89 COMPUTIST

Contents

WELCOME TO EAMON
THE COMPUTER FANTASY ROLE-PLAYING SYSTEM FOR THE APPLE.
CREATED BY DAVID BROWN.

Readers Data EXchange

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

What is a softkey, anyway?

Softkey is a term which we coined to describe a procedure that removes, or at least circumvents, any copy-protection on a particular disk. Once a softkey procedure has been performed, the resulting backup copy can usually be copied by the normal copy programs (for example: COPYA, on the DOS 3.3 System Master disk).

Commands and control keys

Commands which a reader is required to perform are set apart by being in boldface and on a separate line. The **return** key must be pressed at the end of every such command unless otherwise specified. Control characters are preceeded by "ctrl". An example of both is:

6 ctrl P

Type 6. Next, place one finger on the ctrl key and then press P. Don't forget to press the return key.

Other special combination keypresses include **ctrl reset** and **open-apple ctrl reset**. In the former, press and hold down the ctrl key then press the reset key. In the latter, press and hold down both ctrl and **open-apple** then press **reset**.

Software recommendations

The Starter Kit contains most of the programs that you need to "Get started". In addition, we recommend that you acquire the following:
• Applesoft program editor such as "Global Program Line Editor (GPLE)".
• Assembler such as "Merlin/Big Mac".
• Bit-copy program such as "Copy II Plus", "Locksmith" or "Essential Data Duplicator".
• Word-processor (such as AppleWorks).
• "COPYA", "FID" and "MUFFIN" from the DOS 3.3 System Master disk.

Super IOB and Controllers

This powerful deprotection utility (in the COMPUTIST Starter Kit) and its various Controllers are used in many softkeys. (It is also on each Super IOB Collection disk.)

Reset into the Monitor

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

Laser 128: Your ROM includes a forced jump to the monitor. Press **ctrl return reset**.

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

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

Apple //e, //c: Install a modified CD ROM on the computer's motherboard that changes the open-apple ctrl reset vector to point to the monitor. *(This will void an Apple //c warranty since you must open the case to install it.)*

Apple //gs: If you have the 2.x ROM, there is a hidden Classic Desk Accessory (CDA) that allows you to enter the monitor. In order to install the new CDA, you should enter the monitor (CALL -151) before running any protected programs and press **# return** . This will turn on two hidden CDAs, Memory Peeker and Visit Monitor. Thereafter press **openapple ctrl esc** to go to the Desk Accessories menu. Select Visit Monitor and there you are. Use **ctrl Y** to exit.

Recommended literature

• Apple II Reference Manual (or IIe, IIc, etc.)
• DOS 3.3 & ProDOS manual
• Beneath Apple DOS & Beneath Apple ProDOS, by Don Worth and Pieter Lechner, from Quality Software

Typing Applesoft programs

BASIC programs are printed in a format that is designed to minimize errors for readers who key in these programs. If you type:

10HOME:REMCLEAR SCREEN

The LIST will look like:

10 HOME : REM CLEAR SCREEN

Applesoft inserts spaces into a program listing before and after every command word or mathematical operator. These spaces don't pose a problem except when they are inside of quotes or after a DATA command. There are two types of spaces: those that have to be keyed and those that don't. Spaces that must be typed appear in COMPUTIST as special characters (◊). All other spaces are there for easier reading.

NOTE: If you want your checksums to match, only type spaces within quotes or after DATA statements if they are shown as (◊) characters. SAVE the program at periodic intervals using the name given in the article. All characters after a REM are not checked by the checksum program so typing them is optional.

Typing Hexdumps

Machine language programs are printed in COMPUTIST as hexdumps, sometimes also as source code.

Hexdumps are the shortest and easiest format to type in. You must first enter the monitor:

CALL -151

Key in the hexdump exactly as it appears in the magazine, ignoring the four-digit checksum ($ and four digits) at the end of each line. When finished, return to BASIC with:

3D0G

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

Typing Source Code

The source code is printed to help explain a program's operation. To enter it, you need an "Assembler". Most of the source code in older issues is in S-C Assembler format. If you use a different assembler, you will have to translate portions of the source code into something your assembler will understand.

Computing checksums

Checksums are 4-digit hexadecimal numbers which tell if you typed a program correctly and help you locate any errors. There are two types of checksums: one created by the CHECKBIN program (for machine language programs) and the other created by the CHECKSOFT program (for BASIC programs). Both are on the "Starter Kit".

If your checksums do not match the published checksums then the line where the first checksum differs is incorrect.

CHECKSOFT instructions: Install Checksoft (BRUN CHECKSOFT) then LOAD your program. Press & to get the checksums. Correct the program line where the checksums first differ.

CHECKBIN instructions: Enter the monitor (CALL -151), install Checkbin at some out of the way place (BRUN CHECKBIN, A$6000), and then LOAD your program. Get the checksums by typing the Starting address, a period and the Ending address of the file followed by a **ctrl Y** .

SSSS.EEEE ctrl Y

Correct the lines where the checksums differ.

Writing to the RDEX editor

RDEX (are-decks) stands for: Reader's Data EXchange. We print what you write. When you send in articles, softkeys, APTs, etc., you are submitting them for *free* publication in this magazine. RDEX does *not* purchase submissions nor do we verify data submitted by readers. If you discover any errors, please let us know so that we may inform our other readers.

Remember that your letters or parts of them may be used in RDEX even if not addressed to the RDEX editor. Correspondence that gets published may be edited for clarity, grammar and space requirements.

Because of the great number of letters we receive and the ephemeral and unpredictable appearance of our volunteer staff, any response to your queries will appear only in RDEX, so it would be more appropriate for you to present technical questions to the readers and ask for their responses which will then be placed in the Apple-RDEX.

How to get a free library disk

Whenever possible, send everything on Apple format (5.25" - DOS/ProDOS or 3.5" - ProDOS) or IBM format (3.5") disks. Other formats are acceptable but there may be some delay as we look for someone to translate it for us. *(If you use a 5.25" disk, when we print your letter, we will return your disk with the current library disk copied onto it.)* Use whatever text editor you like, but tell us which one. Put a label on the disk with your name (or pseudonym) and address (if you want to receive mail). Don't reformat any programs or include them in the text of your letter. Send Applesoft programs as normal Applesoft files and machine language programs as normal binary files. We have programs to convert them to the proper format for printing. If you are sending source code files, and you are not using the S-C Assembler, send them as normal text files.

When to include a printed letter

Don't include hardcopy (printout) unless:

a. You are writing about a bug or other printing error.

b. You are writing to ask for help.

c. You are answering another readers help request.

d. You are writing about your subscription or sending an order for back issues or software.

Bugs, requests for help and answers to requests for help are bumped to the head of the line and go in the very next issue. All other letters are printed in the order that we receive them.

Writing to get help

When writing to request help, be sure to include ALL relevent information. The more information you include, the easier it is to find a solution. There's an old saying that goes "A properly framed question includes 90% of the answer".

How to get mail

If you are interested in receiving mail from other readers, be sure that we have a current address. If you use a pen name and want to receive mail, we need to have your address. Our readers privacy is important, so we will not print your address unless you specifically say too.

How to write to RDEX authors

When writing to one of the RDEX authors. Write your letter and seal it in an envelope. Put your return address, the authors name (as it appears in RDEX) and *the correct postage* on the envelope. Put this envelope into another and send it to RDEX. We will put the correct address on your letter and mail it for you. Check to the right of the authors name to see if the author is writing from a foreign country and include the proper postage.

Help Line

These readers have volunteered their time to help you. Please call only within the given time frames (corrected for your time zone). No collect calls. (You can write anytime!)

Jack Nissel (Disk Protection, 7-10PM EST) (215) 365-8160

Marc Batchelor, 6025 Coker St., Cocoa, FL 32927

Rich Etarip, 824 William Charles Ct. #2, Green Bay, WI 54304-4018

The BBS
(Bulletin Board System)

Vince Andrews is the sysop for the Computist BBS. The NEW number is:

(206) 964-8730

If you already have a User ID# and password, sign-on using the User ID#. If you are a new user, it may take a day or so to validate your new ID# and password.

COMPUTIST

U.S. $3.75

Canada & Mexico $7

89

Table of Contents

WELCOME
TO
EAMON

THE COMPUTER FANTASY
ROLE-PLAYING SYSTEM
FOR THE APPLE.

CREATED
BY
DAVID BROWN

Features, Notes and such:

Advanced Playing Techniques:

Softkeys:

IBM Softkeys

Jeff Hurlburt   TX

The PRODUCT MONITOR

RATINGS

Superb ★★★★★
Excellent ★★★★
Very Good ★★★
Good ★★
Fair ★
Poor ☹
Bad ☠
Defective ✳

Eamon MCMXCII

Origins

In the mid-70's, if you had a 300-baud modem terminal, you could dial up most 'time share' systems and play the first computer adventure. "Adventure Land" ("The Adventure", etc.) was— in fact, still IS— a text-only game which challenges the player to solve puzzles, overcome hazards, and collect goodies. In a 'Text Adventure', you type in commands and get text descriptions of what happens as a result. For instance, GO NORTH might produce something like "You are in a rocky gulch with high cliffs to the east and west. The way continues north and south. An iron grate covers an opening in the ground." (Hint: OPEN GRATE!)

With the arrival of for-real home computers like the Apple II, text adventuring really 'took off'. We had the Scott Adams quests (e.g. "The Pyramid of Doom"), Infocom's "Zork" (I-III), "Enchanter", "Wishbringer", etc., assorted adventures from other vendors, and 'home brew' efforts published in magazines like Computist and SoftSide. The early '80's also saw the beginnings of Eamon adventuring.

Eamon?

"Eamon" is not a single adventure. It is a text adventure play and authoring system originated by Don Brown. His idea was to provide both a character creation/jumping-off point for many adventures AND the tools by which he and fellow enthusiasts would produce an unending stream of new scenarios. Don wrote "The Beginner's Cave" and a few other early adventures; but, most of the over 200 Eamon's are by numerous creators who joined in later.

Indeed, "Eamon" is people. It is chiefly due to efforts of the many "fellow enthusiasts" that Eamon adventuring continues to flourish. The Apple Avocation Alliance was largely responsible for organizing, numbering, and distributing the early Eamons. Besides writing several adventures, John Nelson helped collect submissions and expanded the capabilities of MAIN PGM (the main game program) through version 6.2.

Via the Eamon Adventurer's Guild (EAG), Tom Zuchowski deserves most of the credit for adapting to Apple II series improvements, assuring product quality, and, in general, maintaining order in modern Eamonville. Tom has also produced a number of scenarios; and, his version 7.0 of MAIN PGM is just one of many contributions to Eamon game development. Chuck Haight has been a solid Eamon supporter from its earliest days. Computist not only publishes Guild ads, but, as a distributor, supplies Eamon adventures and authoring materials (complete with fancy diskette labels!) at unbeatable prices.

The Guild does not pay for submissions and all Eamon materials are public domain. Anyone can copy them, place

WELCOME TO
EAMON

THE COMPUTER FANTASY ROLE-PLAYING SYSTEM FOR THE APPLE.

CREATED BY DAVID BROWN

them in user group libraries, or post them to bulletin boards. (Of course, club libraries and BB's are not nearly so likely to have the latest version of each adventure as is an official distributor like Computist.) Possibly, Eamon's best 'selling point' is that the many people who make it work do so, mainly, for fun.

"Eamon" is also a place, one dominated by an imposing structure known as The Main Hall. Here you can buy and sell arms and armor, bank your gold, and hire a wizard to teach you Eamon's four 'stock' spells (Blast, Heal, Power, Speed). Presiding over the Main Hall is a burly Irishman whose first question is always "What's yer name?"

To get started, you enter a character name. If the name is not found in the CHARACTERS file, the character is created— you pick gender and the program assigns Hardiness, Agility, and Charisma plus skills numbers for five Weapon types (Spear, Axe, Club, Bow/Gun, Sword) and Armor use. There is no strict limit on the number of

characters you and/or others who use the same "Eamon Master" (Main Hall) diskette can create; but, you can bring in only one hero per adventure. Since scenarios often allow some development of skills and attributes, players usually stick with a single character throughout their Eamon questing.

The Main Hall's most important attribute is that it leads to so many interesting places. Just select "Go on an adventure", pop in the diskette for your chosen quest, and you're off! As far as I know, no one has ever explained how leaving the Hall in this manner can place you at the mouth of a cave, before the gates of a city, on a Star Base launching platform, ... ; or, indeed, how returning from such disparate locales so often lands you "on the road leading to the Main Hall". Tom Zuchowski has speculated about "strange forces" brought to bear by the influence of orbiting suns. My guess is that magic, super science, and time portals are all involved.

The Adventures

When Computist published its first major Eamon article in 1983 there were only about 25 adventures; but, even then, quality varied wildly from scenario to scenario. Thanks to the EAG, serious bugs have, for the most part, been tracked down and eliminated; and new submissions are checked out more thoroughly. The Guild publishes a 10-page journal (four issues per year/ $7) where fixes for any newly discovered bugs appear. Along with news and reviews of late releases, each issue also lists authors and ratings for all 200+ adventures.

By far, the majority of Eamons are built around swords & sorcery themes running the gamut from whimsically Zorkian wanderings through Conanesque quests and Dragonlance-style crusades. Others may send you back to the time of dinosaurs, into the future to battle robots and mad scientists, drop you into a modern city to solve a crime, deposit you at the entrance to a haunted manse, or ... . There really is no limit to the possibilities. In terms of both numbers and variety, the challenge is practically endless.

Relatively few scenarios incorporate actual mazes (i.e. places where you are supposed to get lost); but, with up to 200 locations allowed via the standard game program, exploration is a goodly portion of every Eamon challenge. Most players rely upon simple line-and-box maps to keep track of treasures, puzzle-locked doorways, and other key features of the gamescape.

More so than in most other text adventures, critical achievements in Eamon tend to involve overcoming dangerous adversaries. In combats you trade blows to ATTACK MINOTAUR, BLAST GOLEM, etc. until all the bad guys are dispatched or flee. Most scenarios allow you to flee as well (one or more particularly aggressive monsters may follow) and a few permit three or four restorations should YOU be the one dispatched. Usually, getting killed ends the game and you must either create a new character or use a resurrection utility to restore the old one.

Some Infocom devotees may be turned off by Eamon's emphasis upon combat. Understandable; however, there are numerous adventure-enhancing side benefits. First, the need to overcome adversaries, makes character development and acquisition of better weapons 'mean something'. It greatly increases the stake you have in your character. Second, not all "monsters" are unfriendly; some will join you for the duration of a quest. (In Eamon, every character is, technically, a "monster". This includes critters, as well as guards, wizards, elf bowmen, princesses, etc..) Acquiring helpful recruits adds another dimension to the challenge; and, besides, it's more fun to go questing with stalwart companions at your side. Finally, as a Russian revolutionary once observed: "You can't make a soup without slicing monsters." Crusading against evil wizards, wicked queens, and other powerful foes naturally tends to be a rough business.

In text adventuring, "parsing" is equated with how well the game seems to understand your typed-in commands. A very good parser, such as the one you find in Infocom games, can make sense out of inputs that 'sound' the way people really speak. Since you don't have to fish around for just the right command, it is much easier to try out ideas. You play the game, not the game program.

Most Eamon adventures employ a two-word parser. To move, you enter commands like GO NORTH (or NORTH, or N). You may also OPEN CHEST, GET JEWEL, ATTACK VAMPIRE, DRINK POTION, READY LONG BOW, READ SIGN, ... Typically, an adventure recognizes 50-75 command words and the names of any

monsters and artifacts. (The reason for the variation in number of available commands is that many authors modify MAIN PGM, which is in BASIC, to add words.) Entering an invalid command automatically produces a list of commands the particular game understands.

The down side of Eamon's simple parser is that you must sometimes experiment to find the command that does what you know must be done. Scenario development is also impacted. To assure playability, Eamon authors have tended to avoid building-in complex puzzles. The up side is that, by the time you've gotten through your first two or three adventures, you have a good feel for how the commands are used and pretty well know what works and what doesn't. Then, too, many pre-Version 7.0 games employ author-originated mods which let the program understand synonyms, make sense of command abbreviations, and otherwise behave more intelligently. All of these are standard 7.0 features, as are improved capabilities for checking the status and inventory of any ally, trading items, and handling puzzle-oriented entries like PUT CUBE IN SLOT.

With so many games produced by so many authors, quality continues to vary a great deal. EAG nails bugs; but there's no question that some worlds are better 'filled-out' with richer prose and more interesting challenges. Playing time is another inconsistent variable. Most of the early adventures offer no GAME SAVE; yet, some may be good for barely an hour while others can run to four or five. The range is even larger for the newer games (including several multi-diskette quests) which often do, happily, offer a SAVE option.

Picking a 'good one' is a lot like shopping for something to read: you look for a juicy sounding title and a favorite author. EAG's reviews and ratings are a fair guide, too. (Each adventure is rated on a 1-10 scale; 10 is highest.) Predictably, however, many fun-to-play early adventures tend to be down-rated due to lower playing times. Most scenarios fit on a single diskette; and, at $1 each, you can afford to go for 'good prospects' AND experiment with 'unknowns'. A ten or twenty-dollar investment is guaranteed to deliver several evenings of adventuring challenge.

Eamoneering

Back in the early days of computer adventuring we collected Eamons more or less automatically from our local user group (HAAUG) library. Incredibly, I did not become a player until just the last couple of months, when curiosity about the adventures included on Computist Library Disks took hold. So far I've made it through thirty

or so scenarios. Fun (and more than 170 still ahead!); but, the real hook turned out to be the 'other side' of Eamon: programming and game development.

Among our vintage "AAA Eamon" adventures I came upon one not listed by EAG! (Kind of like finding an unpublished Brahms symphony.) "The City of Sorcerers" was playable; but, one notable bug and some undeveloped areas of the gamescape made it obvious the adventure was not quite finished. (Also, a dead give away, an early version of the dungeon editor was on the same diskette.)

I fixed the bug and began to fill-in some gaps and polish a few rough edges with the idea of sending "'City'" to EAG for registration under the original author's name. Well, one thing led to another and it soon became clear that 1. the version 6.x DUNGEON.EDIT I was using needed a user-friendliness transplant; 2. the version of MAIN PGM (which wasn't even named "MAIN PGM" in those days) imposed too many restrictions BECAUSE 3. I was making so many changes that a completely new adventure was taking shape! With the latter realization I copied the original, added the REMed bug fix, and mailed it to the Guild. "The City of Sorcerers" was where it belonged in the first place; and I was free to develop my own scenario.

First, I needed a better editor. NUEDIT began as a program intended to allow a player or designer to 'walk through' a gamescape (using normal N, S, etc. commands) and read descriptions without having to fight monsters. With a nice text editing routine lifted from DUNGEON.EDIT and using Beagle's Program Writer ($49.95) running on the IIgs, it wasn't long before the new editor was ready for serious work. (Writing NUEDIT was fun and a valuable refresher course in BASIC. In case you are thinking of learning a new programming language: Step One is to find something interesting to program.)

Anyway, armed with NUEDIT.6 and a more modern version of MAIN PGM, I was enmeshed in sculpting my new scenario when what should arrive in the mail but Tom Zuchowski's version 7.0 package! Complete with on-disk manual, this IS a major upgrade (still DOS 3.3-based) offering a larger gamescape, better speed, built-in GAME SAVE, and optional 80/40 column display. To which you can add a host of new artifact types (e.g. containers, keys, hidden doors, ...), better parsing, and other features guaranteed to enhance puzzle development. My 'old format' files were soon moved to 7.0. As for EAG's 7.0 editor, it is an improvement, but still too cum-

bersome. NUEDIT.6 became NUEDIT.7.

Along with the version 7.0 game creation materials, Tom included a beta version of a new "Multi-disk Supplement" (which, he notes, has never actually been used to create a game) and materials for authoring in 40 or 80 column ProDOS. Since my adventure is not seriously pushing 7.0's normal 1-diskette limit, and DOS 3.3 works on our II+ and IIgs, and our Eamons are all DOS 3.3 versions, I haven't tried the MDS or ProDOS stuff yet. (Speed would be a decisive ProDOS advantage except that most "DOS 3.3" users actually run some version of Beagle's Prontodos.)

Granted, ProDOS may be 'the wave of the future'; and, in fact, EAG now has ProDOS versions of roughly half the adventures. For now, 7.0 under '3.3 is, by far, the best tested, best documented format for Eamon authoring. Most of the newly released 'Guide for Eamon Adventure Designers and Players is devoted to version 7.0 play, utilities, and programming.

Note: The 21-page 'Guide' is available from Computist. It's the best introduction yet to Eamon adventuring AND it saves the bother of printing out the on-disk 7.0 manual. You'll find plenty of nuts & bolts detail covering Eamon text files, MAIN PGM, and LEADIN PROGRAM (a 'raw' game startup program where you can place introductory text; etc.). Evidently, the booklet is free with every order. There is one error— one which occurs in the on-diskette manual, too— on page 14, artifact Format 1 shows parm 5 as "Weapon type" and parm 6 as "Complexity"; it should say parm 5 is "Complexity" and parm 6 is "Weapon type".

The 'Guide' recommends that a game author have most of his/her scenario mapped-out BEFORE even booting up the editor. Perhaps experienced Eamon makers actually do it that way. Newcomers, however, can learn a great deal by starting with a rough idea, creating a few rooms with neat things to find and monsters to fight; and, then, giving the new adventure a try as a player. Changing (a copy of) an existing adventure is another good way to get started. Whatever your approach, once into creating an Eamon world you won't want to stop. Eamoneering is addictive stuff!

No PC?!

As far as I know, there is no PC or other non-Apple II equivalent for the Eamon Guild of Adventurers. (And, after speaking with Tom Zuchowski, my impression is that Hell may well freeze over before EAG changes its II-only policy.) Clearly, converting and test-playing 200+

adventures for another machine format is the sort of task one normally assigns to a monastery. It isn't likely to happen.

Happily, our PC-only readers can get into Eamon via a low-cost 'Eamon Kit'! According to flyers from some Apple-stuff vendors, an Apple II+ with mono monitor sells for $58. Add a disk drive and 80-column card (required for just a few adventures) and your total cost should fall somewhere around $200. (Comparable IIe and IIgs 'kits' sell for about $450.)

The Programs

NUEDIT.792 lets you edit a version 7.0 adventure or create one from scratch on a formatted diskette. (It will work in 40 or 80 column mode, whichever is active when the program is run.) When started, NUEDIT first asks you to insert the diskette to be edited. If NUEDIT detects an incompatible version it will tell you and then exit. If a compatible (7.x) version is detected NUEDIT takes you to the main menu ready to do editing.

If no adventure is found, NUEDIT will ask if you want to start a new adventure; if "YES", you will be asked for a name and the number of movement directions (6 for NSEWUD or 10 to permit NE, NW, etc. movement). NUEDIT then creates the standard Eamon TEXT files (7.0 default values are in parentheses):

EAMON.NAME: Name of the adventure, Number of directions (6 or 10), and Version number (7.0)

EAMON.DESC: descriptions of Rooms (records 1-200), Artifacts (201-400), Effects (401-600), Monsters (601-800). 'DESC Record length is 242; max length of each description is 238. Record 0 of this file has current counts of rooms, artifacts, effects, and monsters followed by record lengths for the 'ROOMS (72), 'MONSTERS (92), and 'ARTIFACTS (72) files.

EAMON.ROOMS: Each record has Name of a "room"/ location (max length= 38), room # Destination for each movement direction (for each direction, 0= none), and room Lumination (1= lighted, 0 means the player must have a lighted torch, etc. to see).

EAMON.ARTIFACTS: Each record has Name of the artifact (max length= 30), Value in gold, Type #, and Starting location plus four additional parms. The meaning of the latter depends upon artifact type. The 7.0 MAIN PGM recognizes 11 artifact types (e.g. Treasure, Weapon, Container, Door, etc.). NUEDIT includes formats/labels for all 11 and allows 4 additional user-defined types.

EAMON.MONSTERS: Each record has Name of the "monster" (max length= 30) followed by 11 parms to set hardiness, friendliness, starting room #,

etc.. Each record can describe just one monster or a group of same-kind monsters.

There is no separate 'EFFECTS' file because an effect is a text message (max length= 238) contained entirely in 'DESC.

At the start of new adventure creation, NUEDIT writes just one dummy record for each game element (room #1, artifact #1, effect #1, and monster #1).

Once at the main menu you press 'R' to view/edit Rooms, 'A' for Artifacts, etc.. When 'R', 'A', 'E', or 'M' is selected you are given the choice of entering the number of the room, etc. to view/edit OR pressing '+' to create one or more new records. (i.e. after selecting 'M', pressing '+' and entering "25" will create 25 dummy, ready-to-edit monster records.)

Most of the time you will be viewing/editing. For example, pressing 'R' and entering "73" will produce a display showing the room #, Room 73's name, the room's description, and the # of the room or door/gate you can reach by movement in each direction. Prompts at the bottom of the screen indicate that you may move to Room 72 or Room 74 using the arrow keys; OR, you may move to a destination room by pressing a directions key (like 'N', 'S', etc.). If the destination is a door/gate artifact, NUEDIT checks to see that the door and 'room beyond' both exist and automatically moves you to the room on the other side of the door. (*Note:* NUEDIT ignores any attempt to view or do anything to a room, artifact, etc. which does not yet exist.)

Entering 'C' and then, say, "142" copies the current room to Room 142. Entering '#' and, say, "55" is a quick way to hop to Room 55.

'Z' lets you edit displayed information. The cursor is set at the beginning of each item's field and you enter text or numbers as appropriate. (Pressing 'RETURN' with no changes retains the original and moves you to the next item.) Text editing is the same as in EAG's editor (CTRL-I inserts a space, etc.). One important change is that any text character at the length limit for a Name or Description is automatically pushed out when you do an insert. When done, NUEDIT lets you 'Save' the changed Room, 'Redo' editing without throwing away your work, or scrap the changes (the original display appears). 'SPC' exits to the main menu and CLOSEs all open files.

Viewing and editing Artifacts, Effects, and Monsters works much the same. When you have fixed choices, as in picking an artifact's Type, the current selection is shown (e.g. "Container") and the "<— —>" prompt reminds you to make your selection using the arrow keys.

Other main menu options let you change your adventure's name or get a Room Connections display. The Name Change option affects the name stored in EAMON.NAME and, possibly, the name of the your adventure's startup program. (The startup program is a user-made program or renamed LEADIN PROGRAM which has the same name as the adventure. To start play MAIN HALL reads EAMON.NAME to get the adventure's name and runs the startup program, which ends by running the main game program.) If a startup program has not yet been placed on the adventure diskette or it is still named LEADIN PROGRAM, etc., then Name Change affects only the name in EAMON.NAME.

The Connections option is very helpful for quickly checking your adventure's layout. It shows the selected room's destinations (or "exits") by direction ALONG WITH the exits for each destination. (Due to clutter, this option is not available in 10-direction mode.)

THE WONDERFUL WORLD OF EAMON and MAIN HALL programs have changed very little over the years. One enterprising author did produce an Ultima I-style map-graphics version of MAIN HALL. "Graphics MAIN HALL" seeks to patch a serious Eamon deficit, the absence of a cumulative Experience number, by letting you convert gold into attribute and skills boosts at various training centers and shops. GMH never 'caught on', partly because 1. It is too easy to over-enhance your character; and, 2. Having to steer a figure around on the screen maps slows down transactions. Mainly, though, the problem is format. A graphics interface doesn't 'belong' in the Hall. The TEXT Main Hall remains the standard, accepted version because that's the way Eamon players like it.

Modified versions of Eamon's two core programs plus the new EXP.UPDATE program are listed in this issue. They add several enhancements without disturbing Eamon's traditional 'look' and feel:

Experience- The new MAIN HALL automatically runs EXP.UPDATE each time you return alive from an adventure. Your character gains Experience based upon gold obtained, new/better weapons brought back, and any gains in attributes and skills awarded by the scenario. Experience translates into Level advancement, whereupon you can select an attribute or skill to increase. The same program also records and displays the name of each adventure you have experienced along with the number of times played.

Roster- in 'WORLD OF EAMON, a new option lets you display names of all characters in the CHARACTERS file. You can also obtain an Experience display for any character showing adventures played and number of times for each.

Restoration- It is no longer necessary to run a separate utility to restore a killed-off character. In 'WORLD OF EAMON you need only visit The Temple of Restoration.

Locker- in MAIN HALL, a character who meets the Level requirement can buy a space in the Adventurer's Lockers. This space holds up to eight weapons. You can visit your locker at any time when at the Main Hall to deposit, retrieve, or swap-in items.

Status- the MAIN HALL character status display shows Experience and Level along with more detailed information on Magic skills and Armor Expertise. Similarly, displays for the Banker, Weapon Shop, and Magician now include useful information (e.g. gold in your possession, spells you already know, etc.) to help speed-up and simplify transactions.

Since installing the above enhancements, my dad and I have played through something like fifty adventures including ancient "AAA Eamon" versions of the earliest scenarios, newer 6.2 games with and without GAME SAVE, and the latest 7.0 scenarios. That the mods work and are compatible with any CHARACTERS file and characters already in existence is not especially remarkable given that information is passed back and forth among Eamon programs via TEXT files. Inputs and outputs are clear-cut and easy to keep track of. The same applies to the two new files each active character acquires: a name.EXP file (for experience data) and a name.LOC file (for locker data).

Probably, the most important mod is the incorporation of a true Experience attribute and recording of a character's adventures. The purpose is to up the stakes and deliver a more tangible payoff for player successes than mere gold. Advancing in Level and getting to increment Hardiness, Agility, etc. certainly qualifies. So does the 'status symbol' value of a high Level ranking. Now, even 'weakie' scenarios are worth completing; and, of course, a character with a long record of adventure credits (i.e. a genuine, provable history) becomes more real and more fun to play.

So far, so good. However, as veteran Eamoners well know, there are some adventures which shower the character with enormous wealth, hefty attribute boosts, and double-digit skills gains. Other scenarios may be just as extreme in the other direction. Either way, Experience would be devalued.

Consequently, EXP.UPDATE includes a 'change filter' which suppresses scenario-induced shifts in attributes and skills to just a point or so. In the same spirit, beyond roughly 10000 gold, wealth increase has little effect on calculated Experience gain; and, to squelch the 'honey pot' effect, potential Experience gain is less each time you replay any adventure.

Experience is also the one change likely to evoke some suspicion of risky, bomb-prone trickery. There are just three 'tricks', all fairly straightforward. First, to derive an experience gain, EXP.UPDATE compares the character's status and possessions before the adventure (saved in name.PRE) with status and possessions after.

To remember the name of the adventure, MAIN HALL copies it into the DOS error message area before running the adventure. This does no harm; and EXP.UPDATE is able to retrieve the name when you return. (If the adventure is restarted from a SAVEd game, EXP.UPDATE detects an invalid name and asks you to reinsert the adventure diskette long enough for it to read the name.) Finally, MAIN HALL needs to know whether it was RUN by an adventure's MAIN PGM (if you are alive, it sends you to EXP.UPDATE) or by either 'WORLD OF EAMON or EXP.UPDATE (you stay in the Main Hall). It can tell by PEEKing at the command buffer. The "RUN MAIN HALL" commands in 'WORLD' and 'UPDATE' have extra spaces inserted.

Of the other changes, only the Locker offers benefits which may not be apparent, especially to new players. The old MAIN HALL ownership limit of four weapons looks okay UNTIL you consider that 1. Weapons break! It's nice to have a good backup; 2. You want to develop skills for all FIVE weapon types; 3. It is handy to have some high quality weapons to carry into an adventure for use by allies you may acquire; and 4. When you've got your Locker and it's full, you are certain to complain that the eight extra spaces are not nearly enough! (I settled for eight to avoid having to scroll the Lockers display.)

To install the enhanced Eamon play programs, first make a 'just in case' backup of your "Eamon Master" play diskette. Now, copy the new programs. The 'WORLD OF EAMON and MAIN HALL will replace versions already there. EXP.UPDATE uses 21 sectors; and, for the new .EXP and .LOC files which will be created, you should figure on 8 or so sectors per active character. That's it. Veteran Eamoners can look forward to a new dimension of play. As for newcomers, the Wonderful World of Eamon awaits!

The Wonderful World of Eamon

```
0 REM THE WONDERFUL WORLD
   OF EAMON
1 REM BOOT-UP PROGRAM FOR
   EAMON UNIVERSE
4 REM ORIG. RULES & CORE
   PROGRAMS BY DON BROWN
6 REM
7 REM ON-GOING
   DEVELOPMENT: TOM
   ZUCHOWSKI, E.A.G.
8 REM EAMON ADVENTURER'S
   GUILD 7625 HAWKHAVEN DR.
   CLEMMONS, NC 27012
9 REM (919) 766-7490
10 REM
11 REM PLAYER INPUTS
   CHARACTER NAME AND (IF
   EXISTING CHAR) GOES TO
   MAIN HALL.
12 REM IF NAME NOT IN
   'CHARACTERS' FILE THEN
   'NEW CARACTERS' PROGRAM
   IS RUN 1ST.
13 REM
14 REM 7/92 MODS BY JEFF
   HURLBUT (COMPUTIST
   MAGAZINE) 713-645-8680:
15 REM 1- PLAYER CAN
   RESTORE DEAD CHAR AT THE
   'TEMPLE'.
17 REM 2- PLAYER CAN VIEW
   RXPERIENCE RECORDS OF
   CHARACTERS VIA THE
   'ROSTER'.
24 REM
25 REM DRAGON PICTURE
   SHOWN
26 REM DURING PROGRAM BY
27 REM R.L. PHILLIPS &
   FRIENDS
28 REM OF ANN ARBOR, MICH.
29 REM THE BEAUTY IS
   GREATLY
30 REM ENJOYED.
31 REM
50 GOTO 100
55 PRINT DK$; "OPEN◊CHAR
   ACTERS,L150" : PRINT
   DK$; "READ◊CHARACTERS,
   R0" : INPUT NC: PRINT :
   RETURN
60 HOME : HTAB INT ((40 -
   LEN (Q$)) / 2): PRINT
   Q$: FOR I = 1 TO 39:
   PRINT "_" ;: NEXT I:
   PRINT : PRINT : RETURN
67 Q$ = "R◊O◊S◊T◊E◊R◊◊◊F
   ◊H◊E◊R◊O◊E◊S":GOSUB 60
70 GOSUB 55: FOR R = 1 TO
   NC: PRINT DK$; "READ◊CH
   ARACTERS,R" ;R: INPUT A$
72 PRINT "[" ; SPC( R <
   10);R; "]◊" ;: IF A$ =
   "" THEN INPUT "" ;A$:
   PRINT "-" ;
73 INPUT "" ;HD,AG,CH:
   PRINT A$;
75 HTAB 28: PRINT HD; "◊"
   ;AG; "◊" ;CH
77 NEXT R: PRINT DK$
   "CLOSE"
78 PRINT : INPUT "<CR>◊TO◊
   EXIT◊OR◊#◊TO◊VIEW>>◊"
   ;A$: IF A$ = "" THEN 140
79 QQ = ASC (A$): IF QQ <
   49 OR QQ > 57 THEN 140
80 QQ = VAL (A$): IF QQ >
   NC THEN 140
81 GOSUB 55: PRINT DK$;
   "READ◊CHARACTERS,R" ;QQ:
   INPUT NA$: PRINT DK$
   "CLOSE" : HOME : IF NA$
   = "" THEN PRINT : PRINT
   "FIRST,◊YOU◊MUST◊VISIT◊
   THE◊TEMPLE" : PRINT "TO◊
   RESTORE◊THIS◊CHARACTER."
   : GOSUB 580: GOTO 67
82 PRINT "THE◊ADVENTURES
   ◊OF◊" NA$: PRINT
83 Z$ = NA$ + ".EXP"
84 ONERR GOTO 95
85 PRINT DK$ "UNLOCK" Z$:
   PRINT DK$ "OPEN" Z$:
   PRINT DK$ "READ" Z$
86 FOR I = 0 TO 8: INPUT
   P(I): NEXT I
87 POKE 216,0
88 PRINT "LEVEL:◊" ;P(2);
   "◊◊◊◊EXPERIENCE:◊" ;P(1):
   PRINT : FOR I = 1 TO 39:
   PRINT "_" ;: NEXT I:
   PRINT
89 POKE 34,6: VTAB 7:QQ =
   0:J = 1
90 FOR I = J TO P(0):QQ =
   QQ + 1: INPUT E$,N%:
   PRINT E$; SPC( 31 - LEN
   (E$));N%; LEFT$ (
   "◊TIMES" ,6 - (N% = 1))
91 IF QQ > 14 THEN 93
92 NEXT I
93 QQ = 0: PRINT DK$
   "CLOSE" : PRINT
   "◊<KEY>◊" ;: GET Q$:
   HTAB 1: PRINT SPC( 9):
   PRINT :J = I + 1: IF J <
   = P(0) THEN PRINT DK$
   "OPEN" Z$: PRINT DK$
   "POSITION" Z$ ",R" ;I *
   2 + 9: PRINT DK$ "READ"
   Z$: GOTO 90
94 PRINT : POKE 34,0: GOTO
   98
95 PRINT "THIS◊CHARACTER
   ◊DOES◊NOT◊YET◊HAVE" :
   PRINT "AN◊ADVENTURING
   ◊RECORD.◊" :
97 GET Q$: PRINT
98 POKE 216,0: PRINT DK$
   "CLOSE" : GOTO 140
100 TEXT : HOME : NORMAL
105 PRINT : PRINT "EAMON
   ◊ADVENTURER'S◊GUILD" :
   PRINT "7625◊HAWKHAVEN
   ◊DR." : PRINT "CLEMMONS,
   ◊NC◊27012" : PRINT
   "(919)766-7490" : FOR I
   = 1 TO 250: NEXT : PRINT
110 DK$ = CHR$ (4): PRINT
   DK$; "BLOAD◊EAMON.PIC,
   A$4000" :ZZ = PEEK ( -
   16299) + PEEK ( - 16297)
   + PEEK ( - 16302) + PEEK
   ( - 16304)
120 X = RND ( - PEEK (78)
   - PEEK (79))
125 PRINT DK$ "OPEN◊SNAPPY
   ,L64" : PRINT DK$ "READ◊
   SNAPPY,R0" : INPUT NS:R
   = INT ( RND (1) * NS +
   1): PRINT DK$ "READ◊SNAP
   PY,R" ;R: INPUT CM$:C1$
   = LEFT$ (CM$,23):C2$ =
   MID$ (CM$,24): PRINT DK$
   "CLOSE◊SNAPPY"
127 ONERR GOTO 131
130 PRINT DK$; "OPEN◊CHAR
   ACTERS,L150" : PRINT
   DK$; "READ◊CHARACTERS
   ,R0" : INPUT NC:R = 1 /
   NC: PRINT DK$; "READ◊
   CHARACTERS,R1" : INPUT
   A$: PRINT DK$ "CLOSE" :
   GOTO 135
131 POKE 216,0: PRINT DK$
   "CLOSE" : PRINT DK$
   "DELETE◊CHARACTERS" :
   PRINT DK$ "OPEN◊CHAR
   ACTERS,L150" : PRINT DK$
   "WRITE◊CHARACTERS,R0" :
   PRINT 1: PRINT DK$
   "WRITE◊CHARACTERS,R1"
132 PRINT : PRINT
   "NUGUY92"
133 FOR I = 1 TO 12: PRINT
   "11" : NEXT I: PRINT "0"
   : PRINT "M" : PRINT
   "200" : PRINT "0" :
   PRINT "0"
134 FOR I = 1 TO 4: PRINT
   "NONE" : PRINT "0" :
   PRINT "0" : PRINT "0,0"
   : NEXT I: PRINT DK$
   "CLOSE"
135 POKE 216,0: FOR X = 1
   TO 500:I = RND (1): IF
   PEEK ( - 16384) < 128
   THEN NEXT X
137 X = PEEK ( - 16368):
   TEXT
140 HOME : PRINT "◊◊◊YOU
   ◊ARE◊IN◊THE◊OUTER◊CHAM
   BER◊OF◊THE" : PRINT
   "HALL◊OF◊THE◊GUILD◊OF◊
   FREE◊ADVENTURERS." :
   PRINT "MANY◊MEN◊AND◊
   WOMEN◊ARE◊GUZZLING◊BEER"
145 PRINT "AND◊THERE◊IS
   ◊LOUD◊SINGING◊AND◊LAUGH
   TER." : PRINT : PRINT
   "◊◊◊ON◊THE◊NORTH◊SIDE◊OF◊
   THE◊CHAMBER◊IS"
150 PRINT "A◊CUBBYHOLE◊
   WITH◊A◊DESK.◊OVER◊THE◊
   DESK" : PRINT "IS◊A◊SIGN
   ◊WHICH◊SAYS◊'REGISTER
   ◊HERE"
152 PRINT "OR◊VISIT◊TEMPLE
   ◊OF◊RESTORATION" : PRINT
   "OR◊VIEW◊ROSTER" : PRINT
   "OR◊ELSE!'" : PRINT
160 PRINT : PRINT "DO◊YOU◊
   GO◊OVER◊TO◊THE◊DESK,◊JOIN
   ◊THE" : PRINT "◊◊BEER◊
   DRINKERS,◊VISIT◊THE◊
   TEMPLE,◊OR" : PRINT
   "◊◊CHECK◊OUT◊THE◊ROSTER
   ◊OF◊HEROES?"
190 PRINT : PRINT "◊◊(HIT◊
   'D'◊'B'◊'T'◊OR◊'R'.)◊◊"
   ;: GET A$: PRINT : IF A$
   = "D" THEN 220
195 IF A$ = "T" THEN GOSUB
   500: PRINT : GOTO 140
197 IF A$ = "R" THEN 67
200 IF A$ < > "B" THEN 140
210 PRINT : PRINT
   "◊◊◊AS◊YOU◊GO◊OVER◊TO◊THE
   ◊MEN,◊YOU◊FEEL" : PRINT
   : PRINT "A◊SWORD◊BEING◊
   THRUST◊THROUGH◊YOUR
   ◊BACK"
215 PRINT : PRINT "AND◊YOU
   ◊HEAR◊SOMEONE◊SAY,◊'YOU
   ◊REALLY" : PRINT : PRINT
   "MUST◊LEARN◊TO◊FOLLOW◊
   DIRECTIONS!'" : GOTO 345
217 Q$ = "T◊H◊E◊◊M◊A◊I◊N◊◊
   D◊E◊S◊K" : GOSUB 60:
   RETURN
220 GOSUB 217: PRINT "YOU
   ◊ARE◊GREETED◊HERE◊BY◊A
   ◊BURLY" : PRINT : PRINT
   "IRISHMAN◊WHO◊LOOKS◊AT◊
   YOU◊WITH◊A◊SCOWL" :
   PRINT : PRINT "AND◊ASKS
   ◊YOU,◊'WHAT'S◊YOUR◊NAME
   ?'" : GOTO 235
230 PRINT
235 PRINT : PRINT "YOU◊
   GIVE◊HIM◊YOUR◊NAME." :
   PRINT : INPUT "◊>>>◊"
   ;NAME$
239 IF NAME$ = "" THEN
   GOSUB 217: VTAB 5: PRINT
   "'NO◊NAME,◊NO◊GAME.◊YE◊
   CANNA◊BE◊NO◊SIN'" :
   PRINT : PRINT "ABOUT◊THE◊HALL◊
   NAMELESS!◊COME◊BACK
   ◊WHEN" : PRINT : PRINT
   "YE◊BE◊READY◊TO◊MAKE◊A◊
   BIT◊O'◊SENSE.'" : GOSUB
   580: GOTO 140
240 PRINT : IF LEN (NAME$)
   < 2 OR LEFT$ (NAME$,1) <
   "A" OR LEFT$ (NAME$,1) >
   "Z" THEN GOSUB 400: GOTO
   230
250 IF LEN (NAME$) > 20
   THEN GOSUB 217: PRINT
   "'" NAME$ "?!!" : PRINT
   : PRINT "◊◊◊HE◊SCOWLS◊AT
   ◊YOU◊AND◊SAYS,◊'YER" :
   PRINT : PRINT "NAME'S◊
   TOO◊LONG◊FER◊ME◊BOOK.◊◊I
   ◊CANNA" : PRINT : PRINT
   "USE◊MORE◊THAN◊TWENTY
   ◊LETTERS." : PRINT :
   GOTO 230
260 PRINT "◊◊HE◊STARTS◊
   LOOKING◊THROUGH◊HIS◊
   BOOK," : PRINT : PRINT
```

```
   "MUTTERING◊ABOUT◊" ;C1$:
   PRINT : PRINT C2$: PRINT
270 GOSUB 55: FOR R = 1 TO
   NC: PRINT DK$; "READ◊
   CHARACTERS,R" ;R: INPUT
   A$: IF A$ = NAME$ THEN
   330
280 NEXT R: PRINT DK$:
   PRINT "◊◊◊HE◊EVENTUALLY◊
   LOOKS◊AT◊YOU◊AND◊SAYS,"
   : PRINT : PRINT "'YER
   ◊NAME'S◊NA◊IN◊HERE.◊HAVE
   ◊YE◊GIVEN" : PRINT :
   PRINT "IT◊TO◊ME◊ARIGHT
   ?'" : PRINT : PRINT
   "HOW◊DO◊YOU◊ANSWER◊(HIT◊
   'Y'◊OR◊'N')◊◊" ;
290 GET A$: PRINT A$:
   PRINT : IF A$ = "N" THEN
   GOSUB 400: GOTO 230
310 PRINT "◊◊◊HE◊HITS◊HIS◊
   FOREHEAD◊AND◊SAYS,◊'AH,"
   : PRINT : PRINT "YE◊MUST
   ◊BE◊NEW◊HERE!◊◊WELL,◊WAIT
   ◊JUST" : PRINT : PRINT
   "A◊MINUTE◊AND◊I'LL◊BRING◊
   SOMEONE◊OUT◊TO" : PRINT
   : PRINT "TAKE◊CARE◊OF
   ◊YE.'" : PRINT
320 GOSUB 350: PRINT DK$;
   "RUN◊NEW◊CHARACTERS
   ,S6,D1"
330 PRINT DK$: PRINT :
   PRINT "◊◊◊HE◊LOOKS◊UP◊
   AND◊SAYS,◊'AH,◊HERE◊YE"
   : PRINT : PRINT "BE!◊
   WELL,◊GO◊AND◊HAVE◊FUN◊
   IN◊THE◊HALL!'" : PRINT
340 GOSUB 350: PRINT DK$;
   "RUN◊◊◊MAIN◊HALL,S6,D1"
344 REM ///END: CLOSE
   ADDED
345 PRINT DK$ "CLOSE" :
   PRINT : END
350 PRINT DK$; "OPEN◊THE
   ◊ADVENTURER" : PRINT
   DK$; "WRITE◊THE◊ADVEN
   TURER" : PRINT NAME$:
   PRINT R: PRINT DK$;
   "CLOSE" : RETURN
399 REM ///NO NAME ROUTINE
400 GOSUB 217: VTAB 5
410 CNT = CNT + 1: IF CNT
   > 1 THEN 450
420 PRINT "◊HE◊PULLS◊OUT◊
   A◊SWORD◊AND◊BEGINS◊TO" :
   PRINT "SHARPEN◊IT,◊SAY
   ING◊'YE'D◊BEST◊BE◊
   GIVIN'" : PRINT "ME◊YER◊
   NAME◊LADDIE,◊IF◊YE◊KNOW
   ◊WOTS" : PRINT "GOOD◊FER
   ◊YE!!!"
430 RETURN
450 IF CNT = 2 THEN PRINT
   "I'VE◊'BOUT◊HAD◊ME◊FILL◊
   O'◊YER◊SICK" : PRINT
   "SENSA◊'UMOR!!" : PRINT
   "NOW◊GIMME◊YER◊NAME!!" :
   RETURN
460 IF CNT = 3 THEN PRINT
   "THE◊MAN◊CUTS◊ONE◊OF◊
   YOUR◊FINGERS◊OFF!!" :
   PRINT "HE◊THEN◊EATS◊IT!!
   !!" : PRINT "THEN◊HE◊
   SAYS◊'ARE◊YE◊READY◊T'
   ◊TALK◊NOW?" : RETURN
470 IF CNT < 13 THEN PRINT
   "THE◊MAN◊CUTS◊OFF◊
   ANOTHER◊FINGER!!!" :
   PRINT "HE◊EATS◊THIS◊ONE
   ◊TOO!!" : RETURN
480 PRINT "THE◊MAN◊STARTS◊
   SLOWLY,◊'WELL◊YE◊BE◊" :
   PRINT "OUTTA◊FINGERS!'"
   : PRINT "THE◊MAN◊THEN◊
   SPINS◊AROUND◊AND◊RUNS
   ◊YOU" : PRINT "THROUGH◊
   WITH◊A◊SPEED◊YOU◊HAVE
   ◊NEVER" : PRINT "SEEN◊
   BEFORE!◊(AND◊NEVER◊WILL
   ◊AGAIN.)" : GOTO 345
499 REM THE TEMPLE
500 Q$ = "THE◊TEMPLE◊OF◊
   RESTORATION" : GOSUB 60
505 PRINT "WHO◊IS◊TO◊BE
   ◊RESTORED?" : PRINT :
```

```
INPUT "◊>>◊" ;NA$
510 IF NA$ = "" OR NA$ =
    "◊" THEN 590
530 GOSUB 55: FOR R = 1 TO
    NC: PRINT DK$ "READ◊
    CHARACTERS,R" ;R: INPUT
    N$: INPUT M$
535 IF N$ = NA$ THEN PRINT
    : PRINT N$ "◊IS◊IN◊GOOD
    ◊SHAPE!" : PRINT DK$
    "CLOSE" : GOTO 580
540 IF N$ < > "" THEN 555
550 IF M$ = MID$ (NA$,2)
    THEN 560
555 NEXT R: GOTO 570
560 PRINT DK$ "WRITE◊
    CHARACTERS,R" ;R: PRINT
    NA$: PRINT DK$ "CLOSE" :
    PRINT : PRINT "OKAY,◊"
    ;NA$; "◊IS◊RESTORED!" :
    GOTO 580
570 PRINT "I◊CAN'T◊FIND
    ◊ANY◊DEAD◊" ;NA$: PRINT
    DK$ "CLOSE"
580 PRINT : PRINT "<KEY>◊"
    ;: GET Q$: PRINT Q$
590 RETURN
```

Checksums

0-$92CC	86-$AE33	235-$BC93
1-$DB99	87-$EF54	239-$A457
4-$8756	88-$3639	240-$E971
6-$9CE1	89-$DF9B	250-$E70C
7-$2AFA	90-$6282	260-$C0AF
8-$D0BB	91-$4C59	270-$D082
9-$4029	92-$0E8B	280-$E968
10-$17DA	93-$DDC7	290-$152F
11-$9535	94-$CA4C	310-$FEB2
12-$CB6C	95-$704C	320-$DE26
13-$03FA	97-$C0D1	330-$6EE6
14-$8039	98-$51C4	340-$9A60
15-$5232	100-$097E	344-$E076
17-$9D8D	105-$2E40	345-$97A8
24-$8316	110-$CD05	350-$E475
25-$8144	120-$F060	399-$0C1F
26-$86ED	127-$D003	400-$4CE3
27-$B6A6	130-$699C	410-$0CD7
28-$4CCD	131-$08EF	420-$837F
29-$5E0B	132-$7654	430-$9023
30-$1DDA	133-$00D8	450-$354C
31-$9535	134-$BE80	460-$65EA
50-$52D6	135-$7800	470-$7D68
55-$4B64	137-$4869	480-$D0E3
60-$4180	140-$351F	499-$9211
67-$2AED	145-$6214	500-$E351
70-$30BA	150-$B068	505-$B2FB
72-$BFE3	152-$D9E3	510-$653C
73-$5BC6	160-$14B1	530-$282A
75-$B0AA	190-$43AA	535-$6775
77-$DC8A	195-$1B5D	540-$DC11
78-$271B	197-$B401	550-$25CC
79-$5E26	200-$D3FF	555-$4618
80-$24AF	210-$AAF4	560-$5E6F
81-$ABB5	215-$1DB0	570-$D39B
82-$739E	217-$4AD1	580-$DCBD
83-$70FE	220-$8950	590-$2537
84-$82A7	230-$7A6F	
85-$7E2B		

Main Hall

```
0 REM
1 REM MAIN HALL (BUY STUFF
  & GO ON ADVENTURES)
  ORIG. BY DONALD BROWN
2 REM
3 REM REV. 9/92: ADDS
  LOCKER/ IMPROVED CHAR
  STATS DISPLAYS/
  EXPERIENCE ...
4 REM BY JEFF HURLBURT
  (COMPUTIST MAGAZINE)
  713-645-8680
5 REM 8000 SUB: Q9=1 MEANS
  HALL RUN FROM AN
  ADVENTURE
7 GOSUB 8000
8 PRINT CHR$ (4)
  "MAXFILES◊3"
9 TEXT : HOME : NORMAL :
  SPEED= 255
```

```
10 DK$ = CHR$ (4):U$ =
   "_____"
```

Editors note: "In line #10, U$ contains 39 underlines."

```
16 VTAB 5: PRINT "EAMON◊
   ADVENTURER'S◊GUILD" :
   PRINT "7625◊HAWKHAVEN
   ◊DR." : PRINT "CLEMMONS,
   ◊NC◊27012"
17 DIM SA%(5),WA%(6),WN$
   (6),WT%(6),WO%(6),WD%(6)
   ,WS%(6),LN$(8),LT%(8),
   LO%(8),LD%(8),LS%(8)
20 ONERR GOTO 7000
30 PRINT DK$; "OPEN◊THE◊
   ADVENTURER" : PRINT DK$;
   "READ◊THE◊ADVENTURER" :
   INPUT NAME$: INPUT REC:
   PRINT DK$; "CLOSE"
35 IF Q9 = 1 THEN PRINT
   DK$ "RUNEXP.UPDATE" :
   END
37 ONERR GOTO 49
38 Q$ = NAME$ + ".EXP" :
   PRINT DK$ "OPEN" Q$:
   PRINT DK$ "READ" Q$:
   INPUT XN,XE,XL: PRINT
   DK$ "CLOSE"
40 REM OLD ONERR GOTO100
42 Q$ = NAME$ + ".LOC" :
   PRINT DK$ "OPEN" Q$:
   PRINT DK$ "READ" Q$: FOR
   I = 1 TO 8: INPUT LN$(I)
   ,LT%(I),LO%(I),LD%(I)
   ,LS%(I): NEXT I:LQ = 1
49 PRINT DK$ "CLOSE"
50 PRINT DK$; "OPEN◊CHAR
   ACTERS,L150" : PRINT
   DK$; "READ◊CHARACTERS,R"
   ;REC: INPUT NAME$: INPUT
   HD,AG,CH: FOR X = 1 TO
   4: INPUT SA%(X): NEXT :
   FOR X = 1 TO 5: INPUT
   WA%(X): NEXT : INPUT AE
60 INPUT SEX$
70 INPUT GOLD, BANK: INPUT
   AC: FOR W = 1 TO 4:
   INPUT WNAME$(W),WTYPE%
   (W),WODDS%(W),WDICE%(W)
   ,WSIDES%(W): NEXT W:
   PRINT DK$; "CLOSE"
73 FOR I = 1 TO 4: IF
   WN$(I) = "" THEN WN$ =
   "NONE"
75 IF WN$(I) = "NONE" THEN
   WO%(I) = 0:WD%(I) =
   0:WS%(I) = 0
77 NEXT I
80 DEF FN P(X) = INT (X *
   RTIO + .5)
90 POKE 216,0
100 HOME :Q = FRE (0)
102 HTAB 9: PRINT "T◊H◊E◊
    ◊M◊A◊I◊N◊◊H◊A◊L◊L" : FOR
    I = 1 TO 39: PRINT "_"
    ;: NEXT I: PRINT : PRINT
105 PRINT "AS◊YOU◊WANDER◊
    ABOUT◊THE◊HALL,◊YOU" :
    PRINT "REALIZE◊YOU◊CAN
    ◊DO◊ONE◊OF◊SEVEN◊THINGS:"
    : PRINT : PRINT "◊1.◊◊GO
    ◊ON◊AN◊ADVENTURE." :
    PRINT : PRINT "◊2.◊◊VIS
    IT◊THE◊WEAPON◊SHOP◊FOR
    ◊ARMS."
110 PRINT : PRINT "◊3.◊
    ◊HIRE◊A◊WIZARD◊TO◊TEACH
    ◊YOU◊SPELLS." : PRINT :
    PRINT "◊4.◊FIND◊THE◊
    BANKER◊TO◊DEPOSIT◊OR◊" :
    PRINT "◊◊◊◊◊WITHDRAW
    ◊SOME◊GOLD." : PRINT :
    PRINT "◊5.◊◊EXAMINE◊
    YOUR◊ABILITIES."
115 PRINT : PRINT "◊6.◊◊GO
    ◊TO◊THE◊ADVENTURERS'◊
    LOCKERS." : PRINT :
    PRINT "◊7.◊◊TEMPORARILY
    ◊LEAVE◊THE◊UNIVERSE."
120 PRINT : PRINT : PRINT
    "(HIT◊THE◊KEY◊FOR◊YOUR
    ◊CHOICE,◊1-7)" ;
```

```
130 GET A$:CMD = VAL (A$):
    IF CMD < 1 OR CMD > 7
    THEN 130
140 PRINT CMD: PRINT : ON
    CMD GOTO 1000,2000,3000,
    4000,5000,5500,6000
200 PRINT NAME$
202 PRINT HD: PRINT AG:
    PRINT CH: FOR X = 1 TO
    4: PRINT SA%(X): NEXT :
    FOR X = 1 TO 5: PRINT
    WA%(X): NEXT : PRINT AE:
    PRINT SEX$
210 PRINT GOLD: PRINT
    BANK: PRINT AC: FOR W =
    1 TO 4: PRINT WNAME$(W):
    PRINT WTYPE%(W): PRINT
    WOODS%(W): PRINT
    WDICE%(W): PRINT
    WSIDES%(W): NEXT W
220 RETURN
300 IF LQ THEN Q$ = NAME$
    + ".LOC": PRINT DK$
    "OPEN" Q$: PRINT DK$
    "WRITE" Q$: FOR I = 1 TO
    8: PRINT LN$(I): PRINT
    LT%(I): PRINT LO%(I):
    PRINT LD%(I): PRINT
    LS%(I): NEXT I: PRINT
    DK$ "CLOSE"
310 RETURN
1000 HOME :Q = LEN (NAME$)
1003 PRINT DK$; "OPEN◊
     CHARACTERS,L150" : PRINT
     DK$; "WRITE◊CHARACTERS
     ,R" ;REC: PRINT : PRINT
     RIGHT$ (NAME$,Q - 1):
     GOSUB 202: PRINT DK$;
     "CLOSE"
1005 Q$ = NAME$ + ".PRE" :
     PRINT DK$ "OPEN" Q$:
     PRINT DK$ "WRITE" Q$:
     GOSUB 200: PRINT DK$
     "CLOSE" Q$
1007 GOSUB 300
1010 HOME : VTAB 5: PRINT
     "INSERT◊DISKETTE◊WITH◊
     ADVENTURE◊(OR◊KEEP" :
     PRINT : PRINT "THIS◊DISK
     ETTE◊FOR◊BEGINNERS◊CAVE
     ◊IN" : PRINT : PRINT
     "DISK◊DRIVE◊IN◊SLOT◊SIX,
     ◊DRIVE◊ONE◊THEN" : PRINT
     : PRINT "HIT◊'C'" :
     PRINT : PRINT : PRINT :
     PRINT "◊◊◊◊◊" ;: POKE -
     16368,0
1020 GET A$: IF A$ < > "C"
     THEN 1020
1030 PRINT
1040 PRINT DK$; "OPEN◊
     EAMON.NAME" : PRINT DK$;
     "READ◊EAMON.NAME" :
     INPUT ADNAME$: PRINT
     DK$; "CLOSE"
1045 GOSUB 8100
1047 HOME : FOR I = 1 TO
     12:J = 25 - I: VTAB J:
     PRINT U$: VTAB J: PRINT
     U$: NEXT I: FOR I = 12
     TO 1 STEP - 1:J = 25 -
     I: VTAB I: PRINT SPC(
     40): VTAB J: PRINT SPC(
     40): NEXT I
1050 VTAB 10: PRINT "YOU◊
     ARE◊ABOUT◊TO◊ADVENTURE
     ◊IN" : PRINT : INVERSE :
     PRINT ADNAME$: NORMAL :
     PRINT : PRINT
1060 PRINT DK$; "OPEN◊
     FRESH◊MEAT" : PRINT DK$;
     "WRITE◊FRESH◊MEAT" :
     PRINT REC: GOSUB 200:
     PRINT DK$; "CLOSE" :
     PRINT DK$; "RUN◊"
     ;ADNAME$
1920 FOR W2 = ZW TO
     4:WN$(W2) = WN$(W2 +
     1):WT%(W2) = WT%(W2 +
     1):WO%(W2) = WO%(W2 +
     1):WD%(W2) = WD%(W2 +
     1):WS%(W2) = WS%(W2 +
     1): NEXT W2:WN$(4) =
     "NONE" : RETURN
1940 HOME : HTAB 4: PRINT
     "I◊G◊M◊O◊◊S◊O◊L◊O◊◊C◊K◊E◊
```

```
R◊S,◊◊I◊N◊C."
1941 PRINT U$: GOSUB
     5060:W = 0: FOR I = 1 TO
     4:W = W + WN$(I) =
     "NONE" : NEXT I: PRINT :
     IF LQ < 1 THEN GOSUB
     5008: PRINT : GOTO 1944
1942 X = 0: PRINT
     "IN◊STORAGE----" : FOR I
     = 1 TO 8:X = X + LN$(I) =
     "NONE" :Q = ABS
     (LO%(I)): PRINT LN$(I);:
     HTAB (24 - (LO%(I) < 0)
     + (Q < 10) + (Q < 100))
     : PRINT LO%(I); "◊◊◊◊"
     ;LD%(I); "◊D◊" ;LS%(I):
     NEXT I: PRINT
1944 RETURN
1945 FOR I = 1 TO 4: IF Q$
     < > WN$(I) THEN NEXT I
1946 RETURN
1947 FOR I = 1 TO 8: IF Q$
     < > LN$(I) THEN NEXT I
1948 RETURN
1950 HOME : HTAB 10: PRINT
     "T◊H◊E◊◊B◊A◊N◊K◊E◊R" :
     PRINT U$: GOSUB 5008:
     PRINT : PRINT : RETURN
1970 HOME : HTAB 10: PRINT
     "T◊H◊E◊◊W◊I◊Z◊A◊R◊D" :
     PRINT U$: GOSUB 5008:
     GOSUB 5010: PRINT :
     PRINT : RETURN
1990 HOME : HTAB 9: PRINT
     "W◊E◊A◊P◊O◊N◊◊S◊H◊O◊P" :
     PRINT U$: GOSUB 5040:
     PRINT : GOSUB 5008:
     PRINT : RETURN
1998 PRINT : PRINT
     "<KEY>◊" ;: GET Q$:
     PRINT Q$: RETURN
1999 GOSUB 1998: PRINT :
     GOTO 100
2000 GOSUB 1990: GOTO 2005
2002 GOSUB 1998: PRINT
2003 GOSUB 1990: GOTO 2015
2005 PRINT "AS◊YOU◊ENTER◊
     THE◊WEAPON◊SHOP,◊MARCOS"
     : PRINT "CAVIELLI◊(THE◊
     OWNER)◊COMES◊FROM◊OUT◊
     OF" : PRINT "THE◊BACK◊
     ROOM◊AND◊SAYS,◊'WELL,
     ◊AS◊I"
2010 PRINT "LIVE◊AND◊
     BREATH,◊IF◊IT◊ISN'T◊MY◊
     OLD◊PAL" : PRINT NAME$
     "!" : PRINT
2015 PRINT "SO,◊YOU◊WANT
     ◊TO◊BUY◊A◊WEAPON,◊SELL" :
     PRINT "A◊WEAPON,◊OR◊GET
     ◊SOME◊BETTER◊ARMOUR?"
2017 PRINT : PRINT "(HIT
     ◊SPC◊TO◊EXIT◊OR◊B,◊S,◊
     A)◊" ;
2020 C2 = CH + INT (11 *
     RND (1) - 5):C2 = C2 +
     (C2 = 0):RTIO = 10 / C2
2030 GET A$: FOR TR = 1 TO
     3: IF A$ < >
     MID$ ( "BSA" ,TR,1) THEN
     NEXT TR: GOSUB 1990: ON
     1 + INT ( RND (1) * 1.9)
     + (GOLD < 300) GOTO
     2210,2289,2113
2040 PRINT : ON TR GOTO
     2050,2220,2490
2050 IF WN$(4) < > "NONE"
     THEN GOSUB 1990: PRINT
     "MARCOS◊SMILES◊AT◊YOU◊
     AND◊SAYS,◊'THATSA" :
     PRINT "GOOD,◊BUT◊FIRST
     ◊YOU◊GOTTA◊SELL◊ME◊A" :
     PRINT "WEAPON.◊YOU◊
     KNOW◊THE◊LAW--NO◊MORE
     ◊THAN" : PRINT "FOUR◊
     WEAPONS◊PER◊PERSON!'" :
     PRINT : GOTO 2002
2060 GOSUB 1990: PRINT
     "MARCOS◊SMILES◊AT◊YOU
     ◊AND◊SAYS,◊'GOOD." :
     PRINT "I◊GOTTA◊THE◊BEST.
     ◊YOU◊WANTA◊AXE,◊BOW," :
     PRINT "MACE,◊SPEAR,◊
     OR◊SWORD?" : PRINT :
     GOSUB 2650: IF A$ = "X"
     THEN 2003
```

```
2070 GOSUB 1990: PRINT
     "MARCOS◊SAYS,◊'WELL,◊I◊
     JUST◊HAPPEN◊TO" ;N$; "SO◊IN
     ,◊OF◊VARYING" : PRINT
     "QUALITY.◊I'VE◊GOT◊A◊
     VERY◊GOOD◊ONE◊FOR" :
     PRINT FN P(2 * BP); "◊GP
     ,◊A◊FAIR◊ONE◊FOR◊" ; FN
     P(BP); "◊GP,"
2080 PRINT "AND◊A◊KINDA◊
     SHABBY◊ONE◊FOR◊" ; FN
     P(BP / 2): PRINT
     "WHICH◊DO◊YOU◊WANT?" :
     PRINT : PRINT "(HIT◊THE
     ◊KEY,◊G◊F◊OR◊P)◊◊◊"
2090 GET A$: PRINT A$: FOR
     Q = 1 TO 3: IF A$ < >
     MID$ ( "GFP" ,Q,1) THEN
     NEXT : GOTO 2287
2100 P = BP + BP * (Q = 1)
     - BP / 2 * (Q = 3):P =
     FN P(P)
2110 IF P < = GOLD THEN
     2120
2111 GOSUB 1990: PRINT
     "MARCOS◊SHAKES◊A◊FINGER◊
     AT◊YOU◊AND◊SAYS," :
     PRINT "'YOU◊SHOULDN'T◊
     PLAY◊TRICKS◊ON◊AN◊OLD" :
     PRINT "FRIEND!" : PRINT
2113 PRINT "'COME◊BACK◊
     WHEN◊YOU◊GOTTA◊MORE" :
     PRINT "GOLD◊OR◊YOU◊WANT
     ◊SOMETHING◊YOU◊CAN" :
     PRINT "AFFORD.'◊MARCOS
     ◊WAVES◊AS◊YOU◊LEAVE." :
     GOTO 1999
2120 FOR W = 1 TO 4: IF
     WN$(W) = N$ THEN N$ = N$
     + "#" : GOTO 2120
2130 NEXT W: FOR W = 1 TO
     4: IF WN$(W) < > "NONE"
     THEN NEXT
2140 WN$(W) = N$:WT%(W) =
     TP:WO%(W) = 10 * (2 -
     Q):WD%(W) = DICE:WS%(W)
     = SIDES:GOLD = GOLD - P
2150 GOSUB 1990: PRINT
     "MARCOS◊HANDS◊YOU◊YOUR◊
     WEAPON◊AND◊TAKES" :
     PRINT "THE◊PRICE◊FROM◊
     YOU." : PRINT
2160 GOSUB 1998
2170 GOSUB 1990: PRINT
     "HE◊NOW◊ASKS◊YOU,◊'NOW◊
     HOW◊ABOUT◊SOME" : PRINT
     PRINT "(HIT◊Y◊OR◊N)◊◊◊" ;
2180 GET A$: PRINT A$: IF
     A$ = "N" THEN PRINT : ON
     1 + SH GOTO 2580,2210
2190 IF A$ = "Y" THEN 2490
2200 GOTO 2003
2210 PRINT "MARCOS◊SMILES◊
     AND◊SAYS,◊'COME◊BACK" :
     PRINT "AGAIN◊SOON!'◊AS
     ◊HE◊SHOOS◊YOU◊OUT◊OF◊HIS"
     : PRINT "SHOP." : PRINT
     : GOTO 1999
2220 GOSUB 1990: PRINT
     "MARCOS◊ASKS◊YOU,◊'IS◊
     THIS◊WEAPON◊YOU" : PRINT
     "WANT◊TO◊SELL◊ME◊A◊STAN
     DARD◊WEAPON◊LIKE" :
     PRINT "I◊SELL?◊◊(HIT◊'Y'
     ◊OR◊'N'◊◊◊"
2230 GET A$: IF A$ = "Y"
     THEN 2350
2240 IF A$ < > "N" THEN
     2003
2250 PRINT A$: PRINT :P =
     INT (50 / RTIO + .5):
     PRINT "MARCOS◊SAYS,◊'
     THEN◊I◊CAN◊ONLY◊GIVE◊YOU"
     : PRINT P; "◊GOLD◊PIECES
     ◊FOR◊IT,◊TAKE◊IT◊OR" :
     PRINT "LEAVE◊IT!'" :
     PRINT : PRINT "(HIT◊T
     OR◊L)◊◊◊" ;
2260 GET A$: PRINT A$: IF
     A$ < > "T" THEN 2003
2280 PRINT : GOSUB 1990:
     PRINT "MARCOS◊SAYS,◊'OK
     AY,◊WHAT'VE◊YOU◊GOT?" :
     PRINT : INPUT
```

```
     "(WEAPON◊NAME)◊" ;W$: IF
     W$ = "" THEN 2287
2283 FOR W = 1 TO 4: IF
     WN$(W) < > W$ THEN NEXT
     : PRINT : PRINT "MARCOS
     ◊FROWNS◊AT◊YOU◊AND◊SAYS,
     ◊'YOU" : PRINT "CAN'T◊
     SELL◊A◊WEAPON◊YOU◊DON'T
     ◊OWN!'" : GOTO 2002
2285 IF W$ < > "NONE" THEN
     2290
2287 GOSUB 1990: PRINT
     "MARCOS◊SAYS,◊'I◊THINK◊
     YOU◊ARE◊WASTING◊MY◊
     TIME!"
2289 PRINT : PRINT "MARCOS
     ◊SHOOS◊YOU◊OUT◊OF◊HIS
     ◊STORE." : GOTO 1999
2290 L = 5 * ( LEFT$(W$,5)
     = "SWORD" OR LEFT$(W$,5)
     = "SPEAR") + 4 * ( LEFT$
     (W$,4) = "MACE" ) + 3 *
     ( LEFT$ (W$,3) = "BOW"
     OR LEFT$ (W$,3) = "AXE"
     ): IF L = 0 THEN 2300
2292 GOSUB 1990:N$ = LEFT$
     (W$,L): PRINT "MARCOS◊
     FROWNS◊AT◊YOU◊AND◊SAYS,"
     : PRINT : PRINT "'THIS◊"
     N$ "◊IS◊A◊WEAPON◊LIKE◊I
     ◊SELL," : PRINT
     "REMEMBER!" : GOSUB
     1998: GOTO 2420
2300 GOLD = GOLD +
     P:ZW = W: GOSUB 1920
2303 GOSUB 1990: PRINT
     "MARCOS◊GIVES◊YOU◊YOUR
     ◊MONEY◊AND◊TAKES" :
     PRINT "YOUR◊WEAPON." :
     GOSUB 1998
2310 GOSUB 1990: PRINT
     "MARCOS◊ASKS◊YOU,◊'HOW◊
     ABOUT◊BUYING◊A" : PRINT
     "WEAPON?'" : PRINT :
     PRINT "(HIT◊Y◊OR◊N)◊◊◊" ;
2320 GET A$: PRINT A$: IF
     A$ = "Y" THEN PRINT :
     GOTO 2050
2340 PRINT : GOTO 2170
2350 GOSUB 1990: PRINT
     "MARCOS◊ASKS,◊'WELL,
     ◊WHAT◊WEAPON◊YOU" :
     PRINT "WANT◊A◊RETURN?" :
     PRINT : GOSUB 2650:
     PRINT : IF A$ = "X" THEN
     2003
2360 W2 = W: FOR W = 1 TO
     4: IF LEFT$ (WN$(W), LEN
     (N$)) < > N$ THEN NEXT :
     PRINT "MARCOS◊LAUGHS◊
     AND◊SAYS,◊'YOU◊NEVER" :
     PRINT "BOUGHT◊ONE◊FROM◊
     ME,◊REMEMBER?'" : PRINT
     : GOTO 2002
2370 BP = BP + BP * (WO%(T
     ) > 5)- BP/2 * (WO%(T )
     <- 5):P = INT (BP/RTIO +
     .5): IF P > BP / 4 THEN
     P = INT (BP / 4 + .5)
2380 GOSUB 1990: PRINT
     "MARCOS◊EXAMINES◊YOUR◊"
     N$ "◊AND" : PRINT "SAYS,
     ◊'LOOKS◊LIKE◊YOU'VE◊
     DINGED◊'ER" : PRINT
     "A◊BIT.◊I◊CAN◊OFFER◊" ;P;
     "◊GOLD◊PIECES.'" : PRINT
2381 PRINT "'TAKE◊IT◊OR
     ◊LEAVE◊IT.'" : PRINT :
     PRINT "(HIT◊T◊OR◊L)◊◊" ;
2390 GET A$: IF A$ < > "T"
     THEN 2003
2410 PRINT : GOTO 2300
2420 PRINT : FOR X = 1 TO
     4: IF MID$ ( "ABMS"
     ,X,1) = LEFT$ (W$,1)
     THEN ON X GOTO
     2440,2450,2460,2470
2430 NEXT X
2440 BP = 25: GOTO 2370
2450 BP = 40: GOTO 2370
2460 BP = 20: GOTO 2370
2470 IF MID$ (W$,2,1) =
     "P" THEN BP = 25: GOTO
     2370
2480 BP = 50: GOTO 2370
```

```
2490 A2 = INT (AC / 2):SH
     = AC - A2 * 2:B1 = 25 *
     (A2 = 1) + 60 * (A2 = 2)
     + 100 * (A2 = 3):TI =
     INT (B1 / RTIO + .5)
2495 GOSUB 1990: PRINT
     "MARCOS◊TAKES◊YOU◊TO◊THE◊
     ARMOUR◊SECTION" : PRINT
     "OF◊HIS◊SHOP◊AND◊SHOWS
     ◊YOU◊SUITS◊OF" : PRINT
     "LEATHER◊ARMOUR,◊CHAIN◊
     ARMOUR,◊AND◊PLATE." :
     GOSUB 1998: IF TI > B1 *
     2 THEN TI = B1 * 2
2500 GOSUB 1990: PRINT
     "HE◊SAYS,◊'I◊CAN◊PUT◊YOU
     ◊IN◊ANY◊OF◊THESE" : PRINT
     "VERY◊CHEAPLY.◊◊I◊NEED◊"
     ; FN P(100); "◊GOLD" :
     PRINT "PIECES◊FOR◊THE
     ◊LEATHER,◊" ; FN P(250);
     "◊FOR," : PRINT "THE◊CHA
     IN,◊AND◊" ; FN P(500);
     "◊FOR◊THE◊PLATE.": PRINT
2510 IF TI THEN PRINT
     "ALSO,◊I◊CAN◊GIVE◊YOU◊A◊
     TRADE-IN◊ON◊YOUR" :
     PRINT "OLD◊ARMOUR◊OF◊"
     ;TI; "◊GOLD◊PIECES." :
     PRINT
2520 PRINT "WELL,◊WHAT◊
     WILL◊IT◊BE?" : PRINT :
     PRINT "(HIT◊SPC◊OR◊L,
     ◊C,◊P)◊" ;
2530 GET A$: PRINT A$: FOR
     A = 1 TO 4: IF A$ < >
     MID$ ( "NLCP" ,A,1) THEN
     NEXT : ON 1 + (SH > 0)
     GOTO 2580,2003
2540 PRINT : IF A = 1 THEN
     2570
2550 P = FN P(100 * (A =
     2) + 250 * (A = 3) + 500
     * (A = 4)) - TI: IF P >
     GOLD THEN GOSUB 1990:
     PRINT "MARCOS◊FROWNS◊
     WHEN◊HE◊SEES◊THAT◊YOU◊DO"
     : PRINT "NOT◊HAVE◊ENOU
     GHT◊TO◊PAY◊FOR◊YOUR" :
     PRINT "ARMOUR◊AND◊SAYS,◊
     'I◊DON'T◊GIVE◊CREDIT!'"
     : PRINT : GOTO 2570
2560 GOLD = GOLD - P:A2 =
     A - 1:AC = A2 * 2 + SH:
     GOSUB 1990: PRINT "MARC
     OS◊TAKES◊YOUR◊OLD◊ARMOUR
     ◊AND◊YOUR" : PRINT "GOLD
     ◊AND◊HELPS◊YOU◊INTO◊YOUR
     ◊NEW" : PRINT "ARMOUR."
2570 GOSUB 1998: IF SH
     THEN 2003
2580 GOSUB 1990: PRINT
     "MARCOS◊SMILES◊AND◊SAYS,
     ◊'NOW◊HOW◊ABOUT" : PRINT
     "A◊SHIELD?◊I◊CAN◊LET◊YOU
     ◊HAVE◊ONE◊FOR" : PRINT
     "ONLY◊" ; FN P(50); "◊GO
     LD◊PIECES!'" : PRINT :
     PRINT "(HIT◊Y◊OR◊N)" ;
2590 GET A$: PRINT A$: IF
     A$ = "N" THEN PRINT : ON
     1 + INT ( RND (1) * 1.9)
     GOTO 2289,2210
2600 IF A$ < > "Y" THEN
     2003
2610 PRINT A$: PRINT : IF
     GOLD < FN P(50) THEN
     GOSUB 1990: PRINT "WHEN
     ◊HE◊SEES◊THAT◊YOU◊DO◊NOT
     ◊HAVE" : PRINT "ENOUGH◊
     GOLD◊TO◊BUY◊THE◊SHIELD,◊
     MARCOS" : PRINT "FROWNS
     ◊AND◊SAYS,◊'I'M◊SORRY,◊
     BUT◊I" : PRINT "DON'T◊
     GIVE◊CREDIT!'" : PRINT :
     GOTO 2113
2620 GOLD = GOLD - FN
     P(50):SH = 1:AC = A2 * 2
     + SH: GOSUB 1990: PRINT
     "MARCOS◊TAKES◊YOUR◊GOLD◊
     AND◊GIVES◊YOU" : PRINT
     "A◊SHIELD." : GOTO 2003
2640 END
2650 PRINT "(HIT◊THE◊KEY
     (S),◊A◊B◊M◊SP◊OR◊SW)◊◊"
     ;;
```

```
2660 GET A$: FOR W = 1 TO
     4: IF A$ < > MID$ (
     "ABMS" ,W,1) THEN NEXT
     W: GOTO 2755
2670 ON W GOTO 2680,2690
     ,2700,2710
2680 PRINT "AXE" :BP =
     25:N$ = "AXE" :TP =
     1:SIDE = 6: GOTO 2750
2690 PRINT "BOW" :BP =
     40:N$ = "BOW" :TP =
     2:SIDE = 6: GOTO 2750
2700 PRINT "MACE" :BP =
     20:N$ = "MACE" :TP =
     3:SIDE = 4: GOTO 2750
2710 PRINT A$;:
2720 GET A$: IF A$ = "P"
     THEN PRINT "PEAR" :BP =
     25:TP = 4:SIDE = 5:N$ =
     "SPEAR" : GOTO 2750
2730 IF A$ < > "W" THEN
     2755
2740 PRINT "WORD" :BP =
     50:N$ = "SWORD" :TP =
     5:SIDE = 8
2750 DICE = 1: RETURN
2755 A$ = "X" : PRINT A$:
     RETURN
2760 END
3000 GOSUB 1970: GOTO 3005
3002 PRINT : GOSUB 1998
3003 GOSUB 1970: GOTO 3015
3005 PRINT "AFTER◊A◊FEW◊
     MINUTES◊DILIGENT◊SEARCH
     ING," : PRINT "YOU◊FIND◊
     HOKAS◊TOKAS,◊THE◊OLD◊
     MAGE.◊HE" : PRINT "LOOKS
     ◊AT◊YOU◊AND◊SAYS,◊'SO◊YOU
     ◊WANT◊OLD" : PRINT "TOK
     EY◊TO◊TEACH◊YOU◊SOME◊
     MAGIC,◊HEH◊HEH?"
3010 PRINT "WELL,◊IT'LL
     ◊COST◊YOU.◊" ;
3015 PRINT "TODAY◊MY◊FEES
     ◊ARE:" :C2 = CH + INT
     (11 * RND (1) - 5):C2 =
     C2 + (C2 = 0):RTIO = 10
     / C2
3020 PRINT "◊◊◊BLAST◊◊" ;
     FN P(3000); "GP" : PRINT
     "◊◊◊HEAL◊◊◊" ; FN P(1000)
     ; "GP" : PRINT "◊◊◊SPEED
     ◊◊" ; FN P(5000); "GP" :
     PRINT "◊◊◊POWER◊◊" ; FN
     P(100); "GP" : PRINT
3030 PRINT "WELL,◊WHICH◊
     WILL◊IT◊BE?" : PRINT :
     PRINT "(SPC◊TO◊LEAVE◊OR◊
     PICK◊B,◊H,◊S,◊P)◊" ;
3040 GET A$: FOR SP = 1 TO
     4: IF A$ < > MID$ (
     "BHSP" ,SP,1) THEN NEXT
     : PRINT : GOTO 3200
3050 ON SP GOTO 3060,3070
     ,3080,3090
3060 A$ = "BLAST" : PRINT
     A$:P = FN P(3000): GOTO
     3100
3070 A$ = "HEAL" : PRINT
     A$:P = FN P(1000): GOTO
     3100
3080 A$ = "SPEED" : PRINT
     A$:P = FN P(5000): GOTO
     3100
3090 A$ = "POWER" : PRINT
     A$:P = FN P(100)
3100 PRINT : IF P > GOLD
     THEN GOSUB 1970: PRINT
     "WHEN◊HOKAS◊SEES◊THAT◊
     YOU◊DON'T◊HAVE" : PRINT
     "ENOUGH◊TO◊PAY◊HIM,◊HE◊
     STALKS◊TO◊THE" : PRINT
     "BAR,◊MUTTERING◊ABOUT◊
     YOUNGSTERS◊WHO" : PRINT
     "SHOULD◊BE◊TURNED◊INTO◊
     FROGS." : GOTO 1999
3110 IF SA%(SP) < = 0 THEN
     3120
3113 GOSUB 1970: PRINT
     "HOKAS◊SAYS,◊'I◊OUGHT◊
     TO◊TAKE◊YOUR" : PRINT
     "GOLD◊ANYWAY,◊BUT◊HAVE
     N'T◊YOU◊FORGOTTEN" :
     PRINT "SOMETHING?◊I◊
     ALREADY◊TAUGHT◊YOU◊" A$
     "!'"
```

```
3114 IF CH > RND (1) * 28
     THEN 3002
3115 PRINT : PRINT "SHAK
     ING◊HIS◊HEAD◊SADLY,◊HE◊
     RETURNS◊TO" : PRINT
     "THE◊BAR." : GOTO 1999
3120 GOLD = GOLD -
     P:SA%(SP) = INT (50 *
     RND (1) + 26): GOSUB
     1970: PRINT "HOKAS◊TEACH
     ES◊YOU◊YOUR◊SPELL,◊TAKE
     S" : PRINT "HIS◊FEE,◊AND
     ◊RETURNS◊TO◊HIS◊STOOL◊
     ON" : PRINT "THE◊BAR."
3125 PRINT "AS◊YOU◊WALK◊
     AWAY◊YOU◊HEAR◊HIM" :
     PRINT "ORDER◊A◊DOUBLE
     ◊DRAGON◊BOMB." : GOTO
     1999
3200 GOSUB 1970: PRINT
     "AS◊YOU◊LEAVE,◊YOU◊HEAR◊
     HOKAS◊MUTTERING" : PRINT
     "ABOUT◊CHEAPSKATE◊ADVENT
     URERS◊ALWAYS" : PRINT
     "WANTING◊SOMETHING◊FOR
     ◊NOTHING." : GOTO 1999
4000 GOSUB 1950: GOTO 4005
4002 PRINT : GOSUB 1998
4003 GOSUB 1950: GOTO 4015
4005 PRINT "YOU◊HAVE◊NO◊TR
     OUBLE◊SPOTTING◊SHYLOCK"
     : PRINT "MCFENNEY,◊THE◊
     LOCAL◊BANKER,◊DUE◊TO◊HIS
     " :PRINT "LARGE◊BELLY."
4007 PRINT : PRINT "YOU◊
     ATTRACT◊HIS◊ATTENTION
     ,◊AND" : PRINT "HE◊COM
     ES◊OVER◊TO◊YOU◊AND◊SAYS:"
     : PRINT
4010 PRINT NAME$ ",◊MY◊DE
     AR◊" ; MID$ ( "GIRLBOY"
     ,(SEX$ = "M" ) * 4 +
     1,4); "," : PRINT "THIS
     ◊IS◊A◊GENUINE◊PLEASURE!"
4015 PRINT "DO◊YOU◊WISH◊TO
     ◊MAKE◊A◊DEPOSIT(!)◊OR,"
     : PRINT "PERHAPS,◊A◊TINY
     ◊WITHDRAWAL?'◊(SPC/D/
     W)◊"
4030 GET A$: PRINT A$: IF
     A$ = "D" THEN PRINT :
     GOTO 4090
4040 IF A$ < > "W" THEN
     100
4045 IF BANK < 1 THEN
     PRINT "HA!◊GOOD◊TRY,◊"
     NAME$ "." : PRINT
     "UNFORTUNATELY,◊YOUR
     ◊ACCOUNT" : PRINT
     "BALANCE◊IS◊ZERO!" :
     GOTO 4002
4050 PRINT : PRINT
     "SHYLOCK◊SAYS,◊'WELL,◊
     YOU◊HAVE◊" ;BANK: PRINT
     "GOLD◊PIECES◊STORED◊
     WITH◊ME.◊HOW◊MANY◊DO" :
     INPUT "YOU◊WANT◊TO◊
     TAKE◊BACK?'◊>>◊" ;A$
4053 IF A$ = "" THEN 4003
4055 MNY = ABS ( INT ( VAL
     (A$))): IF MNY < = BANK
     THEN 4080
4070 GOSUB 1950: PRINT
     MNY; "!!!!◊(GASP)" :
     PRINT : PRINT
     "THE◊BANKER◊THROWS◊YOU
     ◊A◊TERRIBLE◊GLANCE" :
     PRINT "AND◊SAYS,◊'THAT'S
     ◊MORE◊THAN◊YOU'VE◊GOT!"
4075 IF CH > 28 * RND (1)
     THEN 4002
4078 PRINT "YOU◊KNOW◊I◊
     DON'T◊MAKE◊LOANS◊TO◊
     ◊YOUR" : PRINT "KIND!'◊◊
     WITH◊THAT◊HE◊LOSES◊HIM
     SELF" : PRINT "IN◊THE◊
     CROWD." : GOTO 1999
4080 GOLD = GOLD +
     MNY:BANK = BANK - MNY:
     GOSUB 1950: PRINT "THE◊
     BANKER◊HANDS◊YOU◊YOUR◊
     GOLD◊AND" : PRINT "SAYS,
     ◊'THAT◊LEAVES◊YOU◊WITH◊"
     ;BANK: PRINT "PIECES◊IN◊
     MY◊CARE.'◊◊HE◊SHAKES
     ◊YOUR" : PRINT

     "HAND◊AND◊WALKS◊AWAY." :
     GOTO 1999
4090 GOSUB 1950: PRINT
     "SHYLOCK◊GETS◊A◊WIDE◊
     GRIN◊ON◊HIS◊FACE" :
     PRINT "AND◊SAYS,◊'EXCEL
     LENT!◊HOW◊MUCH◊DO◊YOU" :
     INPUT "WANT◊TO◊DEPOSIT?
     '◊>>◊" ;A$
4095 IF A$ = "" THEN 4003
4097 MNY = ABS ( INT ( VAL
     (A$))): IF MNY < = GOLD
     THEN 4120
4110 GOSUB 1950: PRINT
     "THE◊BANKER◊IS◊VERY◊
     PLEASED◊WHEN◊YOU" :
     PRINT "TELL◊HIM◊THE◊SUM,
     ◊BUT◊WHEN◊HE◊DISCOVERS"
     : PRINT "THAT◊YOU◊DON'T◊
     HAVE◊THAT◊MUCH◊ON◊YOU,"
4112 IF CH > 28 * RND (1)
     OR GOLD > 2999 THEN
     PRINT "HE◊HELPFULLY◊
     SUGGESTS◊YOU◊SETTLE◊FOR"
     : PRINT "A◊'MODEST'◊DEP
     OSIT◊OF" ;GOLD; "." :
     GOTO 4002
4113 PRINT "HE◊WALKS◊AWAY,
     ◊SHOUTING◊ABOUT◊FOOLS" :
     PRINT "WHO◊PLAY◊TRICKS◊
     ON◊'KNIDLY◊BANKERS'." :
     GOTO 1999
4120 GOLD = GOLD - MNY:
     BANK = BANK + MNY: GOSUB
     1950: PRINT "SHYLOCK◊TAK
     ES◊YOUR◊MONEY,◊PUTS◊IT
     ◊IN" : PRINT "HIS◊BAG,
     ◊LISTENS◊TO◊IT◊JINGLE,
     ◊THEN" : PRINT "THANKS
     ◊YOU◊AND◊WALKS◊AWAY." :
     GOTO 1999
5000 GOSUB 5004: GOSUB
     5010: GOSUB 5030
5002 VTAB 23: POKE -
     16368,0: PRINT "(HIT◊ANY
     ◊KEY◊TO◊CONTINUE)◊◊" ;:
     GET A$: PRINT : GOTO 100
5004 Q$ = "MIGHTY◊" : IF
     SEX$ = "F" THEN Q$ =
     "FAIR◊"
5005 HOME :Q$ = "THE◊" +
     Q$ + NAME$:I = INT ((32
     - LEN (Q$)) / 2): HTAB
     I: PRINT Q$; SPC( I);
     "LVL◊" ;XL: FOR I = 1 TO
     39: PRINT "_" ;: NEXT I:
     PRINT
5007 PRINT "EXPERIENCE:◊"
     ;XE;: HTAB 20 + (XE >
     99999): PRINT "ADVENT
     URES:◊" ;XN: PRINT "ATTR
     IBUTES:◊" ;"◊◊AG="
     ;HD; "◊◊AG=" ;AG; "◊◊CH="
     ;CH: PRINT
5008 PRINT "GOLD◊IN◊HAND="
     ;GOLD; ",◊BANK=" ;BANK:
     PRINT
5009 RETURN
5010 PRINT "MAGIC◊SKILLS:
     ◊" ;:SM = 0
5015 FOR S = 1 TO 4: IF
     SA%(S) THEN PRINT MID$ (
     "BLAST◊HEAL◊SPEED◊POWER"
     ,(S - 1) * 5 + 1,5); "◊"
     ;:SM = 1
5017 NEXT S: IF NOT SM
     THEN PRINT "◊ZILCH!" :
     PRINT : GOTO 5025
5020 PRINT : PRINT SPC(
     12);: FOR S = 1 TO 4: IF
     SA%(S) THEN PRINT SPC(
     3);SA%(S); "%" ;
5022 NEXT S: PRINT : PRINT
5025 RETURN
5030 PRINT "YOUR◊WEAPON◊
     ABILITIES◊ARE--" : PRINT
     "◊◊AXE◊◊◊BOW◊◊◊CLUB◊◊
     SPEAR◊SWORD"
5035 FOR I = 1 TO 5: PRINT
     SPC(1+(I = 3) - (WA%(I) <
     0)+( ABS (WA%(I)) < 10)
     + ( ABS (WA%(I)) < 100))
     ;WA%(I); "%" ;: NEXT I:
     PRINT : PRINT
5040 A2 = INT (AC / 2):Q$
     = MID$ ( "SKIN◊◊◊LEATHER
     ◊CHAIN◊◊PLATE◊◊" ,A2 * 7

     + 1,7): FOR I = 1 TO 7:
     IF MID$ (Q$,I,1) < > "◊"
     THEN NEXT I
5043 Q$ = LEFT$ (Q$,I - 1)
5045 PRINT "ARMOUR:◊" Q$;:
     IF AC > A2 * 2 THEN SH =
     1: PRINT "◊&◊SHIELD" ;
5050 PRINT TAB( 26); "AE="
     ;AE; "%◊AC=" ;AC: PRINT
5060 PRINT "WEAPONS"; TAB(
     23); "COMPLEX◊DAMAGE"
5080 FOR I = 1 TO 4:Q =
     ABS (WO%(I)): PRINT
     WN$(I);: HTAB (24 -
     (WO%(I) < 0) + (Q < 10)
     + (Q < 100)): PRINT
     WO%(I); "%◊◊◊" ;WD%(I);
     "◊D◊" ;WS%(I): NEXT I
5095 RETURN
5499 REM //LOCKERS
5500 GOSUB 1940: GOTO 5505
5502 GOSUB 1998
5503 GOSUB 1940: GOTO 5515
5505 PRINT "THE◊LOCKER◊
     ATTENDANT◊GREETS◊YOU:◊
     'GOOD" : PRINT "AFTER
     NOON,◊" NAME$ "!'" : IF
     LQ THEN 5514
5506 IF XL > 6 AND GOLD >
     6999 THEN 5511
5507 PRINT : PRINT "'COME
     ◊BACK◊AS◊SOON◊AS◊YOU"
5508 IF XL > 6 THEN 5510
5509 PRINT "REACH◊LEVEL◊7.
     ◊REMEMBER,◊THE◊PRICE◊IS"
     : PRINT "JUST◊7000◊GOLD◊
     FOR◊A◊ROOMY◊LOCKER!'" :
     GOTO 1999
5510 IF GOLD < 7000 THEN
     PRINT "HAVE◊THE◊7000◊
     GOLD◊TO◊PURCHASE" :
     PRINT"YOUR◊OWN◊LOCKER.'"
     : GOTO 1999
5511 PRINT : PRINT "ARE◊
     YOU◊READY◊TO◊BUY◊A◊
     SPACE?◊(Y/N)◊" ;: GET
     Q$: PRINT Q$: IF Q$ < >
     "Y" THEN 1999
5513 PRINT : PRINT
     "'EXCELLENT!'" ;:GOLD =
     GOLD - 7000: FOR I = 1
     TO 8:LN$(I) = "NONE" :
     NEXT I:LQ = 1: PRINT
     "YOUR◊LOCKER◊IS◊READY.'"
5514 PRINT "YOU◊SMILE,◊
     PICK◊UP◊YOUR◊KEY," :
     PRINT "AND◊GO◊TO◊YOUR◊
     LOCKER." : GOTO 5502
5515 PRINT "DO◊YOU◊STORE◊
     AN◊ITEM◊OR◊RETRIEVE
     ◊ONE?" : PRINT "◊(SPC
     ◊TO◊EXIT◊OR◊S,◊R)◊" ;:
     GET A$: PRINT A$: GOSUB
     1940: IF A$ < > "S" THEN
     5540
5525 INPUT "STORE◊WHICH
     ◊ITEM?◊" ;Q$: GOSUB
     1945:ZW = I: IF Q$ = ""
     OR Q$ = "NONE" THEN 5503
5527 IF ZW < 5 THEN 5533
5530 GOSUB 1940: PRINT
     "YOU◊HAVE◊NO◊'" Q$ "'!"
     : GOTO 5502
5533 Q$ = "NONE" : GOSUB
     1947:ZX = I: IF ZX < 9
     THEN 5550
5535 INPUT "SWAP◊FOR◊
     WHAT◊ITEM?" ;Q$: GOSUB
     1947:ZX = I: IF ZX < 9
     THEN 5550
5537 GOSUB 1940: PRINT
     "NO◊'" Q$ "'◊IS◊IN◊STOR
     AGE!" : GOTO 5502
5540 IF A$ < > "R" THEN
     PRINT "'COME◊BACK◊
     ANYTIME!'" : GOTO 1999
5542 INPUT "GET◊WHICH◊
     ITEM?◊" ;Q$: GOSUB
     1947:ZX = I: IF Q$ = ""
     OR Q$ = "NONE" THEN 5503
5543 IF ZX > 8 THEN 5537
5545 Q$ = "NONE" : GOSUB
     1945:ZW = I: IF ZW < 5
     THEN 5550
5547 INPUT "SWAP◊FOR◊
     WHAT◊ITEM?◊" ;Q$: GOSUB

     1945:ZW = I: IF ZW > 4
     THEN 5530
5550 LN$(0) = WN$(ZW): LT%
     (0) = WT%(ZW): LO%(0) =
     WO%(ZW):LD%(0) = WD%
     (ZW):LS%(0) = WS%(ZW)
5552 WN$(ZW) = LN$(ZX):
     WT%(ZW) = LT%(ZX): WO%
     (ZW) = LO%(ZX): WD%(ZW)
     = LD%(ZX): WS%(ZW) =
     LS%(ZX)
5555 LN$(ZX) = LN$(0):LT%
     (ZX) = LT%(0): LO%(ZX) =
     LO%(0):LD%(ZX) = LD%(0)
     :LS%(ZX) = LS%(0)
5557 IF WN$(ZW) = "NONE"
     THEN GOSUB 1920: GOTO
     5580
5559 Q$ = WN$(ZW)
5560 FOR I = 1 TO 4: IF I
     = ZW THEN 5570
5565 IF Q$ < > WN$(I) THEN
     5570
5567 IF RIGHT$ (Q$,1) =
     "#" THEN Q$ = LEFT$ (Q$,
     LEN (Q$) - 1) + CHR$ (65
     + INT ( RND (1) * 25)):
     GOTO 5560
5568 Q$ = Q$ + "#" : GOTO
     5560
5570 NEXT I:WN$(ZW) = Q$
5580 IF LN$(ZX) = "NONE"
     THEN 5503
5583 Q$ = LN$(ZX)
5585 FOR I = 1 TO 8: IF I
     = ZX THEN 5595
5587 IF Q$ < > LN$(I) THEN
     5595
5590 IF RIGHT$ (Q$,1) =
     "#" THEN Q$ = LEFT$ (Q$,
     LEN (Q$) - 1) + CHR$ (65
     + INT ( RND (1) * 25)):
     GOTO 5585
5592 Q$ = Q$ + "#" : GOTO
     5585
5595 NEXT I:LN$(ZX) = Q$:
     GOTO 5503
6000 HOME : VTAB 11
6007 PRINT "AS◊YOU◊LEAVE◊
     THE◊HALL,◊THE◊IRISHMAN"
     : PRINT "COMES◊UP◊TO◊YOU
     ,◊SLAPS◊YOU◊ON◊THE◊BACK"
     :PRINT "AND◊SAYS,◊'Y'ALL
     ◊COME◊BACK◊REAL◊SOON," :
     PRINT "YA◊HEAH?'"
6010 PRINT DK$; "OPEN◊CHAR
     ACTERS,L150" : PRINT DK$;
     "WRITE◊CHARACTERS,R"
     ;REC : GOSUB 200: PRINT
     DK$; "CLOSE"
6015 GOSUB 300
6020 PRINT DK$; "DELETE◊
     THE◊ADVENTURER"
6030 END
7000 PRINT DK$ "CLOSE" :
     PRINT "EXIT◊ON◊" ; PEEK
     (222); "◊@◊" ; PEEK
     (218) + 256 * PEEK
     (219);
7017 IF PEEK (222) < > 5
     THEN PRINT : PRINT
     "KEY◊>>◊" ;: GET Q$:
     PRINT Q$
7019 PRINT CHR$ (4); "RUN◊
     THE◊WONDERFUL◊WORLD◊OF◊
     EAMON"
7020 END
8000 Q9 = 0:F$ = "" : FOR
     I = 4 TO 8:F$ = F$ +
     CHR$ ( PEEK (512 + I) -
     128): NEXT I
8010 IF F$ < > "◊◊◊MA"
     THEN Q9 = 1
8090 RETURN
8099 REM START ADV (SAVE
     NAME)
8100 L = 43382: POKE L -
     1,40:F$ = ADNAME$:L9 =
     LEN (F$): IF L9 < 7 THEN
     8130
8110 IF LEFT$ (F$,4) =
     "THE◊" THEN L9 = L9 -
     4:F$ = RIGHT$ (F$,L9)
8130 POKE L,L9: FOR I = 1
     TO L9
```

Column 1

```
8140 POKE L + I, ASC (
     MID$ (F$,I,1)): NEXT I
8150 RETURN
```

Checksums

0-$92CC	16-$38F1	50-$14B5
1-$DB99	17-$C92D	60-$D925
2-$9F59	20-$3B18	70-$E13E
3-$7A8F	30-$24AD	73-$B3D0
4-$C15F	35-$1F66	75-$6D88
5-$8CAA	37-$678F	77-$9353
7-$CF81	38-$656C	80-$3CDA
8-$142B	40-$E6B4	90-$3B3C
9-$C02D	42-$B354	100-$2227
10-$7A8F	49-$6769	102-$39C8

EXP.UPDATE

```
10 REM EXP.UPDATE 9/92:
   CALC/RECORD EXPERIENCE &
   LEVEL AFTER AN ADVENTURE
20 REM BY JEFF HURLBUT
   (COMPUTIST MAGAZINE)
   713-645-8680
100 DIM E$(300),N%(300),
    P(8)
110 DIM TA%(4),XA%(5),XNA
    ME$(4),XTYPE%(4),XODDS%
    (4),XDICE%(4),XSIDES%(4)
120 DIM SA%(4),WA%(5),WNA
    ME$(4),WTYPE%(4),WODDS%
    (4),WDICE%(4),WSIDES%(4)
130 PRINT :DK$ = CHR$ (4)
140 PRINT DK$ "OPEN◊THE
    ◊ADVENTURER" : PRINT DK$
    "READ◊THE◊ADVENTURER" :
    INPUT NA$: INPUT REC:
    PRINT DK$ "CLOSE"
150 Q$ = NA$ + ".PRE" :
    PRINT DK$ "OPEN" Q$:
    PRINT DK$ "READ" Q$:
    INPUT NA$: INPUT HV, AV,
    CV: FOR I = 1 TO 4:
    INPUT TA%(I): NEXT I:
    FOR I = 1 TO 5: INPUT
    XA%(I): NEXT I: INPUT
    VAE
160 INPUT SEX$
170 INPUT OGOLD,BANK:
    INPUT OAC: FOR I = 1 TO
    4: INPUT XN$(I),XT%(I),
    XO%(I),XS%(I): NEXT I:
    PRINT DK$ "CLOSE" :
    PRINT DK$ "DELETE" Q$
180 PRINT DK$ "OPEN◊CHAR
    ACTERS,L150" : PRINT DK$
    "READ◊CHARACTERS,R" ;REC
    : INPUT NA$: INPUT HD,AG
    ,CH: FOR I = 1 TO 4:
    INPUT SA%(I): NEXT I:
    FOR I = 1 TO 5: INPUT
    WA%(I): NEXT I: INPUT AE
190 INPUT SEX$
200 INPUT GOLD,BANK: INPUT
    AC: FOR I = 1 TO 4:
    INPUT WN$(I),WT%(I),WO%
    (I),WD%(I),WS%(I): NEXT
    I: PRINT DK$ "CLOSE"
210 REM (GET ADV NAME)
220 L = 43382: IF PEEK (L
    - 1) = 40 THEN 270
230 HOME : VTAB 5: PRINT
    "THE◊SECRETARY◊OF◊HALL◊
    RECORDS◊LOOKS" : PRINT
    "AT◊YOUR◊PRINTOUT◊AND◊
    SIGHS.◊'FOR◊SOME"
240 PRINT "REASON,◊'SHE◊
    SAYS,◊'THE◊NAME◊OF◊YOUR"
    : PRINT "LATEST◊QUEST◊
    WAS◊NOT◊SENT◊IN.'"
250 PRINT : PRINT "<PLEASE
    ◊RE-INSERT◊ADVENTURE◊
    DISKETTE>◊" ;: GET Q$:
    PRINT
252 ONERR GOTO 267
260 PRINT DK$ "UNLOCK◊
    EAMON.NAME" : PRINT DK$
    "OPEN◊EAMON.NAME" :
    PRINT DK$ "READ◊EAMON
    .NAME" : INPUT F$: PRINT
    DK$ "CLOSE"
262 IF LEFT$ (F$,4) =
    "THE◊" THEN F$ = RIGHT$
    (F$, LEN (F$) - 4)
```

Column 2

```
263 PRINT : PRINT "'AH!◊
    HERE◊IT◊IS.'" : PRINT
265 GOTO 268
267 PRINT DK$ "CLOSE" :
    HOME : VTAB 5: PRINT
    "'HMMMM...◊NO◊HELP◊THE
    RE.◊I'LL◊JUST" : PRINT
    "CALL◊IT◊ADVENTURE◊X!'"
    :F$ = "ADVENTURE◊X"
268 PRINT : PRINT "<PLEASE
    ◊INSERT◊MAIN◊HALL◊DISK
    ETTE>◊" ;: GET Q$: PRINT
    : GOTO 290
270 L9 = PEEK (L):F$ = ""
    : FOR I = 1 TO L9:F$ =
    F$ + CHR$ ( PEEK (L +
    I))
280 NEXT I: POKE L - 1,32
290 POKE 216,0:Z$ = NA$ +
    ".EXP"
300 ONERR GOTO 520
310 PRINT DK$ "UNLOCK" Z$
320 PRINT DK$ "OPEN" Z$
330 PRINT DK$ "READ" Z$
340 FOR I = 0 TO 8: INPUT
    P(I): NEXT I
350 FOR I = 1 TO P(0):
    INPUT E$(I),N%(I): NEXT
    I
360 PRINT DK$ "CLOSE" :
    POKE 216,0
370 FOR I = 1 TO P(0): IF
    E$(I) < > F$ THEN NEXT
    I: GOTO 390
380 N%(I) = N%(I) + 1:
    GOTO 410
390 IF I > 300 THEN I =
    300
400 N%(I) = 1:E$(I) =
    F$:P(0) = I
410 POKE 216,0:NZ = N%(I):
    GOSUB 540: GOSUB 670:
    GOSUB 910
420 PRINT DK$ "OPEN" Z$:
    PRINT DK$ "WRITE" Z$
430 FOR I = 0 TO 8: PRINT
    P(I): NEXT I
440 FOR I = 1 TO P(0):
    PRINT E$(I): PRINT
    N%(I): NEXT I
450 PRINT DK$ "CLOSE"
460 HOME : PRINT "ADVENT
    URES◊OF◊" NA$: PRINT
470 PRINT "LEVEL:◊" ;P(2);
    "◊◊◊◊EXPERIENCE:◊" ;P(1):
    FOR I = 1 TO 39: PRINT
    "_" ;: NEXT I: PRINT
480 POKE 34,4: VTAB 5
490 FOR I = 1 TO P(0):
    PRINT E$(I); SPC( 31 -
    LEN (E$(I)));N%(I);
    LEFT$ ( "◊TIMES" ,6 -
    (N%(I) = 1)): NEXT I
500 POKE 34,0: PRINT :
    PRINT "ANY◊KEY>>◊" ;:
    GET Q$: PRINT : PRINT
    DK$ "RUN◊◊◊MAIN◊HALL"
510 END
520 P(0) = 1:E$(1) =
    F$:N%(1) = 1:I = 1: GOTO
    410
530 REM ...CHANGE FILTER
540 PRINT :L = 0:Q = HD -
    HV:J = 5 - 3 * (HV > 13)
    - (HV > 25): IF ABS (Q)
    > J THEN HD = HV + J *
    SGN (Q): GOSUB 640
550 Q = AG - AV:J = 5 - 3
    * (AV > 13) - (AV > 25):
    IF ABS (Q) > J THEN AG =
    AV + J * SGN (Q): GOSUB
    640
560 Q = CH - CV:J = 5 - 3
    * (CV > 13) - (CV > 25):
    IF ABS (Q) > J THEN CH =
    CV + J * SGN (Q): GOSUB
    640
570 FOR I = 1 TO 4:Q =
    SA%(I) - TA%(I):J = 11 -
    4 * (TA%(I) > 25) - 3 *
    (TA%(I) > 50) - 2 *
    (TA%(I) > 80): IF ABS
    (Q) > J THEN SA%(I) =
    TA%(I) + J * SGN (Q):
    GOSUB 640
580 NEXT I
```

Column 3

```
590 FOR I = 1 TO 5:Q =
    WA%(I) - XA%(I):J = 15 -
    5 * (XA%(I) > 25) - 4 *
    (XA%(I) > 50) - 3 *
    (XA%(I) > 80): IF ABS
    (Q) > J THEN WA%(I) =
    XA%(I) + J * SGN (Q):
    GOSUB 640
600 NEXT I
610 Q = AE - VAE:J = 20 -
    10 * (VAE > 30) - 5 *
    (VAE > 50) - 3 * (VAE >
    75): IF ABS (Q) > J THEN
    AE = VAE + J * SGN (Q):
    GOSUB 640
620 L9 = L: IF L THEN
    PRINT : PRINT
630 RETURN
640 L = L + 1: PRINT "*"
    CHR$ (48 + L) "◊" ;:
    RETURN
650 REM ...CALC EXPERIENCE
660 REM GOLD DIF
670 DG = GO - OG: IF DG <
    0 THEN DG = 1
680 IF DG = 0 THEN 720
690 Q = 0: IF DG > 100000
    THEN Q = INT ((DG -
    100000) / 100000): IF Q
    > 9 THEN Q = 9
700 DG = Q + INT (DG /
    200): IF DG < 1 THEN DG
    = 1
710 REM WEAP DIF
720 DW = 0: FOR I = 1 TO 4
730 IF WN$(I) < > XN$(I)
    OR WT%(I) < > XT%(I) OR
    WO%(I) < > XO%(I) OR
    WD%(I) < > XD%(I) OR
    WS%(I) < > XS%(I) THEN
    DW = 1: GOTO 760
740 NEXT I: GOTO 800
750 REM BETTER WEAP
760 BW = 0: FOR I = 1 TO 4
770 FOR J = 1 TO 4: IF
    (WO%(I) > XO%(J)) OR
    (WD%(I) + WS%(I) >
    XD%(J) + XS%(J)) THEN
    NEXT J:BW = BW + 1
780 NEXT I
790 REM ATTRIB DIF
800 DA = (HV < > HD) + (AV
    < > AG) + (CV < > CH)
810 REM SPELL DIF
820 DS = 0: FOR I = 1 TO
    4:DS = DS + (SA%(I) < >
    TA%(I)): NEXT I
830 REM WEAPSKILL DIF
840 DF = 0: FOR I = 1 TO
    5:DF = DF + ABS (WA%(I)
    - XA%(I)): NEXT I
850 REM ARMOR DIF
860 DZ = ABS (AC - OA) +
    ABS (AE - VA)
870 QE = (DA * 50) + (DZ *
    11) +(DW * 3) +(BW * 25)
    +(DS * 33) + (DF * 7)
880 QE = QE + 3 * INT (DG
    / NZ) + (DG > 0)
890 RETURN
900 REM ...LEVEL ADVANCE
910 W = P(2):QE = P(1) +
    QE:J = QE
920 I = 0:L = 0
930 I = I + 100:Q = INT (J
    / I): IF Q > = 16 THEN L
    = L + 16:J = J - (I *
    16): GOTO 930
940 L = L + Q: IF L - W <
    3 THEN 1010
950 REM (ADJUST OVER-EXP)
960 PRINT : PRINT "*" ;QE;
    "◊*" ;L; "◊" ;:I = 0:J =
    0:L = W + 2
970 I = I + 100:Q = INT (L
    / 16): IF Q > = 1 THEN J
    = J + 16 * I:L = L - 16:
    GOTO 970
980 J = J + L * I:L = 18
990 Q = (QE - J) / L:
    PRINT "*" ;: IF Q > I /
    2 THEN L = L + L: GOTO
    990
1000 QE = J + INT (Q + Q /
     3):L = W + 2
1010 P(1) = QE:P(2) = L
```

Column 4

```
1020 PRINT : PRINT
1030 IF P(2) > W THEN 1060
1040 IF L9 > 0 THEN 1150
1050 GOTO 1180
1060 HOME : PRINT NA$
     "◊ADVANCES◊A◊LEVEL!!" :
     PRINT CHR$ (7): PRINT :
     PRINT "◊<KEY>◊" ;: GET
     Q$: PRINT
1070 HOME : RESTORE
1080 PRINT "[A]◊HARDINESS"
     ;: HTAB 17: PRINT HD:
     PRINT "[B]◊AGILITY" ;:
     HTAB 17: PRINT AG: PRINT
     "[C]◊CHARISMA" ;: HTAB
     17: PRINT CH
1090 FOR I = 1 TO 5: READ
     Q$: PRINT "[" ; CHR$ (67
     + I); "]◊" ;Q$;: HTAB
     17: PRINT SPC( WA%(I) <
     10 AND WA%(I) > -
     1);WA%(I): NEXT I
1100 FOR I = 1 TO 4: READ
     Q$: PRINT "[" ; CHR$ (72
     + I); "]◊" ;Q$;: HTAB
     17: PRINT SPC( SA%(I) <
     10);SA%(I): NEXT I
1110 PRINT : VTAB 16:
     PRINT "SELECT◊ATTRIBUTE
     ◊OR◊SKILL◊TO◊ENHANCE:◊"
     ;: GET Q$: PRINT Q$:
     GOSUB 2000
1120 PRINT : IF Q = 99
     THEN 1070
1130 VTAB X: HTAB 25:
     PRINT I; "◊<<◊DONE◊" ;:
     GET Q$: VTAB 16: HTAB 1:
     PRINT : PRINT
1140 W = W + 1: IF P(2) >
     W THEN 1060
1150 PRINT DK$ "OPEN◊CHAR
     ACTERS,L150" : PRINT DK$
     "WRITE◊CHARACTERS,R"
     ;REC
1160 PRINT NA$: PRINT HD:
     PRINT AG: PRINT CH: FOR
     I = 1 TO 4: PRINT
     SA%(I): NEXT I: FOR I =
     1 TO 5: PRINT WA%(I):
     NEXT I: PRINT AE: PRINT
     SEX$
1170 PRINT GOLD: PRINT
     BANK: PRINT AC: FOR I =
     1 TO 4: PRINT WN$(I):
     PRINT WT%(I): PRINT
     WO%(I): PRINT WD%(I):
     PRINT WS%(I): NEXT I:
     PRINT DK$ "CLOSE"
1180 RETURN
2000 Q = ASC (Q$) - 64: IF
     Q < 1 OR Q > 12 THEN
     2120
2010 X = Q: ON Q GOTO
     2020,2030,2040,2050,2050
     ,2050,2050,2050,2080,
     2080,2080,2080
2020 HD = HD + 1:I = HD:
     RETURN
2030 AG = AG + 1:I = AG:
     RETURN
2040 CH = CH + 1:I = CH:
     RETURN
2050 Q = Q - 3: IF WA%(Q)
     > = 100 THEN 2110
2060 WA%(Q) = WA%(Q) + 3 -
     (WA%(Q) > 66) * 2 + INT
     ( RND (1) * 3): IF
     WA%(Q) > 100 THEN WA%(Q)
     = 100
2070 I = WA%(Q): RETURN
2080 Q = Q - 8: IF SA%(Q)
     < 1 OR SA%(Q) > 99 THEN
     2110
2090 SA%(Q) = SA%(Q) + 4 +
     INT ( RND (1) * 4): IF
     SA%(Q) > 100 THEN SA%(Q)
     = 100
2100 I = SA%(Q): RETURN
2110 PRINT
     "NOT◊ALLOWED!◊>>◊" ;:
     GET Q$: PRINT
2120 Q = 99: RETURN
2130 DATA AXE,BOW,CLUB,
     SPEAR,SWORD,BLAST,HEAL,
     SPEED,POWER
```

Checksums

10-$BADD	460-$B08D	900-$A429
20-$9B13	470-$BCD7	910-$5120
100-$54E7	480-$A21E	920-$93AC
110-$9D2C	490-$0DE9	930-$0C4C
120-$D711	500-$B016	940-$2AAF
130-$6820	510-$B49E	960-$85E5
140-$7108	520-$6BB4	970-$71F6
150-$0329	530-$C359	980-$EE7F
160-$0514	540-$6C0F	990-$3845
170-$15A9	550-$8A6A	1000-$A42B
180-$5F2D	560-$953F	1010-$1EC7
190-$C13F	570-$4795	1020-$4D9B
200-$AC6E	580-$0D4A	1030-$A373
210-$4CD4	590-$E98B	1040-$ABCA
220-$7959	600-$DD2C	1050-$3E10
230-$191B	610-$3E28	1060-$605C
240-$93E6	620-$DC4B	1070-$5223
250-$2BD0	630-$EBB7	1080-$A2C3
252-$702A	640-$B365	1090-$D7A5
260-$A1BA	650-$F63F	1100-$7EA0
262-$2BDF	660-$94E9	1110-$BCF0
263-$BA40	670-$0B87	1120-$52ED
265-$B71B	680-$EABF	1130-$2EDC
267-$567A	690-$7172	1140-$7512
268-$3E10	700-$09A3	1150-$B2DB
270-$AF5B	710-$C85F	1160-$FBCC
280-$6454	720-$4F50	1170-$0B50
290-$5C0C	730-$64E0	1180-$83AD
300-$DE22	740-$F0EA	2000-$C1AB
310-$FD8F	750-$CD94	2010-$5644
320-$0FA8	760-$9262	2020-$DD92
330-$EAF2	770-$047C	2030-$0163
340-$5DA7	780-$CB3D	2040-$B538
350-$5FFF	790-$7C60	2050-$E592
360-$56EE	800-$8585	2060-$2499
370-$901A	810-$A400	2070-$B277
380-$FAAC	820-$59CE	2080-$5040
390-$F7D4	830-$7318	2090-$6904
400-$BD6C	840-$139F	2100-$D4AE
410-$F1FA	850-$8636	2110-$FFFE
420-$87AD	860-$C8EF	2120-$683E
430-$8D37	870-$1853	2130-$6A9E
440-$3B25	880-$21FD	
450-$3DC9	890-$5FCD	

NUEDIT.792

```
0 REM NUEDIT.792 IS AN
  EDITOR FOR "NEW FORMAT"
  EAMON GAME MOD & DESIGN
  9/5/92
1 REM BY JEFF HURLBERT
  (USES MOD OF DON BROWN
  ROUTINE @ LINES 280-310)
2 PRINT CHR$(4)
  "MAXFILES◊5"
3 D$ = CHR$ (4):RE$ = CHR$
  (4)+"READ◊EAMON.DESC,R"
  :WE$ = CHR$ (4) + "WRITE
  ◊EAMON.DESC,R" :OE$=CHR$
  (4) + "OPEN◊EAMON." :CC$
  = CHR$ (4) + "CLOSE" :NX
  = 200:NF = 7:MF = 4:TZ =
  15:PZ = 6:FM = 10
4 DIM N%(NF),DR%(11),D$
  (11),M$(10),M%(10),TY$
  (TZ),TF(TZ),LO$(6),AF$
  (FM,4),WT$(1,6),R%(11),
  F$(NF)
5 FOR I = 1 TO NF: READ
  F$(I): NEXT I: FOR I = 1
  TO 11: READ D$(I): NEXT
  I
6 FOR I = 0 TO TZ: READ
  TY$(I),TF(I): NEXT I:
  FOR I = 0 TO 6: READ LO$
  (I): NEXT I: FOR I = 0
  TO FM: FOR J = 0 TO 3:
  READ AF$(I,J): NEXT J:
  NEXT I: FOR I = 0 TO 1:
  FOR J = 0 TO PZ: READ
  WT$(I,J): NEXT J: NEXT
  I: FOR I = 0 TO 10: READ
  M$(I): NEXT I
7 GOTO 700
9 V = 1: PRINT "#◊"; : IF
  QC<= MF THEN PRINT "C◊";
12 PRINT "<-◊->◊Z=ED◊SPC
  =EXIT:◊" ; : GET Q$:I =
  ASC (Q$): IF I = 13 OR
```

```
  Q$ = "◊" THEN PRINT :
  POP : GOTO 770
15 IF Q$ < > "#" AND Q$ <
  > "C" THEN 33
18 Z$ = "" : IF Q$ = "C"
  AND QC < = MF THEN Z$ =
  "COPY◊TO◊"
21 PRINT Q$: VTAB 23: CALL
  - 958: PRINT Z$F$(QC)
  "#◊(1-" ;NT;: INPUT
  ")◊" ;Q$: IF Q$ = ""
  THEN POP : GOTO 770
24 I = 1:Q = VAL (Q$): IF
  Q > 0 AND Q < = NT THEN
  N2 = Q:I = 2
27 IF Z$ = "" OR I = 1 OR
  QC > MF THEN RETURN
30 GOSUB 61: RETURN
33 N2 = N2 - (N2 > 1) * (I
  = 8) + (N2 < NT) * (I =
  21):I = (I = 8) + (I =
  21)
36 IF Q$ = "Z" THEN V = 0
37 IF ASC (Q$) < 32 THEN
  Q$ = "-"
39 PRINT Q$: RETURN
49 REM **READ ARTIF INFO**
50 PRINT RE$;N2 + NX:
  INPUT AD$
51 PRINT D$ "READ◊EAMON
  .ARTIFACTS,R" ;N2: INPUT
  AN$,VL,TY,RO: FOR I =
  0 TO 3: INPUT PA(I):
  NEXT I: PRINT D$: RETURN
57 REM WR.CHANGE
58 GOSUB 64: IF Q$ = "Y"
  THEN 61
59 VTAB 23: PRINT "REDO?◊
  (Y/N)◊"; : GET Q$: PRINT
  Q$: IF Q$ = "Y" THEN Q$
  = "R" : RETURN
60 Q$ = "X" :V = 1: RETURN
61 ON QC GOSUB 110,135,
  160,175: PRINT D$
62 V = 1:Q$ = "S" : RETURN
63 PRINT "RMS=" ;N%(1);
  "◊ART=" ;N%(2); "◊EFF="
  ;N%(3); "◊MONS=" ;N%(4):
  RETURN
64 VTAB 23: PRINT "◊SAVE◊
  (Y/N)◊"; : GET Q$: IF Q$
  < > "Y" THEN Q$ = "N"
65 PRINT Q$: RETURN
66 FOR I = 1 TO 6: HTAB (6
  + I * 5 - (DR%(I) < 0)):
  PRINT DR%(I); : NEXT I:
  PRINT : RETURN
70 REM **OPENS**
71 PRINT OE$ "ARTIFACTS,L"
  ;LA: RETURN
72 PRINT OE$ "DESC,L242" :
  RETURN
73 PRINT OE$ "MONSTERS,L"
  ;LM: RETURN
74 PRINT OE$ "ROOMS,L"
  ;LR: RETURN
75 PRINT OE$ "NAME" :
  RETURN
76 REM **READ ROOM INFO**
77 PRINT RE$;N2: INPUT R2$
78 PRINT D$ "READ◊EAMON
  .ROOMS,R" ;N2: INPUT
  R1$: FOR I = 1 TO ND:
  INPUT DR%(I): NEXT I:
  INPUT LT: PRINT D$
79 RETURN
80 REM **READ DESC PARMS**
81 GOSUB 72: PRINT RE$;0:
  FOR I = 1 TO 4: INPUT
  N%(I): NEXT I: INPUT
  LR,LM,LA: PRINT D$:
  RETURN
87 REM **WRITE DESC
  PARMS**
88 GOSUB 72: PRINT WE$;0:
  FOR I = 1 TO 4: PRINT
  N%(I): NEXT I: PRINT LR:
  PRINT LM: PRINT LA:
  PRINT D$: RETURN
89 REM **WR.NAME**
90 GOSUB 75: PRINT D$
  "WRITE◊EAMON.NAME":
  PRINT DU$: PRINT ND$:
  PRINT DV$: PRINT CC$:
  RETURN
91 REM MAKE.NEW
```

```
92 ON QC GOSUB 105,125,
  155,170
93 FOR K = 1 TO PX: IF
  N%(QC) > = NX THEN 97
94 N%(QC) = N%(QC) + 1:N2
  = N%(QC): VTAB 23: PRINT
  "CREATING◊" F$(QC) "◊#"
  ;N2; ".◊◊"
95 Q$ = STR$ (N2): ON QC
  GOSUB 107,127,157,173:
  PRINT D$
96 NEXT K
97 PRINT D$: GOSUB 88:V =
  1: RETURN
98 REM OPEN ARTIF & RM
99 GOSUB 71
100 REM MK.ROOM
105 GOSUB 74: GOSUB 72:
  RETURN
107 R1$ = "ZILCH◊ROOM" +
  Q$:R2$ = "YOU◊ARE◊IN
  ◊THE◊" + R1$ + "."
110 PRINT WE$;N2: PRINT
  CHR$(34);R2$; CHR$(34):
  PRINT D$ "WRITE◊EAMON.
  ROOMS,R" ;N2: PRINT CHR$
  (34);R1$; CHR$ (34): FOR
  I = 1 TO ND: PRINT
  DR%(I): NEXT I: PRINT LT
115 RETURN
120 REM MK.ARTIFACT
125 GOSUB 72: GOSUB 71:
  RETURN
127 AN$ = "STICK" + Q$:AD$
  = "YOU◊SEE◊AN◊UGLY◊" +
  AN$ + "." :VL = 5:TY =
  2:WT = 3: FOR I = 0 TO
  3:PA(I) = 3: NEXT I
135 PRINT WE$;N2 + NX:
  PRINT CHR$ (34);AD$;
  CHR$ (34): PRINT D$
  "WRITE◊EAMON.ARTIFACTS
  ,R" ;N2: PRINT CHR$ (34)
  ;AN$; CHR$ (34): PRINT
  VL: PRINT TY: PRINT WT:
  PRINT RO
140 FOR I = 0 TO 3: PRINT
  PA(I): NEXT I
145 RETURN
150 REM MK.EFFECT
155 GOSUB 72: RETURN
157 EF$ = "EFFECT◊#" + Q$
  + "◊TEXT◊GOES◊HERE."
160 PRINT WE$;N2 + 2 * NX:
  PRINT CHR$ (34);EF$;
  CHR$ (34): RETURN
165 REM MK.MONST
170 GOSUB 72: GOSUB 73:
  RETURN
173 MN$ = "BUG" + Q$:MD$ =
  "YOU◊SEE◊A◊GROADY◊" +
  MN$ + "!" :FOR I = 0 TO
  10:M%(I) = 1: NEXT I
175 PRINT WE$;N2 + 3 * NX:
  PRINT CHR$ (34);MD$;
  CHR$ (34): PRINT D$
  "WRITE◊EAMON.MONSTERS,R"
  ;N2: PRINT CHR$
  (34);MN$; CHR$ (34): FOR
  I = 0 TO 10: PRINT
  M%(I): NEXT I: RETURN
180 REM ** INIT.FILES
185 PRINT CC$: POKE 216,0:
  HOME : IF DU$ = "" OR
  DU$ = "X" THEN 190
186 PRINT "MISSING◊EAMON
  .NAME◊PARMS"
187 PRINT : PRINT DU$
189 PRINT "USE◊6.X◊EDITOR◊
  FOR◊THIS◊ADV." ; : GET
  Q$: PRINT : GOTO 980
190 VTAB 23: PRINT "MAKE
  ◊NEW◊EAMON◊7.X◊FILES◊&"
  ; : GOSUB 64: IF Q$ < >
  "Y" THEN 980
195 HOME : INPUT "ADV.NAME
  :◊" ;Q$: IF Q$ = "" OR
  ASC (Q$) = 27 THEN 980
200 IF LEN (Q$) > 30 OR
  LEN (Q$) < 3 THEN PRINT
  : PRINT "LENGTH◊ERROR
  ◊>>◊"; : GET Q$: PRINT :
  GOTO 195
202 DU$ = Q$: INPUT "MOVE
  MENT◊DIRECTIONS◊(6◊OR◊
  10):◊" ;ND$: IF ND$ = ""
```

```
  OR ASC (ND$) = 27 THEN
  980
203 ND = VAL (ND$): IF ND
  < > 6 AND ND < > 10 THEN
  980
204 ND$ = STR$ (ND):DV$ =
  "7.0"
205 GOSUB 90
208 LR = 72:LM = 92:LA =
  72: FOR I = 1 TO MF:
  N%(I) = 0: NEXT I: GOSUB
  71: GOSUB 72: GOSUB 73:
  GOSUB 74:PX = 1: FOR QC
  = 1 TO MF: GOSUB 93:
  NEXT QC
210 GOTO 747
219 REM **NAME CHANGE**
220 HOME : PRINT DU$ "/◊"
  ND$ "/◊" DV$: VTAB 2:I$
  = DU$: PRINT : INPUT
  "NEW◊NAME:◊" ;Q$: IF Q$
  = "" THEN 770
225 I = LEN (Q$): IF I >
  30 OR I < 3 THEN PRINT
  "LEN◊ERROR◊>◊" ;: GET
  Q$: PRINT : GOTO 220
240 DU$ = Q$: GOSUB 90
242 ONERR GOTO 255
244 PRINT D$ "UNLOCK" I$:
  PRINT D$ "RENAME" I$ ","
  DU$
255 POKE 216,0: GOTO 770
259 REM *LOC LABEL*
260 LK = (RO > 500) + 2 *
  (RO > 200 AND RO < 500)
  + 3 * (RO > - 999 AND RO
  < - 1) + 4 * (RO = - 1)
  + 5 * (RO = - 999) + 6 *
  (RO = 0): HTAB 1: CALL -
  958: PRINT "LOCATION◊("
  LO$(LK) "):" ; : HTAB 30:
  PRINT RO; : RETURN
278 REM ** ROUTINE TO
  INPUT I$, HAVING BEEN
  PASSED I2$ **
280 I$ = "" : IF I2$ = ""
  THEN I2$ = "MSG◊GOES◊
  HERE"
282 PRINT I2$; : CALL -958:
  FOR A = 1 TO LEN (I2$):
  PRINT CHR$(8); : NEXT
283 GET A$:J = ASC (A$):
  IF J = 13 THEN CALL -
  958: IF I$ = "" THEN I$
  = I2$: PRINT I2$: RETURN
284 IF J = 13 THEN PRINT :
  RETURN
285 IF J > 31 AND LEN (I$)
  > = TL THEN PRINT CHR$
  (7); : GOTO 283
286 IF J = 9 AND LEN (I$)
  > = TL THEN PRINT CHR$
  (7); : GOTO 283
288 IF J > 31 THEN PRINT
  A$; :I$ = I$ + A$:I2$ =
  MID$ (I2$,2): GOTO 283
290 IF J = 5 THEN PRINT
  I2$; :I$ = I$ + I2$:I2$ =
  "" : GOTO 283
292 IF J = 2 AND LEN (I$)
  THEN FOR A = 1 TO LEN
  (I$): PRINT CHR$ (8); :
  NEXT A:I2$ = I$ + I2$:I$
  = "" : GOTO 283
294 IF J = 4 THEN I2$ =
  MID$ (I2$,2): GOTO 282
296 IF J = 9 THEN I2$ =
  "◊" + I2$: IF LEN (I$) +
  LEN (I2$) > = TL THEN
  I2$ = LEFT$ (I2$, LEN
  (I2$) - 1)
297 IF J = 9 THEN 282
298 IF J = 21 AND LEN
  (I2$) THEN A$ = LEFT$
  (I2$,1):J = ASC (A$):
  GOTO 285
300 IF J = 21 THEN A$ =
  "◊" :J = ASC (A$): GOTO
  285
302 IF J = 8 AND LEN (I$)
  > 1 THEN I2$ = RIGHT$
  (I$,1) + I2$:I$ = LEFT$
  (I$, LEN (I$) - 1):
  PRINT A$; : GOTO 283
304 IF J = 8 AND LEN (I$)
  > 0 THEN I2$ = I$ +
```

```
I2$:I$ = "" : PRINT A$;:
    GOTO 283
308 IF J = 27 THEN PRINT
    I2$:I$ = I$ + I2$:
    RETURN
310 GOTO 283
330 VTAB VV: HTAB HH:
    GOSUB 280: IF I$ = "/"
    THEN POP : GOTO 770
350 RETURN
699 REM *** M A I N L I N
    E ***
700 DU$ = "X" : TEXT :
    HOME : PRINT "NUEDIT" :
    PRINT : PRINT "INSERT◊
    DISKETTE◊TO◊BE◊EDITED.◊
    <KEY>◊" ;: GET Q$: PRINT
    : IF ASC (Q$) = 27 THEN
    HOME : GOTO 980
710 ONERR GOTO 185
720 Q$ = "6" : GOSUB 75:
    PRINT D$ "READ◊EAMON
    .NAME" : INPUT DU$:
    INPUT ND$: INPUT DV$:
    PRINT CC$: POKE 216,0:ND
    = VAL (ND$):I = VAL (
    LEFT$ (DV$,1)): IF I < 7
    THEN HOME : GOTO 187
730 GOSUB 81
747 ZL = 238:ML = LM -
    62:AL = LA - 42:RL = LR
    - 34
748 M$ = "N◊S◊E◊W◊U◊D◊◊" :
    IF ND = 6 THEN 770
749 M$ = "NSEWUD◊TVXY"
    :F$(NF - 1) = F$(NF):NF
    = NF - 1
750 GOTO 770
760 PRINT : PRINT "MAX◊
    NUMBER◊OF◊" F$(QC) "S◊IS
    ◊" ;NX; ".◊" ;: GET Q$:
    PRINT
770 PRINT CC$:X = FRE (0):
    HOME : GOSUB 8000: PRINT
    Q$ "◊◊VREF:◊" ; PEEK
    (175) + 256 * PEEK (176)
780 PRINT : PRINT DU$:
    PRINT : PRINT "DIRECT
    IONS:◊" ND$: PRINT "VERS
    ION◊IS◊◊" DV$: PRINT
790 V = 1:N2 = 1:PX = 0:
    GOSUB 63: PRINT
800 FOR I = 1 TO NF: PRINT
    "[" LEFT$ (F$(I),1)
    "]◊◊" F$(I): NEXT I
810 PRINT : PRINT "[◊]◊◊
    SELECT" ;: HTAB 2: GET
    Q$: PRINT Q$
820 FOR I = 1 TO NF: IF Q$
    < > LEFT$ (F$(I),1) THEN
    NEXT I: GOTO 770
830 QC = I
840 HOME : GOSUB 63: VTAB
    7: PRINT F$(QC): IF QC =
    NF THEN 980
845 IF Q$ = "N" THEN 965
850 Q$ = "ROOM" : IF QC <
    = MF THEN Q$ = F$(QC):
    PRINT : PRINT "ENTER◊+
    ◊TO◊ADD◊NEW◊" Q$ "(S)◊OR"
860 PRINT : PRINT "◊ENTER
    ◊" Q$;: INPUT "◊NUMBER◊
    ◊>>◊" ;Q$: IF Q$ = ""
    THEN 770
870 IF QC > MF THEN N%(QC)
    = N%(1): GOTO 960
875 IF Q$ < > "+" THEN 960
880 IF (N%(QC) > = NX)
    THEN 760
890 PRINT : PRINT "CREATE
    ◊HOW◊MANY◊" F$(QC);:
    INPUT "S?◊◊>>◊" ;Q$: IF
    Q$ = "" THEN 770
900 PX = VAL (Q$): IF PX <
    2 THEN PX = 1
910 GOSUB 92: GOTO 770
960 Q = INT ( ABS ( VAL
    (Q$))): IF Q < 1 OR Q >
    N%(QC) THEN 770
965 ON QC GOSUB 99,125,
    155,170,115,99
970 N2 = Q: ON QC GOTO
    1010,2010,3010,4010,
    220,1510
980 PRINT CC$: PRINT CHR$
    (4) "MAXFILES◊3" : PRINT
```

```
    : END
1000 REM *** ROOMS ***
1010 GOSUB 77
1020 HOME : PRINT "ROOM◊"
    ;N2;: HTAB 15: PRINT
    "LIGHTING:◊" ;LT;: IF V
    THEN PRINT : GOTO 1025
1022 HTAB 25: INPUT ""
    ;Q$: IF Q$ = "" THEN
    VTAB 1: HTAB 25: PRINT
    LT: GOTO 1025
1023 LT = VAL (Q$)
1025 VTAB 3: IF V THEN
    PRINT "◊" R1$: GOTO 1070
1030 VV = 3:HH = 2:TL =
    RL:I2$ = R1$: GOSUB 330
1040 IF I$ = "" THEN 1070
1060 R1$ = I$
1070 VTAB 5: IF V THEN
    PRINT R2$: GOTO 1110
1080 VV = 5:HH = 1:TL =
    ZL:I2$ = R2$: GOSUB 330
1090 IF I$ = "" THEN 1110
1100 R2$ = I$
1110 VV = 14: VTAB VV - 2:
    PRINT "EXITS◊(0=NONE◊
    NEG=STRANGE◊+500=DOOR)--
    " : FOR I = 1 TO ND
1120 VTAB VV + INT ((I -
    1) / 2): HTAB (1 + 19 *
    (I / 2 = INT (I / 2)):
    PRINT D$(I); "◊--->◊RM◊"
    ;DR%(I);: IF V THEN
    PRINT : GOTO 1150
1130 HTAB (15 + 19 * (I /
    2 = INT (I / 2))): INPUT
    "" ;Q$: IF Q$ = "" THEN
    VTAB VV + INT ((I - 1) /
    2): HTAB (15 + 19 * (I /
    2 = INT (I / 2))): PRINT
    DR%(I): GOTO 1150
1140 DR%(I) = VAL (Q$)
1150 NEXT I
1160 Q$ = "" : IF NOT V
    THEN QC = 1: GOSUB 58:
    IF Q$ = "R" THEN 1020
1170 IF Q$ = "X" THEN 1010
1200 VTAB 23: PRINT M$;:NT
    = N%(1): GOSUB 9
1210 Q = (QC = 1) + 2 *
    (QC < > 1): IF I THEN ON
    Q GOTO 1010,1510
1220 IF NOT V THEN ON Q
    GOTO 1020,1010
1230 FOR I = 1 TO ND: IF
    Q$ < > LEFT$ (D$(I),1)
    THEN NEXT I: GOTO 1200
1240 IF ABS (DR%(I)) < 501
    OR ABS (DR%(I)) > 700
    THEN 1245
1242 ZZ = N2:N2 = ABS
    (DR%(I)) - 500: IF N2 >
    N%(2) THEN 1244
1243 GOSUB 51: IF PA(0) <
    > 0 AND ABS (PA(0)) < =
    N%(1) THEN N2 = ABS
    (PA(0)):◊GOTO 1247
1244 N2 = ZZ: GOTO 1200
1245 IF DR%(I) < > 0 AND
    ABS (DR%(I)) < = N%(1)
    THEN N2 = ABS (DR%(I)):
    GOTO 1247
1246 GOTO 1200
1247 ON Q GOTO 1010,1510
1250 GOTO 1200
1500 REM *** DISP RM
    CONNECTS ***
1510 ZZ = N2: GOSUB 78:
    FOR I = 1 TO 6:R%(I) =
    DR%(I): NEXT I
1520 HOME : PRINT "ROOM◊"
    ;N2;: HTAB 11: FOR I = 1
    TO 6: PRINT LEFT$
    (D$(I),1); SPC( 4);:
    NEXT I: PRINT
1530 FOR I = 1 TO 39:
    PRINT "_" ;: NEXT I:
    PRINT
1540 FOR J = 1 TO 6:N2 =
    R%(J):K = J * 2 + 3:
    VTAB K: PRINT LEFT$
    (D$(J),1) "◊#" ; SPC( 2
    - ( ABS (N2) > 9) - (N2
    < 0));N2;
1550 IF N2 = 0 THEN HTAB
    11: PRINT "...◊NONE◊..."
```

```
    : GOTO 1590
1555 IF N2 > 500 AND N2 <
    600 THEN HTAB 11: PRINT
    "...◊A◊DOOR◊..." : GOTO
    1590
1560 IF N2 > N%(1) THEN
    HTAB 11: PRINT "...◊UND
    EFINED◊..." : GOTO 1590
1570 IF N2 < 0 THEN HTAB
    11: PRINT "...◊SPECIAL
    ◊..." : GOTO 1590
1580 PRINT : GOSUB 74:
    GOSUB 78: VTAB K: GOSUB
    66: GOTO 1590
1590 NEXT J
1600 N2 = ZZ: FOR I = 1 TO
    6:DR%(I) = R%(I): NEXT
    I: GOTO 1200
2000 REM *** ARTIFACTS ***
    (RD & WR 8 PARMS FOR ALL
    FMTS)
2010 GOSUB 50
2040 HOME : PRINT "ART◊#"
    ;N2; "◊◊◊◊" ;: HTAB 9:
    IF V THEN PRINT AN$:
    GOTO 2090
2050 VV = 1:HH = 9:TL =
    AL:I2$ = AN$: GOSUB 330
2060 IF I$ = "" THEN 2090
2080 AN$ = I$
2090 VTAB 3: IF V THEN
    PRINT AD$: GOTO 2140
2100 VV = 3:HH = 1:TL =
    ZL:I2$ = AD$: GOSUB 330
2120 IF I$ = "" THEN 2140
2130 AD$ = I$
2140 VV = 7: VTAB VV + 3:
    PRINT "VALUE◊IN◊GOLD◊
    PIECES:◊" ;: HTAB 30◊
    PRINT VL;: IF V THEN
    PRINT : GOTO 2170
2150 HTAB 30: INPUT ""
    ;Q$: IF Q$ = "" THEN
    VTAB VV + 3: HTAB 30:
    PRINT VL: GOTO 2170
2160 VL = VAL (Q$)
2170 VTAB VV + 4: HTAB 1:
    PRINT "TYPE:◊" TY$(TY)
    "◊◊◊◊" ;: HTAB 30: PRINT
    TY;: IF V THEN PRINT :
    GOTO 2220
2180 PRINT "◊<-◊->◊" ;:
    GET Q$:I = ASC (Q$): IF
    I = 13 THEN PRINT : GOTO
    2220
2190 IF I = 8 THEN TY = TY
    - (TY > 0) + TZ * (TY =
    0)
2200 IF I = 21 THEN TY =
    TY + (TY < TZ) - TZ *
    (TY = TZ)
2210 PRINT : GOTO 2170
2220 VTAB VV + 5: PRINT
    "WEIGHT◊IN◊GRONDS:◊" ;:
    HTAB 30: PRINT WT;: IF V
    THEN PRINT : GOTO 2250
2230 HTAB 30: INPUT ""
    ;Q$: IF Q$ = "" THEN
    VTAB VV + 5: HTAB 30:
    PRINT WT: GOTO 2250
2240 WT = VAL (Q$)
2250 IF V THEN 2255
2253 VTAB VV + 7: PRINT
    "IN◊ROOM◊X:◊X◊◊◊◊EMBED
    ◊IN◊RM◊X:◊X+200" : PRINT
    "CARRIED:◊◊-100◊◊◊ON◊MON
    STER◊X:◊-X-1" : PRINT
    "WORN:◊◊◊◊-999◊◊◊IN◊ARTI
    FACT◊X:◊X+500"
2255 VTAB VV + 12 - V * 6:
    GOSUB 260: IF V THEN
    PRINT : GOTO 2290
2260 HTAB 30: INPUT ""
    ;Q$: IF Q$ = "" THEN
    2275
2270 RO = VAL (Q$)
2275 VTAB VV + 6: GOSUB
    260: PRINT
2290 ZZ = TF(TY): FOR I =
    0 TO 3
2292 VTAB VV + 7 + I: HTAB
    1: PRINT AF$(ZZ,I) ":◊"
    ;: IF (I < > 1) OR (ZZ <
    > 1 AND ZZ < > 9) THEN
    2298
```

```
2294 PRINT WT$((ZZ =
    1),PA(I));
2298 HTAB 30: PRINT
    PA(I);: IF V THEN PRINT
    : GOTO 2320
2300 IF (I < > 1) OR (ZZ <
    > 1 AND ZZ < > 9) THEN
    2308
2301 PRINT "◊<-◊->◊" ;:
    GET Q$:J = ASC (Q$): IF
    J = 13 THEN PRINT : GOTO
    2320
2302 IF J = 8 THEN PA(I) =
    PA(I) - (PA(I) > 0) + PZ
    * (PA(I) = 0)
2303 IF J = 21 THEN PA(I)
    = PA(I) + (PA(I) < PZ) -
    PZ * (PA(I) = PZ)
2305 PRINT : GOTO 2292
2308 HTAB 30: INPUT ""
    ;Q$: IF Q$ = "" THEN
    VTAB VV + 7 + I: HTAB
    30: PRINT PA(I): GOTO
    2320
2310 PA(I) = VAL (Q$)
2320 NEXT I
2430 Q$ = "" : IF NOT V
    THEN GOSUB 58: IF Q$ =
    "R" THEN 2040
2440 IF Q$ = "X" THEN 2010
2470 VTAB 23:NT = N%(2):
    GOSUB 9
2480 IF I THEN 2010
2490 IF NOT V THEN 2040
2500 GOTO 2470
3000 REM *** EFFECTS ***
3010 PRINT RE$;N2 + 2 *
    NX: INPUT EF$: PRINT D$
3020 HOME : PRINT "EFFECT
    ◊#";N2:VTAB 3: IF V THEN
    PRINT EF$: GOTO 3090
3030 VTAB 3: HTAB 1:VV =
    3:HH = 1:TL = ZL:I2$ =
    EF$: GOSUB 330
3040 IF I$ = "" THEN 3060
3050 EF$ = I$
3060 GOSUB 58: IF Q$ = "R"
    THEN 3020
3070 IF Q$ = "X" THEN 3010
3090 VTAB 23:NT = N%(3):
    GOSUB 9
3100 IF NOT V THEN 3020
3110 IF I THEN 3010
3120 GOTO 3090
4000 REM *** MONSTERS ***
4010 PRINT RE$;N2 + 3 *
    NX: INPUT MD$: PRINT D$
    "READ◊EAMON.MONSTERS,R"
    ;N2: INPUT MN$: FOR I =
    0 TO 10: INPUT M%(I):
    NEXT I: PRINT D$
4040 HOME : PRINT "MON◊#"
    ;N2; "◊◊◊◊" ;: HTAB 9:
    IF V THEN PRINT MN$:
    GOTO 4090
4050 VV = 1:HH = 9:TL =
    ML:I2$ = MN$: GOSUB 330
4060 IF I$ = "" THEN 4090
4080 MN$ = I$
4090 VTAB 3: IF V THEN
    PRINT MD$: GOTO 4140
4100 VV = 3:HH = 1:TL =
    ZL:I2$ = MD$: GOSUB 330
4120 IF I$ = "" THEN 4140
4130 MD$ = I$
4140 VTAB 10: FOR I = 0 TO
    10
4150 PRINT M$(I) ":◊" ;:
    HTAB 20: PRINT M%(I);:
    IF V THEN PRINT : GOTO
    4180
4160 HTAB 20: INPUT ""
    ;Q$: IF Q$ = "" THEN
    VTAB 10 + I: HTAB 20:
    PRINT M%(I): GOTO 4180
4170 M%(I) = VAL (Q$)
4180 NEXT I
4190 Q$ = "" : IF NOT V
    THEN GOSUB 58: IF Q$ =
    "R" THEN 4040
4200 IF Q$ = "X" THEN 4010
4230 VTAB 23:NT = N%(4):
    GOSUB 9
4240 IF I THEN 4010
4250 IF NOT V THEN 4040
4260 GOTO 4230
```

```
8000 Q$ = "" :J = PEEK
   (103) + 256 * PEEK (104)
   + 4: FOR I = 1 TO 11:Q$
   = Q$ + CHR$ (128 + PEEK
   (J + I)): NEXT I: RETURN
8999 REM *F$(I)
9000 DATA ROOM,ARTIFACT,
   EFFECT,MONSTER,NAME◊
   CHANGE,CONNECTIONS,X-IT
9010 REM *D$(I)
9020 DATA NORTH,SOUTH,EAST
   ◊,WEST◊,UP◊◊◊,DOWN◊,T-
   NE◊,V-NW◊,X-SE◊,Y-
   SW◊,NONE
9030 REM *TY$(TZ) TF(TZ)
   <TYPE NAMES & FMT #'S>
9040 DATA GOLD/MONEY,0,STD
   .◊TREASURE,0,STD.◊WEAPON
   ,1,ODD◊WEAPON,1,CONTAIN
   ER,2,LIGHTABLE,3,DRINK
   ABLE,4,READABLE,5,DOOR/
   GATE,6,KEY/OPENER,7,BOUN
   D◊MNST,8,WEARABLE,9,DOOR
   -12,10,CUSTOM-13,0,CUST
   OM-14,0,CUSTOM-15,0
9043 REM *LO$(I)*
9045 DATA IN◊ROOM,INSIDE◊
   CONTAINER,EMBEDDED◊IN◊
   ROOM,ON◊MONSTER,CARRIED
   ,WORN,???
9050 REM *AF$(FM,J) <ARTIF
   PARMS #5-#8>
9060 DATA #5,#6,#7,#8,WEA
   PON◊COMPLEXITY,WEAPON
   ◊TYPE,HIT◊DICE,HIT◊SIDES
9070 DATA KEY◊#,STRENGTH,
   OPEN?,#8,DURATION,#6
   ,#7,#8
9080 DATA HEAL◊AMT,DOSES,
   OPEN?,#8,1ST◊EFFECT,NUM.
   ◊OF◊EFFECTS,OPEN?,#8
9090 DATA ROOM◊BEYOND,KEY
   ◊#,STRENGTH,HIDDEN?,#5
   ,#6,#7,#8
9100 DATA MONSTER◊#,KEY◊#
   ,GUARD◊#,#8,ARMOR◊CLASS,
   TYPE,#7,#8,ROOM◊BEYOND,
   KEY◊#,OPEN?,HIDDEN?
9110 REM *WT$(PZ) <ARTIF.
   PARM #6 WEAP/ARMOR
   TYPES>
9120 DATA ARMOR◊,CAPE◊◊,
   BOOTS◊,GLOVES,HELM◊◊,
   CTYP#5,CTYP#6,NONE◊◊,AXE
   ◊◊,BOW◊◊◊,CLUB◊◊,SPEAR◊
   ,SWORD◊,CUSTOM
9130 REM *M$(I) <MONSTER
   PARMS>
9140 DATA HD,AG,GROUP◊SIZE
   ,COURAGE,ROOM,WEIGHT/
   PRM5,ARMOUR,WEAPON#,HIT
   ◊DICE,HIT◊SIDES,FRIEND
   LINESS
```

Checksums

0–$92CC	24–$CC06	60–$5BCC
1–$DB99	27–$0652	61–$BC83
2–$EB48	30–$1845	62–$25B9
3–$7A80	33–$C60A	63–$2B49
4–$C71B	36–$5C8A	64–$B145
5–$BE89	37–$38E7	65–$D6CD
6–$0217	39–$D453	66–$4DC7
7–$71FC	49–$C881	70–$1981
9–$6DEC	50–$8682	71–$2CB6
12–$726A	51–$8941	72–$D75A
15–$D129	57–$5A98	73–$FA8B
18–$CE28	58–$D3D1	74–$BBB4
21–$DE1A	59–$1DC0	75–$340D

Vendors

Eamon Adventurer's Guild
7625 Hawkhaven Drive
Clemmons, NC 27012
   atten: Tom Zuchowski (910-766-7490)

Product Monitor
7814 Santa Elena
Houston TX 77061
   atten: Jeff Hurlburt (713-645-8680)

Jack R. Nissel   PA

Softkey for...
U.S. Atlas Action
Boppie's Great Word Chase
Verb Viper
Word Man
DLM

Softkey for...
Space Math
Learning Well

Softkey for...
Reading for Meaning with Mother Goose
Metric Skills I
Metric Skills II
Compound Words and Contractions
Nouns/Pronouns 1
Antonyms/Synonyms
Patterns and Sequence
Size and Logic
Observation and Classification
Conservation and Counting
Parquerty and Pictures
Roots/Affixes
Verbs
Chariots, Cougars, & Kings
Money! Money!
Hartley Courseware

Softkey for...
Peanut's Maze Marathon
Fixit
Galaxy Math Basic Facts
Random House

Softkey for...
Dragon's Keep
Sierra On-Line

Softkey for...
Piece of Cake Math
Fraction Factory
Springboard Software

Softkey for...
Sweet Shoppe
Grabit Factory
Eric Software

Softkey for...
Rhymes & Riddles
Spinnaker Software

Softkey for...
Learn About Sounds
American Educational Computer

Softkey for...
Computer Literacy Adventures of the Lollipop Dragon
Cursor Control Adventures
Plotting and Programming Adventures
Letter and Key Number Adventures
Function Key Adventures
The Society for Visual Education

Requirements:
The original disk(s)
Blank disk(s) for each title
COPYA from your DOS 3.3 system master
A sector editor

(This last title was on the most wanted list, as of issue #62, as separate titles. They are Lollipop Dragon: Cursor Control Adventures, Lollipop Dragon: Plotting and Programming Adventures, Lollipop Dragon: Letter and Key Number Adventures, and Lollipop Dragon: Function Key Adventures.)

Notes: On Peanuts Maze Marathon make the sector edit to both sides. On Rhymes & Riddles you do not need to do any sector edits. I found that when you write to the Hartley Courseware and American Educational titles it locks up certain sectors on your unprotected copy. To avoid this I put Diversi-DOS on my copies.

Step-by-step

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24   *ignore checksum & epilogu errors*
RUN COPYA
2. Copy your original disk to your blank disk.
3. Make the following sector edit to your copy.

Trk	Sct	Byte	From	To
$00	$03	$42	38	18

4. Write the sector back to the disk.

Softkey for...
States and Traits
Crypto Cube
DesignWare

Requirements:
The original disk(s)
A blank disk for each title
Any fast copy program
A sector editor

   Jim S. Hart's Softkey, in issue #52, for European Nations and Locations works for these titles as well.
1. Fast copy your original disk.
2. Search for the byte string A9 38 8D 9E B9 and change to EA EA EA EA EA.
3. Write the sector back to the disk.

Softkey for...
Maps & Globes: Understanding Maps & Globes
Troll Associates

Requirements:
The original Maps & Globes: Understanding Maps & Globes disks
2 blank disks
DOS 3.3 system master

   Paul D. Dillion's softkey, in issue 60, for Maps & Globes Latitude & Longitude works for this title.
1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
CALL-151   *gets you into the monitor*
B954:29 00 *ignore first byte of address header*
B988:18 60 *ignore both bytes of address epilogue*
3D0G *gets you back to Applesoft BASIC prompt*
RUN COPYA

---

2. Copy both disks to your blanks
   That's it.

Softkey for...
Math Tutor Fractions Part I
Math Tutor Multiplication
Math Tutor Division
Scholastic

Requirements:
The original disks
3 blank disks for each title
COPYA from your DOS 3.3 system master
Copy II Plus

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24 *ignore checksum and epilogues*
RUN COPYA
2. Copy your original program disks to your blank disks. You can use any fast copy program to copy the management disk.
3. Boot Copy II Plus and use the CHANGE BOOT PROGRAM function to change the boot program to HELLO on your copy of the program disks.

Softkey for...
Math Tutor Fractions Part II
Scholastic

Requirements:
The original disks
3 blank disks
COPYA from your DOS 3.3 system master

   This title was slightly different then Math Tutor Fractions Part I. On even tracks the address prologues were D4 AA 96 and the address epilogues were AB AF. On odd tracks everything was normal. To deprotect this disk we will patch COPYA to allow both D4 and D5 in the address prologues and to ignore the address epilogues.

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
CALL-151 *gets you into the monitor*
B954:4A C9 6A D0 EF *allows D4 and D5 in address prologues*
B988:18 60 *ignores address epilogues*
3D0G *gets you back to the Applesoft prompt*
RUN COPYA
2. Copy your original program disks to your blank disks. You can use any fast copy program to copy your management disk.
   That's it.

Softkey for...
Microzine Jr.
Scholastic

Requirements:
The original disks
4 blank disks
COPYA from your DOS 3.3 system master
Copy II Plus

   Disk 1 was like Math Tutor Fractions Part II. On even tracks the address prologues were D4 AA 96 and the address epilogues were AB AF. On odd tracks everything was normal. We will make the same patches

to COPYA to allow both D4 and D5 in the address prologues and to ignore the address epilogues.

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:

CALL-151 *gets you into the monitor*
B954:4A C9 6A D0 EF *allows D4 and D5 in address prologues*
B988:18 60 *ignores address epilogues*
3D0G *gets you back to the Applesoft prompt*
RUN COPYA

2. Copy both sides of your original disk 1 to your blank disks.

3. Disk 2 is not protected so use Copy II Plus DISK COPY to copy both sides of disk 2 to your remaining blank disks.

Softkey for...

Freddy's Puzzling Adventures
DLM Software

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:

CALL-151 *gets you into the monitor*
B954:4A C9 6A D0 EF *allows D4 and D5 in address prologues*
B942:18 *ignores checksum and epilogue errors*
3D0G *gets you back to the Applesoft prompt*
RUN COPYA

2. Copy your original disks to your blank disks.
That's it.

Softkey for...

The Factory
Sunburst Communications

Requirements:
The original disk
A blank initializied disk
Demuffin Plus
DOS 3.3 system master
A way to reset into the monitor
A blank initializied slave disk with no HELLO program

The softkey in COMPUTIST #57 would not work for me. The controller would die at track $11, sector $08. It's possible that this sector was bad and since there was no information on it the program was not affected. In order to get a COPYA version I did the following.

1. Boot your DOS 3.3 system master and at the Applesoft prompt put in your blank disk and enter;

NEW
INIT LOGO
DELETE LOGO

2. Boot your original disk and at the Applesoft prompt get into the monitor and enter;

6800<B800.BFFFM

3. Boot your slave disk.
C600G

4. At the Applesoft prompt put in your disk containing Demuffin Plus and enter;

CALL-151
BLOAD DEMUFFIN PLUS,A$803
B800<6800.6FFFM
A851G

803G

5. Copy the files using the wild-card (=) option from your original to your blank disk.

Softkey for...

Max Dublin's Treasure
Grolier

Requirements:
The original disk
A blank initialized disk
A blank initialized slave disk with no HELLO program
Super IOB v1.5
A way to reset into the monitor

This title can be deprotected by using Super IOB with the Swap Controller that uses the RWTS of the protected disk to read the original disk then uses a normal RWTS to write the information back to your blank disk.

1. INITialize your blank disk
INIT HELLO
DELETE HELLO

2. Boot your original disk and at the Applesoft prompt reset into the monitor.

3. Move the RWTS to a safe place by entering:
1900<B800.BFFFM

4. Put in your slave disk and boot it by entering:
C600G

5. After the disk boots and the Applesoft prompt appears insert your Super IOB disk and save the RWTS to it by entering:
BSAVE RWTS.MAX DUBLIN, A$1900, L$800

6. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer NO when asked if you want to INITialize the blank disk.

CONTROLLER

```
1000 REM MAX DUBLIN.CON
1010 TK = 3:ST = 0:LT =
     35:CD = WR
1020 T1 = TK: GOSUB 490:
     GOSUB 360: ONERR GOTO
     550
1030 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:
     IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST
     = 0: GOSUB 360
1070 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1070
1080 ST = 0:TK = TK + 1:
     IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT
     "COMPLETED,◊DONE,◊
     THROUGH,◊FINISHED" : END
10010 IF PEEK (6400) < >
     162 THEN PRINT CHR$ (4)
     "BLOAD◊RWTS.MAX◊
     DUBLIN,A$1900"
```

Checksums

1000-$356B	1040-$6342	1080-$6CA2
1010-$3565	1050-$ABA3	1090-$9DCA
1020-$6170	1060-$20C0	1100-$8BBF
1030-$7771	1070-$28C5	10010-$4616

Softkey for...

Bike Hike
Learning Technologies

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master
A sector editor

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24
RUN COPYA

2. Copy your original to your blank disk

3. Make the following sector edits to your copy.

Trk	Sct	Byte	From	To
0E	03	76-77	D0 ED	60 EA
11	00	01	11	03 .

Softkey for...

Math in a Nutshell
Clowning Around
Learning Technologies

Requirements:
The original disk(s)
A blank disk for each title
DOS 3.3 system master
A sector editor

1. Boot your DOS 3.3 system master and at the] prompt enter:
POKE 47426,24 *ignore checksums and epilogues*
RUN COPYA

2. Copy your original to your blank disk.

3. Make the following sector edit to your copy.

Trk	Sct	Byte	From	To
00	03	42	38	18

4. Write the sector back to the disk.

Softkey for...

Let's Learn about the Library
Troll Courseware

Requirements:
The original program
3 blank sides
COPYA from your DOS 3.3 system master

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
CALL-151
B954:29 00
B942:18
3D0G
RUN COPYA

2. Copy all 3 disks to your blank disks.
You now have a COPYA copy.

Softkey for...

Grade Manager v2.3
MECC

Requirements:
The original disk
A blank disk
Copy II Plus
Advanced COPYA

1. Boot Advanced COPYA and use the following setup.

Question	Response
Use foreign RWTS	NO
Start track	01
End track	34
Read as half tracks	NO

Disable DOS error checking	NO
Continue reading on error	NO
All of the above correct	YES
Change address/data bytes	YES
A1 (normal: 213/D5)	170
A2 (normal: 170/AA)	213
A3 (normal: 150/96)	173
E1 (normal: 222/DE)	Press return
E2 (normal: 170/AA)	Press return
D1 (normal: 213/D5)	Press return
D2 (normal: 170/AA)	150
D3 (normal: 173/AD)	170
E1 (normal: 222/DE)	Press return
E2 (normal: 170/AA)	Press return
All of the above correct	YES
Format target disk	YES

2. Use Copy II Plus SECTOR COPY to copy sector $00 from your original disk to your copy.

3. Make the following sector edits to your copy.

Trk	Sct	Byte	From	To
$00	$02	$F1	96	AA
$00	$02	$FC	AA	AD
$00	$03	$55	AA	D5
$00	$03	$5F	D5	AA
$00	$03	$6A	AD	96
$00	$06	$7A	AA	D5
$00	$06	$7F	D5	AA
$00	$06	$84	AD	96
$0B	$04	$1A-3A	AA D5 96 AA	D5 AA 96 D5
			D5 96 AA D5	AA 96 D5 AA
			AD D5 AA 96	96 D5 AA 96
			D5 AA 96 D5	D5 AA 96 D5
			AA AD D5 AA	AA AD D5 AA
			AD D5 AD AA	AD D5 AA AD
			D5 96 AA D5	D5 AA AD D5
			AD AA D5 AA	AA AD D5 AA
			96	AD

4. Write each sector back to the disk before going to the next sector.

Softkey for...

Algebra Shop
Scholastic

Requirements:
The original disk
2 blank disk sides
COPYA from your DOS 3.3 system master

Upon examining this disk I found that side two was unprotected and side one was protected by altered epilogues and by the first byte in the address header being changed to D4 on every other track, which made this program easy to softkey.

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
CALL-151
B942:18
B954:29 00
3D0G
RUN COPYA

2. Copy side one of your original to one of your blank disks.

3. Use any fast copy program to copy side two.
Nothing else needs to be done.

Softkey for...

Spare Moments
MECC

Requirements:
The original disk
2 blank initialized disk sides
A blank initialized slave disk
Super IOB v1.5
A way to reset into the monitor

The controller listed below

To deprotect both sides of this MECC title we will use Super IOB with the Swap Controller and use the RWTS of the protected disk to read the original disk then use a normal RWTS to write the information back to your blank disk.

1. INITialize your blank disks
INIT HELLO
DELETE HELLO

2. Boot side one of your original disk and at the Applesoft prompt reset into the monitor.

3. Move the RWTS to a safe place by entering:
1900<B800.BFFFM

4. Put in your slave disk and enter:
C600G

5. After the disk boots and the Applesoft prompt appears insert your Super IOB disk and save the RWTS to it by entering:
BSAVE RWTS.SPARE MOMENTS,
    A$1900, L$800

6. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer NO when asked if you want to INITialize the blank disk.

CONTROLLER

```
1000 REM SPARE MOMENTS
1010 TK = 3:ST = 0:LT =
     35:CD = WR
1020 T1 = TK: GOSUB 490:
     GOSUB 360: ONERR GOTO
     550
1030 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:
     IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST
     = 0: GOSUB 360
1070 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1070
1080 ST = 0:TK = TK + 1:
     IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT
     "COPY◊DONE" : END
10010 IF PEEK (6400) < >
     162 THEN PRINT CHR$ (4)
     "BLOAD◊RWTS.SPARE◊
     MOMENTS,$1900"
```

Checksums

1000-$356B	1040-$6342	1080-$6CA2
1010-$3565	1050-$ABA3	1090-$9DCA
1020-$6170	1060-$20C0	1100-$9A4D
1030-$7771	1070-$28C5	10010-$F9EC

7. Repeat steps 1-6 for side 2.

Softkey for...

Mr. Pixel's Cartoon Kit
Mindscape

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master
Copy II Plus

This disk is on the most wanted list, (I should know, I asked for it to be put there.)

If you look at this disk with a nibble editor you will see that the only thing that has been

changed on the disk is the epilogues. They have been changes to FF FF from the normal DE AA. Well, this should be easy, I thought. I used COPYA after telling it to ignore checksums and epilogues but when I booted the copy I got a re-boot after it read the first couple of tracks. Hmmm, the dreaded nibble count. I scanned the disks for BD 8C C0 and found it in several places on the disk. The area of track $02, sector $04 was the area I was suspicious of and after putting in a return at the beginning of the track the disk worked fine.

Step-by-step

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24
RUN COPYA

2. Copy your original to your blank disk.

3. Make the following sector edit.

Trk	Sct	Byte	From	To
$02	$04	$00	A9	60

4. Write the sector back to the disk.

Note: If you want to see what putting the 60 does make another copy of your original and on the same track and sector put EA's from address $00-$76. At address $77 you will see a 60 there. On the original disk, after the nibble count is satisfied, the routine winds up at the 60 which tells the program to go back where it came from, before it started the nibble count. When we put the 60 at the beginning of the nibble count we did the same thing, at the start of the routine, that was done at the end of the routine.

Softkey for...

Treasure Island
Below the Root
Windham Classics

Requirements:
The original disks
4 blank disk sides for Treasure Island
2 blank disk sides for Below the Root
Any fast copy program that can ignore read errors

I'm not completely sure, but it seems that these two programs are not protected. I got them for $6.99 each and the disks seem different than what has been said about the Windham Classic programs in any of the issues of Computist. I have tried both of these games and although I haven't completed either of them I have gotten a good way into each of the programs and they seem to work fine. If anyone has these programs and they have the same bad tracks on them I would appreciate them writing in if they have found out different than what I have said.

Treasure Island

1. Fast copy all four sides of Treasure Island and ignore the following read errors.
   Side A - no errors
   Side B - tracks $02 and $03
   Side C - tracks $02 to $0D
   Side D - tracks $02 to $0B

Below the Root

1. Fast copy both sides of Below the Root and ignore the following read errors.
   Side A - tracks $03 and $04
   Side A - tracks $0A to $10
   Side B - no errors

Advanced Playing Technique for...

Heavy Barrel
?

Requirements:
A deprotected copy
A sector editor

Make the following sector edits to side 1 of your copy to make your man invincible.

Trk	Sct	Byte	From	To
$04	$09	$21-23	CE E0 FE	EA EA EA

Advanced Playing Technique for...

Expedition Amazon
Penguin

Requirements:
A deprotected copy
A sector editor

I did this about 2 years ago, (when I wasn't too sure what I was doing), but maybe this will be of some help to someone.

Note: All of the sector edits shown here are to be done to side 2 of your copy.

You will find the prices for the equipment on track $0B, sector $02, starting at byte $83 and on track $1A, sector $05 starting at byte $00.

To modify the amount of money you have, the amount of equipment you have, how many lives have been lost, etc. sector edit the following areas.

Track	Sector
$02	$0D
$06	$09
$16	$00
$18	$06
$1F	$0C

It appears that the disk writes back to two of the five tracks at random so you will have to write down your current stats and check all five of the areas until you find the two that contain the current stats. Once you play the game, as your stats are written back to the disk, if you decide to edit them again they will probably not be on the same track and sectors that they were before.

Softkey for...

Where in North Dakota is Carmen Sandiego
Broderbund

Requirements:
The original disk
2 blank sides
Copy II Plus

1. Fast copy both side of your original disk to your blank disks

2. Make the following sector edit.

Trk	Sct	Byte	From	To
$00	$06	$43-79	D0 F4 BD 8C	A9 FC 85 F0
			C0 10 FB C9	85 F3 85 F6
			E7 D0 4A BD	A9 EE 85 F1
			8C C0 10 FB	85 F2 85 F5
			C9 E7 D0 41	A9 E7 85 F4
			BD 8D C0 A0	85 F7 EA EA
			10 24 80 BD	EA EA EA EA
			8C C0 10 FB	EA EA EA EA
			88 F0 32 C9	EA EA EA EA
			EE D0 F4 EA	EA EA EA EA.
			EA A0 07 BD	EA EA EA EA
			8C C0 10 FB	EA EA EA EA
			99 F0 00 EA	EA EA EA EA
			88 10 F4	EA EA EA

3. Write the sector back to the disk so you don't have to re-enter all of this again.

Softkey for...

Stellar 7
Penguin Software

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master
Copy II Plus

This disk carries the same protection as a lot of the other Penguin programs do. Looking at the disk with a nibble editor shows that the first byte of the address header changes on every other track from D5 to D4 and the address and data epilogues are DA AA instead of the normal DE AA.

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
RUN COPYA

2. After COPYA loads press CONTROL-C. This will stop the program. Now enter:
70
CALL-151
B954:29 00 *this tells DOS to ignore the first byte of the address header*
B993:00 *this tells DOS to ignore the first byte of the address epilogue*
B925:18 60 *this tells DOS to ignore the data epilogues*
3D0G
RUN

3. Copy your original disk to your blank disk.

4. Make the following sector edits to your copy.

Trk	Sct	Byte	From	To
$00	$02	$9E	DA	DE
$00	$03	$35	DA	DE
$00	$03	$91	DA	DE
$00	$03	$54-58	4A 49 6A D0 EF	C9 D5 D0 F0 EA

5. Write each sector to the disk before going to the next sector.

Softkey for...

The Writing Workshop
Milliken

Requirements:
The original disks
11 blank disk sides
DOS 3.3 system master
Copy II Plus

Super IOB v1.5
A way to reset inti the monitor
The controller shown below
A blank initialized slave disk

Since there are quite a few disks in this program let's get rid of a few of them quickly.

Fast copy the following disks to some of your blank disks.

File Cabinet Disk
Postwriting Spelling Checker Disk 1
Postwriting Spelling Checker Word List
Mechanics Checker Word List
Help Disk

Now for the remaining disks.

1. Boot your DOS 3.3 system master and after the Applesoft prompt appears put in one of your blank disks and enter:

INIT BOOT
DELETE BOOT

Label this disk Word Processing and put it aside.

2. Make a change so that the rest of the disks will be initialized to BRUN the boot file.

CALL-151
9E42:34
3D0G

3. Initialize the rest of the disks.

INIT BOOT
DELETE BOOT

4. Boot the original Word Processing disk and at the Applesoft prompt reset into the monitor.

5. Move the RWTS to a safe place by entering:

1900<B800.BFFFM

6. Put in your slave disk and enter:

C600G

7. After the disk boots and the Applesoft prompt appears insert your Super IOB disk and save the RWTS to it by entering:

BSAVE RWTS.WRITING WORK-SHOP, A$1900, L$800

8. Install the controller into Super IOB, run it and copy your original Word Processing disk to your blank disk that you labeled Word Processing. Answer NO when asked if you want to INITialize the blank disk.

9. Repeat step 8 for the remaining disks. You can use the RWTS that you captured from the first disk for the other disks.

CONTROLLER

```
1000 REM THE WRITING
     WORKSHOP.CON
1010 TK = 3:ST = 0:LT =
     35:CD = WR
1020 T1 = TK: GOSUB 490:
     GOSUB 360: ONERR GOTO
     550
1030 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:
     IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST
     = 0: GOSUB 360
1070 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1070
```

```
1080 ST = 0:TK = TK + 1:
     IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "COPY◊
     DONE" : END
10010 IF PEEK (6400) < >
     162 THEN PRINT CHR$ (4)
     "BLOAD◊RWTS.WRIT
     ING◊WORKSHOP,A$1900"
```

Checksums

1000-$356B	1040-$6342	1080-$6CA2
1010-$3565	1050-$ABA3	1090-$9DCA
1020-$6170	1060-$20C0	1100-$9A4D
1030-$7771	1070-$28C5	10010-$B7D1

You are finally done!

Softkey for...

Steps to Comprehension
Educational Publishing Concepts

Requirements:
The original Steps To Comprehension disks
8 blank disks
DOS 3.3 system master
A sector editor

I left out some sector edits in my softkey that appeared in issue 68 of Computist. I am repeating the entire softkey here with all of the sector edits.

1. Boot your DOS 3.3 system master and at the] prompt enter:

POKE 47426,24 *ignore checksums and epilogues*
RUN COPYA

2. Copy all 8 sides of your original to your blank disks.

3. Make the following sector edits to your copy of your master disk.

Trk	Sct	Byte	From	To
$00	$02	$A3	DE	AA
$00	$03	$3F	DE	AA
$00	$03	$42	38	18

4. Write each sector back to the disk before going to the next sector.

My apologies to whoever used my original softkey.

Softkey for...

Steps to Comprehension Password Repair Disk
Educational Publishing Concepts

Requirements:
The original Steps To Comprehension Repair disk
A blank disk
DOS 3.3 system master
Copy II Plus

1. Boot your DOS 3.3 system master and at the] prompt enter:

POKE 47426,24 *ignore checksums and epilogues*
RUN COPYA

2. Copy your original to your blank disk.

3. Make the following sector edit to your copy. *NOTE:* Some of these edits might not be needed if the "TO" bytes are on your copy already.

Trk	Sct	Byte	From	To
$00	$03	$42	38	18
$00	$03	$35	AA	DE
$00	$03	$91	AA	DE
$00	$03	$9B	DE	AA
$00	$02	$9E	AA	DE
$00	$02	$A3	DE	AA

4. Write each sector back to the disk before going to the next sector.

Softkey for...

Children's Writing and Publishing Center
The Learning Company

Requirements:
The original disks
5 blank sides
COPYA from your DOS 3.3 system master
Any fast copy program

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:

POKE 47426,24
RUN COPYA

2. Copy side 1 of your program disk to a blank disk.

3. Use any fast copy program to copy the remaining sides to the rest of your blank disks.
   That's it.

Softkey for...

Cut and Paste
Electronic Arts

Requirements:
The original disk
A blank disk
Copy II Plus
Super IOB v1.5
The controller shown below

The softkey in issue 13, page 26, of Computist wouldn't work with my copy of the program. I was able to modify the controller and get it to work. If your copy wouldn't work with it either then try this softkey.

1. Use Copy II Plus to format your blank disk.

2. Use the SECTOR COPY function of Copy II Plus to copy tracks $00 to $02 from your original to the disk you just formatted.

3. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer NO when asked if you want to INITialize the blank disk.

CONTROLLER

```
1000 REM CUT AND PASTE.CON
1010 TK = 7:ST = 0:LT =
     35:CD = WR
1020 T1 = TK: GOSUB 490:
     IF TK > 3 THEN RESTORE :
     GOSUB 210
1030 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1030
1035 IF TK = 2 THEN GOSUB
     210
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:
     IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST
     = 0: GOSUB 230: IF TK =
     0 THEN GOSUB 1110
1070 GOSUB 430: GOSUB
     100:ST = ST + 1: IF ST <
     DOS THEN 1070
1080 ST = 0:TK = TK + 1:
     IF BF = 0 AND TK < LT
     THEN 1070
1090 IF TK < LT THEN 1020
1100 HOME : PRINT "COPY◊
     DONE" : END
1110 POKE 19015,170: POKE
     19025,173: POKE 18024,24
```

```
1120 POKE 18025,96: POKE
     18026,235: RETURN
62010 DATA 213,187,207
```

Checksums

1000-$356B	1040-$2C83	1090-$37F7
1010-$3F61	1050-$8A4C	1100-$E224
1020-$237A	1060-$6A59	1110-$E706
1030-$357B	1070-$625C	1120-$C6D2
1035-$AADA	1080-$F5DA	62010-$0042

4. Make the following sector edits to your copy.

Trk	Sct	Byte	From	To
$02	$03	$47	BB	AA
$02	$03	$51	CF	AD

Enjoy.

Softkey for...

Word-A-Mation
Sunburst Communications

Requirements:
The original disk
A blank disk formatted in ProDOS (format the disk with the volume name /WORK)
A ProDOS formatted disk with only ProDOS 8 and Basic System on it.
A blank work disk
ProDOS Super IOB
Copy][Plus
The controller listed below
A way to reset into the monitor
A normal ProDOS file

This Sunburst product will softkey very easily using Bill Jetzer's ProDOS Super IOB. You should have the ProDOS Super IOB, along with the support files, already on a ProDOS disk. These files can be found in issue 62 of Computist starting on page 31.

1. Boot the original disk and write down what version number of ProDOS is being used. Then when you see the Applesoft prompt press CONTROL-RESET to stop the boot.

2. Enter the following. The routine you will enter will move the entire language card, except for bank 2, into memory at $8000.

CALL-151
300:AD 88 C0 A2 00 BD 00 D0
:9D 00 80 E8 D0 F7 EE 0A
:03 EE 07 03 D0 EF AD 8A
:C0 60
300G

If after doing 300G, and pressing RETURN the computer freezes just press CONTROL-RESET to get back to the Applesoft prompt and then enter CALL-151 to get back into the monitor.

3. If the ProDOS version on the original disk was 1.0 do this next step. If it was 1.1 or newer skip to step 4.

8000<A800.AEFFM

4. Put in the ProDOS disk that contains only ProDOS and Basic System and boot it.

C600G

5. At the Applesoft prompt put in the disk that contains ProDOS Super IOB and enter:

CALL-151
1900 < 8000.86FFM
BSAVE DRVR.WORD-A-MATION, A$1900, L$700

6. Install the controller into ProDOS Super IOB, run it and copy your original disk to your blank work disk.

CONTROLLER

```
2000 REM WORD-A-MATION.CON
2010 BK = 2:LB = 279:CD =
     WR
2020 GOSUB 380:CD = RS:
     GOSUB 30
2030 GOSUB 380:CD = WR:
     GOSUB 30
2040 BK = PEEK (BLK) +
     PEEK (BLK + 1) * 256: IF
     BK < LB THEN 2020
2050 A$ = "DONE◊WITH◊COPY"
10010 PRINT CHR$ (4):
      "BLOAD◊DRVR.WORD-A-
      MATION,A$1900"
```

Checksums

2000-$DF03	2030-$7703	10010-$DEC6
2010-$E75A	2040-$9DF2	
2020-$F719	2050-$FFF2	

7. Use Copy][Plus to copy all of the files, except for the ProDOS file, from the copy you just made to your blank, ProDOS formatted disk. Next copy a normal ProDOS file onto the disk you just copied all of the files onto.

8. Rename the disk to the same volume name as the disk that was made with the ProDOS Super IOB.

You're all done.

Softkey for...

Read, Write & Publish
D.C. Heath

Requirements:
The original disks
A blank disk formatted in ProDOS (format the disk with the volume name /WORK)
4 additional blank disk sides
A ProDOS formatted disk with only ProDOS 8 and Basic System on it.
A blank work disk
ProDOS Super IOB
Copy][Plus
The controller listed below
A way to reset into the monitor
A normal ProDOS file

This program will also softkey very easily using Bill Jetzer's ProDOS Super IOB.

1. Boot the original disk and write down what version number of ProDOS is being used. Then when you see the Applesoft prompt press CONTROL-RESET to stop the boot.

2. Enter the following. The routine you will enter will move the entire language card, except for bank 2, into memory at $8000.
```
CALL-151
300:AD 88 C0 A2 00 BD 00 D0
308:9D 00 80 E8 D0 F7 EE 0A
310:03 EE 07 03 D0 EF AD 8A
318:C0 60
300G
```

If after doing 300G, and pressing RETURN the computer freezes just press CONTROL-RESET to get back to the Applesoft prompt and then enter CALL-151 to get back into the monitor.

3. If the ProDOS version on the original disk was 1.0 do this next step. If it was 1.1 or newer skip to step 4.
```
8000<A800.AEFFM
```

4. Put in the ProDOS disk that contains only ProDOS and Basic System and boot it.
```
C600G
```

5. At the Applesoft prompt put in the disk that contains ProDOS Super IOB and enter:
```
CALL-151
1900 < 8000.86FFM
BSAVE DRVR.READ, WRITE &
   PUBLISH, A$1900, L$700
```

6. Install the controller into ProDOS Super IOB, run it and copy your original disk to your blank work disk.

CONTROLLER

```
2000 REM READ, WRITE &
     PUBLISH.CON
2010 BK = 2:LB = 279:CD =
     WR
2020 GOSUB 380:CD = RS:
     GOSUB 30
2030 GOSUB 380:CD = WR:
     GOSUB 30
2040 BK = PEEK (BLK) +
     PEEK (BLK + 1) * 256: IF
     BK < LB THEN 2020
2050 A$ = "DONE◊WITH◊
     COPY"
10010 PRINT CHR$ (4):
      "BLOAD◊DRVR.READ,◊WRITE
      ◊&◊PUBLISH,A$1900"
```

Checksums

2000-$DF03	2030-$7703	10010-$D048
2010-$E75A	2040-$9DF2	
2020-$F719	2050-$FFF2	

7. Use Copy][Plus to copy all of the files, except for the ProDOS file, from the copy you just made to your blank, ProDOS formatted disk. Next copy a normal ProDOS file onto the disk you just copied all of the files onto.

8. Rename the disk to the same volume name as the disk that was made with the ProDOS Super IOB.

9. Use any fast copy program to copy both double sided data disks.

You're finished.

Softkey for...

Muppet Word Book
Sunburst Communications

Requirements:
The original disk
A blank disk formatted in ProDOS (format the disk with the volume name /WORK)
A ProDOS formatted disk with only ProDOS 8 and Basic System on it.
A blank work disk
ProDOS Super IOB
Copy][Plus
The controller listed below
A way to reset into the monitor
A normal ProDOS file

Here is still another Sunburst product that will softkey easily using ProDOS Super IOB.

1. Boot the original disk and write down what version number of ProDOS is being used. Then when you see the Applesoft prompt press CONTROL-RESET to stop the boot.

2. Enter the following. The routine you will enter will move

the entire language card, except for bank 2, into memory at $8000.
```
CALL-151
300:AD 88 C0 A2 00 BD 00 D0
308:9D 00 80 E8 D0 F7 EE 0A
310:03 EE 07 03 D0 EF AD 8A
318:C0 60
300G
```

If after doing 300G, and pressing RETURN the computer freezes just press CONTROL-RESET to get back to the Applesoft prompt and then enter CALL-151 to get back into the monitor.

3. If the ProDOS version on the original disk was 1.0 do this next step. If it was 1.1 or newer skip to step 4.
```
8000<A800.AEFFM
```

4. Put in the ProDOS disk that contains only ProDOS and Basic System and boot it.
```
C600G
```

5. At the Applesoft prompt put in the disk that contains ProDOS Super IOB and enter:
```
CALL-151
1900 < 8000.86FFM
BSAVE DRVR.MUPPET WORD
   BOOK, A$1900, L$700
```

6. Install the controller into ProDOS Super IOB, run it and copy your original disk to your blank work disk.

CONTROLLER

```
2000 REM MUPPET WORD BOOK
2010 BK = 2:LB = 279:CD =
     WR
2020 GOSUB 380:CD = RS:
     GOSUB 30
2030 GOSUB 380:CD = WR:
     GOSUB 30
2040 BK = PEEK (BLK) +
     PEEK (BLK + 1) * 256: IF
     BK < LB THEN 2020
2050 A$ = "DONE◊WITH◊COPY"
10010 PRINT CHR$ (4):
      "BLOAD◊DRVR.MUPPET◊WORD◊
      BOOK,A$1900"
```

Checksums

2000-$DF03	2030-$7703	10010-$3425
2010-$E75A	2040-$9DF2	
2020-$F719	2050-$FFF2	

7. Use Copy][Plus to copy all of the files, except for the ProDOS file, from the copy you just made to your blank, ProDOS formatted disk. Next copy a normal ProDOS file onto the disk you just copied all of the files onto.

8. Rename the disk to the same volume name as the disk that was made with the ProDOS Super IOB.

You're all done.

Softkey for...

The Labeler
Sophisticated Software

Requirements:
The original disk
A blank disk
DOS 3.3 system master
A way to reset into the monitor
Copy][Plus
A blank initialized slave disk with no HELLO program

In this softkey I will tell you how to deprotect this disk on a //gs with the ROM revision 01 and on a 128K //e, //c, or //gs

with the original ROM, (if when you first turn on your //gs it has 01 at the bottom of the screen you have the revision, if it doesn't have anything at the bottom of the screen it is the original ROM.)

For the //gs with revision 01 ROM do the following.

1. Boot your DOS 3.3 system master and at the Applesoft prompt put in your blank disk and enter:
```
INIT HELLO
DELETE HELLO
```

2. Use Copy][Plus to copy all of the files from your original disk to your initialized disk except the HELLO file.

3. Boot your original disk. After the program loads reset into the monitor and enter:
```
1/800 < 0/800.4000M
0/AF.B0  write down what the two bytes
         are at AF and B0
```

3. Boot your slave disk.
```
C600G
```

4. When you get the Applesoft prompt enter:
```
CALL-151
0/800 < 1/800.4000M
0/AF: ?? ??  replace the ?? ?? with the
            two bytes that you wrote down
CTRL C       hold down the CONTROL key,
            press the C key and then press RE-
            TURN
```

5. Put in the disk that you copied the files to and enter:
```
SAVE HELLO
```

That's it.

For the //gs with the original ROM, //c, and a 128K //e do the following.

1. Boot your DOS 3.3 system master and at the Applesoft prompt put in your blank disk and enter:
```
INIT HELLO
DELETE HELLO
```

2. Use Copy][Plus to copy all of the files from your original disk to your initialized disk except the HELLO file.

3. Boot your slave disk. At the Applesoft prompt enter:
```
PR#3
CALL-151
0:8D 03 C0 8D 05 C0 4C 00 C6
```

4. Put in your original disk and enter:
```
0G
```

5. When the drive stops press CTRL reset and enter:
```
CALL-151
300:18 4C 11 C3
3F8:4C 00 03
800 < 800.3FFF CTRL Y  don't type "ctrl
      Y", hold down the CONTROL key,
      press the Y key and then press return
AF < AF.B0 CTRL Y
CTRL C
```

6. Put in the disk that you copied the files to and enter:
```
SAVE HELLO
```

That's it.

Softkey for...

Microzine #27
Scholastic

Requirements:
The original disks
4 blank disk sides

COPYA from your DOS 3.3 system master

Any fast copy program

The protected disk has altered epilogues and the first byte of the address header changes on every other track from D5 to D4.

1. Fast copy both sides of disk one. This disk is not protected.

2. Boot your DOS 3.3 system master and at the Applesoft prompt enter:

CALL-151
B942:18 *ignore checksums and epilogues*
B954:29 00 *ignore the first byte of the address header*
3D0G
RUN COPYA

3. Copy both sides of disk 2 to two blank disk sides.

No other changes are necessary.

Softkey for...

M-ss-ng L-nks English Editor
M-ss-ng L-nks Classics Old and New
M-ss-ng L-nks Young Peoples Literature
Word Quest
A Wrinkle in Time
The Puzzler
High Wire Logic
Getting Ready to Read and Add
Sunburst Communications

Requirements:
The original disk
A blank disk for each program
DOS 3.3 system master
Demuffin Plus
A way to reset into the monitor
A blank INITialized slave disk

These Sunburst titles can be deprotected by using Demuffin Plus and the RWTS of the protected disk to read the files on the original disk then write the information back to your blank disk.

1. Boot your DOS 3.3 system master and INITialize your blank disk by entering the following.

Note: Not all of the programs use the same greeting program name so check the following list.

For:
M-ss-ng L-nks English Editor

INIT HELLO@
DELETE HELLO@

For:
M-ss-ng L-nks Classics Old and New
M-ss-ng L-nks Young Peoples Literature
A Wrinkle in Time
High Wire Logic
Getting Ready to Read and Add

INIT LOGO
DELETE LOGO

For:
Word Quest
The Puzzler

INIT HELLO
DELETE HELLO

2. Boot your original disk and reset into the monitor after the Applesoft prompt appears.

3. Move the RWTS to a safe place, so it won't be destroyed when you boot your slave disk, by entering:

6800<B800.BFFFM

4. Boot your slave disk
C600G

5. Put in your Demuffin Plus disk and enter

CALL-151 *gets you into the monitor*
BLOAD DEMUFFIN PLUS,A$803
FF59G *if you hear a beep ignore it*
B800<6800.6FFFM *moves the protected RWTS back to its original location*
A851G *reconnects DOS*
803G *starts Demuffin Plus*

5. Put your original disk in and copy all of the files to your blank initialized disk. When asked to put in a file name enter = and when asked if you want prompting answer N.

Softkey for...

Computers in Government
Writing an Opinion Paper
Polls and Politics
Energy House
Student Stories
Ducks
Pets, Ltd
Classification Volume 1
Health Maintenance Volume 1
Health Maintenance Volume 2
MECC

Requirements:
The original disk(s)
A blank initialized disk for each title
A blank initialized slave disk
Super IOB v1.5
A way to reset into the monitor
The controller listed below

To deprotect these MECC titles we will use Super IOB with the Swap Controller and use the RWTS of the protected disk to read the original disk then use a normal RWTS to write the information back to your blank disk.

1. INITialize your blank disk(s)
INIT HELLO
DELETE HELLO

2. Boot your original disk and at the Applesoft prompt reset into the monitor.

3. Move the RWTS to a safe place by entering:
1900<B800.BFFFM

4. Put in your slave disk and enter:
C600G

5. After the disk boots and the Applesoft prompt appears insert your Super IOB disk and save the RWTS to it by entering:
BSAVE RWTS.MECC, A$1900, L$800

6. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer NO when asked if you want to INITialize the blank disk.

CONTROLLER

```
1000 REM MECC.CON
1010 TK = 3:ST = 0:LT =
    35:CD = WR
1020 T1 = TK: GOSUB 490:
    GOSUB 360: ONERR GOTO
    550
1030 GOSUB 430: GOSUB
    100:ST = ST + 1: IF ST <
    DOS THEN 1030
1040 IF BF THEN 1060
1050 ST = 0:TK = TK + 1:
    IF TK < LT THEN 1030
1060 GOSUB 490:TK = T1:ST
    = 0: GOSUB 360
1070 GOSUB 430: GOSUB
    100:ST = ST + 1: IF ST <
    DOS THEN 1070
1080 ST = 0:TK = TK + 1:
    IF BF = 0 AND TK < LT
    THEN 1070
1090 IF TK < LT THEN
    10201100 HOME : PRINT
    "COPY◊DONE" : END
10010 IF PEEK (6400) < >
    162 THEN PRINT CHR$ (4)
    "BLOAD◊RWTS.MECC,A$1900"
```

Checksums

1000-$356B	1040-$6342	1080-$6CA2
1010-$3565	1050-$ABA3	1090-$496E
1020-$6170	1060-$20C0	10010-$2D2E
1030-$7771	1070-$28C5	

Softkey for...

Creature Creator
Designware

Requirements:
The original disk
A blank disk
A sector editor that has search capability
Any fast copy program

1. Fast copy the original to your blank disk.

2. With the sector editor make the following edit to your copy.

Trk	Sct	Byte	From	To
$00	$03	$64-65	A5 1E	A9 F0

4. Write the sector back to the disk

Softkey for...

The Flying Carpet
Learning Technologies

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master
A sector editor

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24
RUN COPYA

2. Copy your original to your blank disk.

3. With the sector editor make the following edit to your copy.

Trk	Sct	Byte	From	To
$11	$00	$01	11	03
$20	$02	$75-76	D0 ED	18 60

4. Write the sector back to the disk

Softkey for...

Let's Go Fishing
Learning Technologies

Requirements:
The original disk
A blank disk
COPYA from your DOS 3.3 system master
A sector editor

1. Boot your DOS 3.3 system master and at the Applesoft prompt enter:
POKE 47426,24
RUN COPYA

2. Copy your original to your blank disk.

3. With the sector editor make the following edit to your copy.

Trk	Sct	Byte	From	To
$11	$00	$01	11	03
$09	$09	$75	D0 ED	18 60

4. Write the sector back to the disk

Softkey for...

Fun from A - Z
MECC

Requirements:
The original disk
A blank disk
A sector editor
Super IOB v1.5
The controller shown below

1. Install the controller into Super IOB, run it and copy your original disk to your blank disk. Answer YES when asked if you want to INITialize the blank disk.

CONTROLLER

```
1000 REM FUN FROM A - Z
1010 TK = 0:LT = 1:ST =
    15:LS = 15:CD = WR:FAST
    = 1
1020 GOSUB 430: GOSUB 490:
    GOSUB 610:T1 = TK:TK =
    PEEK (TRK):TK = T1
1030 GOSUB 430: GOSUB 490:
    GOSUB 610: IF PEEK (TRK)
    = LT THEN 1050
1040 TK = PEEK (TRK):ST =
    PEEK (SCT): GOTO 1020
1050 TK = 1:LT = 35:ST =
    15:LS = 15:CD = WR:FAST
    = 1
1060 RESTORE : GOSUB 190:
    GOSUB 210: GOSUB 430:
    GOSUB 490: GOSUB 610
1070 GOSUB 230: GOSUB 430:
    GOSUB 490: GOSUB 610: IF
    PEEK (TRK) = LT THEN
    1090
1080 TK = PEEK (TRK):ST =
    PEEK (SCT): GOTO 1060
1090 HOME : PRINT
    "COPY◊DONE" : END
5000 DATA 170,213,173,
    213,150,170
```

Checksums

1000-$356B	1040-$AB41	1080-$1C99
1010-$EA41	1050-$A4EA	1090-$A212
1020-$9224	1060-$36FF	5000-$083D
1030-$A1B2	1070-$03CB	

2. Make the following sector edits to your copy.

Trk	Sct	Byte	From	To
$00	$02	$58	96	AA
$00	$02	$5D	AA	AD
$00	$02	$F1	96	AA
$00	$02	$FC	AA	AD
$00	$03	$55	AA	D5
$00	$03	$5F	D5	AA
$00	$03	$6A	AD	96
$00	$06	$7A	AA	D5
$00	$06	$7F	D5	AA
$00	$06	$84	AD	96

3. Write each sector to the disk before going to the next sector.

The Basics of Kracking

Part #16

Softkey for...

Kameari

Ado-Soft (Japan)

Both because Mr. KRAC-MAN was generous enough to send me an original of the program, and because the original puckman was the first real protected disk I ever broke, it was fun to get my hands on the new "super puckman", or Kameari. It turned out to be not quite as hard to break as the first one, but it provided some interesting challenges.

If you catch the program at the right point, it's small enough to fit in a normal DOS BFILE, so we won't get to go through the theory and practice of program packing on this one. The sequencing used to load the game and access the disk later is a little unusual, and would have been tougher if they hadn't made a few mistakes.

A first-stage boot trace reveals the first interesting trick—the contents of $814-$8FF are eXclusive-OR'ed (XOR) with the address low byte and stuffed into page one with some cute code:

```
0801-    LDX    $26
0803-    TXS
0804-    DEC    $27
0806-    LDA    ($26),Y
0808-    EOR    $26
080A-    TSX
080B-    PHA
080C-    DEC    $26
080E-    CPX    #$14
0810-    BNE    $806
0812-    RTS
```

Now, this is not bad for the first part of a protection scheme, because it requires a reasonable knowledge of the DOS boot process as well as 6502 stack/page one usage. The tricks are:

• You have to know (or guess) that $26 contains 0 and $27 contains 9 after the first stage boot.

• You have to understand how the indexed, indirect load works at $806.

• You need an understanding of the TSX and TXS instructions.

• You need to interpret the final RTS correctly.

(If you know all these, skip this explanation and go on to the meat of the protection scheme below).

In the order listed above, location $26 contains 0 from the boot ROM at location $C652, where the accumulator was stored there after calling the "WAIT" routine at $FCA8 (acc = 0 on exit from "WAIT"). Location $27 is the high byte of the two-byte storage pointer, and it is incremented from $08 to $09 in case there's more than one sector to load in on the first stage boot. LDA ($26),Y means look at the location pointed to by $26 and $27, add the contents of the Y-register to it, and load the accumulator with the contents of that location: If $26=32, $27=08, and the Y-reg=17, the address is $832 + $17, or $849. Next, as those of you who stayed awake through the description of the stack and stack pointer in the Arcade Machine file will recall, the TSX and TXS instructions refer to transferring a byte between the ->stack pointer<- and the X-register, not between the stack and the register.

The first byte fetched from $26 through the X-reg is used to initialize the stack pointer at $00, meaning that the next byte pushed on the stack will be placed in location $100. Since the stack pointer is a nine bit hardware register with the most significant bit set, it will always contain a value between $100 and $1FF. If you "push" (PHA) another byte onto the stack, it goes not into $FF, but into $1FF. Successive bytes go into $1FE, $1FD, etc. This is known as "stack wraparound", and was used by IDSI in their "JUGGLER" protection, among others. After the first time through, each byte from $8FF down to $814 is exclusive-ored with the address low byte ($FF-$14), and pushed on the stack in the corresponding location from $1FF to $114. Each time through, the stack pointer is loaded into the X-register to compare it with number #$14 to find out if enough bytes have been transferred. When $14 is found, they do an "RTS". this takes the two bytes above the stack pointer, increments the low byte, and places them into the program counter. The program continues to run at the new location (a variation of "jumping through the stack").

The new starting location is $116 (it may seem a little strange to execute code out of what is normally the stack page, but there's nothing illegal about it. Applesoft, in fact, has a short subroutine called "CHRGET" at $B1-C8 in zero page!). Briefly, the program runs a checksum on $120-$1FF to make sure those nasty Krackists haven't changed anything, then clears all of memory from $800-$B7FF. After setting up the screen to view Hires Page two (so you can't see the read routine loading in across the text screen memory), track 0 of the disk is searched for the byte sequence "DD AD DA." Astute readers of this column will recall that this is the old Sirius trademark, and just the beginning of the rip-off of Sirius protection techniques used by the publisher (apparently, it's all right to plagiarize code from a competitor's protection scheme, but not to make backup copies of software protected with the stolen code!). The real loader program is loaded into $400-$7FF (remember Cyclod and Friends?), and after checking for a single epilog byte of $EE on the track, we do a checksum on zero page and jump to $400 with the checksum byte in the accumulator. The old "4+4" nibblizing from SIRIUS is used, and the program is contained in a single record which is $800 nibbles long and follows sector 0, which is in normal DOS format, on track 0 (the nice things about 4+4 nibblizing are that individual bytes can be located and changed, as described in the "WAY OUT" file, and the number of nibbles is always exactly equal to twice the number of bytes in the record).

At $400, the checksum of zero page is repeated and compared (they only need to be the same), and there is a bunch of language card deprotection and checking of the reset and NMI vectors. If any of the checks fail, an error message is printed and the illegal opcode $12 is executed to cause the system to hang. True to the SIRIUS heritage, the loader then fills up memory by reading tracks $1-$D (twelve pages each) into $0800-$A3FF, using an address marker of "DD AD DA" and the $EE epilog byte. After jumping to $612, the main screen is moved from $8000-$9FFF to $4000-$5FFF, and the main program is entered at $800. Source code for the reader is shown below:

```
         ORG    $0579
         STA    $05       ;destination
         PHA              ;high byte
         LDY    H03FE     ;Number of
         STY    $06       ;pages to read.
         LDY    #$00      ;Clear destination
         STY    $04       ;low byte.
         LDX    H03FF
H0588    LDA    HC08C,X   ;Begin to search
         BPL    H0588     ;for the "DD AD DA"
H058D    CMP    #$DD      ;sequence.
         BNE    H0588
H0591    LDA    HC08C,X
         BPL    H0591
         CMP    #$AD
         BNE    H058D
H059A    LDA    HC08C,X
         BPL    H059A
         CMP    #$DA
         BNE    H058D     ;After header,
H05A3    LDA    HC08C,X   ;get the first
         BPL    H05A3     ;nibble, set the
         SEC              ;carry, rotate
         ROL              ;left, and store
         STA    $0F       ;it in $0F.
H05AC    LDA    HC08C,X   ;Get the second
         BPL    H05AC     ;nibble: AND it
         AND    $0F       ;with the first.
         STA    ($04),Y   ;Store complete
         INY              ;byte and go on
         BNE    H05A3     ;to the next.
         INC    $05       ;Dest'n address
         DEC    $06       ;Page counter
         BNE    H05A3
H05BE    LDA    HC08C,X   ;Check for
         BPL    H05BE     ;epilog byte
         CMP    #$EE
```

```
         BNE    H0578
         PLA
```

At this point, all the program resides in $0000-$8000, so it's a good place to interrupt and save it. It seems strange that, with all the other SIRIUS-type protection, there's no checksum on the loader, so we can go in and change bytes on a copy of the disk. It's easy to copy the disk by using Nibbles Away with an address marker of "DD AD DA" for tracks $0-$E, but you can actually get by with only copying track 0 onto a separate disk (NA or Locksmith will both copy it without parms, since it is a standard DOS 3.3 sector on it). There is no disk error handling, so a disk with only track zero on it just sits and spins, allowing you to remove it and insert the original to load in tracks $1-$D. As in days of old, read in track zero using the track editor from Nibbles Away, then type "Z" to make it analyze the track. Set the display to the pointer page with "G6800", then search for the string "AA EE AA AA AE AA", which is "4C 00 08" or "JMP $0800" in 4+4 nibblize. Change this to "AE EE AE FB FF FF", which means "JMP $FF59", or "AE EE EE EF FF FE" which is "JMP $FECD" for use with a KrakROM (the right one to use here is KRAKROM4, since $2000-$3FFF contains program code and $4000-$5FFF has only a Hi-Res picture). Write the altered track to a blank disk with the "W" command.

Boot the new disk, and when it spins, insert the original. after the normal load, the banner will be displayed for about 5 seconds before your modification at $66E redirects the program into the monitor. Assuming that you used a Krakrom, the entire program is now contained in $900-$7FFF and can be saved as a BFILE after booting a slave diskette.

Modifications are easy now, and this is one set of "cleanup" activities that will bring the program into condition to BRUN:

Boot a slave disk, then move page 8 back from $4800-$48FF.

Move the stored zero page memory from $4000-$40FF to $8000-$80FF.

Write a memory move routine at $8050 which will restore zero page to $0-$FF (see below). don't forget to set up Hi-Res page 2 and clear the keyboard strobe.

Replace the picture in $4000-$5FFF with one containing your own advertising (you can reset the original after the boot and save the picture as a binary file for modification).

Put "4C 50 80" or "JMP $8050" at $7FD to start the program. **BSAVE KAMEARI, A$7FD, L$7880.**

Move Routine (L$38)

```
8050:A0 00 B9 00 80 99 00 00 $9BD3
8058:C8 D0 F7 A2 60 9A AD 50 $20B1
```

```
8060:C0 AD 55 C0 AD 52 C0 AD   $1A3A
8068:57 C0 AD 10 C0 A9 80 A2   $C2AD
8070:60 A0 00 4C 00 08         $B047
```

Kameari Move Routine.src

```
TXTCLR  =  $C050
HISCR   =  $C055
MIXCLR  =  $C052
HIRES   =  $C057
STROBE  =  $C010
           ORG  $8050
           LDY  #$00
H8052      LDA  H8000,Y  ;return zero
           STA  H0000,Y  ;page to $0-FF
           INY
           BNE  H8052
           LDX  #$60     ;Set up stack
           TXS           ;pointer and
           LDA  TXTCLR   ;graphics.
           LDA  HISCR
           LDA  MIXCLR
           LDA  HIRES
           LDA  STROBE
           LDA  #$80     ;Load up the
           LDX  #$60     ;registers.
           LDY  #$00
           JMP  H0800    ;Begin program
```

The resulting program will run just fine until you clear a board and advance to the next level. At that point, the disk starts to spin and the system refuses to respond to any inputs. The reason is the instruction at $B5C which jumps to $403, which jumps to $5D5:

```
           ORG  $05D5
           TYA
           PHA
           LDY  #$00
           STA  H03FE
H05DC      LDX  H03FF
           LDA  HC089,X  ;Start the drive
           LDA  #$30
           JSR  WAIT
           LDA  #$7F
           JSR  H0579    ;Read the "track"
           LDX  H03FF    ;into 7F00-up.
           LDA  HC088,X  ;Stop drive
           LDA  #$00
           TAY
H05F5      EOR  H7F00,Y  ;Checksum 7F00-7FFF
           INY
           BNE  H05F5
           CMP  #$44
           BNE  H05DC
           JSR  H7F00    ;Do subroutine
           LDY  #$00
H0604      CLC
           ADC  #$45     ;Wipe out
           STA  H7F00,Y  ;the code so
           INY           ;it must be read
           BPL  H0604    ;in each time.
           PLA
           TAY
           JMP  H0CE8
```

This routine loads the single page contained on track $E into $7F00-$7FFF, executes the subroutine at $7F00, and mangles the code in page $7F for good measure. By loading the code in once and nop'ing the mangle routine, you can avoid the unnecessary disk access and have a 122-sector KAMEARI program to use as you see fit. Change $B5C from "4C 03 04" to "4C 80 1A", and put this short substitute routine at $1A80:

Kameari Substitute Routine (L$10)

```
1A80:98 48 20 00 7F 68 A8 4C   $CFE9
1A88:E8 0C                      $0919
```

Kameari Substitute Routine.src

```
           ORG  $1A80
           TYA
           PHA
           JSR  H7F00
           PLA
           TAY
           JMP  H0CE8
```

Adding a "Pause" routine to Kameari

Advanced Playing Technique for...

Kameari

Ado Soft

Requirements:
Softkeyed version

Kameari is a decent enough Pacman, but it lacks the "pause" control with the escape key that's become standard in games from the U.S.A. You can add one by changing locations $1717-1719 to "4C 40 14" (JMP $1440), and adding this short routine at $1440:

Pause Routine(L$27)

```
1440:C9 9B F0 07 C9 CB D0 12   $F532
1448:4C 5D 17 AD 10 C0 AD 00   $F676
1450:C0 10 FB C9 9B D0 F7 AD   $3DE9
1458:10 C0 60                  $E5C3
```

Kameari Pause Routine.src

```
H175D   =  $175D
STROBE  =  $C010
KEY     =  $C000
           ORG  $1440
           CMP  #$9B     ;Was it "esc"?
           BEQ  H144B
           CMP  #$CB     ;No, check for K
           BNE  H145A    ;Nothing, so exit.
           JMP  H175D    ;It was K, goto 175D.
H144B      LDA  STROBE   ;It was esc, clear
H144E      LDA  KEY      ;the strobe and
           BPL  H144E    ;wait for another
           CMP  #$9B     ;"esc" to be pressed.
           BNE  H144E
           LDA  STROBE   ;Must clear here!
H145A      RTS
```

It's a pleasant bit of nostalgia to see someone using the old techniques with a new twist, and it provides us a chance to review some of the Kracking approaches that used to be "State-of-the-Art." See you in a "bit" or so.

Unknown

Diagram Master v5.0

?

If you, like myself paid out your hard earned cash for diagram master only to find out it's install to hard disk will not work on any drive but C: (contrary to what their support staff says), Follow the following procedure and remove the silly copy protect altogether:

1. Make a copy of DM.EXE
2. Rename it to DM.XXX
3. Use DEBUG

DEBUG DM.XXX
D CS:3FC7

You should see the following:
75 68 ...

If not then you've got a different version than 5.0 and this won't work.

4. If you see 75 68, then
E CS:3FC7 EB 68
W
Q
RENAME DM.XXX DM.EXE

You now have a copy of Diagram Master without the copy protect.

Chessmaster 2000 v1.01

?

Kids, this one is real simple. The program disk has on it a hidden file which is formatted in weird and wonderful ways, ways that your common, ordinary, run of the mill copy programs can't handle. I wasn't even able to get CopyIIPC or CopyWrite to work on this one. But then, I have a strange home-brew computer. Anyway, the program CM.EXE on your Chessmaster disk looks for verify failures from these bizarre sectors, and when found, runs the program. If the sectors aren't found then the program runs as a demo, and shuts down after a few minutes (nice touch, more genteel than calling you a slimy bastard and disemboweling your hard disk).

1. Make a copy of the distribution disk using DISKCOPY. Ignore the dire warnings of unrecoverable read/write errors. The copy will have all normal DOS sectors, with all of the necessary files on it. Only the weird stuff won't get copied, but we don't need it. Put your original disk away in a nice safe place where the Thought Police won't get at it.
2. Rename the file CM.EXE to something like CM.XXX, so we can use DEBUG. Startup DEBUG.

DEBUG CM.XXX

3. Search for the byte string B8 00 00 EB E7.

S 0100 FFFF B8 00 00 EB E7

XXXX : YYYY *debug gives you these* #

This looks for a mov ax,0000 instruction that gets executed if the program fails to find the weird sectors.

4. Using the numbers from step 3, assemble a new instruction here to fool the program.

A XXXX:YYYY
XXXX:YYYY MOV AX,0001
XXXX:YYYY *press <enter> key to exit Assem.*

This instruction will return a code that indicates the presence of weirdness, when in fact the sectors are normal.

5. Write the program back to disk.

W *Writing ZZZZZ bytes*
Q

6. Rename the file back to .EXE and you now have a copy of Chessmaster that will run from the hard disk, and that can be copied for archival purposes

The Dambusters

Accolade

Use the following debug procedure. Note that even recent version versions of CopyIIPC cannot copy this software, and to protect your investment you will want to use this unprotect.

1. Copy DAMB.EXE to a formatted blank diskette. Then:
REN DAMB.EXE DAMB.BIN
DEBUG DAMB.BIN

2. Now search for the use of INT 13
S 0000 FFFF CD 13

I found this call at 13BC:0116 You may find it at a different location, but as long as the number ends in 16 you have the correct location. If it doesn't end with a 16 then you may have a different version and this unprotect may not work.

3. Next look at the code after this call
U xxxx:xx16 *the number from above*

You should see the following:

```
XXXX:XX16 CD13 INT  13
XXXX:XX18 7306 JNB  0120
```
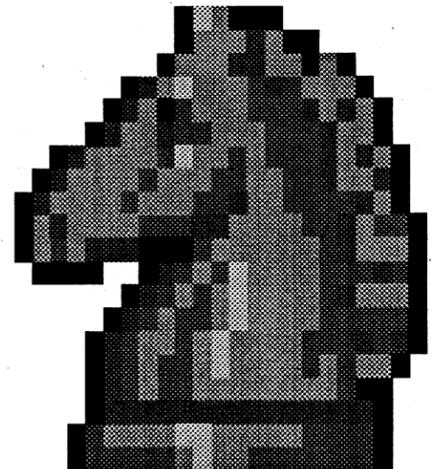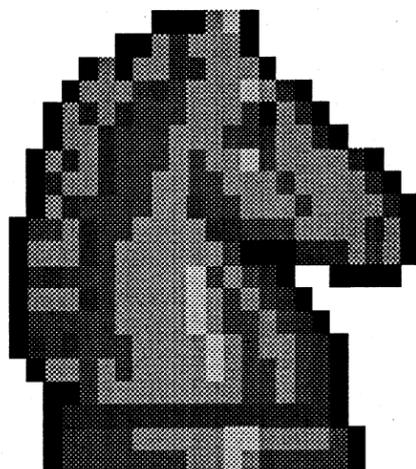
If this is different, you may have a different version. Stop now.

4. Now change the jump to an unconditional jump
E xxxx:xx18 EB 06

5. Now write the changes back to disk and end the debug session
W
Q
REN DAMB.BIN DAMB.EXE

Now place your original diskette in a safe place.

unClassifieds

How to place an UnClassified Ad

Send a typed sample copy with appropriate instructions. (If possible, send text on a 5.25" Apple format disk.) Use up to 40 characters per line, we will adjust word wrap.

Special Graphics Instructions: The first three words of the first line are printed in bold for free. If you want other words bolded, use 5 characters less per line. Use 10 characters less per line if you have a lot of uppercase bold letters. Bold letters are wider than normal. If the typed copy does not show bold, circle the words you want bolded and, on the side, write BOLD. If you want a line centered, write **CENTER** next to that line. There is no charge for centering any line.

You must check your ad for errors, the first time it runs. Errors on our part will be corrected, then, for free. Errors or changes on your part will be charged a $5 processing fee.

★★★★ New Rates (per line) ★★★★
Computist club member .... 25¢
All others .... 35¢

The minimum order is $5.

- Our liability for errors or omissions is limited to the cost of the ad.
- We reserve the right to refuse any ad.
- Washington state residents add 7.8% sales tax.
- Send a check or money order (funds drawn on US bank only) for the entire amount to:

COMPUTIST unCLASSIFIEDS
33821 East Orville Road
Eatonville, WA 98328

WANTED

"Most Wanted List" Software

Need help to deprotect a disk?

Softkey hobbist is interested in acquiring copy protected software to deprotect. Good track record, many successful attempts. Original disk will be returned along with softkey for COMPUTIST. Especially interested in older software (pre-1988) but will give any disk a shot. I'm especially interested in:

Drol ---Broderbund
Serpentine ---Broderbund
Spare Change ---Broderbund
Wings of Fury ---Broderbund
Star Cruiser ---Sirius
Space Eggs ---Sirius
Falcons ---Picadilly
Microwave ---Cavalier

System: Apple IIe, 128K. Send disk to:

Rich Etarip
824 William Charles, Apt #2
Green Bay, WI 54304

For Sale:

Complete collection of COMPUTIST. From #1 thru 83 except #27 & #43. It is in excellent condition except for two issues that have slight tears and a few have slight pen marks. I am asking $80 for the set and will not sell single issues.

Ronald H Stein
126 Roxbury Park
East Amherst NY 14051

Wanted! Information

on Cracking IBM protection. Also software like Copywrite, Locksmith PC, Central Point Software's PC Option board and PC Watch. Also send list of Apple II items for sale.

Joe Torzewski
51625 Chestnut Road
Granger IN 46530

Apple II Nerd?

The **Ultimate** reference library!
Complete Call A.P.P.L.E.!
Complete InSider!
Complete A+!
Complete COMPUTIST
All 10 Nibble Expresses!
Many, many extras!

Shipping alone would cost a fortune. The highest bid received by year-end 1992 takes them all. Send to:

The Amazing Spiderman
PO Box 4851
Carson CA 90745

For sale:

Applied Engineering RamKeeper. in good working order. Includes: cable, transformer, 5hr battery, manual and Appleworks expansion software. All reasonable offers will be considered.

Kip Harker
P.O. Box 600
Eagle River AK 99577

RDEX Contributors

Apple Most Wanted

IBM Most Wanted

79 The Product Monitor •Bitkeys: •Kabul Spy •Softkeys: •ABM •Algebra 1-6 •Cause and Effect •Chemistry: Series I •Computer Generated Mathematics Vol. 2 •Cribbage •Designer Puzzles •Dungeon Master Assistant Vol. 2 •Economics •Genesis •Gin King •Go •Graphmaster •Hard Hat Mack •Hi Res Computer Golf •Integer Arcade •Laser Bounce •Mammals Reptiles and Insects •MasterGrades •Mickey's Crossword Puzzle Maker •Mind Benders •Missing Links •Non-Western Cultures •RoboCOP •Safari Search •SAT Score Improvement Series •Special Product and Algebraic Factors •Stickybear GS Talking series -Talking Alphabet-Talking Opposites-Talking Shapes •Task Force •Teacher's Toolkit version 3.1 •The Great Knowledge Race •The History of Europe •The Solar System •The Time Tunnel •Time Thief •TrianGO •US History •Wasteland •Water and Weather •Who Am I? •Word Problems for Algebra •Worksheet Generator •Writing Chemical Formulas •Your Body •Your Body: Series II •Playing Tips: •Baneful Tales •Elite •Mac Features: •Mac Hard Disk Ejection Fix •Mac Softkeys and other Patches: •ABCBase •Animation Toolkit1 •Aztec C 1.0 •Aztec C version 1.00c •Championship Boxing •Chart •Checkminder •Cutthroats •Cutthroats alternate •Deja Vu •DeskToppers •Dollars & Sense •Dollars & Sense alternate •Electric Checkbook •Excel •Excel alternate fix •Fact Finder 1.0 •Factfinder •Farenheit 451 •Feathers & Space •File •FileMaker •Filevision •Filevision alternate •Forecast •Frogger •FunPak •Gato •Grid Wars •Griffin Terminal •Haba-Comm •Haba-Comm alternate •HabaCheckMinder •Habadex 1.1 •Harrier Strike Mission •Hayden Speller •Hayden Speller alternate •Hippo/C Level 1 •Hitchhiker's alternate •Hitchhiker's Guide to the Galaxy •Home Accountant •Legacy •Lode Runner •Mac Fortran •Macattack •MacChkrs/Rvrsi •MacCommand •MacDraft1.0 •MacDraft1.1 •MacGammon/Cribbage •MacJack/Poker II •MacLabeller •MacMatch •MacPascal (version) •MacPoker •MacType •MasterType •MasterTypealternate •MouseStampede •Multiplan alternate •Multiplan version 1.02 •OverVue •PageMaker •PageMaker 1.0 •Pensate •PFS •PFS File/Report •PFS version A.03 •Real Poker •Rogue •Sargon III •SkyFox •Smooth Talker •The Quest •Think Tank •ThinkTank 1.1 •ThinkTank 128 •ThinkTank512 •Transylvania •TriplePlay1.0 •Trivia Arcade •Trivia Fever •Typing Intrigue •Ultima][•UltimaIII •VideoWorks1.0 •WellTris •Winter Games •Xyphus •Features, Notes & such: •COPYA-able Questron II •How to make Thief into a BRUNable file •How to run Task Force on your hard drive •Making Genesis into a single BRUNable file •Making Hard Hat Mack into a single BRUNable file •Making PLATO software run on the Enhanced //e •Multi-Column Print Utility (MCP) •Notes on Battle Chess •Notes on Silent Service GS •Notes on Wildcard II card •Object Module Format (OMF) •ORCA/Disassembler Scripts •ORCA/Disassembler utilities •Other Notes •Running Teacher's Toolkit v3.1 (3.5") on a Laser 128 •Task Force on a hard drive and Wings by Vitesse •The Basics of Kracking (part 5): Deprotection of Modified DOS disks •The Basics of Kracking Part 6: Mating Zone & Nibblizing Mysteries •Update on the Silent Service GS v9.25.01 crack •Xternal Commands for BASIC: CWD (Change Working Directory)/ONLINE

80 The Product Monitor •Features, Notes & such: •Add Copy II Plus file handling to your BASIC program. See A Copy II Plus PLUS! •Comments on the Beginner's Book •Formatting 720K disks as 1.44M HD. See Holey Megabytes! •How to SAVE hexdumps as CDA's •Logging ProDOS Drives •The Basics of Kracking (part 7) •The Basics of Kracking (part 8) •Bitkeys: •Black Magic •Guild of Thieves •Gunslinger •King's Quest Series •Leisure Suit Larry •Man Hunter: New York •Police Quest •Realms of Darkness •Saracen •Sierra Boot Disks •Silicon Dreams •Space Quest Series •Ultima V •Wizardry Series •Xyphus •Softkeys: •Ancient Art of War •Battle Chess •Bridge 6.0 •Captain Blood GS •Dinosaur Days v1.0 •Empire •Fahrenheit

451 •Fay's Word Rally •GATO v1.3 •Greeting Card Maker •Hostage •Keef The Thief •Magic Spells v1.2 •MagicSpellsv2.1 •Mickey's Crossword Puzzle Maker •Monsters and Make Believe v1.1 •Pipe Dream •Pipe Dreams •Rear Guard •Rendezvous with Rama •Same or Different •Teacher's Tool Kit •Teacher's Tool Kit (IIc) •War of the Lance •Where in the USA is Carmen Sandiego? •WindwalkerGS •WindwalkerIIe •APTs: •Space Rogue •Wizardry III. See Keeping the Wiz in Wiz III •Playing Tips: •Countdown •Space Rogue •IBM Softkeys: •Serve and Volley •Welltris

81 The Product Monitor •Bitkeys: •Micro Typewriter •Softkeys: •Backyard Birds •Balance of Power •Chemistry: Balancing Equations •Chemistry: The Periodic Table •Chuck Yeager's AFT •Equation Math •Estimation: Quick Solve I •Estimation: Quick Solve II •Five-Star Forecast •Fossil Hunter •Grammar Toy Shop •Instant Survey •Micro Typewriter v4.0 •Murphy's Minerals •Patterns •Picture Chompers •Probability Lab •Professor Al's Sequencing Lab •Stickybear Shapes (ProDOS 1.5) •Studymate (the grade booster) •Sun and Seasons •The Duel: Test Drive II •Time Navigator •Tomahawk •Windwalker •APTs: •Where in Europe is Carmen Sandiego? •Where in the USA is Carmen Sandiego? •Where in the world is Carmen Sandiego? •Where in Time is Carmen Sandiego? •Playing Tips: •Windwalker •IBM Softkeys: •Crime Wave •Gauntlet II •Stunt Driver •Thexder II •Wing Commander

82 The Product Monitor •Features, Notes & such: •GS/DSP: Update on IIgs Enhancement Board •Two-World Transfers: Apple to IBM and Back •Softkeys: •Advanced Applesoft Basic •Apple Assembly Language •Bad Dudes •Bubble Bobble •Captain Goodnight •Chemistry - Balancing Equations •Chemistry - Periodic Table •Comics •CommuniKeys •County Fair •Electric Crayon (This Land is your Land) •Gorgon •Hi Res Tool Kit •High Rise •Invisible Bugs •Jawbreaker •King's Bounty •Might and Magic II •Miner's Cave •Mission Control •Mystery Matter •Mystery Objects •Note Card Maker •O'Riley's Mine •Outliner •Platoon •Risk v1.4 •Space Station Freedom •To Preserve, Protect, and Defend •Wood Car Rally •APTs: Captain Goodnight •Elite Plus •The Maps of Kings Quest V •Playing Tips: •Elite Plus •IBM Softkeys: •ABC Monday Night Football •Abrams Battle Tank •Battlehawks 1942 •Bop'n Wrestle •Centurion •Champions of Krynn •Command HQ •Indianapolis 500 •Jack Nicklaus' Greatest 18 Holes of Major Champion •Life & Death II: The Brain •Lowblow Boxing •M1 Tank Platoon •Might and Magic •Might And Magic II •Nuclear War •Pipe Dream •Populous •Railroad Tycoon •Red Storm Rising! •Silpheed v2.2 •Street Rod •Test Drive II •Vette •Wing Commander

83 The Product Monitor •Features, Notes & such: •COREEDIT •Installing Defender of the Crown on a Hard Drive •Notes on COPY II PLUS 9.0 •Notes on RAMUP - The "Cruising" Utility For RAM Cards •Notes on Teacher Grading programs •Notes on Test Writing programs •The Basics of Kracking Part 9 •Ultima V Dungeon Copier v1.0 •Softkeys: •ABM •Cribbage •Firebird •Lunar Leepers •Milliken Software •Snoggle •Solitaire •Playing Tips: •Defender of the Crown •Eye of the Beholder. See Beholder Lore •Might & Magic III. See The Lost Journals of M&M III •Spellbreaker. See Spellbreaker Break •IBM Partial Softkey: •Bargames •IBM Softkeys: •Alley Cat •Archipelagos •Balance of Power v1.10 •BattleHawks: 1942 •California Games •Carrier Command •Caveman Ugh-Lympics •Colonel's Bequest •Continuum •Crime Wave •Crimewave v1.1 •Curse of the Azure Bonds •Dragons Lair •Dragon's Lair •Dragon's Lair II •Earl Weaver's Baseball v1.5 •Earthrise •Escape From Hell •F-15 •Grand Slam Bridge •Gunship •One on One •Where in the U.S.A. is Carmen Sandiego? •Where in the World is Carmen Sandiego? •Where in Time is Carmen Sandiego? •IBM Playing Tips: •A10 Tank Killer •Space Quest IV

#84 The Product Monitor •Bugs: Another bug in

AppleWorks 3.0 •Bug in Captain Goodnight Softkey •Bug in Gorgon Softkey •Bugs in "Where In The USA Is Carmen Sandiego GS" •Features, Notes & such: A fix for "Putting... Super Boulder Dash on a Hard Disk •An ELITE Craft ...without NMI •Boot code tracing Star Maze •EZ APT's with Compare Disk program •Notes on Hacker II •Run MECC On Hard Disk •SuperBoulderdash APT Explanation •The Basics of Kracking Part 10 •WindwalkerGS Editor •Advanced Playing Techniques: Eidolon •Elite •Super Boulderdash APT Explanation •Bitkeys: MECC Copy System/Label Utility •Midwest Software •Softkeys: Alge - Blaster Plus •Arcade Machine (The) •Axis Assassin •Backyard Birds •Bandits •Bill Budge's Space Album •Borg •Chemistry: Balancing Equations •Chemistry: The Periodic Table •Cleanwater Detectives •Clock •Communikeys •Conquering Decimals (+,-) •Conquering Decimals (X,/) •Conquering Fractions (+,-) •Conquering Fractions (X,/) •Conquering Math Worksheet Generator •Conquering Percents •Conquering Ratios & Proportions •Coordinate Math •Decimal Concepts •Dungeon Master's Assistant vol2 •Early Skills (2 diskettes) •Equation Math •Estimation Quicksolve I •Estimation Quicksolve II •Estimation Strategies •Exploring Gas Laws •Five-Star-Forecast •Flip Out •Fossil Hunter •Fraction Concepts, Inc. •Fraction Practice Unlimited •Grammar Gazzette •Grammar Toy Shop •Instant Survey •Instant Survey Sampler •Invisible Bugs •Keyboarding Klass •Kinder Koncepts •LittleTown Zoo •Living Cell (The) •Lunar Greenhouse •Mastertype's Writer •Math Facts Tracker •Measureworks •MECC 3.5" ProDOS disks •MECC Outliner •Miner's Cave •Minotaur •Money •Mystery Matter •Mystery Objects •Paper Plane Pilot •Patterns •PictureChompers •Probability Lab •Problem-Solving With Nim •Rings of Saturn •Space Station Freedom •Spell It •Spellevator •Spelling Puzzles and Tests •Spelling Series ToolKit •Star Maze •Sun & Seasons •Teaching Scientific Inquiry •Time Navigator •Time Navigator Around The World •Time Navigator Leaps Back •To Preserve, Protect & Defend •Type Attack •Weeds To Trees •What's First? What's Next? •Wood Car Rally •Wooly Bounce •Playing Tips: Gemstone Healer •IBM Softkeys: Battle Chess II •Carrier Command •Colonel's Bequest •Continuum •Crime Wave •Crimewave v1.1 •Curse of the Azure Bonds •Dragon's Lair •Dragon's Lair II •Earl Weaver's Baseball v1.5 •Earthrise •Escape From Hell •F-15 •Where in U.S.A. is Carmen Sandiego?

#85 •The Product Monitor •Bugs: A bug in AppleWorks 3.0 •Bugs in the Copy II Plus •Bugs in "Where in the USA is Carmen Sandiego GS" •Features, Notes and such: A "LISTable" version of Warship & WWI Battlecruise •Copy II Plus discontinued? •Dazzle Draw Patch to Save the Configuration •Fastcopya Enhancement problem •Locksmith Fastcopy (2 GS) Help •PDOS (ProDOS RDOS) v1.1 •PDOS Examples •PDOS Updates: Changes for v1.1 of PDOS •PDOS Updates: Changes for v1.1 of RDOS Transfer •Question and Help for Wings Of Fury by Broderbund •RDOS Transfer Utility v1.1 •The Basics of Kracking Part 11 & 12 •Advanced Playing Techniques: The Duel: Test Drive II GS •Bitkeys: Midwest Software •Softkeys: Airheart •Battle Cruiser •Battle For Normandy •Battle Group •Battle of Antietam •Battle of Shiloh •Bomb Alley •Breakthrough in the Ardennes •Calendar Crafter v1.3 •Carrier Force •Cartels and Cutthroats •Clue •Computer Air Combat 1.1 •Computer Air Combat Data Disk •Computer Ambush 2.0 •Computer Baseball •Computer Bismarck 1.1 •Cosmic Balance •Cosmic Balance II •Cyclod •Cytron Masters •Eagles •Epidemic •Fifty Mission Crush •Fighter Command •Galactic Adventures •Galactic Gladiators •Geopolitique 1990 •Guadalcanal Campaign •Imperium Galactum •Kampfgruppe •Kampfgruppe Scenario Disk One •Kinder Koncepts •Knights of the Desert •Mech Brigade •Napoleon's Campaigns:1813 & 1815 •North Atlantic '86 •Objective: Kursk •Operation: Market Garden •Phantasie •President Elect •Pursuit of the Graf Spee •Reforger '88 •Ringside Seat •Risk 1.4 •Road to

Ghettysburg •Roadwar 2000 •Roadwar Europa •Six-Gun Shoot Out •The Scoop •Tigers In The Snow •Torpedo Fire •U.S.A.A.F. •War In Russia •Warp Factor •Warship •Word Attack Plus •WWI Battlecruiser •IBM Softkeys: Carrier Command •Caveman Ugh-Lympics •Colonel's Bequest •Continuum •Crime Wave •Crimewave v1.1 •Curse of the Azure Bonds •Dragon's Lair •Dragon's Lair II •Earl Weaver's Baseball v1.5 •Earthrise •Escape From Hell •F-15 •Firehawk Thexder II •Gunship •Where in the U.S.A. is Carmen Sandiego?

#86 •The Product Monitor •Bugs: Bug in Airheart softkey •Bug in Minotaur softkey •Features, Notes and such: IIgs Resources •PaintWorks Gold on a hard drive? •PC Keyboard Fix •Placing Battle Chess on a Hard Drive •The Basics of Kracking Part 13 •The Basics of Kracking Part #14 •Softkeys: Analogies Tutorial •Arithmetic Facts Learning Aid •Book Worm •Cause & Effect •Circus Math •Comparison Kitchen •Early Addition •Epoch •Extra! Extra! •Factor Opinion •Figurative Language •Foundations in State History: Focus on Indiana •Go to the Head of the Class •Indiana Jones in Revenge of the Ancients •Intellectual Software disks •Introductory Genetics •Kids at Work •Magic Candle •Mastery Arithmetic Games (1989) •Medalists: Create Your Own •Medalists: Presidents •Microcomputer Workshops disks •Microzine #19 •Milt's Math Drills •Money Works •Monsters and Make Believe •Mrs. Wigglesworth's Secret •Multiplication Puzzles •Mystery of the Witch's Shoes •Paint with Words •Parts of Speech Nouns •Parts of Speech Verbs 1 •Punctuation Skills Commas •Punctuation Skills End Marks, Semicolon, and Colon •Questron II •Quotient Quest •Rounding •SAT Score Improvement System Algebra •SAT Score Improvement System Geometry •Sea Speller •Sheila •Superstar Ice Hockey •Test Drive •The Case of the Missing Chick •The Skeletal System •Trivia Machine •Way Out •Who, What, When, Where •Word Scrambler and Spelling Tutor •IBM Softkeys: Artwork •Faces •Heat Wave •Hoverforce •Indy! The Graphic Adventure •Kings Quest III •Space Harrier •Teenage Mutant Ninja Turtles

#87 •The Product Monitor •Bugs: Bug in Algebra vol 3 & 4 •Bug in Matchmaker World Geography Facts •Bug in PSAT Word Attack Skills •Bug in PSAT/SAT Analogies •Bug in SAT Word Attack Skills •Features, Notes and such: Basics of Kracking Part #14, The •Build Your Own Elements! •Kracking SSI's RDOS •MECC 3.5" disk softkeys •MECC 5.25" disk softkeys •Notes on Test Drive II cars disks •Questron II CDA v2.0 •Softkeys: Blue Grass Bluff (5.25") •Bluegrass Bluff (3.5") •Botanist's Apprentice (The) •Cause & Effect: Mountain Climbing Blue Level •Cleanwater Detectives (3.5") •Cleanwater Detectives (5.25") •Context Clues: Hidden Treasure Red Level •Drawing Conclusions: Chief of Detectives Blue Level •Drawing Conclusions: Chief of Detectives Red Level •Estimation Activities (3.5") •Estimation Activities (5.25") •Fact or Opinion: Smart Shopper Blue Level •Fact or Opinion: Smart Shopper Red Level •Field Zoologist, The •The Following Directions: Behind the Wheel Red Level •Getting the Main Idea: Around the World Blue Level •Getting the Main Idea: Around the World Red Level •Grammar Gazette (3.5") •Grammar Gazette (5.25") •Grammar Monsters (3.5") •Inference: School Days Blue Level •Inference: School Days Red Level •Journey to the Age of the Dinosaurs •Littletown Zoo (3.5") •Littletown Zoo (5.25") •Living Cell (3.5"), The •Living Cell (5.25"), The •Outpost •Paper Plane Pilot (3.5") •Paper Plane Pilot (5.25") •Pensate •Problem Solving With Nim (3.5") •Problem Solving With Nim (5.25") •Reading for Detail: Race Track Blue Level •Reading for Detail: Race Track Red Level •Sequence: What Comes First Blue Level •Sequence: What Comes First Red Level and much more. . .

For a complete back issue list, send a 75¢ stamp to Computist.

Some disks apply to more than one issue and are shown as taller boxes.

☆ Limited supply — first-come-first-serve basis.
○ Out-of-print — only "Zeroxed" copies for sale.
* Issue 66 is laser printed on 8½ by 11 paper.