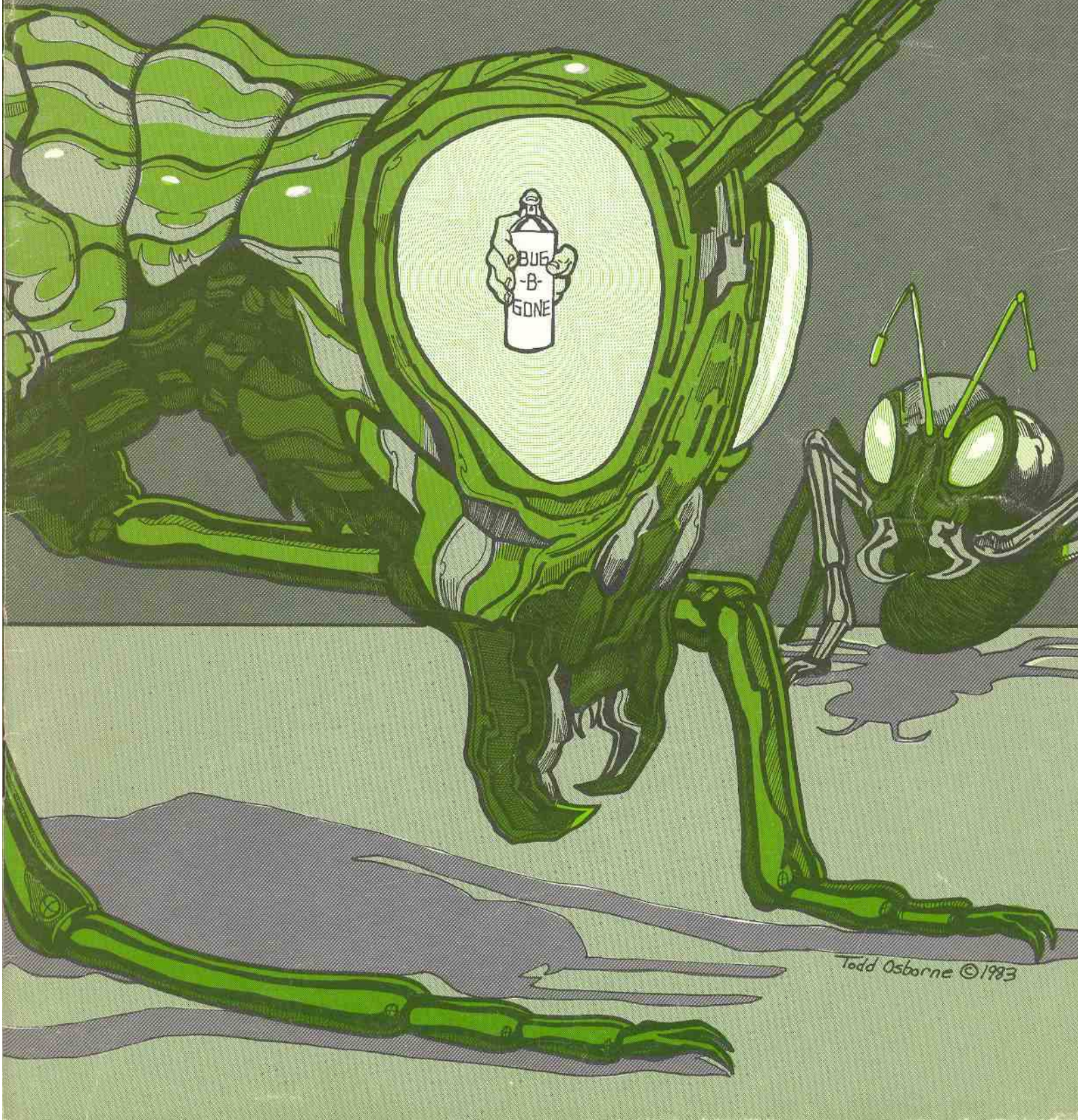


**NO MORE BUGS!**

\$2.50

# hardcore

for the serious Apple-user and hard-core **computist** 1



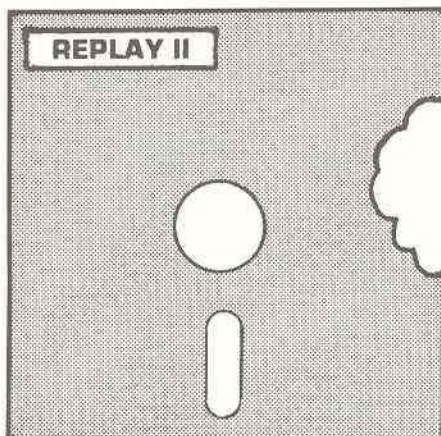
Todd Osborne ©1983

**NEW**

# REPLAY II

## APPLE PROGRAM COPY SYSTEM

- **Disk Formatting Irrelevant**
- **Does not interfere with other cards**
- **Menu driven**
- **Card is transparent until copy desired.**
- **Copy \*and\* restart in under 15 seconds**
- **Will copy all 64k with \*no\* compression**
- **Copies total load programs**



### TO ORDER OR REQUEST INFO:

Write or Call

**MICRO ANALYST, INC.**

**P.O. Box 15003**

**Austin, TX 78761**

[512] 926-4527

Dealer Inquiries Welcome

### COST:

\$150.00 (includes postage)

Outside U.S./Canada add \$10 shipping

Texas residents add \$7.50 sales tax

VISA/MASTERCARD Accepted!!

**REPLAY II** is intended for archival backup purposes only.

\*Apple is a registered trademark for Apple Computers Inc.

## MICRO ANALYST, INC.

**REPLAY II** is an interface card that is slot independent. Users can stop the program, examine and change memory, then restart without changing disks. Control of the system is obtained by pressing the remote switch which comes on an 18 inch cord outside the Apple\*. If a copy is desired a blank disk is inserted in Drive 1 and the copy option is selected from the menu. **REPLAY II** does not copy the disk, rather it copies the program executing in memory.

**REPLAY II** does not change ANY memory - unlike other copy cards. It faithfully reproduces the lower 48k of memory in a fast load format. The upper 16k of memory can also be copied for a 64k copy. Standard DOS 3.3 files can be created for storage on hard disk, although the RAM card is needed.

**REPLAY II** can move protected APPLESOFT programs to standard DOS 3.3 disk automatically.

Utility programs supplied with the **REPLAY II** card include Program Analysis, Packing, and Compression. A language card is not needed to run packed programs. Because most programs are written in Assembly language, the user should be familiar with Assembly in order to fully utilize the Analysis and Packing programs.

NOW game player can save a game at ANY LEVEL and QUICKLY restart it with the **REPLAY II** card.

AT LAST!!! A full sized card with full sized capabilities!!!



# hardcore

for the serious Apple-user and hard-core **computist 1**

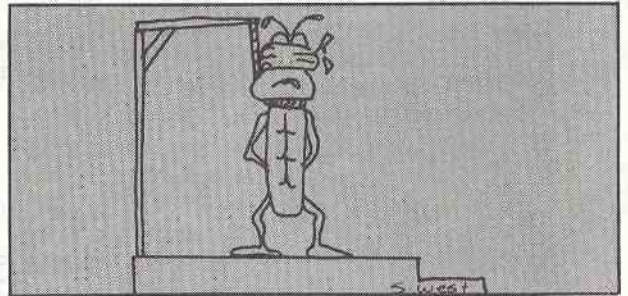
## Features

### SOFTKEYS

Data Reporter, by Don Halley . . . . .	4
Multiplan . . . . .	4
Zork . . . . .	5

### PARAMETERS

Parms for Copy II Plus . . . . .	6
----------------------------------	---



### SPECIAL FEATURE

No More Bugs, by Robb Canfield . . . . .	10
Checksoft . . . . .	11
Checkbin . . . . .	21

### A.P.T.'s

Choplifter . . . . .	20
Cannonball Blitz . . . . .	24

## Reviews

### COPY CARDS

Replay . . . . .	27
Crackshot, Snapshot, Wildcard. . . . .	28
<i>(reprinted from Peelings II)</i>	

## Departments

Letters . . . . .	2
Publisher's note . . . . .	3
Checksums for CORE . . . . .	30
Ad Index . . . . .	32

# INPUT INPUT INPUT

## Fed Up Down Under

The whole business of copy-protection sickens me—I resent being treated like a child and told I can only look and not touch! (All my CP/M programs which are all much more expensive, or average, than Apple 6502 programs, have not one bit of copy-protection!) Consequently, even though I am not a reseller of "pirated" programs, I do like to have backups and so have copied now most of my software (which runs to more than 300 diskfuls, some even D/S). Even Locksmith seems to be back-peddalling a little and not supplying parm changes on some programs anymore.

Dr. Leigh Rowan-Kelly  
Australia

## Breaking Software Wastes Time

Keep up the good work! Although protection may be justified for games, I feel that the use of copy-protection for serious applications has greatly damaged the usefulness of the Apple as a business machine. Having had much experience in running critical medical applications (on large mini s) I could not

tolerate becoming dependent on software which I could not back up to the extent that I (or my auditors) felt adequate. This always exceeds the "1 backup per user" currently allowed by the vendors of protected program packages. While my use of the Apple is currently mainly recreational, my biases carry over from the "real world" systems. As I become more involved in Apple applications, I begin to resent the time I must spend to "break" the software that I have already spent good money to own. CP/M packages now appear more attractive, because they are not protected (WHY???). I am seriously considering buying a data base program; and will probably choose dBASE over DBmaster because (among other things) dBASE can be backed up.

Mitchell Pilot, M.D.  
Atlanta, GA

## Pac-Man Clones Hardly "Grand Piracy"

I must take exception to your "Grand Piracy" editorial [**HARDCORE COMPUTING #3**]. Nothing could have missed the mark more. Now I'm no expert on copyright law, but I cannot see where Atari's copyright is being infringed by anyone publishing a Pac Man look-alike.

Are we to assume that all Whatsit-Calc spreadsheet programs are a ripoff

of someone else's? And what about all those screen oriented text editors that all do more-or-less the same thing on all those different machines? If the Atari precedent stands, a lot of people are in BIG trouble.

You want Grand Piracy? Copy someone else's ROM and put it in your Z-80 card or Apple-compatible computer. There's Grand Piracy. Run off copies of someone else's copyrighted program and sell them. There's Grand Piracy. But to see a game and then write your own version of it? Sorry, that doesn't qualify in my book.

Unless I'm mistaken, even in board games the same board cannot be copyrighted. The name may be a trademark but neither the playing pieces (dots? ghosts?) nor the board (a maze?) can be protected by a copyright. Only the rules of play can be copyright protected. So how does Atari get away with claiming copyright protection for their "images"? They can pull off pretty much anything they want as long as practically no one can afford to muster a battalion of attorneys to do battle against Warner Communications in court. Let's face it, most anyone else would be outgunned.

Even if everything does come out in Atari's favor, have they really gained anything? As an Apple owner, I'm not about to go out and buy either an Atari video game or computer just to play Pac Man. I suppose they might come

*continued on page 25*

**Publisher:**  
Charles R. Haight

**Editor:**  
Bev R. Haight

**Business and  
Circulation Manager:**  
Karen Fitzpatrick

**Assistant Editor:**  
Julie Joringdal

**Production:**  
David C. Smith  
Cheri J. Harrell

**Technical Editor:**  
Robb Canfield

**Cover Art:**  
Todd Osborne

**Illustrators:**  
Todd Osborne  
Luke West  
Steve West  
Ryuji

**Typeset via Modem:**  
Graphic Services  
Tacoma, WA

**Printing:**  
Grange Printing, Inc.  
Seattle, WA

**Publishing:**  
Softkey Publishing  
P.O. Box 44549  
Tacoma, WA 98444

erence (without the 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 or Softkey Publishing.

Address all editorial, advertising, and subscription inquiries to the proper department, HARDCORE COMPUTIST, P.O. Box 44549, Tacoma, WA 98444. (For subscription information see page 25.) Return postage must accompany all manuscripts, drawings, photos, disks, or tapes if they are to be returned. No responsibility can be assumed for unsolicited manuscripts. We suggest you send only copies.

DOMESTIC DEALER RATES sent upon request, or call (206) 581-6038.

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

Entire contents copyright 1983 by Softkey Publishing. All rights reserved. Copying done for other than personal or internal ref-

## ANOTHER PREMIERE ISSUE?!?

Now you've really gone and done it. You got the editor so mixed up with your divergent opinions that he (it?) has split in two, taking the magazine with him. But since it's difficult to sell a magazine torn in halves, the remainder of the editorial staff has mended both parts (yes, both, as in two instead of one).

This traumatic division of the editor has two bright sides. The new Hardcore Computing is now both CORE and **HARDCORE COMPUTIST**. CORE, a quarterly, will tackle the little problems, like, "Tell me all about graphics." **HARDCORE COMPUTIST**, which will come out eight times a year, will handle the b-i-g problems, like, "How do I make a copy of *Zork*?"

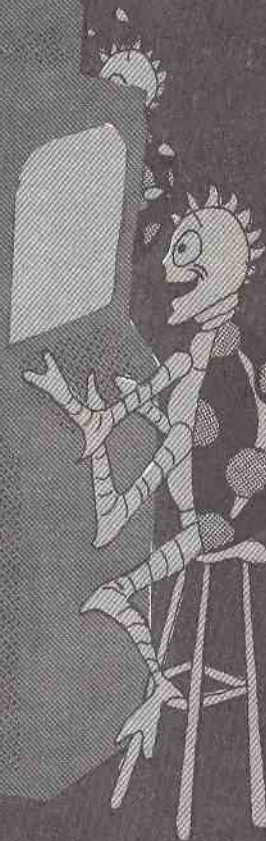
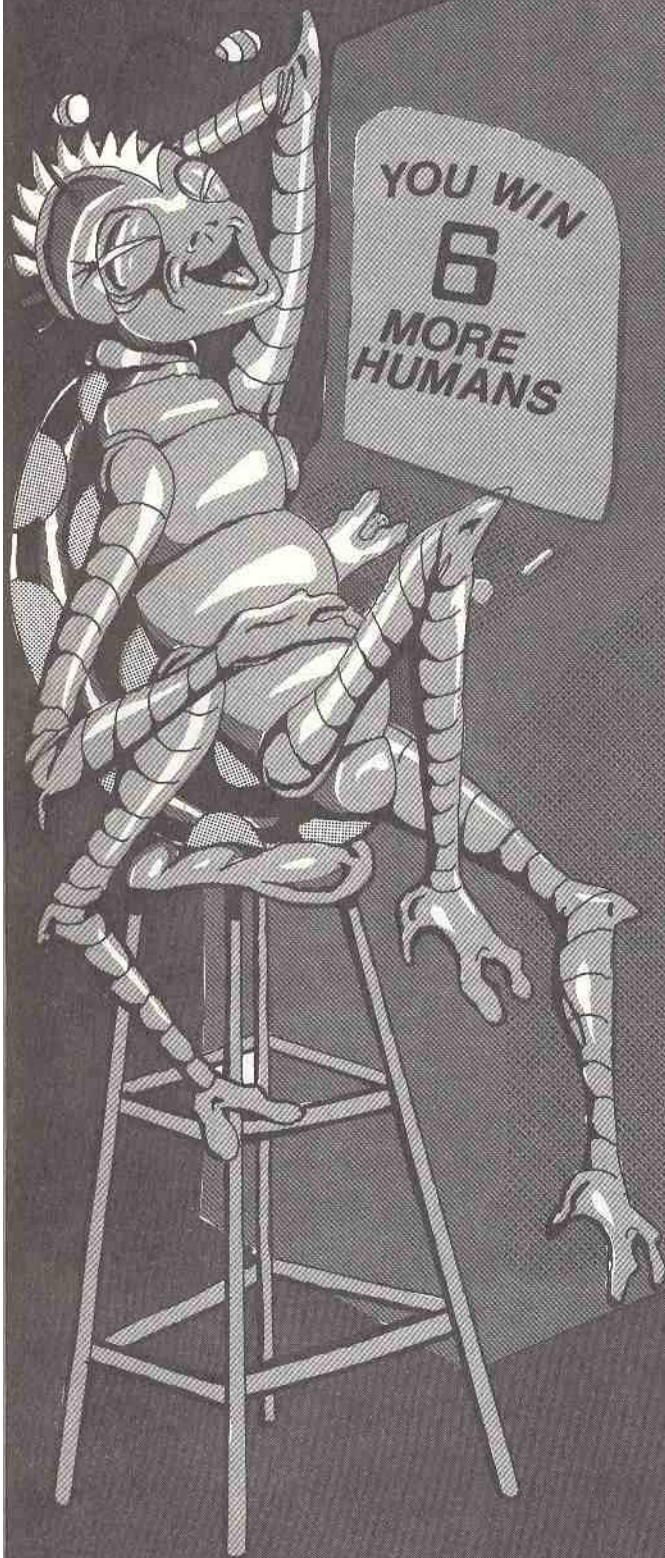
Remember the mythical Hardcore Computing number 4, the graphics special? Well, if you take the premiere issue of CORE and add the premiere issue of **HARDCORE COMPUTIST**, you'll get a total that equates as that mythical special.

Confused? Not yet? Well, let me reword it. Your subscription to the old Hardcore Computing (pre-schism) is equal to a subscription to both CORE and **HARDCORE COMPUTIST** (post-schism). While CORE is aimed at all Apple users (yes, that includes users of the Franklin Ace and other pseudo-Apples and Apple clones), **HARDCORE COMPUTIST** is aimed at those interested in making back-ups of locked software and examining everything under the sun that results from that practice, including the debates, reviews, A.P.T.s, mods, and fixes.

Welcome to the new **HARDCORE COMPUTIST**

It's not flashy, but it has what it takes to keep hardcore computists informed about what they say they want. So keep the articles, letters, and programming tips coming. The more we get, the more we print.

The feature for this premiere issue is Robb Canfield's checksum programs: Checksoft and Checkbin. Also included are parameters for *Copy II Plus*, and some softkeys and A.P.T.s.



# SOFTKEYS

SOFTKEY FOR

## Data Reporter

by Don Halley

**Data Reporter**, Synergistic Software, 5221 120th Avenue S.E., Bellevue, Washington 98006, (206) 226-3216, \$220.00

*The Data Reporter*, from Synergistic Software, is advertised as Version 2 of the popular modifiable data base system. It is basically a data storage and retrieval system with graph plotting capabilities, a text editor, and many data management features.

It is offered at a suggested retail price of \$220.00. There is a \$5.00 charge for a back-up disk.

Documentation is fairly complete, although it suffers from the same unimaginative approach to organization as does the documentation for its competitors. There is no index, and the table of contents offers no help beyond one- or two-word references to program features. A tutorial section would be appreciated by the uninitiated user, and a reference table containing pointers to key sections would help.

### Normal Copy

The documentation suggests that a copy of the original disk be made for general use, in order to prolong the original's life. The COPYA program on the DOS System Master may be used, but it will encounter a read error on the last track. This track has been written with a modified DOS, and its contents are read into memory via a short machine language program appended to the last line of the HELLO program. This means that you must always boot from the original, then swap to your application disk for processing.

### Unprotected Copy

That is, only if you want to. A little PEEKing around will reveal that the protected sectors are from \$00 to \$06 on track \$22, and that the information contained there is loaded into memory from \$9400 to \$9AFF. Examination of the HELLO program shows that it does not touch this area upon exit.

This means that both a way of reading the protected portion of the disk into the proper memory locations (HELLO) and a clean exit have been provided.

Try the following sequence:

1. RUN COPYA to create a copy of the original disk. (Remember to ignore the read error.)
2. RUN the HELLO program on the original disk.
3. Choose the QUIT option from the primary menu.

SOFTKEY FOR

## Multiplan

by Bobby

**Multiplan**, Microsoft Corporation, 10700 Northup Way, Bellevue, Washington 98004 \$275.00

### REQUIREMENTS:

The Multiplan disk

The IOB program from Issue 3 of *Hardcore Computing* or IOB+ from *Hardcore Computing Update 3.1* (for single drive users)

**Multiplan** is an excellent spreadsheet program by Microsoft. It includes an unusually complete manual with a reference guide, and an auto-help mode from within the program. Multiplan allows one and only one back-up

NOTE: Softkey Publishing was unable to test the authenticity of this Softkey. Reader response is encouraged.

4. Replace the original disk with your copy.
5. Issue these commands:

```
BSAVE HELLO.OBJ,  
A$9400,L$06FF
```

```
LOCK HELLO.OBJ
```

```
UNLOCK HELLO
```

```
63999 PRINT D$;  
"BLOAD HELLO.OBJ" : RETURN
```

```
SAVE HELLO
```

```
LOCK HELLO
```

```
PR#6
```

You now have a fully operational back-up, for only the cost of the disk itself. Of course, you may make as many additional back-ups as you like from this disk.

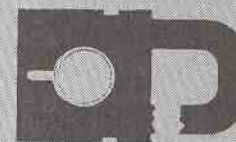
to be made, which I found to be an insufficient guarantee (three is my minimum back-up policy for commercial software).

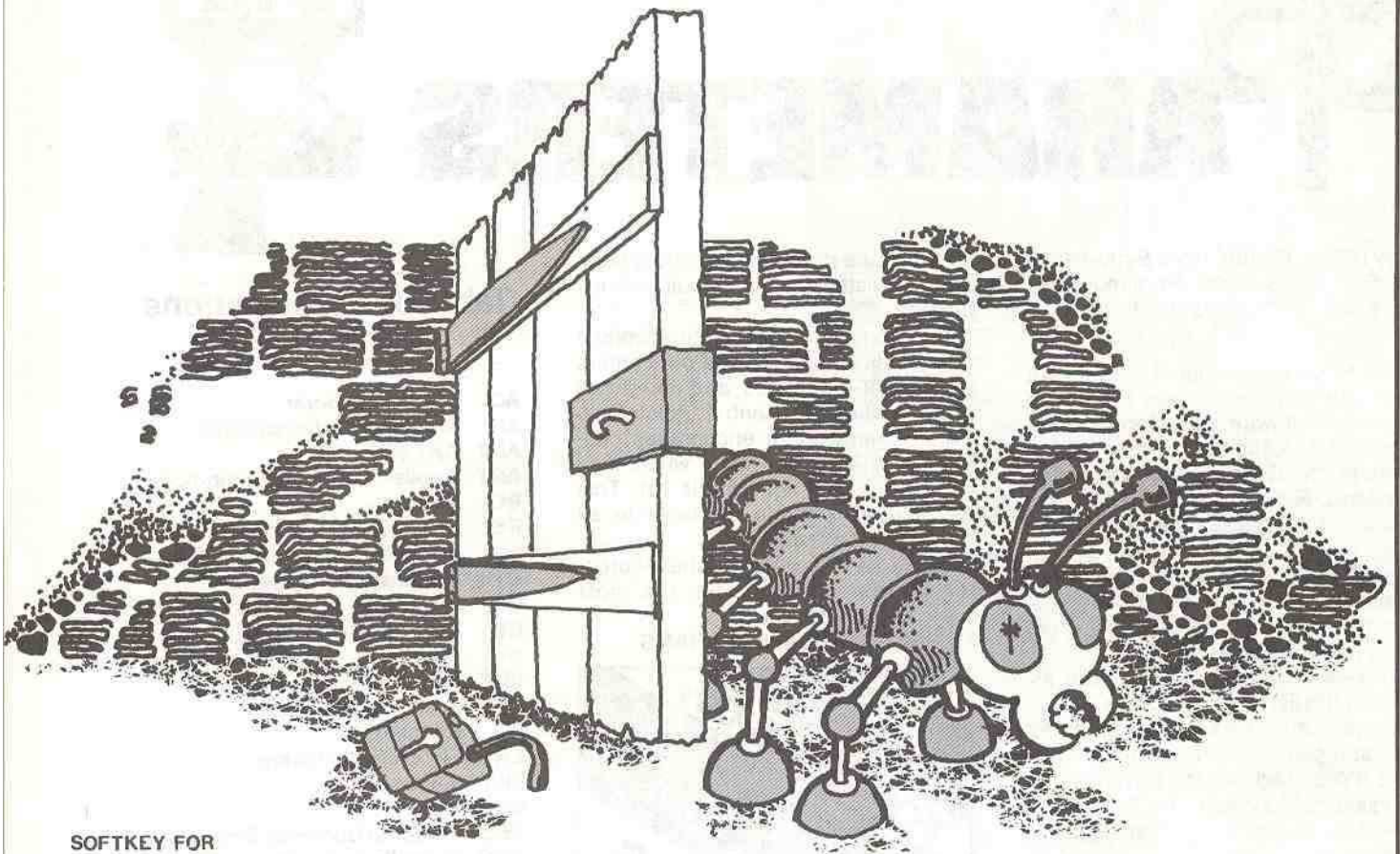
The program is only protected on tracks zero through four. The protection scheme is to change the end of the address mark on those tracks from 0E to CB. To allow the Multiplan DOS to read the unprotected disk, a mod must be done to track 0, sector A, changing byte D from CB to DE. This mod is automatically done by the IOB program.

Copy Multiplan by using the IOB program and the controller listed below. The IOB copy of Multiplan can then be duplicated with COPYA, or any number of other copy programs.



*Hardcore Computist* invites our readers to submit Softkeys (methods of unlocking programs) for a Reader's Softkey Exchange to begin in **HARDCORE #3**. Softkey Publishing will be unable to authenticate every submission, although we will test as many as possible.





## SOFTKEY FOR **Zork**

by Bobby

**Zork**, Infocom, Inc.,  
55 Wheeler Street  
Cambridge, MA 02138  
\$39.95

### REQUIREMENTS:

An Apple II  
At least one disk drive  
A copy of Zork  
COPYA  
A disk editing program, such as  
DiskEdit

**Zork** is a challenging adventure game from Insoft. Although a hi-res picture is lacking (since it is a text game), Zork is one of the best adventure games I have ever attempted to solve.

While trying to solve some of the puzzles, I started to do a little APT and found that Zork was on a protected disk. I set it aside until I had the time to examine the program. Then a reader called, explaining a way to unprotect Zork. Believe it or not, the COPYA program on the system master disk can be used.

## How to Copy Zork

This copy method works on Zork versions I, II, and III.

1. Boot the 3.3 master disk.

### PR#6

2. Run COPYA
3. Once it is in memory, press ctrl C to halt the program.
4. Delete line 70. This stops the program from reloading a machine language routine each time it is run.
5. Drop into the monitor.

### CALL -151

6. Create the following modes.
  - B925:18 60 Kill end of data marks.
  - B988:18 60 Kill end of address marks.
  - BE48:18 Clear errors.
  - B8FB:29 00 Kill last byte of beginning of data mark.

7. Return to Applesoft.

### 3D0G

8. Run the program.

### RUN

9. After the copy is done, reboot the disk with the 3.3 master.
10. Run your disk editing program.
11. Read track 0, sector 2.
12. Modify the following values:

### Change location:

**5D from BC to AD**

**FB from C9 to 29**

**FC from BC to 00**

You now have an unprotected version of Zork. It can be duplicated by any of the numerous copy programs available, even COPYA, without the use of modes. The disk cannot be cataloged, nor may separate files be run; it must be booted to play the game. But Zork is now open to inspection by those wishing to participate in the rapidly growing hobby of APT.

# PARAMETERS



**Copy II Plus**, Central Point Software  
P.O. Box 10730, #203, Portland, Ore-  
gon 97219, (503) 244-5782 \$39.95.

The following is a list of parameters to change in order to back up certain pieces of software with **Copy II Plus version 4.1**. To the right of the program name is the abbreviated name of the publisher. For a complete list of the publishers, please see the table on this page.

When making a back-up, be sure to follow the steps in order. Often a parameter will not be relisted if it is set for a prior range of tracks.

To back up a program, first find its name in the list of parameters. Directly below the name is a list of the tracks to copy and parameters to change. If the word BY is used, set the increment to the value that follows it. Use the default increment of one if no other figure is given.

When the word SECTMOD appears, it means that a sector should be changed using the Track Sector-Editor. Be sure to patch the read/write routines if the listing shows PATCHED and to use the correct DOS (3.2 or 3.3). Place the destination disk in drive one, then perform the changes listed.

The command format is:

```
SECTMOD {F=n, C=n, S=n, T=n}
DOS 3.n PATCHED
CHANGE ADDRESS A1 FROM A2 TO
A3
```

The meaning of F, C, S, T and A1, A2, A3 are explained below:

- F** Disk format to be used. The value (n) will be either 13 or 16.
- C** Toggle. The value (n) will be either on or off.
- S** Sector to be read.
- T** Track to be read.
- A1** Location to be changed in the buffer.
- A2** Old value.
- A3** New value.

The middle line from the example gives the DOS (3.2 or 3.3) patched.

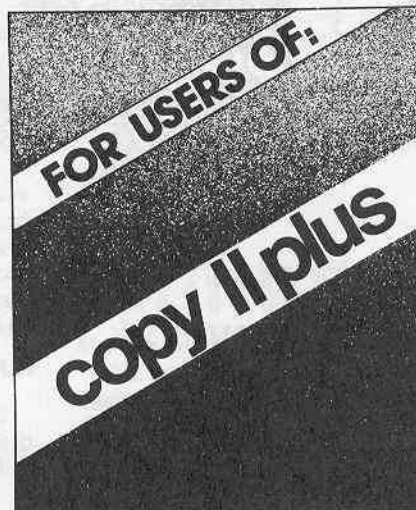
Some diskettes can be duplicated using the default parameters (select the Bit Copy option from the main

menu). If the diskette you wish to back up is not listed, try the default settings anyway.

An asterisk (\*) next to the product name indicates that these parameters were user-submitted and have not been verified by Central Point Software. Central Point encourages customers to let them know when they back up a disk not on this list. This information is made available to all Copy II Plus owners.

If you have access to The Source, the command to type out the most recent parameter list is

```
TY SFILES > ST2008 > PARAMS
```



## Table of Abbreviations of Publishers

AC	Apple Computer
AI	Adventure International
ART	ARTSCI
AST	Applied Software Technology
BC	Budgetco
BS	Broderbund Software
CC	Cavalier Computer
CTS	Continental Software
DM	Data Most
EW	Edu-Ware
HN	Hayden
ISM	ISM
KL	Krell
KN	Kensington
LNS	Lightning Software
MIS	Microsoft
MU	Muse
PBS	Personal Business Systems
PDS	Picadilly Software
SEN	Sensible Software
SL	Sub Logic
SOL	Sierra On-Line
SPC	Software Publishing Corp.
SRS	Sirius Software
SVS	Silicon Valley Software
SW	Stoneware
TKS	Turnkey Software
UNK	Unknown
USA	USA
VCP	Visicorp

## Parameters for Copy II Plus

### ALIEN RAIN \* (BS)

```
0-5.....9=0, 31=0, D=D5,
F=0
6-D.....E=DE
```

### APPLE ADVENTURE \* (UNK)

```
0-22.....D=1, 10=96, 24=96
```

### APPLE LOGO \* (AC)

```
0-22
1.....A=1, 4B=1, 50=1
(ERROR 6 OK)
```

### APPLE PANIC \* (BS)

```
0-D
```



APPLE WORLD \* (USA)

0-23

APPLEWRITER II (AC)

0-22.....10=96

APPLEWRITER /// (AC)

0-22.....D=1, 10=96, 24=96

A2-PB1 (PINBALL) (SL)

0.....10=96

1-15.....A=3, E=DB, F=AB,  
10=BF, 44=1, 45=D,  
46=F

AZTEC \* (DM)

0-22.....D=1, 10=96, 24=96

BACK-IT-UP II \* (SEN)

0.....10=96, 9=0

1.5-B.5.....10=B5, A=3

BEER RUN (SRS)

0.....9=0

1.5-D.5.....D=1, 3B=40

CANNONBALL BLITZ \* (SOL)

0-22

3-F.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

CANNONBALL BLITZ (alternate)

0-22.....10=96

SECTMOD [S=E, T=17]

DOS 3.3 PATCHED

CHANGE ADDRESS CD FROM  
49 TO 60

CASTLE WOLFENSTEIN (MU)

0-22.....D=1

CEILING ZERO \* (TKS)

0-2

3-11.....9=0, E=D6, 1C=D6,  
34=1, 38=F9, 4F=1

CHESS 7.0 \* (OD)

0-22.....10=96, 9=0

CHOPLIFTER (BS)

0.....A=3, 44=1, 45=D,  
9=0, 0=F, 50=3

1-8.....4=FD, 31=0, 43=0,  
45=10, 4F=1, 46=12

9.....45=8, 46=D

A-B.....45=2



We HATE to think of you  
typing in all the programs in  
CORE and HARDCORE!

A disk with the programs  
contained in HARDCORE  
COMPUTIST #1 and #2 and in  
the first issue of CORE is  
available from Softkey  
Publishing for only

**\$19.95**

(postage and handling complimentary)

These disks are available:

Library Disk #1 ..... \$19.95

Spring 1983 CORE {Graphics issue}:

Scruncher      Design Plus  
Quick Draw     Faster Shapes  
QD.Editor       Space Raid

Hardcore Computist #1:

Checksoft      Checkbin

Hardcore Computist #2:

Page Flipper   Wall Draw

DISK CONTROL ..... \$18.00

DiskEdit        Menu  
DiskView       I.O.B.  
SpeedDOS

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

Sorry,  
no credit cards.  
no phone orders.  
no purchase orders.  
U. S. funds only.

Send check or money order to:  
HARDCORE PROGRAM LIBRARY  
P.O. Box 44549  
Tacoma, WA 98444

(advertisement)

IF YOU LIKE



YOU'LL LOVE

# SPARKEE!

## Son of FIRE ORGAN

FOR THOSE WHO MAY NOT BE AWARE, FIRE ORGAN WAS THE FIRST 'ALBUM' OF DYNAMIC (EVER-CHANGING) VISUAL COMPOSITIONS TO BE PRODUCED BY VAGABONDO ENTERPRISES. FIRE ORGAN WAS, AND STILL MAY BE, AVAILABLE FREE FROM MOST APPLE DEALERS.

NOW YOU CAN SEND FOR SPARKEE, WITH TWO DOZEN NEW DYNAMIC WORKS OF ART, GUARANTEED TO DELIGHT YOU WITH THEIR BRILLIANT COLORS AND GRACEFUL FLOWING PATTERNS OF CONTINUALLY MOVING SHAPES, LINES AND CURVES.

SPECIAL OFFER TO READERS OF HARDCORE: SEND JUST \$19.95 (PA RESIDENTS ADD \$1.20 SALES TAX) TO

KEN SHERWOOD  
117-B N. 25TH ST.  
READING, PA 19606

IN RETURN, SPARKEE WILL BE SENT TO YOU BY FIRST CLASS MAIL.

DETAILS: SPARKEE REQUIRES A 48K APPLE WITH DISK DRIVE. SPARKEE AND FIRE ORGAN ARE WRITTEN IN THE LANGUAGE OF CBEMAC. FIRE ORGAN, SPARKEE, CBEMAC AND THE FIRE ORGAN LOGO ARE ALL (TM) OF VAGABONDO ENTERPRISES.

SPARKEE IS NOT COPY-PROTECTED.

C-1E.5 BY.5....45=8, 10=D4, 51=1, D=1

20.....45=6, D=0, 4F=0

NOTE: Choplifter, Serpentine, David's Midnight Magic and Starblazer use track arcing and are very sensitive to drive speed. If you have problems, try reversing drives.

### COLOSSAL CAVE ADVENTURE \* (AC)

0-22

### CRANSTON MANOR (SOL)

0-22

18.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

### CROSSFIRE (SOL)

0-B.....9=0

1.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

### DAVID'S MIDNIGHT MAGIC (BS)

0.....A=3, 44=1, 45=D,  
9=0, 0=F, 50=3

1-A.....44=0

B.....44=1, 31=0, 43=0,  
45=8

C-19 BY .5.....10=F5, F=FD, 51=1,  
4F=1, D=1

See notes for Choplifter.

### DB MASTER (SW)

0-5.....10=96, 24=96, D=1  
6.5-22.5.....D=0

### DISK ORGANIZER \* (SEN)

0

1.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

2-4.....D=1

A-B

### DESKTOP PLAN II (VCP)

0-22.....10=96, 34=1, 36=2A

### ESCAPE (SL)

0-22

### EXECUTIVE SECRETARY \* (PBS)

0-22.....9=0, 8=1, 10=96

**EXPEDITOR (SOL)**

0-22.....10=96  
3 & 1F.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

**FORMAT II \* (KN)**

Use Copy Disk from Main Menu

**FS-1 (FLIGHT SIMULATOR) (SL)**

0.....10=96  
1.5-21 BY 1.5...E=DB, F=AB,  
10=BF, A=3, 4E=1  
7-8  
9.5

**GORGON (SRS)**

0.....10=96, 9=0  
1.5-E.5.....D=1, 24=96, A=3,  
E=DD, F=AD, 10=DA,  
3B=40

**HYPERSPACE WARS \* (CTS)**

0-22.....9=0

**JAW BREAKER \* (SOL)**

0-22.....9=0  
3.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

**KRELL LOGO \* (KL)**

0-22

**MAGIC WINDOW \* (ART)**

0-22

**MASTER TYPE (old) \* (LNS)**

0-2.....8=1  
3-22.....E=D4

**MICRO WAVE \* (CC)**

0-22  
11.....3B=1, A=1, 4B=1,  
4D=8, 50=1

**MOUSKATTACK \* (SOL)**

0-22.....10=96  
SECTMOD [S=3, T=18]  
DOS 3.3 PATCHED  
CHANGE ADDRESS B1  
FROM 49 TO 60

**MULTI PLAN (MIS)**

0-22.....10=96

**PFS & PFS REPORT (SPC)**

Use "Copy Disk" from Main Menu.  
After copying and before using,  
put a TAB over the write protect  
notch or you will turn into a  
lizard.

**PHANTOMS FIVE (SRS)**

0.....9=0  
2-10.....3A=0, 50=20

**PRISM \* (ISM)**

0-22

**PRISONER \* (EW)**

0-22

**RASTER BLASTER**

**(Old & New Versions) (BC)**

0.....10=96  
5-11 BY 4.....D=1, 9=0, 31=0,  
A=2, E=AD, F=DE,  
3B=40

6-12 BY 4

7.5-F.5 BY 4

1.5-3.5 BY 2

**SABOTAGE \* (SOL)**

0-22

3.....3B=1, A=1, 4B=1,  
4D=8, 50=1  
(ERROR 6 OK)

**SARGON \* (HN)**

0-1A

**SERPENTINE (BS)**

Same as Choplifter

**SNACK ATTACK (DM)**

0-12

SECTMOD [T=0, S=3]

DOS 3.2 PATCHED

CHANGE ADDRESS 63

FROM 38 TO 18

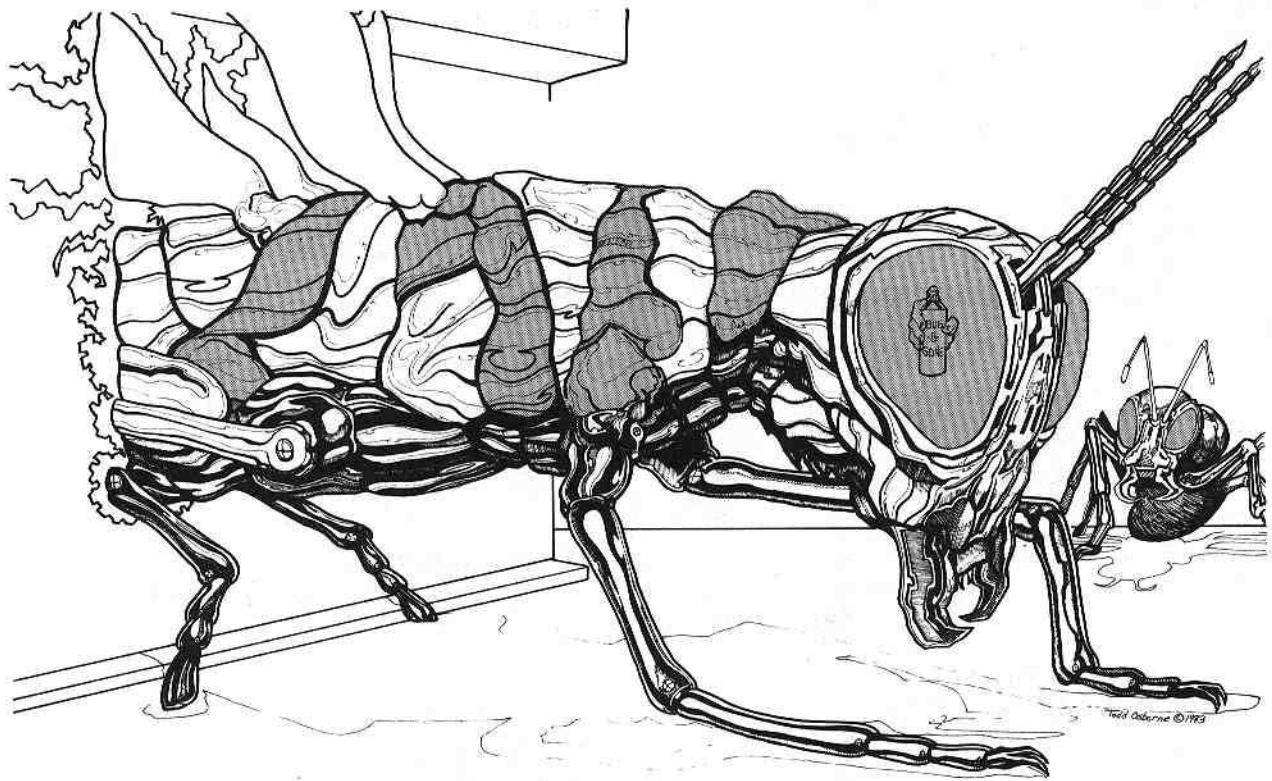
**SNEAKERS (SRS)**

0.....9=0, 10=96, 44=1,  
45=10, D=1  
1.5-C.5.....44=0  
D.5.....44=1

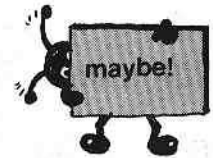
**SOFTPORN ADVENTURE (old) (SOL)**

0-22.....9=0

*continued on page 32*



# NO MORE BUGS!



One of the most inexpensive methods of accumulating software (if your monetary status is not in the "rich" category) is to subscribe to magazines that print program listings and laboriously type them in and save them. And the most frustrating thing that can occur after typing in a lon-n-n-g listing is to find that it doesn't work.

And the reason it doesn't work is that it has bugs.

Some of the bugs are improperly printed listings. These are our bugs. They most often occur when we try to format a listing to make it easier to read and understand. Other bugs occur when the listing is sliced up and pasted onto boards in order to be photographed and printed.

We've eliminated our bugs. Listings will no longer be formatted. They will be printed just as they are seen on the screen (usually). And we will have all line numbers in consecutive increments of ten so that if a line is missing it will be spotted by the editorial staff.

Most bugs occur at the user's keyboard. The program is not typed in exactly as it was listed.

Syntax errors in BASIC listings are easily found because you are notified of this fact by the computer.

But many errors are not found because they are errors that are not really syntax-related. Most of these mistakes fall into this category. These include:

1. Transposition (typing 536 instead of 563).
2. Missing characters (forgetting to put in parentheses).
3. Mistaken characters (typing a colon instead of a semicolon).
4. Missing arguments (typing IF A THEN . . . , instead of IF A > 5 THEN . . .).
5. Missing GOSUBs and GOTOs, and their line numbers.
6. Mistyped data statements.

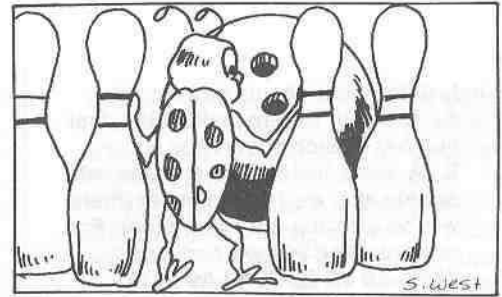
And a profusion of others . . .

These bugs are annoying, and require an inordinate amount of patience and determination to eliminate. Thus, we will now be using a solution that seems to have worked with other magazines: a program that will help spot bugs.

These programs are called "checksum" programs. *Nibble* uses one called *Key Perfect*. Our own are *Checksoft* (for Applesoft programs) and *Checkbin* (for machine programs).

# NO MORE BUGS!

# CHECKSOFT



**REQUIREMENTS:**  
 An Apple II or Apple II+  
 Applesoft  
 One disk drive

Checksoft is an ampersand utility for those who type in the programs listed in CORE and HARDWARE COMPUTIST. Checksoft will allow you to verify that the listing you have typed and the original one in the magazine are the same.

Any errors found can be easily traced to the individual lines and corrected. All that is necessary once Checksoft is in memory is to press the & (ampersand) key and compare the numbers displayed on the screen with those shown in the magazine.

Checksoft is designed to work only with Applesoft programs and will not work correctly on programs written in other languages such as Integer or binary.

## Unique Options in Checksoft

Checksoft provides options that are not found in most other checksum programs.

1. Checksoft is written in machine language.
2. Checksoft provides a checksum for every line in a program and will detect the smallest differences in a program and the listing.
3. The beginning of the program affects the checksum. This double checks that the program starts at the correct location.
4. Checksums may be sent to a printer.
5. Checksoft can be configured to ignore:
  - A. REM statements, allowing you to leave them out of a program and still have the same checksums as the original listing that contained REM s.
  - B. Line numbers, so programs can be renumbered without affecting the checksums.

Since Checksoft will only work for Applesoft programs, the program in the listing must be accompanied by a checksum list generated by Checksoft.

Since Checksoft will only work for Applesoft programs, the program in the listing must be accompanied by a checksum list generated by Checksoft.

by Robb Canfield

Checksoft will generate checksums that are different than those of other checksum programs (i.e., *Keyperfect* by Nibble), so a checksum list by other checksum programs will not match those by Checksoft.

Please read through the entire article before entering and using the program. You may be able to save yourself some extra work and frustration.

## Typing In Checksoft

Checksoft works by generating a two-byte hexadecimal (base 16 number) checksum. A checksum is a number based on the collection of certain data; that is to say, a number that will be constantly changing in a predictable way. Checksoft will update its checksum for every line encountered in an Applesoft program. Since the checksum is cumulative, all the previous checksums generated (one for each line) affect the next one. This gives each program its own unique checksum. There is a small chance that two programs will have the same final checksum (since there are only 65,536 different possibilities), but this chance is so small that it's not worth losing sleep over.

I will refer to two different checksums from this point on:

1. The line checksum, which appears for every line of the program.
2. The final checksum, which appears as the last line checksum displayed.

Checksoft must first be typed into memory as a series of bytes (the source code is also included for those interested). After typing the machine language part of Checksoft, enter the Applesoft part of the program.

The directions below explain how to move Checksoft from the published listing to the computer's memory in simple step-by-step procedures.

1. Re-boot your Apple  
**IN#6**

To clear memory and make sure you are in Applesoft, also known as floating point,

- FP**  
 (the Applesoft prompt is "I").

2. Enter the monitor with  
**CALL -151**

The machine language prompt should appear (as an asterisk "\*\*\*"). If it doesn't, try CALL -151 again. If the prompt still refuses to appear, then something is definitely wrong with your Apple (you are using an Apple, aren't you?).

3. Type the following bytes

```

0300: A9 4C 8D F5 03 A9 10 8D
0308: F6 03 A9 03 8D F7 03 60
0310: 20 8E FD A9 10 8D 96 03
0318: A9 FB 8D 97 03 A9 14 85
0320: 0A A5 67 8D C8 03 85 0B
0328: A5 68 8D C9 03 85 0C A2
0330: 00 A0 00 20 C7 03 E0 02
0338: 90 22 E0 04 B0 03 48 90
0340: 0E C9 00 F0 1D C0 FF F0
0348: 13 C9 B2 D0 02 A0 FF 6A
0350: 45 0B 2A 45 0C 85 0B 45
0358: 0C 6A 85 0C 20 BF 03 E8
0360: D0 D4 68 A8 68 AA 98 20
0368: 24 ED 38 A9 06 E5 24 AA
0370: 20 4A F9 A0 02 B9 C8 03
0378: 20 ED FD 88 10 F7 A5 0B
0380: A6 0C 20 41 F9 20 8E FD
0388: C6 0A D0 1F A9 14 85 0A
0390: 20 8E FD AD 00 C0 10 FB
0398: 8D 10 C0 C9 83 F0 1C C9
03A0: 9B D0 08 A9 EA 8D 96 03
03A8: 8D 97 03 A2 00 A0 00 20
03B0: BF 03 D0 82 E0 01 F0 03
03B8: E8 10 F4 20 8E FD 60 EE
03C0: C8 03 D0 03 EE C9 03 AD
03C8: FF FF 60 A4 A0 AD
  
```

4. Double check what you have typed. You should still be in the monitor. Type **300.378**

Compare this to lines 300 to 378. The

only differences should be:

A. There is a zero before the first number in each of the lines.

B. A dash instead of a colon will appear after each line number (there will be a space after the dash). For example, line 300 will appear as:

**0300- A9 4C 8D F5 03 A9 10 8D**

The bytes themselves should be the same. After fixing any errors you may have found, examine the rest of the program by typing

**380.3C8**

and compare this to lines 380 to 3C8 in the listing. Fix all errors before continuing.

5. Return to BASIC

**3D0G**

or if you have AutoStart simply press the RESET key. At this point you should see the Applesoft prompt, (>). If you see the integer prompt (>), type FP to enter Applesoft. You should now see the Applesoft prompt. Checksoft is not affected by language changes.

6. Save Checksoft to the disk

**BSAVE CHECKSOFT,A\$300,L\$D0**

7. Erase the current program

**NEW**

Then type the Applesoft program.

8. Save the Applesoft program

**SAVE START CHECKSOFT**

## How To Use Checksoft

The easiest way to use Checksoft is to run the Applesoft program (the binary part of Checksoft must be on the disk under the name CHECKSOFT). Press RETURN when asked for the version to load (this will automatically load the normal version CHECKSOFT) and answer the next two questions with "Y" and "N", respectively.

If you want to save yourself the extra effort of typing in the Applesoft program you may simply BRUN CHECKSOFT. When Checksoft is loaded in this manner, the Applesoft program in memory will remain intact (the variables will not even be disturbed). The loading of any Applesoft program will not disturb Checksoft (loading binary files — machine language programs — may erase Checksoft if they use the upper half of page \$3).

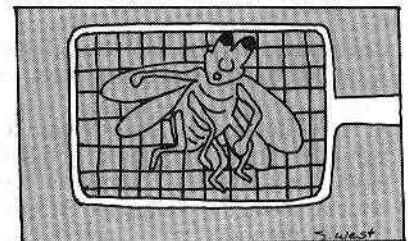
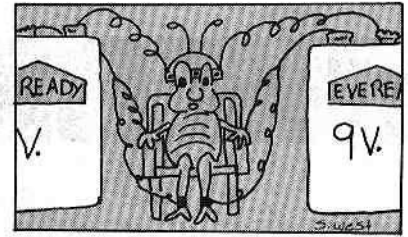
The only reason I have included the BASIC program is so that the various options in Checksoft may be changed easily. These options may be changed manually, as I will explain later. Check-

## Checksoft

```

1000 .....
1010 *
1020 *   CHECKSOFT   *
1030 *
1040 *   BY   *
1050 *
1060 *   ROBB S. CANFIELD *
1070 *
1080 *   COPYRIGHT 1980 *
1090 *   SOFTKEY PUBLISHING *
1100 *
1110 .....
1120 *
1130 * LOCATIONS USED
1140 *
1150
1160 LINPRNT .EQ $E124   PRINT DECIMAL VALUE OF X AND A REG
1170 CROUT  .EQ $FD8E   PRINT A CARRIAGE RETURN
1180 COUT   .EQ $FDED   PRINT A REG IN ASCII
1190 PRINTX .EQ $F941   PRINT A AND X REG AS A TWO HEXBYTE NUMBER
1200 PRINT.X.SPACES .EQ $F94A
1210 BEG    .EQ $67     BEGINNING OF APPLESOFT PROGRAM
1220 HTAB   .EQ $24     HORIZONTAL POSITION OF CURSOR
1230 CLEAR  .EQ $C010   CLEAR KEYBOARD STROBE
1240 READ   .EQ $C000   LOAD A REG WITH KEY PRESSED
1250 COUNTER .EQ $0A    COUNTER FOR NUMBER OF LINES
1260 CHKSUM .EQ $B      CHECKSUM BYTES $B & $C
1270 AMPER  .EQ $3F5    AMPERSAND JUMP VECTOR
1280
1290
1300 .OR $300   PLACE ON PAGE 3
1310 .TF CHECKSOFT
1320
1330 *
1340 * INITIALIZATION FOR AMPERSAND
1350 *
1360
1370 INIT     LDA #$4C   SET UP AMPERSAND JUMP VECTOR
1380         STA AMPER
1390         LDA #SETUP
1400         STA AMPER+1
1410         LDA /SETUP
1420         STA AMPER+2
1430         RTS
1440
1450 *
1460 * PROGRAM STARTS HERE
1470 *
1480
1490 SETUP    JSR CROUT
1500         LDA #$10    OPCODE FOR BNE (SET KEY1 TO NORMAL)
1510         STA KEY1+3

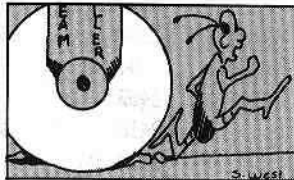
```



```

1520 LDA #SFB FOLLOWS BNE STATEMENT
1530 STA KEY1+4
1540 LDA #20 # OF LINES UNTIL STOPPED
1550 STA COUNTER
1560 LDA BEG GET BEGINNING OF APPLESOFT PROGRAM
1570 STA GET+1
1580 STA CHKSUM USE THIS BYTE FOR CHECKSUM
1590 LDA BEG+1
1600 STA GET+2
1610 STA CHKSUM+1
1620 LDX #000 SET COUNTER TO ZERO
1630 LDY #S00 CLEAR REM FLAG
1640 LOOP JSR GET GET A BYTE
1650 LOOP2 CPX #2 ON OFFSET?
1660 BCC SUMIT3 YES, SO SKIP END OF LINE CHECK
1670 CPX #4 ON LINE #?
1680 BCS SUMIT NO, SO DON'T SAVE LINE #
1690 PHA SAVE LINE #
1700 BCC SUMIT2 SKIP END OF LINE CHECK. DEFAULT IS TO USE LINE #
1710 SUMIT CMP #0 END OF LINE?
1720 BEQ PRINTCHK YES SO PRINT CHECKSUM AND CHECK FOR END OF PROGRAM
1730 CPY #SFF IGNORE REM?
1740 BEQ SUMIT3 YES
1750 CMP #S2 TOKEN FOR REM
1760 BNE SUMIT2
1770 LDY #SFF SET REM FLAG. DEFAULT IS TO IGNORE THEM
1780 SUMIT2 ROR
1790 EOR CHKSUM ;GENERATE A CHECKSUM
1800 ROL
1810 EOR CHKSUM+1
1820 STA CHKSUM
1830 EOR CHKSUM+1
1840 ROR
1850 STA CHKSUM+1
1860 SUMIT3 JSR INCGET INCREMENT COUNTERS
1870 INX
1880 BNE LOOP2 CONTINUE BY GETTING ANOTHER BYTE
1890 PRINTCHK PLA GET HIGH BYTE OF LINE #
1900 TAY SAVE IT TEMPORARILY
1910 PLA GET LOW BYTE OF LINE #
1920 TAX SET UP FOR LINE PRINT
1930 TYA GET HIGH BYTE BACK
1940 JSR LINPRNT PRINT DECIMAL LINE #
1950 SEC CALCULATE THE NUMBER OF SPACES TO PRINT
1960 LDA #S6
1970 SBC HTAB
1980 TAX
1990 JSR PRINT.X.SPACES
2000 LDY #2 NUMBER OF CHARACTERS TO PRINT MINUS 1 (-1)
2010 PRINT1 LDA MSG1,Y
2020 JSR COUT PRINT AREG IN ASCII
2030 DEY DONE WITH PRINT?
2040 BPL PRINT1 NO, SO GET NEXT CHARACTER
2050 LDA CHKSUM PRINT CHECKSUM VALUE
2060 LDX CHKSUM+1

```



soft will disable any ampersand utility you have in memory (such as PRINT USING and RENUMBER), but does not affect PLE (Program Line Editor).

After Checksoft is in memory, enter a program from a listing that has a checksum table created by Checksoft. (I have included the checksum table for the Applesoft program START CHECKSOFT, and will use examples from this program in the rest of the article.) After entering the program, type "&". If there is no Applesoft program in memory, Checksoft may display random line numbers and checksums (if this happens, use the ctrl C command to exit Checksoft).

The first twenty lines of your program will be displayed in the following fashion:

```

line# - $0000
(hexadecimal checksum)

```

For example:

```
10 - $E7C9
```

The checksum number will always be four digits long.

Compare the checksums given for each line of the program to those provided in the magazine. If the checksums match, the line is okay. If the two checksums differ, there is an error in the program.

The first line that has a checksum different than the one in the magazine is the offending line (remember that every checksum is based on the previous one, so if one line is incorrect all the following lines also appear to be incorrect). Fix this line and check the program again to see if any other lines were mistyped.

If the first twenty lines are okay, press the SPACE bar to display the next twenty lines. Proceed in this manner until you are returned to Applesoft.

Double check the program to make sure all the checksums match. When they all match, the program is exactly as given in the magazine.

This process can be reduced to the following:

1. Type the program into memory.
2. Press the ampersand key (&). The first twenty lines of the program will be displayed.
3. Compare the checksums on the screen to those in the listing. The first line encountered that doesn't match the checksum table printed in the magazine is in error and must be fixed before continuing.
4. Continue in this manner until all the checksums displayed on the screen match those in the table.

## Commands

Checksoft has three commands available:

- ctrl C** Exit Checksoft immediately. This command can be used to exit Checksoft without having to page through the remaining checksums.
- SPACE** Display the next twenty lines of the program with checksums.
- ESC** Does not pause after displaying twenty lines. This command can be used to rapidly check a program (see Trick 2 below).

When the Applesoft prompt (>) appears, Checksoft has finished checking the program and has automatically returned you to Applesoft. There is no need to use ctrl C to exit.

## Tricks

The following tricks may be used to help find where the checksums differ:

1. The first time through, press ESC and look at the last checksum printed

(the final checksum). If this one matches the final one in the magazine, you have successfully typed in the program. If this checksum doesn't match, the program is different from the one in the magazine. Try Trick 2.

2. When paging through the checksum tables, only compare the last checksum printed for each set of twenty. When you find that they differ, back-track until you find the offending line (the first line with a different checksum than listed).

## Modifying Checksoft

The following is a discussion of how to modify Checksoft for different types of listings. Checksoft can be configured in a number of different ways. You have the option to use or ignore the following:

1. REMark statements.
2. Line numbers.
3. Next line pointers.

For convenience I am going to refer to each of the three possible changes as three separate parameters. Checksoft

is normally set up to ignore REMark statements, use line numbers, and ignore line pointers.

If any of the above parameters are changed, Checksoft will generate different checksums. Each parameter has its own special function. The REMark parameter can be configured so that everything after a REM statement is ignored. This allows you to simply type the command REM and leave out all the garbage following it without affecting the checksum. Or you could even replace the original REM statement with your own, again without affecting the checksum.

Only existing REMs may be changed; no new REMarks may be created. This is the most useful and powerful parameter.

For example, assume that the following line appears in a program:

```
10 D$ = CHR$(4): REM CTRL "D"
```

If you wanted to, you could change the line to the following:

```
10 D$ = CHR$(4): REM
```

or:

```
10 D$ = CHR$(4): REM ANYTHING CAN GO HERE
```

without affecting the checksums.

2070 JSR PRINTAX	2360 NOSTOP2 JSR INCGET	GET NEXT BYTE
2080 JSR CROUT	2370 BNE LOOP2	IF NO ZERO WAS ENCOUNTERED CONT.
2090 DEC COUNTER DONE WITH 20 LINES	2380 CPX #\$1	IS THIS THE 3RD ZERO?
2100 BNE NOSTOP NO, SO CONTINUE	2390 BEQ END	IF SO END
2110 LDA #20 YES, SO GET KEYPRESS	2400 INX	CHECK FOR LAST ZERO (3RD)
2120 STA COUNTER AND RESET COUNTER	2410 BPL NOSTOP2	ALWAYS DO THIS
2130 JSR CROUT	2420 END JSR CROUT	DONE WITH PROGRAM
2140 KEY1 LDA READ GET A KEYPRESS	2430 RTS	RETURN TO APPLESOFT
2150 BPL KEY1 LOOP UNTIL A KEY IS PRESSED	2440	
2160 STA CLEAR CLEAR KEYBOARD STROBE	2450 *	
2170 CMP #\$83 WAS IT CTRL C?	2460 * SUBROUTINES CALLED FROM MAIN PROGRAM	
2180 BEQ END YES, SO STOP PROGRAM	2470 *	
2190 CMP #\$9B WAS IT ESC?	2480	
2200 BNE NOSTOP NO, SO CONTINUE NORMALLY	2490 INCGET INC GET+1	ADD ONE TO THE VALUE IN GET
2210	2500 BNE GET	IS THERE A CARRY OVER
2220 *	2510 INC GET+2	YES INCREMENT HIGH BYTE
2230 * CHANGES PROGRAM TO IGNORE KEYPRESSES	2520 GET LDA \$FFFF	GETS A VALUE FROM APPLESOFT
2240 *	2530 RTS	
2250	2540	
2260 STOP LDA #\$EA SET BRANCH IN KEY1 TO NOPS	2550 *	
2270 STA KEY1+3	2560 * PRINT MESSAGES	
2280 STA KEY1+4	2570 * THE MESSAGE IS STORED BACKWARDS	
2290	2580 *	
2300 *	2590	
2310 * CONTINUE WITH PROGRAM	2600 MSG1 .AS -"\$ -"	
2320 *	2610 *	
2330	2620 * LOCATION FOR TEMPORARY USES	
2340 NOSTOP LDX #\$00 CHECK FOR END	2630 *	
2350 LDY #\$00 CLEAR FLAG		





The following would be illegal:

**10 D\$ = CHR\$(4):**

since the REM statement is missing. If anything else was changed, such as D\$ to E\$, the checksum would also be changed and therefore would be illegal.

If the REMark parameter is set up to use REM s, the REMark must be typed exactly as shown in the listing.

The line number parameter can also be configured to ignore or use the line numbers. This parameter was originally set up to use line numbers. This means that each line number must be typed exactly as shown.

If this parameter is changed to ignore line numbers, line numbers may be changed when typed. This only applies to the line number, not to line references such as GOTO s or GOSUB s. These line references must still be typed exactly as printed in the magazine.

For example, again assume that the following line exists in a program:

**10 D\$ = CHR\$(4): GOTO 900**

The following change is legal if the line numbers are ignored:

**15 D\$ = CHR\$(4): GOTO 900**

The line number has been changed from 10 to 15. This change would not have been allowed if it had been specified to use line numbers.

The following change is illegal and would affect the checksum in all situations:

**10 D\$ = CHR\$(4): GOTO 980**

The line reference has been changed from 990 to 980.

The line pointer is a special parameter that will be used very rarely. When it is used, the program must have the same number of bytes in each line. That is, each line must take up the same amount of memory as it did originally. Generally, if the line pointer is used the author wants the program typed in exactly as shown with absolutely no changes.

Below are some common configurations that may be used and when to use them. To conserve space, I have abbreviated each of the three parameters to:

**REM**    REMark statements  
**Line**    Line numbers  
**Point**    Next line pointers

### Configuration 1

This is what Checksoft is normally set to. REMark statements can be changed or deleted, but everything else must remain the same.

**REM**    - Ignore (off)  
**Line**    - Use (on)  
**Point**    - Ignore (off)

### Configuration 2

This is especially useful in self-modifying or length-dependent code. The program must be typed in exactly as shown. No changes can be made, no matter how slight they are.

**REM**    - Use (on)  
**Line**    - Use (on)  
**Point**    - Use (on)

### Configuration 3

This can be used if there are no line references within the program. The only real advantage it has is to allow the line numbers to be changed (such as renumbering the program) without affecting the checksums.

**REM**    - Ignore (off)  
**Line**    - Ignore (off)  
**Point**    - Ignore (off)

There are two ways to modify Checksoft: one is to use the BASIC program, and the other is to POKE the changes into memory. The BASIC program will tell if Checksoft is configured normally or not by changing the top line of text. For normal configuration, the top line will read "NORMAL CONFIGURATION IS". If Checksoft has been modified, this line will change to "NON-NORMAL CONFIGURATION IS".

To change Checksoft, simply answer "N" when asked if this is the proper configuration. You will then be asked if you want to use or ignore the three possible parameters. Answer with "Y" for Yes, "N" for No, or press RETURN if you wish it to remain the same.

The modified Checksoft may then be saved to the disk under any name. Pressing RETURN will cause the

*\* due to corrections updated checksums are in No.3, page 4*

## Checksums for Start Checksoft

Configuration 1

1	- \$97CE	130	- \$7AD8	260	- \$7EF8	390	- \$1DD0	530	- \$254A
2	- \$F48B	140	- \$9536	270	- \$68B1	400	- \$A7EE	540	- \$55ED
3	- \$88EC	150	- \$4F85	280	- \$D09E	410	- \$6FAC	550	- \$7402
10	- \$E7C9	160	- \$817F	290	- \$ECFB	420	- \$ABA5	560	- \$E8E8
20	- \$DCE6	170	- \$C8EA	300	- \$8AEC	430	- \$A7AA	570	- \$A88B
30	- \$781C			310	- \$4B42	440	- \$460D		
40	- \$A12C	180	- \$1D6C	320	- \$1A2E	450	- \$1E58	580	- \$79E8
50	- \$C14E	190	- \$58D9	330	- \$816E	460	- \$6861	590	- \$0B27
60	- \$30A3	200	- \$8092	340	- \$1B1A	470	- \$8C18	600	- \$00B7
70	- \$D960	210	- \$9EAB	350	- \$877F	480	- \$100A	610	- \$2668
80	- \$D228	220	- \$F52A	360	- \$7B8C	490	- \$F841	620	- \$D0CC
90	- \$CC76	230	- \$0B17	370	- \$36FA	500	- \$C946	630	- \$6E2C
100	- \$E735	240	- \$736E			510	- \$346D	640	- \$0307
110	- \$FFB5	250	- \$1C37	380	- \$CE92	520	- \$B6B7	650	- \$144C
120	- \$41CB								

BEG: \*300.3CD END:

## Hex Dump and Checksums for Checksoft

0300-	A9 4C 8D F5 03 A9 10 8D	\$2C5A	0368-	24 ED 38 A9 06 E5 24 AA	\$FAF5
0308-	F6 03 A9 03 8D F7 03 60	\$8B3A	0370-	20 4A F9 A0 02 B9 CB 03	\$F441
0310-	20 8E FD A9 10 8D 96 03	\$A383	0378-	20 ED FD 88 10 F7 A5 0B	\$27ED
0318-	A9 FB 8D 97 03 A9 14 85	\$EEB6	0380-	A6 0C 20 41 F9 20 8E FD	\$9356
0320-	0A A5 67 8D C8 03 85 0B	\$8884	0388-	C6 0A D0 1F A9 14 85 0A	\$2DBF
0328-	A5 68 8D C9 03 85 0C A2	\$171C	0390-	20 8E FD AD 00 C0 10 FB	\$3B3A
0330-	00 A0 00 20 C7 03 E0 02	\$EFA7	0398-	8D 10 C0 C9 83 F0 1C C9	\$32D2
0338-	90 22 E0 04 B0 03 48 90	\$82D8			
0340-	0E C9 00 F0 1D C0 FF F0	\$2B27	03A0-	9B D0 08 A9 EA 8D 96 03	\$16EC
0348-	13 C9 B2 D0 02 A0 FF 6A	\$F4C8	03A8-	8D 97 03 A2 00 A0 00 20	\$5056
			03B0-	BF 03 D0 82 E0 01 F0 03	\$0AE2
0350-	45 0B 2A 45 0C 85 0B 45	\$4E77	03B8-	E8 10 F4 20 8E FD 60 EE	\$A485
0358-	0C 6A 85 0C 20 BF 03 E8	\$8698	03C0-	C8 03 D0 03 EE C9 03 AD	\$1A30
0360-	D0 D4 68 A8 68 AA 98 20	\$0224	03C8-	FF FF 60 A4 A0 AD	\$6118



**ORDER TODAY FROM:**  
**GARY BROWN**  
**P.O. BOX 727**  
**PHILOMATH, OR 97370**  
**(503) 929-3666**

**WE ACCEPT CHECKS, MONEY**  
**ORDERS, VISA, MASTERCARD (NO**  
**COD) PLEASE ADD \$3.00 POSTAGE**  
**& HANDLING**  
**OPEN 7 DAYS 9AM - 9PM**



# WE WILL NOT BE UNDERSOLD

## WRITE AWAY

An advanced, professional word processing system with unsurpassed speed and flexibility. Simple to learn commands. Interfaces easily with most printers.

~~\$175.00~~



**LOOK!!!**

**\$99.95**

DISKETTE  
MANUAL

**ORDER YOURS TODAY!!**

Limited Time Only

## ACTION SOUNDS AND HI-RES SCROLLING

Machine language sounds, hi-res scrolling routines and Superfont program available in one useful package. ~~\$18.95.~~

**NOTICE!**

**DISKS  
ARE  
NOT  
COPY  
PROTECTED!**

## HI-RES SECRETS GRAPHICS APPLICATIONS SYSTEMS

Go step by step from ordinary Basic to better Basic programs. Basic to Assembly hi-res graphics. Create business graphs, electronic and architectural designs, arcade and adventure games and more! Fastest color-fill routines available, more colors and patterns than anyone. 240 pages. ~~\$75.00.~~

**56.95**

3  
DISKETTES

## SUPER DRAW AND WRITE

Includes Superfont program featuring 9 sizes and 8 styles of type, scrolling, saving, retrieving and manipulation of characters. Draw practically anything with the Instant Graphics program. ~~\$17.95.~~

**DON'T DELAY—  
ORDER TODAY!**

## CREATIVITY TOOL BOX

Create instant people, instant graphics at the touch of the keyboard. Write poetry, music. 72 type size/style combinations, action sounds and hi-res scrolling are also included. ~~\$44.95.~~

**\$34.95**

## PAINT MASTER SCENE UTILITY

Allows line drawing, scene painting, scene editing, screen compression. Fastest color-fill routines anywhere. Perfect for adventure game creation or computer art. Over 300 colors/patterns. Use creations in your own program. ~~\$34.95.~~

**\$24.95**

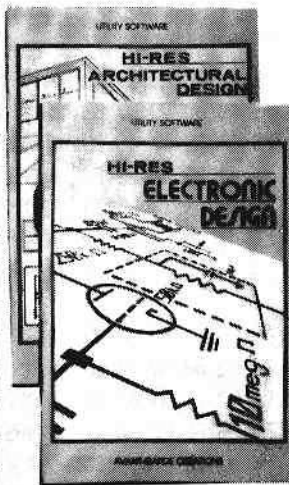
## HI-RES ARCHITECTURAL DESIGN

Plan and design individual room, complete floor plans and total buildings. 75 different floor plan shapes can be easily rotated and moved. Lengths, diagonals and angles are calculated on-screen. ~~\$29.95.~~

**\$24.95 each**

## HI-RES ELECTRONIC DESIGN

Create and print electronic circuit plans. Keyboard selection and rotation of 98 different electronic components. Paddles or joysticks provide easy placement. ~~\$29.95.~~

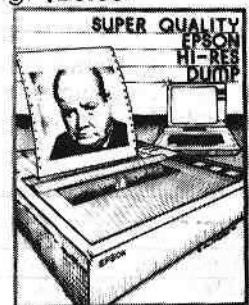


## SUPER QUALITY EPSON HI-RES DUMP

Best quality Epson dump features true black copy. Mirror, negative, flip or rotated images available in three sizes. Automatic horizontal centering. ~~\$25.00.~~



**\$19.95**





**ORDER TODAY FROM:**  
**GARY BROWN**  
**P.O. BOX 727**  
**PHILOMATH, OR 97370**  
**(503) 929-3666**

**WE ACCEPT CHECKS, MONEY**  
**ORDERS, VISA, MASTERCARD (NO**  
**COD) PLEASE ADD \$3.00 POSTAGE**  
**& HANDLING**  
**OPEN 7 DAYS 9AM - 9PM**



# WE WILL NOT BE UNDERSOLD

## THE COMPLETE MAILING LABEL AND FILING SYSTEM

Print labels 1, 2, 3 or 4 across, sort on any field, do range, character and two-level sorts, formatted reports in alphabetical or zip code order, duplicate or customized labels and much more.

~~\$75.00-~~

**\$39<sup>99</sup>**  
 Your Choice  
**EA!**



## ULTRA PLOT

Menu driven, flexible business graphics. Create pie, scatter, bar or stacked bar charts and line or hi-low graphs plus unique U.S. map charts. Many options and user-friendly features.

~~\$70.00-~~

Demo disk available. \$10.00.

## A.E.N. GRADING SYSTEM

Alphabetizes, calculates and grades records of one student, many students, or categories of students. Uses number grades or any of three types of letter grades. Package includes demo, manual and tutorial sheet. ~~\$70.00-~~

## THE SMALL BUSINESS TRILOGY

The Complete Mailing Label and Filing System combines with Ultra Plot, plus an interface that can transform your mailing system data base into useful business graphics. ~~\$475.00-~~

Introductory Price

**99<sup>95</sup>**

**NOTICE!**

**DISKS ARE NOT COPY PROTECTED!**

**DON'T DELAY ORDER TODAY!**

## SOUND EXPANDER

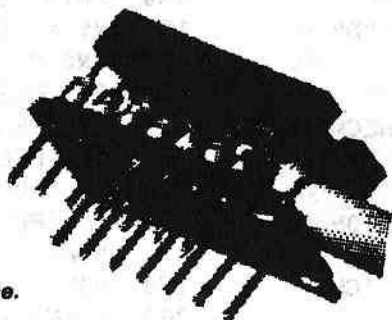
## SOUND EXPANDER

- \* INTERFACES YOUR APPLE TO YOUR HI-FI OR OTHER AMPLIFIER!
- \* SIMPLE PLUG IN INSTALLATION!
- \* ADDITIONAL MEMORY NOT REQUIRED!
- \* WORKS WITH ANY SOFTWARE THAT MAKES SOUND!
- \* GAME PORT NOT USED!
- \* NOW HEAR ALL YOUR APPLE SOUNDS THROUGH YOUR HI-FI!
- \* MUSIC, VOICE, AND GAME PROGRAMS COME ALIVE!

\*Apple is a registered trademark of Apple Computer Inc.

**NEW!**

Prices subject to change without notice.



~~\$39.95~~



**DEALER INQUIRIES WELCOME!**

default name to be used. The default name is CHECKSOFT unless you decided to enter another name when you were asked for the version to load, causing the new name to become the default.

There is one POKE for each of the possible parameters.

**REMARKS:**

- 1. Ignore POKE 846,255 (Standard)
- 2. Use POKE 846,00

**Line numbers:**

- 1. Ignore POKE 832,27
- 2. Use POKE 832,14 (Standard)

**Next line pointers:**

- 1. Ignore POKE 825,34 (Standard)
- 2. Use POKE 825,21

The configuration type must accompany the checksum table, and Checksoft must be configured in that same manner before checking the program, otherwise the checksums won't match. For example, if I used Configuration 2, then your version of Checksoft must be changed to Configuration 2 or the checksums would not match.

## Using Checksoft with a Printer

If you want a printout of the checksums, simply turn on the printer before calling Checksoft (&). To turn on your printer type PR#n, where n is the slot that the printer interface card is in. Since the interface card is usually located in slot one, PR#1 will turn on most printers. PR#0 will turn off the printer. Make sure that the printout

goes to the screen as well as the printer. Your printer manual should have information on how to do this.

## How Does Applesoft Store A Program?

Before discussing how Checksoft works, a review of how Applesoft stores a program is in order. An Applesoft line consists of four parts:

1. The next-line pointer, a two-byte number that points to the place in memory where the next line is located.
2. The line number, also two bytes. Line numbers are converted to hexadecimal before being stored.
3. The tokenized line itself.
4. And 00 to mark the end of the line.

When you type a line in Applesoft, it is tokenized. This is a process whereby a large amount of information may be encoded to fit into a small amount of memory. See page 121 of the *Applesoft Manual* for a list of tokens.

In Applesoft there is a one-byte code for each of the commands (PRINT, REM, POKE, etc.). The one-byte code takes the place of the letters for the command. This means that only one byte, instead of five, is being used each time the PRINT command is required.

That shrinks the amount of space required to store a line tremendously.

There is also a code for each of the letters, numbers, and symbols. This doesn't save space, but it insures that no two commands and/or characters have the same code.

A zero (00) is added to the line being tokenized to mark its ending. Two extra zeros would be added if this was the last line of the program.

After the line is tokenized, Applesoft searches for a line number. If one was not given, it immediately processes the line (immediate mode). Otherwise the line number is converted into its hexadecimal equivalent (deferred mode). After this conversion is done, a check is made to see where in memory the line will go. The next line pointer is then set to point to the end of the line plus one (right after the 00). The line is then stored in memory in the order mentioned above.

The following is an example of how an Applesoft line is stored:

**20 D\$ = CHR\$(4): REM CTRL "D"**

First the line is tokenized. D\$= is tokenized character-by-character into the values \$44 \$24 \$D0. Since CHR\$ is considered a command, it is reduced to one byte, \$E7. The (4): is stored as four separate characters, \$28 \$34 \$29 \$3A. The REM is another command, so it is stored as \$B2.

### Illustration 1

SAMPLE LINE WITH CORRESPONDING HEX TOKENS

**20 D\$ = CHR\$(4) : REM**  
 14 00 44 24 D0 E7 28 34 29 3A B2 20

## Start Checksoft

```

1 REM * START CHECKSOFT
2 REM * WRITTEN BY
3 REM * ROBB CANFIELD
10 D$ = CHR$(4): REM CTRL "D"
20 TEXT : HOME
30 INVERSE
40 PRINT : HTAB 15: PRINT "CHECK
  SOFT"
50 NORMAL
60 VTAB 11: PRINT "ENTER VERSION
  OF CHECKSOFT TO USE."
70 PRINT "DEFAULT VERSION IS 'CH
  ECKSOFT'"
80 PRINT
90 HTAB 7: INPUT " > ":NA$
100 IF LEN(NA$) < 1 THEN NA$ =
  "CHECKSOFT"

```

```

110 INVERSE
120 VTAB PEEK(37): HTAB 10: PRINT
  NA$
130 NORMAL
140 PRINT D$"BLOAD"NA$
150 LN$ = "YES":OF$ = "NO ":RM$ =
  "NO "
160 IF PEEK(825) = 21 THEN OF$
  = "YES"
170 IF PEEK(832) = 27 THEN LN$
  = "NO "
* 180 IF PEEK(846) = 0 THEN RM$ =
  "YES"
190 HOME
200 PRINT : PRINT SPC(10)"NORM
  AL CONFIGURATION IS"
210 VTAB 6
220 PRINT SPC(10)"USE REMARKS"
230 PRINT

```

\*(Hardcore Computist # 3, page 4) -Corrections

The REM is an exception to the rule that each command has a one-byte code. After the REM is stored (\$B2), a space is added to the line (\$20). The DATA command works the same way. The rest of the line is stored as characters — \$43 \$54 \$52 \$4C \$20 \$22 \$44 \$22 —since Applesoft ignores any commands in a REMark statement. A zero is added to mark the end of the line.

This leaves the line consisting of the following bytes:

```
$44 $24 $D0 $E7 $28 $34 $29 $3A
$B2 $20 $43 $54 $52 $4C $20 $22
$44 $22 $00
```

The line was reduced from 23 bytes to 20. This adds up to a lot of space saved in a long program, especially if REM s are left out.

Next, the line number is converted to hexadecimal. The decimal number 20 becomes \$14 in hexadecimal, but since all line numbers are two bytes, a zero is used in the second byte (it is standard in machine language to store all numbers in the low/high format; low byte first, high byte second).

Now the only thing left is the next line pointer. To show this, I will assume that there was a previous line as follows:

```
10 HOME
```

### Illustration 2

```
801:07 08 0A 00 97 00 1E
808:08 14 00 44 24 00 E7 28
810:34 29 3A B2 20 43 54 52
818:4C 20 22 44 22 00 00 00
```

It would be tokenized into the following bytes: \$0A \$00 \$97 00 (the line number and the HOME command). The program will be placed at \$801 in memory (this is where Applesoft normally starts a program).

Illustration 2 is a hex dump (a listing of hexadecimal bytes) obtained by typing the BASIC program given above and then entering the monitor to list the desired bytes. To duplicate my efforts, do the following:

1. Type FP to clear any Applesoft programs in memory and to set the beginning of the program to \$801.
2. Type the basic program used above:  
**10 HOME**  
**20 DS = CHR\$(4):REM CTRL "D"**
3. Enter the monitor with  
**CALL -151**
4. List the bytes  
**801.81F**
5. Re-enter basic with 3D0G or press the RESET key if you have AutoStart ROM.

In Illustration 2, the first two bytes tell where the next line begins (remember that these bytes are in reverse order: 07 08). In this case, the next line starts at \$807. The line number is at \$803 and \$804 with the tokenized line following. The zero marks the end of the line.

The next byte is the beginning of line 20. Line 20 starts at \$807 and begins with the next line pointer pointing to \$814. What follows is the line number, tokenized line, and the zero. Two more zeros are found after this line. This lets Applesoft know where the program ends.

Applesoft also uses four locations in

memory to indicate where the program starts and where it ends. Locations \$67 and \$68 (decimal 103 and 104) point to the beginning of the program (again the bytes are in a reverse order). Locations \$AF and \$B0 (decimal 175 and 176) point to the end of the program (these are also in the reverse order).

## How Does Checksoft Work?

The first thing Checksoft does is find the start of the program by checking locations \$67 and \$68. After it finds the beginning, Checksoft puts every byte of the entire program (until it encounters three zeros in a row) through the checksum routine.

Checksoft keeps track of its location on every line by using a counter (this counter is reset to zero whenever a new line is encountered). If this counter is at two or three, the current bytes are temporarily saved (these two bytes are the line number). When the end of that line is found, the two bytes saved are retrieved and converted to decimal, then printed.

After the line is printed, Checksoft does an HTAB to six and prints the message "CHECKSUM \$" along with the current checksum value. Another counter keeps track of how many lines have been displayed, stopping every 20. The program also checks for the various commands at this time. If the ESC key has been pressed, Checksoft modifies itself so that it will no longer pause and wait for a command before displaying the next 20 lines.

```
240 PRINT SPC( 10)"USE LINE NUM
BERS"
250 PRINT
260 PRINT SPC( 10)"USE OFFSETS"
270 PRINT
280 INVERSE : VTAB 6: HTAB 30: PRINT
RM$
290 VTAB 8: HTAB 30: PRINT LN$
300 VTAB 10: HTAB 30: PRINT OF$
310 NORMAL
320 IF RM$ < > "NO " OR LN$ < >
"YES" OR OF$ < > "NO " THEN
VTAB 2: HTAB 7: PRINT "NON-
": GOTO 340
330 VTAB 2: HTAB 7: PRINT SPC(
4): PRINT
340 VTAB 20: PRINT "IS THIS CONF
IGURATION CORRECT? ";: GET A
$: IF A$ = "Y" THEN 530
350 VTAB 6: HTAB 30: PRINT "Y/N"
:: GET A$
360 IF A$ = CHR$ (13) THEN 380
370 RM$ = "NO ": IF A$ = "Y" THEN
RM$ = "YES"
380 INVERSE
390 HTAB 30: PRINT RM$
400 NORMAL
410 VTAB 8: HTAB 30: PRINT "Y/N"
:: GET A$
420 IF A$ = CHR$ (13) THEN 440
430 LN$ = "NO ": IF A$ = "Y" THEN
LN$ = "YES"
440 INVERSE
450 HTAB 30: PRINT LN$
460 NORMAL
470 VTAB 10: HTAB 30: PRINT "Y/N"
:: GET A$
480 IF A$ = CHR$ (13) THEN 500
```

## How To Generate The Checksums

The Checksum Algorithm is the routine in Checksoft which generates the checksums. Every time a byte is sent to this routine, the checksum will change.

Unless you know machine language, you will not understand the technique I use to generate my checksums. I will continue to use this technique in any later editions I write (unless a bug is found, in which case the routine will be corrected and a note to that effect will be given). In this way I will always generate the same checksums, regardless of how the rest of the program is written.

The method I use will catch missing letters and transposed letters (such as TETX instead of TEXT) along with any other errors that are made. For those interested in machine language, I rotate the bytes numerous times to

catch transposition errors, and exclusively OR it with the last checksums in a manner that will catch even the slightest error (even an error in one bit of a 27-sector program).

## The BASIC Program

I included the BASIC program to allow easy changes to Checksoft. The only thing it does is to make all the POKEs for you. It is a straightforward program that requires no explanations.

Nonetheless, I would like to point out a very useful technique. In lines 120 and 610, I do a VTAB PEEK (37). This VTABs to the text line directly above the current line. For example, if something was printed on line three, normally line four would be printed on next. But this doesn't occur because location 37 (\$25 HEX) contains a value of three (in fact this location contains the current line

minus one, so do NOT VTAB PEEK (37) from line one or an error will occur).

This technique worked very well when I wanted to print right over where the input was, without having to keep track of where I actually was on the text screen. This can add a nice touch to your own BASIC programs.

## Get Some Sleep

Gone forever are those long, sleepless nights spent poring over listings, trying to find that one "little" error. Checksoft will track down that error, no matter what line it's in.

Pleasant dreams.



### ADVANCED PLAYING TECHNIQUES

**Choplifter**, Broderbund  
Software, 1938 Fourth Street,  
San Rafael, California 94901,  
(415) 456-6424.

## Choplifter

Many people don't know that ctrl L will regulate the level of play in Choplifter. Ctrl L followed by these numbers will get the specified results:

- 0 One tank attacking you
- 1 Two tanks and jets
- 2 Two tanks and jets, and two drones
- 3 Two tanks and jets, and two drones that fire

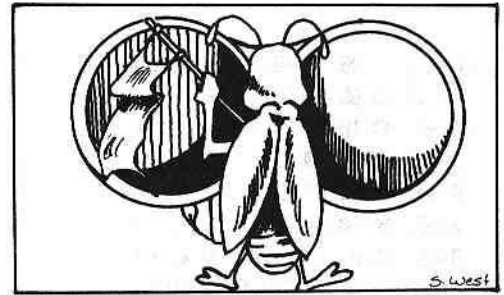
You will also find that making the prisoners walk back to Base, instead of flying them back, will cause those prisoners who go into the Post Office to completely disappear from the game (the scorekeeper will not update the information).

```

490 OF$ = "NO ": IF A$ = "Y" THEN
      OF$ = "YES"
500 INVERSE
510 HTAB 30: PRINT OF$
520 GOTO 310
530 OF = 34: IF LEFT$(OF$,1) =
      "Y" THEN OF = 21
540 LN = 14: IF LEFT$(LN$,1) =
      "N" THEN LN = 27
550 RM = 255: IF LEFT$(RM$,1) =
      "Y" THEN RM = 00
560 POKE 825,OF: POKE 832,LN: POKE
      846,RM
570 PRINT : VTAB 20: PRINT "DO Y
      OU WISH TO SAVE THIS VERSION
      TO THE DISK? ";: GET A$
580 IF A$ = "Y" THEN HOME : PRINT
      : VTAB 12: INPUT "SAVE THIS
      VERSION AS ";A$
590 IF LEN(A$) > 0 THEN NA$ =
      A$
600 INVERSE
610 VTAB PEEK(37): HTAB 22: PRINT
      NA$
620 NORMAL
630 PRINT D$;"BSAVE"NA$",A$300,L
      $D0"
640 CALL 768: HOME : PRINT NA$"
      IS READY"
650 NEW : REM ERASE PROGRAM IN
      MEMORY
  
```

# NO MORE BUGS!

# CHECKBIN



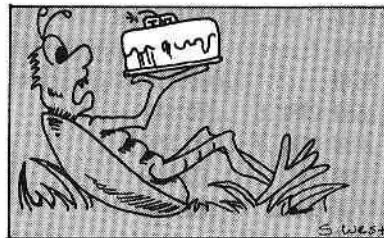
## Checkbin

by Robb Canfield

```

1000 .....
1010 *
1020 *   CHECK-BIN   *
1030 *
1040 *   BY   *
1050 *
1060 *   ROBB S. CANFIELD *
1070 *
1080 *   COPYRIGHT 1980 *
1090 *   SOFTKEY PUBLISHING *
1100 *
1110 .....
1120 *
1130 * LOCATIONS USED
1140 *
1150
1160
1170 *-----*
1180 * LOCATIONS IN THE MONITOR
1190 *-----*
1200
1210 COUT   .EQ $FDED   PRINT ACCUMULATOR IN ASCII
1220 CROUT  .EQ $FD8E   PRINT A CARRIAGE RETURN
1230 GETNUM .EQ $FFA7   CONVERT THE NUMBER AT $200 INTO HEX
1240 INPUT.BUFFER .EQ $200 THE INPUT BUFFER
1250 NXTAL  .EQ $FCBA   INCREMENT ALL ,A1H
1260 PRA1   .EQ $FD92   PRINT ALL ,A1H IN HEX AND A DASH "-"
1270 PRBYTE .EQ $FDDA   PRINT THE ACCUMULATOR IN HEX
1280 PRINT.SPACES .EQ $F948 PRINT THREE SPACES
1290 PRINTAX .EQ $F941  PRINT THE ACCUMULATOR AND X-REG IN HEX
1300 PRINT.X.SPACES .EQ $F94A PRINT X SPACES
1310 RETURN .EQ $FF58   A GUARANTEED RETURN COMMAND (RTS)
1320 STACK  .EQ $100    THE APPLE STACK
1330
1340
1350 *-----*
1360 * LOCATIONS IN I/O AREA (PAGE $C0)
1370 *-----*
1380
1390 CLEAR  .EQ $C010   CLEAR KEYBOARD STROBE
1400
1410 READ   .EQ $C000   READ KEYBOARD STROBE
1420
1430
1440 *-----*

```



REQUIREMENTS:  
An APPLE II or APPLE II+  
One disk drive

Checkbin is a checksum utility which will be used to confirm the accuracy of binary programs typed from future issues of both CORE and HARDCORE COMPUTIST.

Checkbin has many points in common with Checksoft, even though Checksoft is for use with Applesoft programs. They both generate the same type of checksums, they have nearly identical commands, and they must be used on programs that are in memory.

Checkbin has the following features:

1. The program is written in machine language.
2. A checksum is provided for every eight bytes of machine code.
3. Checkbin is relocatable (more on this later).
4. Checksums may be sent to a printer.

Every program that is to be verified by Checkbin must be accompanied by a checksum list generated by Checkbin. This is the same general technique that was used with Checksoft. Checkbin essentially works the same way as Checksoft, and has the same limitations.

Checkbin must be typed into memory as a series of bytes (the source code is included for the curious). There is no Applesoft program for Checkbin like there is for Checksoft. It was written to stand alone.

1. To make sure DOS is intact and working, reboot your Apple

IN#6

2. Enter the monitor

CALL-151

3. Type the following bytes

```

0300: 20 58 FF BA CA BD 00 01
0308: 18 69 1F 8D F9 03 05 62
0316: E8 8D 00 01 69 00 8D FA
0324: 03 05 63 A9 4C 8D F8 03
0332: 60 20 0E FD A9 0A 85 0A
0340: A0 00 84 31 20 A7 FF A9

```

```

0330: FF 05 31 A5 3C 05 0B A5
0338: 3D 05 0C 20 A7 FF A0 55
0340: A9 10 91 62 A9 FB C8 91
0348: 62 A0 00 F0 45 A5 3C 29
0350: 07 D0 42 38 A9 1F E5 24
0358: AA 20 4A F9 A9 A4 20 ED
0360: FD A5 0B A6 0C 20 41 F9
0368: C6 0A D0 26 20 8E FD A9
0370: 0A 05 0A AD 00 C0 EA EA
0378: 8D 10 C0 C9 83 F0 48 C9
0380: A0 F0 BB C9 9B D0 0B A9
0388: EA A0 55 91 62 C8 91 62
0390: A0 00 20 92 FD A9 A0 20
0398: ED FD B1 3C 48 20 DA FD
03A0: 68 6A 45 0B 2A 45 0C 85
03A8: 0B 45 0C 6A 85 0C 20 BA
03B0: FC 90 9A A9 1F E5 24 AA
03B8: 20 4A F9 A9 A4 20 ED FD
03C0: A5 0B A6 0C 20 41 F9 20
03C8: 8E FD 8D 10 C0 60
    
```

\*

4. Double check what you have typed. You should still be in the monitor. Type **300.378**
- Compare this to lines 300 through 378 in the listing.
5. Return to BASIC with **3D0G** or by pressing the RESET key if you have AutoStart ROM.
6. Save Checkbin **BSAVE CHECKBIN,A\$300,L\$D0**

## How To Use Checkbin

As previously mentioned, Checkbin and Checksoft have nearly identical commands. The only major difference is in how they are accessed. To use Checkbin, you must be in machine language, but you should already be there after typing a machine language program into memory. The command to start Checkbin is ctrl Y. This works much the same way as the ampersand key in Applesoft.

Type the beginning address of the machine language program you wish to check (remember, it must be in memory), then a period and the ending address of the program (both these addresses will be given with the listing), and a ctrl Y. The screen will fill with ten lines of code, each line consisting of an address, one to eight hexadecimal bytes, and a checksum value.

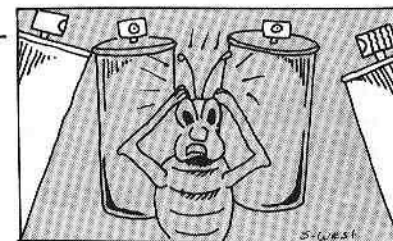
**Example:**

**0300-45 56 56 56 56 56 56 56 \$F987**

The number to the left of the dash is

```

1450 * MONITOR PAGE 0 LOCATIONS
1460 *-----
1470
1480 HTAB .EQ $24 CURSOR LOCATION HORIZONTALLY
1490 MODE .EQ $31 MODE COUNTER FOR GETNUM
1500 A1L .EQ $3C START OF MEMORY TO LIST, LOW BYTE
1510 A1H .EQ $3D START OF MEMORY TO LIST, HIGH BYTE
1520 A2L .EQ $3E END OF MEMORY TO LIST, LOW BYTE
1530 A2H .EQ $3F END OF MEMORY TO LIST, HIGH BYTE
1540
1550
1560 *-----
1570 * LOCATIONS I USE ON PAGE 0
1580 *-----
1590
1600 CHKSUM .EQ $0B THE CHECKSUM VALUES ($0B AND $0C)
1610 COUNTER .EQ $0A COUNTER FOR NUMBER OF LINES PRINTED
1620 RELOCATE .EQ $62 THE LOCATION WHERE CHECKBIN IS
1630
1640 *-----
1650 * THE FOLLOWING ARE KEY CODES
1660 *-----
1670
1680 CTRL.C .EQ $83
1690 CTRL.Y .EQ $99
1700 ESC .EQ $9B
1710 SPACE .EQ $A0
1720
1730 *-----
1740 * PAGE 3 LOCATIONS
1750 *-----
1760
1770 CTRL.Y.VECTOR .EQ $3F8 LOCATION TO JUMP TO WHEN A CTRL.Y IS PRESSED
1780
1790
1800
1810
1820
1830 .OR $300 PLACE ON PAGE 3
1840 .TF CHECKBIN
1850
1860 *-----
1870 * INITIALIZATION FOR CTRL.Y.VECTOR
1880 *-----
1890
1900 INIT JSR RETURN FIND WHERE WE ARE AT
1910 TSX
1920 DEX
1930 LDA STACK,X
1940 CLC
1950 ADC #SETUP-INIT-$2
1960 STA CTRL.Y.VECTOR+1 POINT CTRL.Y JUMP VECTOR HERE
1970 STA RELOCATE
1980 INX
1990 LDA STACK,X
2000 ADC #$00
    
```

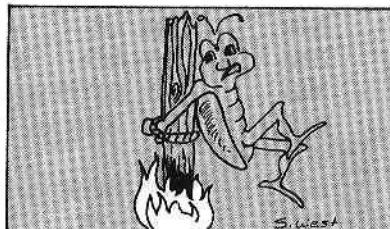




```

2010 STA CTRL.Y.VECTOR+2
2020 STA RELOCATE+1
2030 LDA #$4C
2040 STA CTRL.Y.VECTOR
2050 RTS
2060
2070 *-----
2080 * PROGRAM STARTS HERE
2090 *-----
2100
2110 SETUP JSR CROUT
2120 LDA #10 LIST 10 LINES AT A TIME
2130 STA COUNTER
2140 LDY #$00 GET CHARACTERS FROM INPUT BUFFER
2150 STY MODE CLEAR MODE
2160 JSR GETNUM TRANSLATE ASCII TO HEX AND PUT IN AIL,A1H
2170 LDA #$FF SET MODE
2180 STA MODE
2190 LDA AIL USE FIRST ADDRESS AS STARTING CHECKSUM
2200 STA CHKSUM
2210 LDA A1H
2220 STA CHKSUM+1
2230 CONT JSR GETNUM CONVERT 2ND NUMBER TO HEX
2240 START2 LDY #KEY+3-SETUP RE-ENABLE PAUSE COMMAND
2250 LDA #$10
2260 STA (RELOCATE),Y
2270 LDA #$FB
2280 INY
2290 STA (RELOCATE),Y
2300 LDY #$0 CLEAR Y-REG AND SET ZERO FLAG
2310 BEQ XAM ALWAYS DO THIS BRANCH
2320 MOD8CHK LDA AIL HAVE EIGHT BYTES BEEN PRINTED?
2330 AND #$07
2340 BNE CHECKSUMS NO. SO CONTINUE TO PRINT BYTES
2350 SEC
2360 LDA #31 CALC NUMBER OF SPACES TO PRINT
2370 SBC HTAB
2380 TAX
2390 JSR PRINT.X.SPACES
2400 LDA #'$+$00 PRINT A DOLLAR SIGN ($)
2410 JSR COUT
2420 LDA CHKSUM PRINT THE CHECKSUM VALUE
2430 LDX CHKSUM+1
2440 JSR PRINTAX
2450 DEC COUNTER COUNTER FOR NUMBER OF LINES
2460 BNE XAM IF NOT ZERO THEN CONTINUE
2470 JSR CROUT OTHERWISE PAUSE
2480 LDA #10 RESET COUNTER
2490 STA COUNTER
2500 KEY LDA READ GET KEYPRESS
2510 BPL KEY WAIT FOR A KEY TO BE PRESSED
2520 STA CLEAR CLEAR BUFFER
2530 CMP #CTRL.C STOP PROGRAM?
2540 BEQ DONE YES!
2550 CMP #SPACE RESET PAUSE FUNCTION?
2560 BEQ START2 YES!
2570 CMP #ESC DISABLE PAUSE FUNCTION?

```



the location of the bytes in memory after the line is stored. The eight bytes should appear in both the listing and the checksum value. The number of bytes on the first and last line of the listing may change slightly, but will never be more than eight or less than one.

If the checksum matches the one in the listing, all is okay and you can proceed to the next line. If the checksum doesn't match, there is an error in that line and it should be retyped. The same tricks used in Checksoft can be applied to Checkbin.

Checkbin recognizes the following commands:

**ESC** Stop pausing after every ten lines.  
**SPACE** Re-enable the pause function (this is different from Checksoft) and list the next ten lines.  
**ctrl C** Exit Checkbin.

Any other key will display the next ten lines.

## Using Checkbin with a Printer

If a printout of the checksums and the code is desired, simply turn on the printer before calling Checkbin. To turn on the printer from machine language, type PR#n, where n is the slot that your printer card is in (usually one). Then proceed with the check. When you are done, turn off the printer with PR#0.

Make sure that the printout goes to the screen as well as the printer. Your printer manual will have information on how to do this.

## How Checkbin Works

Checkbin works by running every byte in the program being verified through the checksum routine. The checksums are dependent on where in memory that program is located. Checkbin is relocatable. This means that it may be BRUN anywhere in memory without affecting the checksums. If Checkbin was not relocatable it might interfere with the program being checked.

All programs that have a checksum list will also have the beginning and ending addresses of the program, along with the location where Checkbin should be BRUN.

Checkbin can be used to check itself as well as the machine language part of

\* corrected CHECKSUM in No. 3, pages, due to change in "line" #378.

Checksum. I have included the checksums for both programs.

There are no configurations for Checkbin.

## Checksums for Checkbin

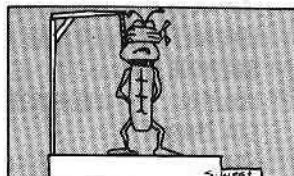
BEG: \*300.3CD END:

0300- 20 58 FF BA CA BD 00 01 \$B2E1  
 0308- 18 69 1F 8D F9 03 85 62 \$286C  
 0310- E8 BD 00 01 69 00 8D FA \$D2ED  
 0318- 03 85 63 A9 4C 8D F8 03 \$68E2  
 0320- 60 20 0E FD A9 0A 85 0A \$2066  
 0328- A0 00 04 31 20 A7 FF A9 \$5284  
 0330- FF 85 31 A5 3C 85 0B A5 \$0223  
 0338- 3D 85 0C 20 A7 FF A0 55 \$D448  
 0340- A9 10 91 62 A9 FB C8 91 \$6CDF  
 0348- 62 A0 00 F0 45 A5 3C 29 \$E502

0350- 07 D0 42 38 A9 1F E5 24 \$E286  
 0358- AA 20 4A F9 A9 A4 20 ED \$3E81  
 0360- FD A5 0B A6 0C 20 41 F9 \$459D  
 0368- C6 0A D0 26 20 8E FD A9 \$2D05  
 0370- 0A 85 0A AD 00 C0 EA EA \$5EB6 \*  
 0378- 8D 10 C0 C9 83 F0 48 C9 \$C54A  
 0380- A0 F0 BB C9 9B D0 0B A9 \$6C5B  
 0388- EA A0 55 91 62 C8 91 62 \$29C6  
 0390- A0 00 20 92 FD A9 A0 20 \$F038  
 0398- ED FD B1 3C 48 20 DA FD \$E122

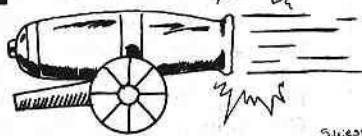
03A0- 68 6A 45 0B 2A 45 0C 85 \$C531  
 03A8- 0B 45 0C 6A 85 0C 20 BA \$0598  
 03B0- FC 90 9A A9 1F E5 24 AA \$C561  
 03B8- 20 4A F9 A9 A4 20 ED FD \$8DF6  
 03C0- A5 0B A6 0C 20 41 F9 20 \$2A83  
 03C8- 8E FD 8D 10 C0 60 \$C406

2580 BNE XAM NO, SO SKIP THE REWRITE SECTION.  
 2590 LDA #SEA DISABLE PAUSE FUNCTION.  
 2600 LDY #KEY+3-SETUP  
 2610 STA (RELOCATE),Y  
 2620 INY  
 2630 STA (RELOCATE),Y  
 2640 LDY #00  
 2650 XAM JSR PRA1 PRINTS AIL,A1H AND A DASH  
 2660 CHECKSUMS LDA #SA0 PRINTS A SPACE  
 2670 JSR COUT  
 2680 LDA (AIL),Y PRINT THE BYTE  
 2690 PHA SAVE THIS VALUE  
 2700 JSR PRBYTE  
 2710 PLA RETRIEVE VALUE  
 2720 ROR RUN THROUGH CHECKSUM ROUTINE  
 2730 EOR CHKSUM  
 2740 ROL  
 2750 EOR CHKSUM+1  
 2760 STA CHKSUM  
 2770 EOR CHKSUM+1  
 2780 ROR  
 2790 STA CHKSUM+1  
 2800 JSR NXTA1 INCREMENT AIL,A1H AND COMPARE TO A2L,A2H  
 2810 BCC MODBCHK IF CARRY CLEAR THEN AIL,A1H (>) A2L,A2H SO CONTINUE  
 2820 LDA #31  
 2830 SBC HTAB  
 2840 TAX  
 2850 JSR PRINT.X.SPACES  
 2860 LDA #'\$+\$00 PRINT FINAL CHECKSUM  
 2870 JSR COUT  
 2880 LDA CHKSUM  
 2890 LDX CHKSUM+1  
 2900 JSR PRINTAX  
 2910 DONE JSR CROUT EXIT PROGRAM  
 2920 STA CLEAR  
 2930 RTS  
 2940



\*

## ADVANCED LAYING TECHNIQUES



### And a Quick Softkey. . .

To copy the entire disk, use COPYA. Then, using a disk edit program such as *DiskEdit*, read Track 17, Sector 0E, and change address CD from 49 to 60. Finally, write the sector back to the disk.

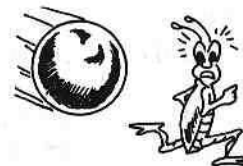
This back-up copy can be copied using any of the numerous copy programs on the market. To run the program, simply boot the disk.

## Cannonball Blitz A.P.T.

Here is a short A.P.T. for **Cannonball Blitz**, which will reduce the hazards encountered on the second level of play.

After finishing Level 1, just press the space bar and the repeat key simultaneously (or press the space bar continuously if you have automatic repeat) until the screen changes to the next level.

When play begins at the second level, the number of cannons will have been reduced to only two.



**Cannonball Blitz**, Sierra On-Line, Inc., 36575 Mudge Ranch Road, Coarsegold, California 93614.

## INPUT

continued from page 2

out dollars ahead if they licensed look-alikes to non-Atari programmers, but I seriously doubt they'll sell much more product as a result.

We as computer consumers are all headed for lean times ahead if we're only to be offered one game program where you have to shoot down space ships dropping bombs on us. Would a programmer be permitted to change the space ships to flying saucers and implement his own version? What if he used lasers instead of missiles, would that be different enough to keep him from getting sued? And who's to decide just how much different is different enough, a judge?!?!?

If this carries over to other types of publishing, just think . . . No more "boy meets girl . . . girl meets girl . . . boy gets girl," sorry, it's been used before. And the western where the new Marshall comes to town to clean it up and can't get help from anyone but the pretty widow . . . Well, you get the idea.

Someone once said, "There are no new ideas, just new combinations of ideas."

Granted, Atari deserves to be rewarded handsomely for creating Pac Man, and they are being so rewarded. But no one copied their source code (at least I don't think it's been alleged) so where have their copyrights been infringed? This is crazy; Atari can stop others from publishing something that merely looks like something of theirs while Apple can't get an injunction to stop the manufacture of computers that are alleged to use an actual copyrighted source code of theirs.

All of this is, of course, just my opinion, and I've already confessed to knowing very little about copyright law, but, in my opinion, if Atari pulls this off, it's bad news for everyone but Atari. We consumers shouldn't be hailing it as Justice.

Warren Michelson  
Page, AZ

## Hardcore Info Valuable to Business

After driving 160 miles (round trip) to obtain a copy of issue #3, and approximately 140 miles to find issues 1 and 2, I determined that a subscription would

# hardcore

*While the price of everything else in Appledom is shooting SKY-HIGH, subscription rates for the new CORE and HARDCORE COMPUTIST magazines are HALF the cover price!*

Instead of 4 issues and 4 updates (a \$20.00 value), a subscription now consists of 12 full-sized magazines (8 of Hardcore Computist and 4 of Core—a \$40.00 value), yet our price for U.S. subscribers is still

**\$20.00**

Current subscribers: Don't wait to renew. Our rates can't stay this low forever!

- New subscriber  
 Renewal

### BACK ISSUES:

Hardcore #1 \_\_\_\_\_  
Update 1.1 \_\_\_\_\_  
~~Hardcore #2~~ **SOLD OUT!**  
Update 2.1 \_\_\_\_\_  
Hardcore #3 \_\_\_\_\_  
Update 3.1 \_\_\_\_\_  
Update 3.2 \_\_\_\_\_

(\_\_\_\_) years  
subscription \_\_\_\_\_

TOTAL \_\_\_\_\_

NAME \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

VISA OR  
MASTERCARD NO. \_\_\_\_\_

EXP.  
DATE \_\_\_\_\_

SIGNATURE \_\_\_\_\_

Sorry . . .  
No purchase orders  
U.S. funds only

Make checks payable to: **HARDCORE**  
P.O. Box 44549  
Tacoma, WA 98444

### SUBSCRIPTION RATES:

U.S.A. \$20.00  
1st Class, APO, FPO \$29.00  
Canada \$29.00  
Mexico \$32.00  
All others \$42.00

### BACK ISSUES:

Issues #1,2,3: \$5 each, U.S.A.  
\$8 each, other  
Updates: \$3 each, U.S.A.  
\$5 each, other

Subscriptions begin with  
the current issue.

Please specify otherwise.  
Back issues available  
only while they last.

take much less of my time and would be much more enjoyable.

I appreciate and applaud the wealth of information you have packed into each issue. As a businessman, I feel that the information that you are providing will enable myself, and others like me, to make back-up copies of valuable business software. Since I no longer support those software publishers who deprive me of my right to protect my software investment, I welcome every opportunity to protect (legally) my company's existing software library.

Keep up the good work that you have begun and "stick by your guns." I believe you'll see the "other community" slowly beating a path to your door.

William E. Noel  
Ontario, CA

## Some Tips for Ultima

I have really enjoyed Issue 3 and the last two Updates. Especially the articles on *HyperDOS*, *Castle Wolfenstein* and *Ultima*. *Ultima* is the subject of this letter.

Using the I.O.B. program and following the excellent directions given by Bobby I was able to make 16-sector copies of both sides of the *Ultima* disk. Next I tried to make a character disk and then realized that the Player Master would no longer do this since it was originally 13 sectors.

After a little searching I discovered that the program named 'PLAYER DISK' included the copy program to make a player character disk. The

changes necessary to make it work with 16-sector DOS 3.3 were minimal and it now works better (faster) than the original.

Just change the following lines to read as shown:

```
531 FOR A = 768 TO 801: READ B:
POKE A,B: NEXT A: RWTS = 768
```

```
532 DATA 32, 227, 3, 32, 217, 3, 176,
25, 238, 241, 183, 206, 237, 183, 173,
237, 183
```

```
533 DATA 201, 255, 208, 235, 169, 15,
141, 237, 183, 238, 236, 183, 198, 0,
208, 223, 96
```

```
590 FOR T = 3 TO 27 STEP 8: GOSUB
640: NEXT T
```

```
790 POKE VO, 0: POKE TR, T: POKE
SE, 15: POKE BL, 0: POKE BH, 20:
POKE CO, C
```

```
800 POKE 0, 8: CALL RWTS: IF PEEK
(0) = 0 THEN RETURN
```

Now save this program PLAYER DISK back to the Player Master. That's all it takes to make the Player Master work as intended.

C.V. Fields  
Sacramento, CA

## A Lyrical Letter to the Editor: Not Pirates, But Pioneers?

I'm sick and tired of hearing all  
The names that some are calling.  
You'd think, for all their bitching, that  
The cops would soon be hauling  
A bevy of us off to jail!  
For breaking copyrights?  
Folks oughta take a closer look.  
They'd see we set our sights  
Much higher than mere pirating.  
Indeed, in coming years,  
We'll not be known as pirates, but  
—COMPUTER PIONEERS!

We hurt the micro industry?  
That's rumored in some quarters.  
When really, we're the industry's  
Most powerful supporters!  
For software can't be run without  
Some hardware. Obvious?  
Then why are all those micro firms  
So paranoid of us?  
They'd welcome us with open arms,  
If only they could see  
That what we do, does benefit  
Their whole darn industry!

I've never seen an issue of  
Your **HARDCORE** publication.  
And yet, I've heard the word of how  
You captivate the nation!  
A lot of folks are very proud  
Of what you're here to do.  
Indeed, you are succeeding! And I  
Praise the Lord for you!

Michael Herbert Shadick  
Minneapolis, MN

## Praise from a Magazine Virtuoso

I subscribe to *Popular Computing*, *Creative Computing*, *Byte*, *Softalk*, *Apple Orchard*, *In Cider*, *Computers and Electronics*, *Softside*, and maybe something else and I've never been impressed enough with one to write and say anything, but yours is different. I've learned more about how my computer works from just one issue of **HARDCORE** than I have from any of the others.

William Wingfield Jr.  
Martinsville, VA

(advertisement)

**Anyone who has purchased an issue of *Hardcore Computing* or any other product from Norman Napier, "Red Rebel," or Pirates Harbor, either by mail or at a computer fair, please contact Bev R. Haight at 14404 East "D" Street, Tacoma, WA, 98445. You will be of great help in collecting information for a comprehensive article on Pirates Harbor.**

# the REPLAY II card

by Robb Canfield

*Replay II* is an updated version of the original *Replay* card, which was among the first copy cards marketed. It is still being produced by the same company, but the name has been changed from Texas Ranch And Shoreline Systems (TRASH) to Micro Analyst, Inc. (a wise decision).

The revised *Replay II* is a definite improvement over the first *Replay*. The card no longer interferes with any other card in the computer and can now copy the entire 64K of memory. The copy switch has been moved from the card to the end of an 18-inch cable extending out from the back of the computer. In addition, the card now makes three different types of copies.

## Making a Copy

The first and easiest way to make a copy of the memory is to press the copy button and select "C" from the menu displayed. The card will prompt you to insert a blank diskette, and the program in memory is then copied in about 15 seconds. The quick copy will reload in about 10 seconds. This method provides a fast and convenient way of making a copy. Unfortunately, the copy is written in a non-standard format and requires that the *Replay* card be in the computer to boot the disk. A RAM card is not necessary.

The second copy method makes a standard DOS file which represents the entire 48K of memory. Its only limitation is that a RAM card is required to boot the disk (the *Replay* card need not be in the computer). The copy will reload in approximately one minute. To make this kind of copy requires no programming skill, and consists of first making a quick copy (the first one mentioned) and then running "DOSMAKER" from the *Replay* utility disk to convert the quick copy to standard DOS.

The third type of copy creates a single binary program that will boot under normal DOS. It does not require the *Replay* card or a RAM card. This kind of copy is called a "packed" copy, and it is made with the use of the packer program on the *Replay* utility disk. A knowledge of machine language is required to understand and use the packer utilities.

## Advanced Topics

The packer and the programs that go along with it make up a powerful set of utilities. Using these utilities, it is possible to analyze and condense the 48K copy into a more compact file that does not require extra hardware to run. Some of the utilities included help scan for used and unused sections of code and eliminate those sections that are unused. This would be the best type of copy if it were not so difficult to understand.

A number of auto-pack utilities are included. These allow the inexperienced user to pack some of the more popular programs without having to know a great deal about machine language. There are also utilities that allow the creation of an auto-pack file if you can figure out how to pack the file in the first place.

## Functions on the Card

When the copy switch is pushed, a menu appears with a choice to copy the disk, view various screens or enter a special monitor. This monitor is one of the more powerful and useful features of the card. The monitor shows where the reset vector is pointed and what the program counter and stack pointer were when the copy switch was pressed. From the monitor, memory may be displayed as hex or ASCII values (there is no disassembler available). Specific values can be changed and a search for a certain byte may be performed.

The only complaints I have are that there is no way to enter into the normal Apple monitor, and that to look at the original pages (0-7) requires the user to look at the memory on the card itself (a rather hit or miss proposition).

## Documentation

The manual makes an effort to explain all the functions of the card, and does well on the first and second copy methods. It falls down on the packer section. The instructions for the packer are hard to understand and the utilities are not at all user-friendly. They require the user to keep referring to a chart that relates logical and physical sectors. This should have been implemented in software, leaving the user free to think about more important matters.

## Summary

The *Replay* card is very easy to use and understand (as long as the packer section is avoided like the plague). The *Replay* card no longer has any conflicts with other cards or slots and it has more features than *Wildcard* and *Snapshot*. Its versatility allows it to be used as a copy card and as a very helpful programming aid (one I have used many times when my program decided to leave this universe). Of all the cards I have used, I have found the *Replay II* to be the most powerful card available.

## a note on Crackshot and Replay

The *Crackshot* card sold by Pirate's Harbor is an earlier version of the *Replay 1.0* card. This card has conflicts with other cards and slots.

The newer *Replay II* card is available from Micro Analyst, Inc. at P.O. Box 15003, Austin, Texas, 78761 for \$150.00.

The *Replay II* has no conflicts with other cards or slots and makes three different types of copies.

Purchasers of the *Crackshot* card may not be able to upgrade to the *Replay II* as Micro Analyst, Inc., has informed us that Pirate's Harbor is not an authorized dealer.

If you are considering the purchase of a copy card, call Micro Analyst, Inc. directly at (512) 926-4527.

Reprinted with permission from *Peelings II*, volume 4, number 1, pages 45 and 48. *Peelings II* is copyrighted by *Peelings II, Inc.* A one-year subscription is available for \$21 from:  
**Peelings II, P.O. Box 188**  
**Las Cruces, NM 88004-0188, (505) 526-8364**

## Hardware Copy Cards

The war goes on. The scenario is similar to the electronic warfare situation. First there were protected programs on cassette. Those were broken. Then came protected diskettes and the nibble copiers. Then more sophisticated protection mechanisms were introduced and more sophisticated copy programs became available. Now we have the hardware copy cards. Each of the three cards reviewed is designed to make copies of the program in memory, for archival purposes. All of these hardware copy methods are subject to similar restrictions. They only easily copy programs which reside entirely in memory; programs requiring multiple disk accesses (except possibly for user data files) will most likely remain uncracked. At present these copy cards will not satisfy all of your archival needs. Total satisfaction can only be achieved by unlocked software, or software which allows a limited number of back-ups to be made. But for single load programs, perhaps one of these cards will be satisfactory. None of the cards is really intended to copy programs running with a Z-80 card. □

## CRACK-SHOT

**PIRATES HARBOR**  
**P. O. Box 8928**  
**Boston, MA 02114**  
**617-738-5051 (modem)**

\$149.95  
Machine language  
48K, DOS 3.3

Unlocked

*Reviewed by Edward Burlbaw*

### INTRODUCTION

CRACK-SHOT can be installed in any slot, provided there is no card in slot 0. If you have a RAM card, you will have to remove it. In order to run copied programs, CRACK-SHOT must be installed in one of the slots (slot 0 can be occupied) or a RAM card can be used. There are known conflicts with certain other cards, such as the Hayes Micromodem II and some video cards. I also had problems using my Grappler card with CRACK-SHOT installed, and vice versa. Any conflicting cards must be removed before using CRACK-SHOT. A good rule would be to take everything out, except CRACK-SHOT and the disk controller card.

### MAKING A COPY

This is a very simple procedure. Once the card is installed, any program can be run as normal. When the program is at the desired point, the CRACK-SHOT copy switch is flipped and a blank diskette placed in drive one. A copy will be made in fifteen seconds. The blank diskette then has a copy of the entire 48K of memory, with the exception of the screen display locations. These must be set, on booting, by a menu within the CRACK-SHOT program. To execute your copy with the CRACK-SHOT card in place, simply CALL a location on the card and it takes over from there. You have essentially created an archival copy of the program which requires a hardware "key" (CRACK-SHOT) to run. If you are satisfied, there is nothing more to do. File the original away and use the CRACK-SHOT copy.

### ADVANCED TOPICS

If you are satisfied with the copy described in the previous paragraph, you will not be interested in the other possibilities of CRACK-SHOT. There are other utilities included on the system disk for analysis of the copy. The EDIT utility will allow you to read tracks into a buffer, disassemble the code in the buffer, write the buffer back to disk, or exit to BASIC (for BSAVE operations). The PACKER utility is perhaps the most powerful of the utilities. For example, if it can be determined that the entire 48K is not being used by the program, the program areas of memory can be consolidated and automatically unfolded once resident in memory. This would allow loading under DOS 3.3 without the CRACK-SHOT card. I would feel more com-

fortable with an archive copy of this nature. The PACKER utility is also the most difficult to understand. It is confusing because it requires one to keep track of physical and logical locations of the code and the location on the diskette. Other features of PACKER will read and attempt to analyze the various sections of the copied memory to locate code or ASCII data sections. This can be done manually or automatically. Packing command files can also be created but, obviously, you have to be able to first do it manually.

The language card is supported in several ways. The utility programs will run with the card and use the extra memory to increase the size of the binary file that is packable. The language card can be used in place of the CRACK-SHOT card, for running the copy.

### DOCUMENTATION

The manual is very extensive and attempts to explain all of the features and possibilities of CRACK-SHOT. The simple copy and re-run procedures are well covered, but the advanced topics are not adequately treated. Perhaps the experienced user will not need any more explanation, but a simple tutorial would improve the usability considerably. There are nine Appendixes with some examples, tips, trouble shooting hints, and packing parameters (sound familiar?).

### SUMMARY

In addition to making archival copies of single-load protected programs, CRACK-SHOT can be used as a gaming tool by making copies which start at higher levels. Once understood, the extensive advanced features could be useful. If you dislike performing surgery on your Apple, you may be annoyed with the slot 0 restrictions and the conflict with other cards. □

## SNAPSHOT

**Dark Star Systems**  
**54, Robin Hood Way**  
**Greenford, Mddx. UB6 7QN**  
**01-900-0104**

**P. O. Box 140**  
**Amherst, MA 01004**  
**413-584-7600**

\$109.95 (plus p&h)  
Machine language  
Dos 3.3, 64K (certain RAM cards only)

Unlocked

*Reviewed by Edward Burlbaw*

### INSTALLATION

SNAPSHOT can be used with only some RAM cards. The more common of these are Apple, Microsoft, and Ramex. Also supported

Reprinted with permission from *Peelings II*, volume 4, number 1, pages 45 and 48. *Peelings II* is copyrighted by *Peelings II, Inc.* A one-year subscription is available for \$21 from:  
**Peelings II, P.O. Box 188  
Las Cruces, NM 88004-0188, (505) 526-8364**

are Digitek, MPC, RH Electronics, and Super Ram II. It also works with the Franklin Ace 100, with 16K card, and the Ace 1000. The list is growing, so if you have any questions concerning the suitability of your RAM card, check it out with Dark Star Systems. Known exceptions are Andromeda, Indigo, and Saturn 32K. You must remove one of the chips from your RAM card and plug the ribbon cable connector, from the SNAPSHOT board, into the empty socket. The SNAPSHOT card can then be inserted into any of the nearby slots on the motherboard. There is an empty socket on the card into which to plug the game paddles or a four-switch dip switch. Either of these will be used to activate the copy program when it is time.

#### MAKING A COPY

Either the game paddle or one of the switches can be used to activate the SNAPSHOT card. If it is not active, the RAM card acts as if SNAPSHOT were not present. This enables the RAM card to be used normally. The SNAPSHOT software is booted into the RAM card. Pushing the paddle switch, or the appropriate dip switch, will activate SNAPSHOT and display its menu. The "subject" diskette can then be booted. (Thirteen-sector diskettes can be used with the BASICS diskette.) Once the subject is in memory, pushing the button again interrupts the program and displays the SNAPSHOT menu. The menu has several options in a logical order. If you proceed from the top to the bottom, you will be doing the steps necessary to create a working backup. The video display settings are selected, both for bootup and run. The memory is then saved on disk. The final step is to use the last option of converting SNAPSHOT disk to a backup disk that does not require the software to be in the RAM card. The backup will now run on any 64K Apple (independent of the type of memory card). It may or may not run on a 48K Apple. By activating the SNAPSHOT card, and disabling the RAM card, the copy can be checked on a 48K machine.

One of the menu options allows the user to exit to the monitor for disassembly of the program in memory or other diagnosis. Some memory is available on the RAM card for user programs to assist in this. An exit to BASIC is also allowed which may make the program in memory LISTable. There is very little treatment of advanced techniques, but I submit that once the DOS 3.3 backup is created, very little more is needed.

#### DOCUMENTATION

The instruction booklet consists of 13 typewritten half-pages. Everything necessary to make a DOS 3.3 backup of a memory resident program is found in it. It's not fancy, but definitely adequate.

#### SUMMARY

SNAPSHOT can be used in a manner similar to CRACKSHOT for creating high level starting games or archival copies of single load programs. The necessity to use one of the compatible RAM cards will be a problem, unless you happen to have one of them. The simplicity of the menu-driven software is very convenient to use. □

The three cards reviewed above are all designed to provide a method of creating archive copies of legitimately acquired software. Each uses a slightly different approach to achieve essentially the same end. CRACKSHOT requires that any card in slot 0 be removed before operation and other interface cards can interfere with its operation. These include some fairly common cards (i.e., D.C. Hayes and Grappler) and, if in place, would interfere with the operation. There are more advanced features supported, but an understanding of some of them is required to create a DOS 3.3 backup. SNAPSHOT places fewer restrictions on existing hardware. Used with one of the supported RAM cards, there are no restrictions, but a chip must be removed from the card. That could be viewed as a disadvantage. The copy procedure is perhaps the easiest and clearest of the three cards. WILDCARD places the least restrictions on the hardware. In purchasing, this could be the deciding factor. The copy procedure is clear and straightforward.

Overall, with one of the supported RAM cards, SNAPSHOT, at \$109.95 is the best buy. WILDCARD places next because of ease of use. CRACKSHOT is the most expensive of the three, has memory conflicts with other cards, and requires the most technical knowledge to use. However, it also has more features available to the advanced user. □

## WILDCARD

East Side Software Co.  
344 E. 63 St., Suite 14-A  
New York City, NY 10021  
212-355-2860

\$129.95 (\$3 s&h)  
Machine language  
Dos 3.3, 64K, Apple II+

Unlocked

*Reviewed by Edward Burlaw*

#### INSTALLATION

With power off, open the Apple cover and place WILDCARD in any empty slot with the cable exiting through one of the slots in the case. Close the cover and turn the power back on. The only requirement is that you have a RAM card and an empty slot.

#### MAKING A COPY

Load the subject program. When it is at the selected point, press the WILDCARD button and RETURN. The WILDCARD menu will appear and allow you to boot, restart, or exit to monitor. To make a copy, use the boot option with the WILDCARD system disk in drive 1. From there follow the menus to create a turnkey autoboot copy of the subject software. You will need a blank diskette and it will take about two minutes to complete the copy. You then have an archive copy which will boot on a 64K machine. It may be possible to compress it to run on a 48K machine. The compression can be attempted automatically by selecting the appropriate option. If this is unsuccessful, it will still run on a 64K machine. As with the other copy cards, the video screen to be displayed must be selected during the copy process.

#### DOCUMENTATION

The thirty half-sized pages contain installation and operation procedures. The copy section leads you through a sample copy procedure. It is simple enough that once or twice through the book will be sufficient. A process is described by which a 64K program may be copied; however, if it cannot be compressed sufficiently it will not be successful. There is a short section discussing some uses of the utility option for the machine language programmer.

#### SUMMARY

WILDCARD like SNAPSHOT and CRACKSHOT provides a convenient method for backing up single load programs. It requires the least modification of, and places the fewest restrictions on, the existing hardware. It is both simple and easy to use. □

# CHECKSUMS FOR

10 - \$BADD	490 - \$20AB	970 - \$42C1	1380 - \$30ED	1790 - \$7EF1	2200 - \$A39F	2600 - \$E5ED	3070 - \$6048
20 - \$32F3	500 - \$1DBA	980 - \$5CAF	1390 - \$0B1D	1800 - \$3B15		2610 - \$2776	3080 - \$5120
30 - \$0CCE	510 - \$70E9	990 - \$110E	1400 - \$E42B		2210 - \$80B0	2620 - \$C5E2	3090 - \$CC76
40 - \$7F9A	520 - \$9D78	1000 - \$E04A		1810 - \$8A9F	2220 - \$6372	2630 - \$3E51	3100 - \$1380
50 - \$601F	530 - \$F7BA		1410 - \$158B	1820 - \$7800	2230 - \$85A6	2640 - \$063F	3110 - \$2F54
60 - \$3A19	540 - \$0AB4	1010 - \$991A	1420 - \$F227	1830 - \$F425	2240 - \$198C	2650 - \$4449	3120 - \$67A7
70 - \$0796	550 - \$7BC6	1020 - \$D303	1430 - \$C9CF	1840 - \$6EC8	2250 - \$56D0	2660 - \$7D6E	3130 - \$9155
80 - \$56C8	560 - \$F563	1030 - \$8159	1440 - \$151B	1850 - \$C5E7	2260 - \$59AB	2670 - \$256D	3140 - \$652E
90 - \$82DE	570 - \$67CB	1040 - \$676F	1450 - \$CDA A	1860 - \$FE39	2270 - \$AE38	2680 - \$6F51	3150 - \$C736
100 - \$D39A	580 - \$588D	1050 - \$8DCA	1460 - \$3EB2	1870 - \$D290	2280 - \$641E	2690 - \$DA03	3160 - \$197F
110 - \$FD44	590 - \$E2D7	1060 - \$FC37	1470 - \$C85D	1880 - \$646A	2290 - \$2153	2700 - \$D166	3170 - \$A52A
120 - \$EE94	600 - \$573D	1070 - \$0DF5	1480 - \$8A5B	1890 - \$8C06	2300 - \$01D9	2710 - \$3EA5	3180 - \$DFBA
130 - \$06CD		1080 - \$EE6F	1490 - \$9505	1900 - \$837C	2310 - \$BDC5	2720 - \$CA9B	3190 - \$8E4B
140 - \$4B03	610 - \$9080					2730 - \$9B30	3200 - \$6D86
150 - \$9693	620 - \$8B3D					2740 - \$5A16	3210 - \$A916
160 - \$7864	630 - \$2BF0					2750 - \$2B12	3220 - \$700B
170 - \$2E88	640 - \$27BE					2760 - \$7369	3230 - \$3BAA
180 - \$75E2	650 - \$80E7					2770 - \$FEB2	3240 - \$EB5A
190 - \$889F	660 - \$90D3					2780 - \$774B	3250 - \$AD29
200 - \$CF16	670 - \$7A9B					2790 - \$A8F6	3260 - \$8E10
	680 - \$94EC					2800 - \$CC32	3270 - \$4FB6
210 - \$CCCB	690 - \$7F70	1090 - \$9B14	1500 - \$869F	1910 - \$375C	2320 - \$82D3	2810 - \$156A	3280 - \$F983
220 - \$504B	700 - \$CE91	1100 - \$33FF	1510 - \$F1C3	1920 - \$80D2	2330 - \$5847	2820 - \$F597	3290 - \$8317
230 - \$3638	710 - \$381B	1110 - \$44AE	1520 - \$702A	1930 - \$2767	2340 - \$6243	2830 - \$42F3	3300 - \$468A
240 - \$DF71	720 - \$E50A	1120 - \$30F6	1530 - \$50B1	1940 - \$9A1D	2350 - \$30D3	2840 - \$8111	3310 - \$7253
250 - \$3886	730 - \$D35D	1130 - \$F1BB	1540 - \$7763	1950 - \$1C0A	2360 - \$5BEA	2850 - \$03BF	3320 - \$C7C2
260 - \$4F24	740 - \$15E4	1140 - \$1C26	1550 - \$057D	1960 - \$2EDD	2370 - \$E7F5	2860 - \$496C	3330 - \$3A38
270 - \$616A	750 - \$D3F9	1150 - \$51A5	1560 - \$7C08	1970 - \$1527	2380 - \$BD62	2870 - \$ED10	3340 - \$9FB8
280 - \$1AD5	760 - \$84E6	1160 - \$D402	1570 - \$7C14	1980 - \$7965	2390 - \$DD57	2880 - \$6DBF	3350 - \$5AE7
290 - \$AC8E	770 - \$9D20	1170 - \$C17D	1580 - \$8732	1990 - \$D80F	2400 - \$7168	2890 - \$78C1	3360 - \$3187
300 - \$F901	780 - \$CD6D	1180 - \$346E	1590 - \$3062	2000 - \$0CB0		2900 - \$3284	3370 - \$8352
310 - \$E122	790 - \$73FC	1190 - \$F585	1600 - \$543F	2010 - \$ABD1	2410 - \$4352	2910 - \$6A0F	3380 - \$CE98
320 - \$F362	800 - \$4E92	1200 - \$41CE		2020 - \$9823	2420 - \$E585	2920 - \$1443	3390 - \$6CC7
330 - \$746B				2030 - \$31CA	2430 - \$4D6A	2930 - \$766E	3400 - \$38BF
340 - \$73AF	810 - \$6DF1	1210 - \$604A	1610 - \$F614	2040 - \$EB38	2440 - \$EEF4	2940 - \$8D32	3410 - \$7E71
350 - \$382A	820 - \$A5D1	1220 - \$CA43	1620 - \$3E31	2050 - \$3736	2450 - \$823A	2950 - \$8D26	3420 - \$E6AE
360 - \$F085	830 - \$DCBC	1230 - \$9BD4	1630 - \$77A8	2060 - \$E5DA	2460 - \$FF47	2960 - \$698D	3430 - \$D138
370 - \$6FD8	840 - \$5131	1240 - \$845E	1640 - \$ADCA	2070 - \$3F2C	2470 - \$FEB1	2970 - \$3CFE	3440 - \$4ADE
380 - \$9F80	850 - \$3A50	1250 - \$2913	1650 - \$8F1A	2080 - \$CCD3	2480 - \$937D	2980 - \$A80B	3450 - \$4844
390 - \$71EB	860 - \$2FE6	1260 - \$1FD8	1660 - \$DF12	2090 - \$D1ED	2490 - \$4765	2990 - \$DFB9	3460 - \$A9FF
400 - \$92F9	870 - \$4087	1270 - \$22D8	1670 - \$70B1	2100 - \$3A3A	2500 - \$7910	3000 - \$6569	3470 - \$8125
	880 - \$488C	1280 - \$12C6	1680 - \$7751	2110 - \$7082	2510 - \$DB06	3010 - \$C96E	3480 - \$7466
410 - \$6F71	890 - \$22AE	1290 - \$88C2	1690 - \$78AF	2120 - \$AB3A	2520 - \$DEF5	3020 - \$EB74	3490 - \$86FA
420 - \$527E	900 - \$2680	1300 - \$60D2	1700 - \$5E45	2130 - \$7EE4	2530 - \$E122	3030 - \$6D06	3500 - \$8D40
430 - \$A403	910 - \$E421	1310 - \$7D4A	1710 - \$3E74	2140 - \$8B66	2540 - \$27A5	3040 - \$DEF4	3510 - \$DE2A
440 - \$8BAF	920 - \$968F	1320 - \$9085	1720 - \$342C	2150 - \$E01F	2550 - \$10E8	3050 - \$082D	3520 - \$DCC6
450 - \$FBC0	930 - \$828B	1330 - \$189E	1730 - \$FA36	2160 - \$3156	2560 - \$BF06	3060 - \$8284	3530 - \$8805
460 - \$4BB5	940 - \$852B	1340 - \$6CFE	1740 - \$835F	2170 - \$D34F	2570 - \$8086		3540 - \$2F44
470 - \$8BA2	950 - \$C22A	1350 - \$84AE	1750 - \$5ED9	2180 - \$5D9A	2580 - \$ABE1		
480 - \$EE40	960 - \$FCF3	1360 - \$CC9C	1760 - \$D331	2190 - \$3469	2590 - \$5EAE		
		1370 - \$737B	1770 - \$0F47				
			1780 - \$D76A				

## QD.Editor

Configuration 1



# **core** Spring Issue

## QUICK DRAW

### Quick Draw. Obj

BEG: 803.8AB END:

0003-	05 D2 06 D1 04	\$1409
0008-	D0 A6 06 06 08 A6 07 BC	\$D405
0010-	0C 08 04 00 C9 0D F0 5F	\$1E33
0018-	25 32 A8 04 01 18 B9 A4	\$7ECB
0020-	08 65 04 0D 4B 08 B9 A4	\$33AB
0028-	09 65 05 0D 4C 08 A6 25	\$CB4D
0030-	BC 0C 08 04 D3 A2 00 C0	\$9821
0038-	C0 00 16 B9 A4 0A 05 D4	\$2055
0040-	B9 64 08 18 65 E6 05 D5	\$C400
0048-	A4 24 0D FF FF 91 D4 91	\$0CAB
0050-	D4 E6 D3 A4 D3 E8 E4 00	\$B688
0058-	D0 D0 C6 08 F0 11 A4 24	\$978A
0060-	C8 C4 21 F0 12 04 24 18	\$2EEF
0068-	A5 07 65 01 4C 1A 08 E6	\$8C14
0070-	24 A4 24 C4 21 90 0E A0	\$6BBB
0078-	00 04 24 E6 25 A4 25 C4	\$D18A
0080-	23 90 02 C6 25 A5 01 A6	\$F411
0088-	D1 A4 D0 60	\$944A

### Editor.Set

BEG: D00.DFF END:

0000-	7F 7F 7F 7F 7F 7F 7F 7F	\$00AD	0088-	3C 25 25 3C 3C 25 25 3C	\$B463
0008-	7F 01 01 01 01 71 11 11	\$9F15	0090-	78 4B 4B 78 78 4B 4B 78	\$5C53
0010-	7F 00 00 00 00 7F 00 00	\$5D95	0098-	70 16 16 70 70 16 16 70	\$783F
0018-	7F 40 40 40 40 47 44 44	\$F1DF	0DA0-	61 2D 2D 61 61 2D 2D 61	\$585F
0020-	44 44 44 44 44 44 44 44	\$51BF	0DA8-	43 5A 5A 43 43 5A 5A 43	\$ECAB
0028-	44 44 47 40 40 40 40 7F	\$CDC1	0DB0-	07 34 34 07 07 34 34 07	\$042B
0030-	00 00 7F 00 00 00 00 7F	\$8FA5	0DB8-	7F 7F 00 00 7F 7F 00 00	\$5C9B
0038-	11 11 71 01 01 01 01 7F	\$999A	0DC0-	7F 7F 00 00 7F 7F 00 00	\$842B
0040-	11 11 11 11 11 11 11 11	\$6112	0DC8-	7F 7F 00 00 7F 7F 00 00	\$DC9B
0048-	00 66 66 00 00 66 66 00	\$156E	0DD0-	7F 7F 00 00 7F 7F 00 00	\$042B
0050-	00 4C 4C 00 00 4C 4C 00	\$4D36	0DD8-	7F 7F 00 00 7F 7F 00 00	\$5C9B
0058-	00 19 19 00 00 19 19 00	\$3912	0DE0-	FF FF 00 00 FF FF 00 00	\$842B
0060-	00 33 33 00 00 33 33 00	\$6192	0DE8-	FF FF 00 00 FF FF 00 00	\$DC9B
0068-	7F 41 41 41 41 41 41 7F	\$C845	0DF0-	FF FF 00 00 FF FF 00 00	\$042B
0070-	00 00 08 1C 08 00 00 00	\$042B	0DF8-	FF FF 00 00 FF FF 00 00	\$5C9B
0078-	0F 69 69 0F 0F 69 69 0F	\$20F7			
0080-	1E 52 52 1E 1E 52 52 1E	\$0027			

### Get.Obj

BEG: 1700.17A2 END:

1700-	A4 0B A6 00 06 FE B9 00	\$A974
1708-	18 08 29 7F A6 FD E0 08	\$2A8E
1710-	B0 0D CA 4A CA 10 FC A6	\$DD0E
1718-	FE 06 00 A2 01 D0 02 A2	\$EF46
1720-	07 20 99 17 4A 05 08 B0	\$3145
1728-	11 A2 00 20 8D 17 A0 03	\$3164
1730-	A2 06 A9 00 20 01 F6 4C	\$5888
1738-	4F 17 A2 03 20 8D 17 A0	\$C8B4
1740-	03 A2 06 A9 00 20 01 F6	\$A324
1748-	28 08 30 03 A2 00 2C A2	\$4178
1750-	03 20 8D 17 A0 03 A2 0E	\$215D
1758-	A9 00 20 01 F6 20 9E 17	\$A329
1760-	18 A5 00 69 04 05 00 A5	\$290F
1768-	08 CA D0 B5 28 18 A5 01	\$C989
1770-	69 04 C5 02 90 0C F0 0A	\$E535
1778-	A6 03 06 01 C8 C4 0C D0	\$CB16
1780-	05 60 05 01 38 A5 00 E9	\$DC42
1788-	1C 05 00 D0 EF 20 EC F6	\$F810
1790-	A5 01 A0 00 A6 00 4C 11	\$8B1F
1798-	F4 06 09 04 0A 60 A6 09	\$D491
17A0-	A4 0A 60	\$1486

### QD.Editor.Util.Obj

BEG: 300.3A9 END:

0300-	02 00 06 00 0E 00 23 2D	\$3318	0358-	02 29 7F 99 00 18 C8 C4	\$70EB
0308-	35 36 3F 27 04 00 35 27	\$DAB8	0360-	0C D0 EB 60 A4 08 B9 00	\$08E6
0310-	00 A4 08 A9 00 99 FF 17	\$0383	0368-	18 49 7F 99 00 18 C8 C4	\$65E9
0318-	08 D0 FA 60 A4 08 B9 FF	\$3CC4	0370-	0C D0 F3 60 A4 08 B9 00	\$C15C
0320-	17 91 1A 08 D0 F8 60 A4	\$3FEF	0378-	18 30 04 09 00 30 02 29	\$CEF2
0328-	08 B1 1A 99 FF 17 08 D0	\$7942	0380-	7F 99 00 18 C8 C4 0C D0	\$827F
0330-	F8 60 A9 0E 0D 3F 03 A9	\$A007	0388-	ED 60 E8 A9 00 9D 04 70	\$B242
0338-	00 A2 08 A0 00 8D 00 0E	\$F775	0390-	60 A6 FB E0 08 F0 08 00	\$A05B
0340-	08 D0 FA CA F0 05 EE 3F	\$DDCF	0398-	03 4C A2 03 0A 0A 0A 0A	\$B595
0348-	03 10 F0 60 A4 08 B9 00	\$8D1C	03A0-	0A 0A 45 FC 05 FC 60 02	\$4C15
0350-	18 A6 19 F0 04 09 00 30	\$00F4	03A8-	00 06	\$D00E

### Make Tables Configuration 1

10	- \$BADD	110	- \$71CE	210	- \$EB59	260	- \$9ADA	310	- \$F984
20	- \$81CA	120	- \$09AE	220	- \$9C92	270	- \$0376	320	- \$80D1
30	- \$1920	130	- \$1B7C	230	- \$9660	280	- \$2DD2	330	- \$8D23
40	- \$5F32	140	- \$88C3	240	- \$D3F2	290	- \$CD1D	340	- \$A705
50	- \$44AB	150	- \$C555	250	- \$6927	300	- \$D98C	350	- \$DAE7
60	- \$F740	160	- \$8887						
70	- \$947A	170	- \$7665						
80	- \$24E0	180	- \$2E03						
90	- \$51B6	190	- \$DF05						
100	- \$6247	200	- \$CB1C						

Due to space limitations, the checksums for Design Plus, UFO Factory, Space Raid, and Scruncher will be in **HARDCORE #2**. The checksums for **Faster Shapes** are on the following page.

### Make Start Configuration 1

10	- \$9F4F	70	- \$47D5	130	- \$D210
20	- \$6BD2	80	- \$DF35	140	- \$9AF9
30	- \$5C5A	90	- \$E4AC	150	- \$5DD0
40	- \$F1A0	100	- \$51C4	160	- \$309A
50	- \$0E02	110	- \$5846	170	- \$B5EC
60	- \$19F8	120	- \$0757	180	- \$D9C1

# Advertising Index

Gary Brown..... 16,17  
 Bev R. Haight..... 26  
 Micro Analyst, Inc..... inside front  
 Ken Sherwood..... 8

# Copy II Plus Parm

continued from page 9

3.....3B=1, A=1, 4B=1,  
 4D=8, 50=1  
 (ERROR 6 OK)

MORE

# CHECKSUMS

## Faster Shapes Configuration 1

10 - \$105F	430 - \$6599	580 - \$80AF	720 - \$A468
20 - \$EE1B	440 - \$260F	590 - \$9C04	730 - \$8A30
30 - \$5C2B	450 - \$5806	600 - \$32CF	740 - \$5698
40 - \$7E49	460 - \$0956		750 - \$3A29
50 - \$72F8	470 - \$4CB5	610 - \$41E4	760 - \$3560
60 - \$091B	480 - \$0891	620 - \$9993	770 - \$6A4A
70 - \$3344	490 - \$8D73	630 - \$8201	780 - \$214A
80 - \$611C	500 - \$9287	640 - \$22E3	790 - \$82AD
90 - \$8E9D	510 - \$0DEF	650 - \$12ED	800 - \$FE34
100 - \$FFAE	520 - \$43F3	660 - \$5CCB	
110 - \$8475	530 - \$FA9D	670 - \$2706	810 - \$9EA9
120 - \$D36B	540 - \$4038	680 - \$3168	820 - \$9F0A
130 - \$FAB5	550 - \$087F	690 - \$D200	830 - \$99C5
140 - \$556B	560 - \$4F51	700 - \$9A4C	840 - \$8C19
150 - \$4CE0	570 - \$706A	710 - \$31C7	850 - \$E0D9

## A.L.Shapes 300.3BA

0300- A9 00 05 FD 05 FE 05 FC	\$D370
0300- A9 0A 05 FB A6 FD 0D 00	\$4695
0310- 60 C9 FF F0 70 C9 00 F0	\$57DE
0310- 28 C9 04 B0 18 20 91 03	\$E564
0320- E0 0C D0 0B A6 FE 9D 04	\$755F
0320- 70 A0 00 04 FC E6 FE 20	\$C474
0330- 67 03 4C 0C 03 A6 FB E0	\$F2FD
0330- 0C D0 0E 20 76 03 4C 0C	\$7BAE
0340- 03 A6 FB E0 0B F0 00 00	\$5826
0340- 2D 20 64 03 4C 0C 03 A6	\$4400
0350- FD E8 0D 00 60 C9 04 00	\$E9E6
0350- 1D 4C 49 03 A5 FC A6 FE	\$851B
0360- 9D 04 70 60 20 91 03 E6	\$9702
0360- FD E6 FB A6 FB E0 0D 00	\$2DAB
0370- 04 A2 0A 06 FB 60 20 5C	\$B16F
0370- 03 A9 00 05 FC E6 FE 20	\$6DFB
0380- 71 03 4C 0C 03 20 5C 03	\$960B
0380- F0 06 E8 A9 00 9D 04 70	\$8834
0390- 60 A6 FB E0 0B F0 00 00	\$8ACD
0390- 03 4C A2 03 0A 0A 0A 0A	\$8FE3
03A0- 0A 0A 45 FC 05 FC 60 02	\$5683
03A0- 00 06 00 0C 00 3C 0C 15	\$819E
03B0- D6 36 00 3C 0C 15 D6 17	\$4DDE
03B0- 0D 05 00	\$730C

210 - \$10D5
220 - \$41E5
230 - \$4DC1
240 - \$FE46
250 - \$26F7
260 - \$67DF
270 - \$91F0
280 - \$0004
290 - \$92E3
300 - \$3D65
310 - \$543E
320 - \$A730
330 - \$5032
340 - \$8FBC
350 - \$630B
360 - \$BE15
370 - \$CD22
380 - \$DABD
390 - \$EBCA
400 - \$CC64
410 - \$9107
420 - \$4436

## SPACE EGGS \* (SRS)

0.....9=0  
 2-6  
 11-1A

## SPACE INVADERS \* (UNK)

0-22.....10=96

## SPACE VIKINGS \* (SL)

0-22

## STARBLASTER \* (PDS)

0.....10=96, 9=0  
 7-20 BY 1.5....E=DF, F=AD, 10=DE

## STARBLAZER (BS)

Same as Choplifter

## THRESHOLD (SOL)

0-22  
 1-23 BY 22.....3B=1, A=1, 4B=1,  
 4D=8, 50=1  
 (ERROR 6 OK)

## TUBE WAY \* (DM)

0-22

## TYPING TUTOR \* (MIS)

Use Copy Disk from Main Menu

## WORD HANDLER \* (SVS)

Use Copy Disk from Main Menu

## VERSAFORM \* (AST)

0-22

## VISICALC (VCP)

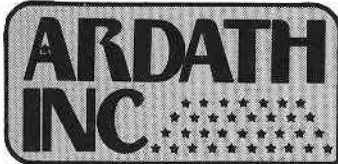
0-16

## VISICALC /// (AC)

0-22.....10=96, 24=96, D=1  
 NOTE: Don't use bit copy on Visidex,  
 Visischedule, Visiterm, or  
 Visitrend/Visiplot (VCP).  
 Use "Copy Disk" from Main Menu.

## VISIFILE (VCP)

0-22.....10=96, 34=1, 36=2A,  
 37=EB, 3E=2



**TOP QUALITY  
COST-EFFICIENT**  
Computer  
Products

# LOW PRICES

## Perform magic with the **Wizard BPO**

Parallel Printer Interface Card with a 32K buffer. Order before May 31, 1983 and you also receive a:

**FREE  
Graphics ROM**

Includes: graphics screen dumps, inverse graphics, double size picture, 90 degree rotation, center graphics, skips

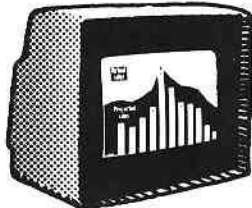
over perforations, left and right margins, variable line length, text screen dumps. Also operable with Pascal and CP/M.

**3 in 1 CARD  
ONLY**

**\$159.<sup>95</sup>**

## Amdeck Color-1 **13-inch Color Monitor**

Composite video signal ideal for Apple II, Atari and other compatible computers. 260(H) x 300(V) line resolution, built-in speaker and audio circuit, front mounted controls. UL/FCC approved.



**NOW ONLY  
\$309.<sup>00</sup>**

NEC 12" color only \$295.<sup>00</sup>

## Transtar-20 **12-inch Green Screen Monitor**

Quality workmanship, professional styling, superior brightness on a no-glare screen. One year limited warranty.

**NOW ONLY  
\$129.<sup>95</sup>**

Economy 12" green screen only \$85.<sup>00</sup>

## Pro/Writer 8510-A **Printer**

80 column, 120 CPS, 2K built-in buffer\*. Hi-Res graphics feature. One year limited warranty.

[\*optional]

**NOW ONLY  
\$385.<sup>00</sup>**

## Solo **Disk Drive**

A cost effective solution for Apple disk storage. Plugs into your Apple controller and boots protected software. 35 track, 143K storage. Full 120 day warranty. Now at our lowest price ever.

**NOW ONLY  
\$285.<sup>00</sup>**

Controller card \$95.<sup>00</sup>

## Franklin Ace-1000 **Home Computer**

They won't last long at these savings! 64K color, numeric keyboard, large power supply, joystick connection.

**NOW ONLY  
\$1010.<sup>00</sup>**



*I really  
want to be  
a CORE  
writer ...*

*... so I  
brought my  
programs over  
personally.*

**We are looking  
for programs, articles,  
columns, and reviews  
on these subjects:**

- CORE #2—Utilities
- CORE #3—Data Base  
(Business software)
- CORE #4—Games

**WRITERS, PROGRAMMERS, ARTISTS:**

Send a stamped, self-addressed envelope for our Writer's guide for more on rates and specs before sending your manuscript, software, or artwork.

**To order and for more information, send a SASE to:  
Ardath, Inc. — 15105 Union Avenue SW  
Tacoma WA 98498 Phone: 584-6177**

**CORE/HARDCORE COMPUTIST**  
P.O. Box 44549  
Tacoma, WA 98444

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

