPYRIGHT 1985 WILLIAM STEPHENS

DISK EDITING SYSTEM
FOR THE APPLE II FAMILY OF COMPUTERS
USING DOS 3.3 OR DOS 3.3 WITH
PRONTO-DOS ENHANCEMENT

• 4 Editors • 13 Sub-editors • Completely menu driven • 1 or 2 drive operation • INSTRUCTIONALS and TECHNICAL INFORMATION on disk
• Sort catalog entries • Hide catalog entries • Create multiple heading catalogs • Create multi-column catalogs • Reassign lock/unlock and file type symbols, spaces, columns, catalog heading and program titles to any combination of NORMAL, FLASH, INVERSE, LOWER CASE or CONTROL characters • Rename all DOS commands • Rewrite all DOS error messages • Make a (non-VTOC dependent) disk map showing the TRACK/SECTOR LISTS, VTOC and CATALOG TRACK/SECTORS
• Trace out individual files • Change the HELLO program • Change any byte anywhere on a disk • Make instructions, disk maps, track/sector displays and traces using your printer

AND MUCH

MUCH MORE

SO WHAT

SOFTWARE

10221 Slater Ave. Suite 103 Fountain Valley, Ca. 92708

TO ORDER: Send a check or money order (U.S. funds) for \$29.95 plus \$2.00 shipping (\$5.00 foreign) California residents add 6%.

WARNING!

You are entering DANGEROUS territory

The
Book of Softkeys

*shows you how to softkey
(remove copy-protection from)
commercial software.*

Volume I 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, Homeward, 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.

Volumes II and III are being compiled now!

YES, I want ___ copies of The Book Of Softkeys Volume 1. I have enclosed \$20 per book. Foreign orders add 20%. U.S. funds drawn on U.S. banks. Washington state orders add 7.8% sales tax. Send your orders to: Softkey Publishing, PO Box 110816, Tacoma, WA 98411

Name _____ HC23

Address _____

City _____ State _____ Zip _____

Country _____ Phone _____

Visa,MC _____ Exp. _____

Signature _____

If you want to make backups, then you want The Book Of Softkeys Volume I, the only resource that teaches you how to backup your expensive Apple)(software.

**Public Domain Software
SPECIAL
From The COMPUTER LEARNING CENTER!**

**GET Over 170
BUSINESS
& FINANCE
Programs FOR ONLY
\$28.⁰⁰**

- Forecast your investment profits, and evaluate the worth of your stock portfolio
- Experiment with Visicalc formulas, real estate and stock plots and design a business or personal letter writing format
- Organize your household expenses
- Balance your checkbook, manage your loans, payroll

The Business & Finance SPECIAL contains the following programs and many more...

Annuity	Real Estate Plot	Savings Growth
Bond Price & Interest	Visicalc Formulas	Treasury Bill Valuation
Budget Monthly	Business Finance	Address File
Decision Matrix	Check Stub	Compound Int. Tables
Financial Pak	Household Exp. Profile	File Manager
Keogh Savings Program	Income Tax 1040 For 77	Home Accounting
Mortgage Calculation	Inventory Company	Names Search
Regular Deposits I	Average Growth Rate	File Cabinet I
Stocks	Check Book Balancer	Family Finance
Calendar Personal	Income Taxes	Gen Ledger Printer
Letter Writer	Interest Earned	Constar
Phone List	Loan Balance	etc...

ORDER NOW!

You will also receive a complete catalog of Public Domain Software available from the CLC with your order.

Yes! Please send me the Business & Finance Special Offer (Volumes 18-25 from the CLC Public Domain Library). I understand that Public Domain Software is not commercial quality and is supplied as-is and that orders are filled on double sided disks.

Name _____ S-4

Address _____

City _____ St _____ Zip _____

Country _____ Phone _____

VISA/MC _____ Exp _____

Signature _____

Send check or money order (US funds drawn on US bank) to: Computer Learning Center, PO Box 110876-HC, Tacoma, WA 98411. Washington residents add 7.8% sales tax. Foreign orders add 20% shipping and handling.

(The Computer Learning Center's PDS Library contains over 175 volumes, which will run on Apple][Plus computers and Apple compatibles. Most will also run on Apple IIe and IIc.)

NEW DON LANCASTER RELEASES
Available ONLY from Synergetics. All software open and unlocked.

ABSOLUTE RESET (IIe/old IIe/new IIe) \$19.50

Get back in total control. No more hole blasting! Access any part of any program at any time for any reason. Passes all diagnostics. Invisibly till activated. Includes EPROM burner adapter details, service listings, and a free bonus book.

APPLEWRITER™ TOOLKITS (DOS 3.3 or ProDOS) \$39.50

EIGHT diskette sides include a complete disassembly script; source code capturing; self-prompting glossaries; Diablo microjustify and proportional space, patches including NULL, shortline, Grappler, IIe detaching, many others; answers to most help line questions; two columns; WPL secrets; space on disk; keyword indexing; multi and even-odd headers; bunches more.

Both TOOLKITS (sixteen disk sides) \$59.50

APPLEWORKS™ DISASSEMBLY SCRIPT \$49.50

TEN diskette sides chock full of Applework's innermost secrets, using Don's unique and powerful "tearing method". Primarily for gonzo hackers and definitely NOT for the faint of heart.

•LASERWRITER/APPLEWRITER UTILITIES \$39.50

Two volume package gives you unmatchably superb IIe text and graphics. Included are automatic formatters, boxes and fancy borders, daisywheel changers, envelope and label routines, unbelievably fast formletters, grids and rulers, HIRE5 converters, DCI patches, demos, powerful justify routines, self-prompting glossaries, help screens, a sign shop, outlined text, more.

AUTOGRAPHED LANCASTER BOOKS:

Apple IIe Assembly Cookbook	\$21.50
All About Applewriter IIe	\$14.50
•Enhancing Your Apple II & IIe, (Volume I or II)	\$15.50
Micro Cookbook (Volume I or II)	\$15.50
CMOS Cookbook	\$14.50
TTL Cookbook	\$12.50
Incredible Secret Money Machine	\$7.50
Complete book and software list	(FREE)

COMPANION DISKETTES:

For Enhancing Your Apple II, Volume I	\$19.50
For Enhancing Your Apple II, Volume II	\$19.50
For Don Lancaster's Assembly Cookbook	\$19.50

SYNERGETICS
746 First Street
Box 809-HC
Thatcher, AZ, 85552

**FREE
VOICE HELPLINE**
(602) 428-4073

Appleworks, Applewriter, and ProDOS are registered trademarks of Apple Computer. VISA and MASTERCHARGE accepted. Please - no COD, foreign, or purchase orders.

THE MOST POWERFUL BACK-UP UTILITY
YOU'VE EVER SEEN...



\$59.95 add \$5 ship.

- * Back-ups 1/2, 1/4, 3/4 tracks.
 - * Automatic back-up options.
 - * No parms needed for most of the back-ups.
 - * Excellent DOS copy on flip side.
- Ask for our other products RAM-LOCK (for L-Smith)
And also SHOUGI (Japanese chess-type game.)

ART GALLERY
Yoshinoya Bldg, 438 Sasu-machi, Chofu-shi
Tokyo 182, Japan
Send money or check. VISA/MASTER CARD accepted.

NEW SUBSCRIBER?

of Hardcore COMPUTIST

and **★ CORE** are **PACKED** with information that you won't want to miss!

BACK ISSUES

Issue	Mag	Disk
	\$4.75	\$9.95
HC 22	<input type="checkbox"/>	<input type="checkbox"/>
HC 21	<input type="checkbox"/>	<input type="checkbox"/>
HC 20	<input type="checkbox"/>	<input type="checkbox"/>
HC 19	<input type="checkbox"/>	<input type="checkbox"/>
HC 18	<input type="checkbox"/>	<input type="checkbox"/>
HC 17	<input type="checkbox"/>	<input type="checkbox"/>
HC 16	<input type="checkbox"/>	<input type="checkbox"/>
HC 13	<input type="checkbox"/>	<input type="checkbox"/>
HC 11	<input type="checkbox"/>	<input type="checkbox"/>
HC 4	<input type="checkbox"/>	<input type="checkbox"/>
HC 1	<input type="checkbox"/>	see below
C 3	<input type="checkbox"/>	<input type="checkbox"/>
C 2	<input type="checkbox"/>	<input type="checkbox"/>
C 1	<input type="checkbox"/>	see below
HC 1 / C 1	disk only	<input type="checkbox"/>

Send me the back issues indicated above:

Name _____

ID# _____

Address _____

City _____

State _____ Zip _____

Phone _____

VISA/MC _____

Exp _____

Signature _____

Send check or money order to:
 Hardcore COMPUTIST, PO Box 110846-T,
 Tacoma, WA 98411. Most orders shipped
 UPS. Please use street address. Washington
 residents add 7.8% sales tax. Foreign orders
 add 20% shipping and handling. US funds
 drawn on US bank.

Hardcore COMPUTIST 22: Softkeys for Miner 2049er, Iode Runner, A2-PB1 Pinball; Readers' Softkeys to the Heist, Old Ironsides, Grandma's House, In Search of the Most Amazing Thing, Morloc's Tower, Marauder, Sargon III; Features: Customized Drive Speed Control, Super IOB version 1.5; Core: The Macro System

Hardcore COMPUTIST 21: Softkeys for DB Master version 4+, Dazzle Draw, Archon, Twerps; Readers' Softkeys to Advanced Blackjack, Megaworks, Summer Games, College Entrance Exam Prep, Appewriter revisited; Features: Demystifying The Quarter Track; Core: Proshadow: A ProDOS Disk Monitor

Hardcore COMPUTIST 20: Softkeys for Sargon III, Wizardry; Proving Grounds of the Mad Overlord and Knight of Diamonds, The Report Card v1.1 and Kidwriter / Apple II Boot ROM Disassembly / The Graphic Grabber v3.0 / Copy II+ 5.0: A Review / The Know-Drive: A Hardware Evaluation / An Improved BASIC/Binary Combo

Hardcore COMPUTIST 19: Softkeys for Rendezvous With Rama, Peachtree's Back To Basics Accounting System, HSD Statistics Series, Arithmetick, Arithmekicks and Early Games for Children / Double Your ROM Space / The Games of 1984: In Review- Part II / Towards a Better F8 ROM / The Nibbler: A Utility Program to Examine Raw Nibbles From Disk

Hardcore COMPUTIST 18: Softkeys for the Scholastic Version of Bank Street Writer, Appewriter IIe, SSI's Non-RODOS Disks, BPI Accounting Programs and DesignWare Programs / Installing a Free Sector Patch Into Appewriter IIe / The Games of 1984: In Review / 65C02 Chips Now Available / Checksoft v2 / Simple Copy Protection

Hardcore COMPUTIST 17: Softkeys for The Print Shop, Crossword Magic, The Standing Stones, Beer Run, Skyfox, and Random House Disks / A Tutorial For Disk Inspection and the Use Of Super IOB / The Graphic Grabber For The Print Shop / The Lone Catalog Arranger Part Two / S-C Macro Assembler Directives (Reprint)

Hardcore COMPUTIST 16: Softkeys for Rescue Raiders, Sheila, Basic Building Blocks, Artsci Programs, Crossfire, Sensible Speller for ProDOS and Sideways / Secret Weapon: RAMcard / The Controller Writer / A Fix For The Beyond Castle Wolfenstein Softkey / The Lone Catalog Arranger Part 1

Hardcore COMPUTIST 13: Softkeys for Laif Pak, Beyond Castle Wolfenstein, Transylvania, The Quest, Electronic Arts, Snooper Troops (Case 2), DLM Software, Learning With Leeper, & TellStar / CSaver: The Advanced Way to Store Super IOB Controllers / Adding New Commands to DOS 3.3 / Fixing ProDOS 1.0.1 BSAVE Bug / REVIEW: Enhancing Your Apple / Locksmith 5.0 and Locksmith: Programming Language

Hardcore COMPUTIST 11: Copy II Plus 4.4C Update / PARS for Essential Data Duplicator / Ultimaker III / Mapping of Ultima III / Ultima II...The Rest of the Picture / Softkeys for Sensible Speller, Ultima III, Softporn Adventure, The Einstein Compiler v5.3, & Mask of the Sun

Hardcore COMPUTIST 4: Ultima II Character Editor / Softkeys for Ultima II, Witness, Prisoner II, & Pest Patrol / Adventure Tips for Ultima II & III / Copy II Plus PARS Update

Hardcore COMPUTIST 1: Softkeys for Data Reporter, Multiplan & Zork / PARS for Copy II Plus / No More Bugs / APT's for Choplifter & Cannonball Blitz / Reviews: Replay, Crackshot, Snapshot & Wildcard copy cards

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 / and Dragon Dungeon

CORE 2 Utilities: 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

(*CORE is no longer published as an independent quarterly magazine.)

Back issues not listed are no longer available. But disks are still available for ALL sold-out issues of Hardcore COMPUTIST upon request.

Don't **TYPE IN** programs that appear in Hardcore COMPUTIST.

Order the Library Disk, instead!

Each month a Library Disk with all the programs that appeared in the previous issue of Hardcore COMPUTIST is prepared for **SMART READERS** like you who have *better* things to do with their time than type in program listings. Please use the order form to the left to order disks.

Offer good while supply lasts.

*By Hackers
For Hackers*

- ELITE BOARD DOWNLOADS
- CRACKING TIPS
- PHREAKING SECTION
- GAME CHEATS
- PARMS
- PROGRAMS
- INTERVIEWS
- ADVENTURE TRIPS
- HACKING TIPS
- MYSTERY SECTIONS

Published on both sides of
an Apple diskette -
4 times a year.

The BOOT-LEGGER MAGAZINE

Subscribe Now!

Send 25 Bucks for a 1-Year Subscription
THE BOOT LEGGER, 3310 Holland Loop
Road, Cave Junction, Oregon 97523.
Overseas Subscriptions \$50.
Canadian \$30 U.S. Currency.

FOR AD INFO. & QUESTIONS
CALL BOOTLEG AT (503) 592-4461

You want the old issues of Hardcore Computing?

Well, WE'VE TORN OFF THE FANCY COVERS!

- And deleted all of the editorial material, out-of-date interviews and letters
- And updated the remaining material and included an enormous list of bit copy parameters
- And **PACKED IT ALL** into a *single volume*

**Don't wait.
GET YOUR COPY NOW!**

**NOW,
WHAT HAVE YOU GOT?**

The Best Of
**Hardcore
Computing**

To order your copy of **The Best Of Hardcore Computing** send \$19.95 check or money order (US funds only) to: Hardcore COMPUTIST, PO Box 110846-B, Tacoma, WA 98411. Washington state residents add 7.8% sales tax. Foreign orders add 20% shipping and handling. VISA and MC orders enclose signature and expiration date.

Pineapple 6502™

SOFTWARES INC.

The only store in TOKYO

Where you can find imported softwares and books and lots
lots more at reasonable cheap prices.

New programs welcomed..

If you have a newly made games, utilities or peripherals,
and wish to sell them in Japan, we welcome your inquiry.

Al Bergo Ochanomizu 718, 2-1-19 Kanda-Surugadai, Chiyoda-ku
Tokyo 101, JAPAN 03-294-6502

We do not deal or have any relation with Pine-Com