$1.50

Washington Apple Pi

Volume 3 November 1981 Number 10

Highlights

USING A FIRMWARE CARD IN SLOT 4
A SIMPLE FULLSCREEN TEXT EDITOR
SORT YOUR DIRECTORY
ON THE OVERALL STRUCTURE OF APPLESOFT

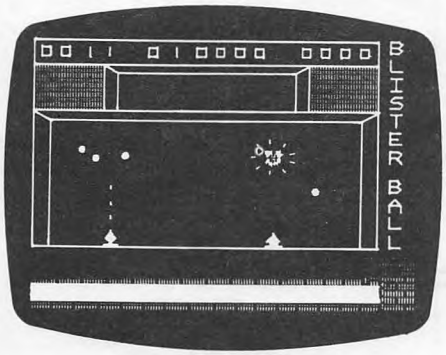
In This Issue

The BEST games are from Creative Computing Software

1978: Adventure
1979: Air Traffic Controller
1980: Super Invader
1981: Blister Ball and Mad Bomber

Blister Ball

Blister Ball is the first completely original arcade-type game for a computer. Not a copy, not an adaptation, not a spinoff. **Blister Ball** is new—it's a new idea—better than Invaders, better than Circus, better than Asteroids, better than Galaxian. If you've played other games for hours, you'll play **Blister Ball** for days.



How does it work? Well, some mean but fun-loving aliens have produced some bouncing bombs. First they drop one and you've got to position yourself under it and zap it with your laser. If you miss, that's OK. It will bounce around, although each bounce is lower, and you have several chances to zap it. Got the hang of it? OK, here come two bouncing bombs. You zap them. Then you're faced with three, then four and five.

As they bounce longer and longer the walls begin to close in so you're faced with either zapping the bombs or being hit. Each hit knocks you a little further toward the gutter. But you can survive two hits which is usually enough to zap all the bombs.

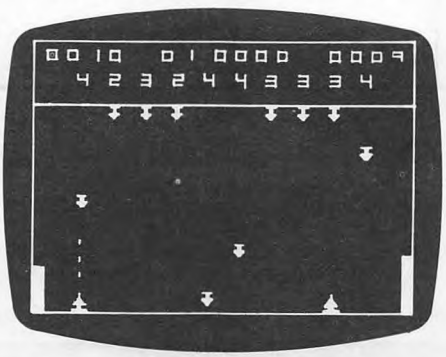
Feeling confident? Don't. Because after 5 bombs the murderous little devils drop 5 bonus bombs, worth ten times as much. These don't bounce, so you get only one shot. You need nerves of steel and the reflexes of a tail gunner.

After you complete one round,the game starts again with bombs that bounce faster and lower (and are worth more) than the previous ones.

Blister Ball is a fantastic solo game. But there are two-player options as well in which players can play as a team or as opponents. Each player can move the entire width of the screen and zap any of the bombs. Here, you're not only trying to survive, but trying to outscore your opponent. The game has two skill levels.

Mad Bomber

In **Mad Bomber** you are faced with aliens in a huge ship hovering overhead. They have bomb racks which they constantly fill with bombs. Your object is to move from side to side on the ground and zap the bombs in the bomb racks or as they fall.



As the game progresses, the aliens fill up their bomb racks more quickly and the bombs fall faster. You lose after ten bombs have hit the area which you are defending.

Mad Bomber can be played by one player solo or by two players as a team or as opponents. Two skill levels.

Order Today

Blister Ball and **Mad Bomber** are available together for $24.95 on disk (DOS 3.2) only and require a 48K Apple with paddle controls. (We recommend using the Super Paddles from Peripherals Plus).

To order send $24.95 plus $2.00 shipping and handling to the address below. Credit card customers should include card number and expiration date of Visa, MasterCard or American Express card. Credit card orders may also be called in to our toll-free number in the continental U.S.

If you also wish to order a set of Super Paddles from our Peripherals Plus subsidiary, the cost is just $39.95. The paddles are backed by a 90-day limited warranty from the manufacturer as well as Peripherals Plus' moneyback guarantee of satisfaction.

Blister Ball and **Mad Bomber** are colorful, challenging, fast and noisy. They are the games of the year from Sensational Software.

creative computing

Attn: Gretta
39 E. Hanover Avenue
Morris Plains, NJ 07950
Toll-free 800-631-8112
In NJ 201-540-0445

CREATURE VENTURE

You have just inherited your Uncle Stashbuck's mansion but first you must rid it of the horrible creatures that have taken it over and find your uncle's buried treasure.

Directing the computer with two word commands such as 'Go North', 'Get Key', 'Look Room', 'Punchout Boogeyman' etc. you will need to explore deep into the mansion to finally find the Stashbuck Fortune.

There are tons of High Resolution graphics plus some clever animation just for fun.

Requires 48K Ram, Applesoft Rom and disk.

All High Resolution characters generated with Higher Graphics II by Robert Clardy.

CREATURE VENTURE on disk

$24.95

GOBLINS

Escape to GOBLIN Country, an adventure to challenge your wits and cunning where you will need skill and sorcery to survive Dragons, Ogres and Goblins.

Myths and Magic, Treasure and Danger.

Fantastic High Resolution Graphics and animation help you on your journey.

Requires 48K, Applesoft ROM and disk

A world apart – enter if you dare!

GOBLINS on disk

$27.50

TARTURIAN

THE TARTURIAN requires 48K RAM, APPLESOFT ROM, and disk. As you explore the 160 rooms (each done in HI-RES), gathering weapons and treasure that will prepare you for the final battle against the TARTURIAN, you will encounter deadly KROLLS, battle the MINOTAUR, try and get by COUNT SNOOTTWEEKER, decipher the YUMMY YAKKY'S secret, make friends with the TULIESWEEP, avoid GHOULS, explore the PILLAR tombs, discover secret passages and more, 5 interlocking programs.

TARTURIAN on disk

$24.95

OLDORF'S REVENGE

OLDORF is a well done and exciting HI-RES game using over 100 HI-RES pictures. OLDORF requires 48K, Applesoft Rom, and Disk. As you explore the caverns and castles (each locale is done in HI-RES) looking for treasure, you must battle the one-eyed, two-thumbed torkie; find the grezzerlips' sword; visit the snotgurgle's palace and get through the domain of the three-nosed ickyup – Plus MORE!

OLDORF on disk

$19.95

CRAE

The finest Global Applesoft program editor on the market today and here's why:

Written entirely in Machine Language. Global Find (Search) for anything in your program; Global Change (Replace) that can substitute anything in your program, anything!; List 40 columns across on screen with no added spaces; The fastest renumber in the business; Quote (Copy command for moving program lines anywhere in your program; Automatic Line Numbering; HEX/ASCII Memory Dump. Plus 5 more powerful commands. CRAE co-resides in memory with your Applesoft program so that all editing of your program is done right in RAM. If you are familiar with TSO then you are familiar with CRAE. Compatible with Neil Konzen's P.L.E. Requires 48K RAM, Applesoft ROM and disk.

CRAE on disk with manual

$39.95

MCAT 2.0

MCAT 2.0 is a fast binary utility which creates a sorted master catalog which is saved on disk as a binary file (Fast). The master catalog can be easily updated a whole diskette at a time (Add, Delete, Replace), List/Print have global search capability and one or two columns. Provisions for duplicate volume numbers. Approximately 1200 file names. 48K or 32K, 13 or 16 sectors DOS supported.

MCAT on disk with 10 page manual

$24.95

SPECIALS

EROM 1, 2, 3

$99.95

CRAE and MCAT on one disk

$59.95 with manuals

EROM #1

Requires Applesoft ROM & ROM-PLUS. CRAE'S powerful Global change/find, optimizes List Command, Hex to Decimal and Decimal to Hex conversion now available on a 2716 EPROM.

EROM #1 with manual

$49.95

EROM #2

Requires Applesoft Rom and Romplus

CRAE's Autoline numbering, formatted memory Dump, Append, Number Conversion (Hex/Dec) on one 2716 EPROM.

EROM #2 with manual

$34.95

EROM #3

CRAE's powerful Renumber and Quote function now on two 2716 EPROMS.

EROM #3 with manual

$34.95

SEE YOUR LOCAL DEALER OR SEND CHECKS TO

HIGHLANDS COMPUTER SERVICES

14422 S.E. 132nd
Renton, Washington 98056
(206) 228-6691

Washington residents add 5.4% sales tax. Applesoft and Apple are registered trademarks of Apple Computers, Inc.

Romplus is a registered trademark of Mountain Computers, Inc.

OFFICERS & STAFF

President	-David Morganstein	(301) 972-4263
Vice President	-Mark Crosby	(202) 488-1980
Treasurer	-Dana Schwartz	(301) 725-6281
Secretary	-Jesse Wagstaff	(301) 937-4215
Members-at-Large	-John Moon	ABBS Sysop
	-Nancy Philipp	(301) 924-2354
	-Rich Wasserstrom	(703) 893-9147
Immed.Past Pres.	-Bernard Urban	(301) 229-3458
Editor	-Bernard Urban	(above)
Associate Editor	-Rich Wasserstrom	(above)
	-Genevie Urban	(301) 229-3458
Librarians:		
Disk Mail Order	-Sarah Lavilla	(301) 926-6355
Disk Pick-up	-Bill Bowie	(301) 924-3455
Group Purchases	-Rich Wasserstrom	(301) 621-2719
	-Bob Peck	(301) 621-2719
Advertising	-Howard Lefkowitz	(301) 649-3373
Membership	-Bob Peck	(301) 621-2719
Volunteer Coord.	-Boris Levine	(301) 229-5730
SIG Chairmen:		
ASMSIG	-Jim Rose	(301) 730-2230
EDSIG	-Chuck Philipp	(301) 924-2354
NEWSIG	-Paul Hoffman	(301) 831-7433
Pascal(PIG)	-Tom Woteki	(202) 547-0984
SIGAMES	-Al Gass	(703) 371-3560
SIG/DISABLED	-Curt Robbins	ABBS WAP428

8805 Barnsley Court
Laurel, MD 20811

Washington Apple Pi
P. O. Box 34511
Bethesda, MD 20817
(301) 621-2719

ABBS (301) 983-9317

Membership dues for Washington Apple Pi are $18.00 per year, beginning in the month joined. If you would like to join, please call the club phone and leave your name and address, or write to the PO Box above. A membership application will be mailed to you.

Members who would like to sign onto the Washington Apple Pi ABBS system should call the club phone and leave your name (first and last), WAP number and phone number. You will be assigned a password and John Moon will take care of signing you on.

EDITORIAL

You say you can't afford a text editor. Well, take a look at the one by Walter Lee. Just a few added instructions here and there (the proverbial exercise left for the reader) and you've got yourself a word processor. And it's not copy protected! You say that Call-A.P.P.L.E.'s "Applesoft in Depth" is good, but you need more detailed information. C.K. Meszytenyi's article comes to the rescue. How about a way to alphabetize your bloated diskette directories, other than recopying file by file? See Andy O'Brien for that one. If games are what you are looking for, see the reviews and comments by both Dave Morganstein and John Alden. Got a leftover firmware card now that you've got a RAM card in Slot 0? How about Dave Morganstein's solution (Slot 4)? All that and lots more in this issue.

What, you say, we are not covering your needs? Well then, get cracking and write on the subject near and dear to you so that you may inspire others to do likewise. For next month, be sure to get copy in by November 6. We have an early deadline because of the Thanksgiving Holidays.

CLASSIFIEDS

FOR SALE (group buy): DC Hayes Smart Modem, $235; Novation APPLE Cat, $300; Lobo Disk with 3.2/3.3 Controller, $510. Dave Kemp (301) 674-2690.

FOR SALE: Locksmith 4.0, $85.00. Kevin Duffy, (202) 363-6245.

EVENT QUEUE

Washington Apple Pi meets on the 4th Saturday of each month at 9:30 AM, at George Washington University, usually in Building C, on G Street at 23rd Street, NW. (To be sure of the exact location call the club phone or ABBS during the week of the meeting.) The October meeting is on the 24th.

Due to the Thanksgiving Holidays, the November meeting will be on the 3rd Saturday, November 21. A flea market will be held. See details elsewhere in the newsletter.

The December meeting will also be held on the 3rd Saturday due to holidays. Beginning with this meeting, on December 19, we will hold our monthly meeting in Building A of the Uniformed Services University of the Health Sciences (USUHS). The address is 4301 Jones Bridge Road, Bethesda, MD and it is located on the campus of the National Naval Medical Center. There is a map showing the new meeting location elsewhere in this issue.

The Executive Board meets on the 2nd Wednesday evening of each month. All members are welcome to attend. Details will be on the club phone and ABBS.

NOVAPPLE meets on the 2nd Saturday of the month at 1:00 PM at Kings Park Library on Burke Lake Road in Fairfax County; and on the 4th Thursday of the month at 7:30 PM at Computerland of Tysons Corner. In addition, tutorials will be presented on the 2nd Wednesday at 7:30 PM at Computers Plus on Franconia Road.

PRESIDENT'S CORNER

by David Morganstein

The Mid-Atlantic Conference is behind us and the Inside Washington Apple Pi is printed. This simple sentence does not convey the mammoth amount of effort put forth by all those involved. First, the show at the Armory: many thanks to Bernie Benson and Bernie Urban for their exhaustive (and exhausting) efforts. They organized well and had an interesting well-attended booth. We found almost a hundred Apple owners who decided to join us and expect many more from those who took home our hand-out. Mark Crosby also should be thanked for his many hours of attendance and for the marvelous display program used to advertise us.

The Inside Washington Apple Pi is a very professional looking publication and Steve Hadley deserves our appreciation for his efforts. Steve spent night and day during the closing hours to insure that it would be ready by the show. Mark Crosby helped Steve as the final moments approached. The collection owes its contents to all the many authors over the past year, both those appearing in the "Inside" and all the other writers as well. (Footnote: where can I get the Nirvana card advertised on p.24?)

A subject I would like to spend a minute on is the following oft-seen quote paragraph:

"XYZ Systems makes no warranties, either expressed or implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability or fitness for any purpose. XYZ Systems software is licensed "as is". The entire risk as to its quality and performance is with the buyer. Should the software prove defective following its purchase, the buyer (not XYZ Systems) assumes the entire cost of all necessary servicing, repair or correction and any incidental or consequential damages resulting from any defect in the software, even if XYZ Systems has been advised of the possibility of such damages."

While I'm not a lawyer, I think this says that if the program destroys my computer, I have no one to blame but myself. This interesting phrase certainly bestows a feeling that the merchant has enormous faith in his product...I guess that this kind of thing is necessary to protect the merchant from numerous legal actions, presumably unjustified, for any little objection the purchaser may express. However, it leaves the buyer with little comfort. Given the large proportion of software available almost exclusively by mail order, where does this leave you and me? Clearly, to make sure something works, you need to see and test it thoroughly or else you may be left without a legal leg to stand on.

My second point regarding this Disclaimer deals with copy protection. It seems that if you buy a program covered by the above umbrella and find that it does not quite do what you want, you have no recourse from the merchant. If the software is not copy protected, you may at least be able to make desired changes your self. After all, the program is now yours to use for your own purposes. If the disk is copy protected, however, this last option may not be available either (at least not to most of us). It seems a shame we can not resolve this issue of rights (the merchant's and the consumer's) in a more workable fashion.

GROUP PURCHASE POWER by Rich Wasserstrom

This will inaugurate an aperiodical column on the club store and group purchase program.

Let me begin by thanking Howard Lefkowitz who ran these activities so well for the last year. Howard's good cheer and ability to choose the best product from among the many will be missed. But mourn not, Howard has opened his own computer store in Beltsville (free plug!). I'll not have any talk about the Peter Principle.

Now, what was I talking about? Oh yes; the club store. I intend to run three purchase programs: (1) club store, (2) consignment sales, and (3) group purchases.

1. Group purchases are those in which several members (as many as possible) pool their money to place a large quantity order for computer products at prices which reflect quantity discounts. I will reserve this for expensive items, such as printers, which the club store cannot stock. Speaking of printers, our recent group purchase of Epsons was well supported by the membership. I will continue the Epson program for as long as you wish. Group purchases offer the lowest prices in the club program since we ask that you advance your money and wait for delivery, generally until the next meeting. The degree of support for group purchase products varies with the club's vendor. Products purchased locally will be supported by those local vendors just as if you bought them from the vendor's retail store. Products purchased from local or out-of-town non-retail vendors, will carry the manufacturer's warranty.

2. Consignment sales are those in which the club invites local retailers to display computer products at the club store during monthly meetings. The

contd.

merchandise remains the property of the dealer and the club merely acts as a vehicle to introduce the product to club members. Members can expect about a 10% discount on consignment sales and the club receives a commission on such sales from the vendor.

I intend to use consignment sales for software and little goodies which are difficult for the club store to stock because I cannot find them at an advantageous price or because demand for such items is not steady (i.e. games).

I invite all local (or not so local) retailers to participate in the consignment program. Washington Apple Pi carries insurance which will cover your merchandise. Please contact me via the club phone or mail box to arrange consignments. I cannot allow "consignors" to appear at the monthly meeting without prior arrangements. Since I invision the consignment program as one offering full after sales support, I will limit the program to legitimate retailers with established places of business.

3. I discuss the club store last since in many ways it is the most problematical. When Bob Peck started the store many moons ago, blank diskettes were the only product. Now, the store STOCKS a large variety of boards (Z-80, Ramcard, CPS Multi-funtion, Smart-term, Micromodem II, Language System, etc.); software (Visi-calc/plot/term/dex, Letter Perfect, Super-scribe, Data-Capture, etc.); books (Beneath Apple DOS, Reference Manual); accessories (Sup'r'fan, Scotch head cleaners, and disk rings); and supplies (Dysan and Memorex diskettes). Each of these products is offered at a substantial discount and was stocked because it performs its intended function. If you want the club store to continue, you must support the store!

To be sure, you can find each of these items for less money, but we have them IN STOCK and this is your store. We use your money to stock the store to provide the membership a service and to help the club treasury. The store cannot compete with the super discounters since we do not buy in large enough quantities and since we must add a margin to build our store stock funds. Without this "up front" money the club store cannot maintain a stock of items for you.

As the man said, "you pays your money and you takes your choice." The club cannot have a stocking store if the members do not support it. For my part, I will try to find vendors offering the lowest possible prices and will look hard at the store prices to see that our members are getting the best possible deal.

SALE, SALE, SALE, SALE, SALE, SALE, SALE
Be sure to check the club store at the next meeting as I have lowered the prices on many items. Some items will be offered at cost!

MORE GROUP PURCHASES !!!

Sign up at the next meeting for group purchases on Epson printers, Apple disk drives, video monitors, and surge/hash suppressors. I am still considering which monitors to purchase, but I find the Amdek Color I and Video 300G (1000 lines at the center for the 300G !) are strong contenders as are several of the Sanyo and Hitachi monitors. The Sanyos are rock steady and Hitachi offers some of the sharpest monitors I have seen. (Incidentally, Hitachi manufactures the Amdek Color I.) Let me hear from you hardware/engineering types as I need all the help I can get on the merits of these and other group purchase items.

MANUFACTURERS, DISTRIBUTORS, AND RETAILERS

Interested in doing buisiness with Washington Apple Pi? We now have almost 1000 members and are prepared to buy Apple and Apple compatible hardware and software in quantity if you offer attractive prices. We offer single point shipping, quantity purchases, a reliable and steady business relationship and CASH. Contact Rich Wasserstrom via the club phone (301-621-2719) or by letter (Washington Apple Pi, P.O. Box 34511, Bethesda, Md. 20817).

SOFT VIEWS: NEW RELEASES

by David Morganstein

Apple Writer Extended (BRILLIG SYSTEMS, Inc.). Do you use Apple Writer? Have you wished you could use it to edit programs or use it to create EXEC files? (Of course, you could have typed in the Applesoft programs written by Paul Sand and published in our newsletter.) This package provides several machine language routines for converting from Apple Writer to Applesoft or to EXEC files. However, the nicest feature is the PRINT EXIT package. This allows several desirable addtions such as: merging a file of names and addresses (or any data) with a letter (or any form) and printing multiple copies of the document with each merged item (e.g. form letters). In addition, it allows you to embed control characters in an Apple Writer file so that you may activate printer commands.

Locksmith 4.0 (Omega Software). The long awaited update is now available. Many valuable new routines have been added to yield a very useful package. Some of the new features are: a hi-res disk/speed test; a disk surface certifier; a disk eraser and a nibble-editor. The basic function of creating archival back-ups has been enhanced via a quickscan feature to examine tracks and half-tracks. The oft discussed parameters are briefly explained and a method is provided for modifying them and storing them. An interface with their excellent utility The Inspector, is a part of the software. (An update on the initial "bug" in the Inspector ROM is that a correctly rewired version has been mailed out to owners of the original production ROMs.)

The Shattered Alliance (Strategic Simulations, Inc.) This is the latest in their war gaming series and it is a hum-dinger. The most notable feature, referred to as Rapidfire, is the speed of play. Most SSI buffs are familiar with the correlation of playing time with complexity. Well the SSI folks have apparently solved this one. Instead of assigning various movement allowances to each piece and requiring that each piece be moved each turn, they have changed their mode of thinking. Pieces which have greater distance capabilities, like cavalry, move more often. The result is a fast-paced game which I have to constantly halt just to appreciate. This first in a series contains not only battle between hi-res multi-color shapes, representing various types of units, but includes the use of magic spells as well!!! The screen is improved over the marvelous multi-color game of Apocalypse. The actual playing field can be viewed from an overall strategic vantage, where the pieces appear as solid colored shapes, or it can be viewed from a close-up scrollable view of a segment of the battlefield, where the pieces appear as horses, men, elves, dwarves, demons, centaurs and more...I really have only one complaint, and it is minor. When starting the game you are given the option of Beginner or Advanced. Selecting Beginner, you are given a choice of various advantages to be assigned the "Solitaire Player". I assumed this was me when I was playing against the Computer. Guess what...the Computer got the advantages, which I assure you it didn't need.

Epoch (Sirius Software). These folks have some of the best arcade games around and I'm sure Epoch will move into a top spot on the sales charts. It is one of the most realistic 3-D space simlulations I have seen on the APPLE. You are flying through space at a speed controllable by you. The elements of the game include: Time, which runs out (...unless); Fuel, which gets consumed (...unless); and Ammo, which you use up (...unless). The objects found in space include: bad guys (in several forms, some of which move and some of which don't); your bases which allow refueling and rearming; and time warps which regain more time to keep playing. To obtain fuel/ammo you must locate one of your bases, manuevering with a joystick (preferred over paddles) and fly through an opening in the bottom of the base. To regain time, again you must locate a time warp and blast through it down a corridor in space, a twisting worm hole. I won't go into the sound effects but will leave a few surprises for you to discover. Suffice it to say that between refueling and "retiming", you can shoot up a myriad of the bad guys, avoiding their return fire (a hit will cost you fuel). Your shots disappear off into the distance in perspective. The Enemy destroyed explode and fragments expand as you approach. It's a very well done arcade game.

International Gran Prix (Riverbank Software, Inc.) What can you expect from the author of Three Mile Island? Realism, tension, an excellent simulation? Gran Prix is all that and more. Remember Bill Budge's Night Driver, we all thought it was amazing what you could do with an APPLE. Rich (who I believe lives in Severna Park), has taken the fine art of hi-res and added a realism you will enjoy. If you are a race car buff you can get more miles per gallon by training on your APPLE with this program. Choose from one of five grand prix style road circuits. The screen displays the track as you view it and your instrumentation along the bottom (including a few gauges you won't find in any race car!!!). The Christmas Tree changes colors from red, to yellow, then green. You're off, whether you select an automatic transmission or a five-speed, you see the revs increase and you can shift up or down as needed. If you get too near the track edge, a warning indicator announces your peril. Crashes are colorful and noisy. You'll soon

contd. on pg. 9

We are pleased to announce the opening

October 15, 1981

of our

NEW COMPUTER SHOWROOM AND TRAINING FACILITY

HLA INC.
5700-J SUNNYSIDE AVE.
BELTSVILLE, MD. 20705
301 345-1123

OPENING SPECIALS
(For September, October and November)

*FREE

WITH THE PURCHASE OF A SYSTEM

- Training Course
  "Beginners Guide to Using Computers"
- 12" Hitachi High Resolution Monitor
  (Requires a cabinet and 12 volt power supply)
  or
  Your choice of software - retail value $75

  *Except Atari 400

10% DISCOUNT

- All Items Except Special Sale Items
- All Courses for One Year (with any system purchased)

EXTENSIVE TRAINING COURSES WILL BE PROVIDED USING TEN STUDENT-OPERATED COMPUTER SYSTEMS. THESE WILL INCLUDE:

- Business and financial use of microcomputers
- Stock market analysis and tracking
- Pascal programming (elementary, intermediate, advanced)
- Basic programming (intermediate, advanced)
- Data base applications and operation on microcomputers
- Computers for the Professional Office
- Use of computers in the Classroom

ADVANCED REGISTRATION IS REQUIRED FOR ALL COURSES

WE INVITE YOU TO VISIT OUR NEW FACILITY just ½ mile north of U.S. 1 and the Beltway at the Beltsville Exit; Right on Sunnyside Ave. for three blocks.

An Authorized Apple Dealer

SIG-NEWS

SIGAMES is the special interest group of computer hobbyists interested in using their APPLEs for entertainment.

This month's newsletter features two new regular SIGAMES columns: HIT PARADE and SIGAMES NEWS, both by John Alden. HIT PARADE is SIGAMES' new buyer's guide to games. Each month a new group of games will be featured.

SIGAMES NEWS will present the agenda for the current month's SIGAMES meeting, the next month's agenda, a synopsis of the prior month's meeting and a review of one or two new games.

PIG, the Pascal Interest Group, meets on the third Thursday of each month at 7:30PM at the Uniformed Services University of the Health Sciences, Bldg. A, Room A2054 (2nd floor), on the campus of the National Naval Medical Center at 4301 Jones Bridge Road, Bethesda, MD.

EDSIG will meet immediately after the regular meeting of Washington Apple Pi.

NEWSIG will meet just after the regular Washington Apple Pi meeting. The meeting seems to best help the new members by answering their questions, and telling them what to do to get their system up and running. We also tell them something about WAP, how to order the disks, what's on the disks, etc.

The following members have agreed to answer questions over the phone when someone gets stuck and needs help between meetings:

Bob Chesley	560-0121
Paul Hoffman	831-7433
Sara Lavilla	926-6355
Boris Levine	229-5730
John H. Smith	439-4388
Steve Sondag	281-5392

MINUTES

EXECUTIVE BOARD MEETING

The Executive Board of Washington Apple Pi met at 7:30 PM on September 9, 1981 at the home of Bernie Urban. President David Morganstein presided; 16 people were in attendance.

Bob Peck and Rich Wasserstrom will operate the club store. The club has applied for a tax exempt number. A motion passed recommending that the club meetings be held at USUHS as soon as possible. A letter confirming the availability of the USUHS facility will be obtained from Col. Dick Hodder. Our booth at the Mid-Atlantic Computer Show under the direction of Bernie Benson was discussed. Motion passed to print 2500 insides and 5000 covers for "Inside Washington Apple Pi". A flea market for the November general meeting was discussed, and Nick Santelli was asked to write detailed instructions for it. Voting procedures at Board meetings were discussed.

The official meeting was adjourned at 10:15 PM and those who could stay longer helped with proofreading for "Inside Apple Pi".

GENERAL MONTHLY MEETING

Washington Apple Pi met at 9:30 AM on September 26, 1981. President David Morganstein presided with 193 people attending.

Announcements were made: Bob Peck and Rich Wasserstrom will run the club store and group purchases; Charles Dow volunteered for membership and mail list; a CP/M SIG is forming; insurance is available for storing club hardware and software in our homes; Inside Apple Pi is published; a flea market is planned for the November meeting on the 3rd Saturday; and a club tutorial is planned at USUHS on October 10 and 17 and on November 7 and 14, for a fee of $20 with an APPLE and $30 without. A number of commerical announcements were made; the appropriateness of these was discussed; the issue will be taken up by the Board. There was a discussion of meeting at USUHS; the majority indicated that they could get to that location but some said they would have difficulty; the issue will be taken up by the Board. Roger Kaufman gave a presentation on the use of APPLE in kinematics.

The meeting adjourned at 11:30 AM.

Jesse Wagstaff, Secretary

NOTICES

COME TO THE FLEA MARKET

****

At our November 19 (3rd Saturday) meeting, there will be a flea market. BUY.. SELL..SWAP. Looking for a bargain, or have a piece of hardware or software that's gathering dust? This is the place to go. There will be limited table space for sellers on a first come, first served basis. No admission. No fees. We ask that sellers observe two rules: (1) All copyrighted software for sale MUST be on original disks or tapes and be accompanied by original documentation; (2) Absolutely no commercial sales. See you there.

NEW MEETING LOCATION

****

Beginning with our December meeting, we are changing our meeting location. The new location is in Building A of the Uniformed Services University of the Health Services. It is located at 4301 Jones Bridge Road, Bethesda, MD and is on the campus of the National Naval Medical Center. For your convenience we are publishing a map elsewhere in this issue to show the new location. There is ample parking.

contd.

EARLY DEADLINES FOR NEXT TWO ISSUES

Writers of articles for the newsletter are reminded that the next two issues have unusually early deadlines due to the Holidays. Deadline for the next issue (December) is November 6, and for the following issue December 4. We would appreciate your getting materials (both articles and ads) in as soon as possible. It is difficult to put out a well-organized and presentable newsletter without ample time for preparation and layout. But, whatever happens, keep the articles coming!

WAP HOTLINE

Have a problem? The following club members have agreed to help. PLEASE, respect all telephone restrictions, where listed, and no calls after 10:00 PM.

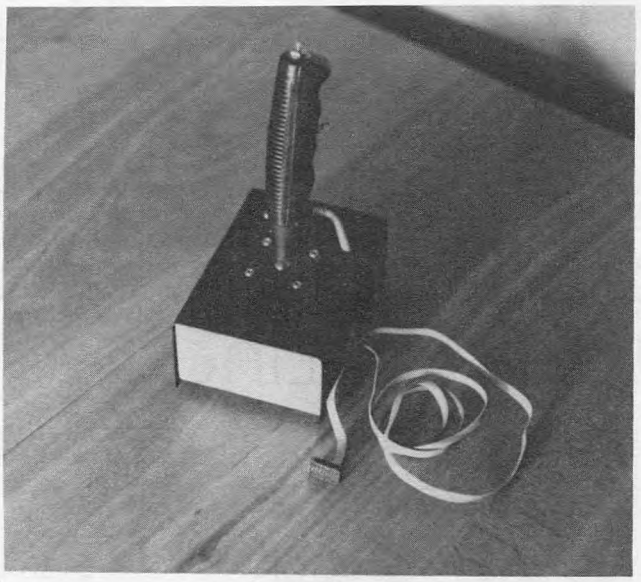
General		
	Ben Acton	972-1533
	Robert Fretwell	971-2621
	Dave Harvey	527-2704
	Tom Jones	460-8773
	Robert Martin	498-6074
Operating Systems		
APPLE DOS	Richard Untied (weekends only)	241-8678
CP/M	Robert Fretwell	971-2621
Languages (A=Applesoft, I=Integer, P=Pascal, M=Machine)		
A,I	Jeff Dillon	422-6458
A,I	Tom Jones	460-8773
A	Mark Pankin	370-9219
A,I,P,M	Bill Schultheis (except Tue.,Thurs.)	538-4575
A,I,M	Richard Untied	241-8678
P	Robert Fretwell	971-2621
Printers	Walt Francis	966-5742
Word Proc.	Walt Francis	966-5742
	Ben Acton	972-1533
VisiCalc	Ben Acton	972-1533
	Walt Francis	966-5742
Time-Sharing	Chuck Reinbrecht	299-6810
	Dave Harvey	527-2704
Graphics	Bill Schultheis (except Tue.,Thurs.)	538-4575
Games	Jim Eatherly	232-6046

SOFTVIEWS contd. from pg. 6

learn to control the car, read your lap counter and fuel gauge. The whole family will get many miles of pleasure from International Gran Prix.

Oo-topos (Sentient Software). This excellent text style adventure game was written by Michael Berlyn, a published science fiction writer. Most adventure programs either add hi-res graphics, which is usually loaded in from the disk, or use greatly expanded dictionaries (also frequently loaded in from the disk). Oo-topos will challenge the latter adventure player with its well written dialogue. Its like acting out a sci-fi story rather than just reading it. As needed, you can save the adventure in mid-game and return to it later (you'll need this feature since you won't complete your adventure in one sitting!!!). Michael's next adventure, yet to be released, is called "Cyborg" and features a symbiotic relationship between you and your APPLE (more on that next month...)

TIRED OF A DISORGANIZED WORK SPACE FOR YOUR APPLE



FEATURES:

- o SPACE FOR 2 DISC DRIVES/MONITOR/RECORDER, ETC.
- o BUILT-IN DISC STORAGE
- o MODEM SHELF
- o BUILT-IN EXTENSION FOR GAME I/O SOCKET - - MOVES YOUR GAME I/O SOCKET TO FRONT PANEL ACCESS AND EXPANDS IT TO TWO SOCKETS
- o TWO A.C. OUTLETS WITH CONTROL SWITCH AND INDICATOR LIGHT ON FRONT PANEL
- o STURDY CABINET COVERED WITH WALNUT TONE PLASTIC LAMINATE
  OTHER COLORS AVAILABLE AT SLIGHT ADDITIONAL COST
- o 3 CONDUCTOR A.C. CORD
- o TIE DOWN STRIPS TO LESSEN CABLE STRAIN
- o FAN AVAILABLE - - RECOMMENDED FOR WARM ENVIRONMENT ($39.95 ADDITIONAL)

CUSTOM UNIT - INTRODUCTORY PRICE - $84.95

BASIC UNIT ALSO AVAILABLE (EXCLUDES ELECTRONICS I.E. EXTENSION SOCKET, A.C. OUTLET & CORD, SWITCH, LIGHT, TIE DOWNS, ETC.)

BASIC UNIT - INTRODUCTORY PRICE - $49.95

ALSO AVAILABLE - - JOYSTICKS

PISTOL GRIP X-Y JOYSTICKS WITH PUSH BUTTON CONTROLS HOUSED IN THE GRIP. ENHANCE THE REALISM OF YOUR COMPUTER GAMES WITH THIS LIFE-LIKE CONTROLLER.

INTRODUCTORY PRICE - $44.95

SEND ORDERS TO: SILVER-FISCH ASSOCIATES
POST OFFICE BOX 4161
SILVER SPRING, MARYLAND 20904

ADD FOR POSTAGE AND HANDLING - - $5.00 CABINET, $2.50 JOYSTICK; MARYLAND RESIDENTS ADD 5% SALES TAX - - ALLOW 4 TO 6 WEEKS DELIVERY.
APPLE IS A REGISTERED TRADEMARK OF APPLE COMPUTER CORPORATION.

COLUMBIA COMPUTER SYSTEMS

The discount house that takes a BIG bite out of game prices.

PROGRAM	SYSTEM	MFG	LIST $	SALE $
Galactic Saga:				
I Galactic Empire	D48+	BS	24.95	**19.95**
II Galactic Trader	D48+	BS	24.95	**19.95**
III Galactic Revolution	D48+	BS	24.95	**19.95**
IV Tawala's Last Redoubt	D48+	BS	29.95	**22.95**
Galactic I, II & III			74.85	**55.00**
Galactic I, II, III & IV			104.80	**75.95**
Adventure	D48+	MS	29.95	**25.95**
Stellar Trek	D48+	RC	24.95	**19.95**
Datestones of Ryn	D48+	AS	19.95	**15.95**
Morloc's Tower	D48+	AS	19.95	**15.95**
Temple of Apshai	D48+	AS	39.95	**32.95**
Hellfire Warrior	D48+	AS	39.95	**32.95**
Zork	D48+	PS	39.95	**32.95**
Hi-Res Adventures:				
#0-Mission Asteroid	D48+	OL	19.95	**15.95**
#1-Mystery House	D48+	OL	24.95	**18.75**
#2-Wizard & Princess	D48+	OL	32.95	**27.95**
#3-Cranston Manor	D48+	OL	34.95	**28.95**
Special Sale				
New #4-Ulsses & Golden Fleece	D48+	OL	34.95	**27.95**
Missile Defense	D48M	OL	29.95	**22.95**
Hires Soccer	D48M	OL	29.95	**22.95**
Hires Football	D48M	OL	39.95	**33.95**
Hires Cribbage	D48M	OL	24.95	**18.75**
Sabotage	D48M	OL	24.95	**18.75**
Paddle Graphics	D48M	OL	39.95	**33.95**
Tablet Graphics	D48M	OL	49.95	**39.95**
Hyper Head-on	D32+	BS	24.95	**18.75**
Galaxy Wars	D32+	BS	24.95	**18.75**
Tank Command	D32+	BS	29.95	**22.95**
Alien Rain	D48M	BS	24.95	**20.95**
Alien Typhoon	D48M	BS	24.95	**20.95**
Golden Mountain	C32+	BS	19.95	**15.95**
Devil's Dungeon	D16I	RC	15.95	**12.95**
Special Sale				
New Goblins	D48+	HC	27.50	**21.95**
Pool 1.5	D48M	ID	34.95	**28.95**
Shuffleboard	D48M	ID	29.95	**22.95**
Ultima	D48+	CP	39.95	**32.95**
Special Sale				
Race to Midnight	D48+	AG	29.95	**21.95**
Special Sale				
New Super Stellar Trek	D48M	RC	39.95	**32.95**
ALF Music Card				
MC1 (9 voice) - hobby			195.80	**169.95**
MC16 (3 voice) - professional			245.00	**199.95**
All ALF albums			14.95	**11.95**

PROGRAM	SYSTEM	MFG	LIST $	SALE $
Snoggle	D48M	BS	24.95	**20.95**
The Prisoner	D48M	EW	29.95	**24.95**
Operation Apocalypse	D48M	ST	59.95	**48.95**
Cartels & Cutthroats	D48M	ST	39.95	**33.95**
Torpedo Fire	D48M	ST	59.95	**48.95**
Warp Factor	D48M	ST	39.95	**33.95**
Computer Quarterback	D48M	ST	39.95	**33.95**
Computer Conflict	D48M	ST	39.95	**33.95**
Computer Air Combat	D48M	ST	59.95	**49.95**
Computer Napoleonics	D48M	ST	59.95	**49.95**
Computer Ambush	D48M	ST	59.95	**49.95**
Computer Bismarck	D48M	ST	59.95	**49.95**
Special Sale				
New Battle of Shiloh	D48M	ST	39.95	**32.95**
New Battle of the Bulge	D48M	ST	39.95	**32.95**
Apple Oids	D48M	CP	29.95	**22.95**
Akalabeth	D48M	CP	34.95	**28.95**
Raster Blaster	D48M	BC	29.95	**22.95**
3-D Graphics System	D48M	BC	39.95	**33.95**
Space Album	D48M	BC	39.95	**33.95**
Trilogy of Games	D48M	BC	29.95	**22.95**
Fender Bender	D48M	BC	24.95	**19.95**
Cosmos Mission	D48M	BC	24.95	**19.95**
Scott Adams'				
Adventure 1, 2 & 3	D48M	AI	39.95	**31.95**
Adventure 4, 5 & 6	D48M	AI	39.95	**31.95**
Adventure 7, 8 & 9	D48M	AI	39.95	**31.95**
Adventure 10, 11 & 12	D48M	AI	39.95	**31.95**
Special Sale				
Robot Wars	D48+	MU	39.95	**33.95**
Gorgon	D48M	SS	39.95	**33.95**
Space Eggs	D48M	SS	29.95	**22.95**
Pulsar II	D48M	SS	29.95	**22.95**
Autobahn	D48M	SS	29.95	**22.95**
Orbitron	D48M	SS	29.95	**22.95**
Gamma Goblins	D48M	SS	29.95	**22.95**
Star Cruiser	D48M	SS	24.95	**18.75**
Cyber Strike	D48M	SS	39.95	**33.95**
Sneakers	D48M	SS	29.95	**22.95**
Both Barrels	D48M	SS	24.95	**18.75**
Phantom's 5	D48M	SS	29.95	**22.95**
EZ Draw 3.3	D48M	SS	49.95	**39.95**
Special Sale				
New Wizardry	D48M	SI	49.95	**39.95**
Game Controllers				
Hand Controller		KB	29.95	**22.95**
Joystick II		KB	49.95	**39.95**

- VISA, MC add 3% (include card no., expiration date, and MC, InterBank no.)
- Allow 3 weeks for personal check to clear.
- COD's require 10% deposit—all COD charges to be paid by customer.
- Prices subject to change without notice—all items subject to availability.
- Money order, certified check, cashier's check or bank wire deposit accepted
- Include full telephone number on mail orders.
- Include $2.50 for postage and handling per software shipment (UPS).
- DC residents add 6% tax. If tax exempt, include number.

VISA

COLUMBIA COMPUTER SYSTEMS

P.O. Box 40813
Washington, DC 20016
Call (202) 364-0273 any time, every day

MasterCard

SIGAMES NEWS by John Alden

Welcome to the SIGAME NEWS. This column (as opposed to certain rumors) is still devoted to three tasks: To present information about upcoming SIGAME events; to present a synopsis of the past month's meeting; and to review new and favorite games.

The next few months still promise to be especially especially exciting (I'll try to change the opening line next month). Each meeting begins with the survey of games for the Hit Parade. (For the schedule of games to be surveyed, see the Hit Parade column in this issue) New games are presented and demonstrated on an Apple in full color. The featured speaker will present the program for the meeting.

Steve Stern is leading this month's program. Steve will be opening the meeting to the Apple Seeds (Stayman, of course!). They will compete against each other with several games. Come one, come all and watch the nasty aliens be blasted into eternity.

Theron Fuller presented last month's program about artifical intelligence. Theron's presentation was one of the best this year. Our thanks to Theron for his excellent program.

Do you have a game you would like demonstrated or explained. Let's hear from you. This is your meeting and we want to help people make educated decisions when buying games.

Recently released software include: Wizardry (Sir-Tech); Copts & Robbers (Sirius); Race for Midnight (Avante-Garde); Goblins (Highlands); High-Res Adventure #4 - Ulysses and the Golden Fleece (On-Line); The Battle of Shilo (SSI); The Battle of the Bulge (SSI); Space Adventure (Sierra); Pool 1.5 (IDSI); Shuffleboard (IDSI); Voyage of the Valkyrie (Advanced); and an expansion module for both Temple of Apshai and Hellfire Warrior (EPYX).

'Copts & Robbers' is a high-res adventure game based in Egypt. You attempt to go through the mazes to find 4 jewels which must be returned to the vault room so you can leave the tomb. So far I have not found the 'Copts' but I have found the 'Robbers': They are the people who designed this game. It has to be worse than 'Beneath the Pyramids'. If you are easily bored and like imaginative games, I recommend you avoid purchasingthis one. From Sirius Software for $29.95.

'Wizardry' is better than 'Ultima'. It combines 3d mazes, multi-level and multi- player characters. It is, also, much longer than 'Ultima'. You start by selecting your character's name and password. The password permits only the owner of the character to use it. When you originally enter the password, all you see on the screen are 'XXXXX', thus you enter the password twice for your safety. You will then create the characteristics for your character. You select whether a human, an elf, a dwarf, a gnome, or a hobbit. Your alignment must be either good, neutral, or evil. The computer rolls the die for your strength, I.Q., piety, vitality, agility, and luck plus bonus points which can be spread across each of the characteristics at your choice. You're finally ready, right. Wrong! You have to assemble a party of 1 to 6 adventurers at Gilgamesh's Tavern. Sir-Tech recommends a begining party should comprise 2 fighters, 2 mages, 1 priest, and 1 thief. After reading the spell books, you are finally ready to begin. Some recommended strategies are to run from stronger monster parties until your party is level 3 or above. You should return to the castle after each battle to regain your strength. When your characters are level 3 and above start longer excursions. Map the dungeons. A quick note about levels. Level refers to 3 items in 'Wizardry': Level of experience for each character; level (or floor) of the dungeon; and level of power for your characters. Thus, you may have a level 10 priest who has attained the 5th level of power fighting on the 3rd level of the dungeon. Each level of experience is almost double the last level. If a character attained level 4 at 12000 experience points, it will require around 24000 experience points to attain level 5. I have a level 11 fighter who attained level 11 at 134,568 points and will require 232,044 points for level 12. I estimate that level 13 will be in the vacinity of 450,000 to 500,000 points. This is a long game. Fortunately there are only 10 levels to the dungeon. But after 500,000 points and 13 levels and 10 levels of the dungeon is it all over? Yes, but only for this scenario. The program is designed to use scenarios like Eamon. 'Wizardry' will be demon strated at this meeting and will be featured at the November meeting. A spectacular game. A highly recommended purchase. From Sir-Tech for $49.95.

'Race for Midnight' is a combination of Lon Chaney and Cranston Manor. There are 75 rooms done in high-res color. "You live in a small town in the 14th centory You were awakened this morning by a terrible pain in your arm. Upon examining your arm, your find a bloody gash in it. Wisely you cover it so that nobody will see it. Later, you find that the townspeople had seen a werewolf last night and one person had shot an arrow at it, but evidently he had missed, because the werewolf continued running. You instantly deduce that you must be the werewolf and realize that you must find an antidote.

contd.

You decide to go to a nearby dungeon that is deserted. Legend says that a powerful wizard, Evro, once lived there, but he became a victim of his own experiments. The rumor is that he had strange and deadly creatures under his power. You decide that you might be able to find some sort of recipe for a potion to cure your affliction." This adventure is unique because you can run it as a text adventure or a high-res color adventure. By pushing 'return' you can switch between each mode. In the text mode, it is formated like a Scott Adams adventure with your possible exits and items seen listed above a line and your commands below the line. Have fun (Howwwww!!!!). From Avante-Garde Creations for $29.95.

'The Battle of Shilo' is an excellent addition to Strategic Simulations' line of computer wargames. 'Battle of Shilo' is somewhat unique even among this advanced group of wargames. While it has the high-res color layout we expect, it allows you to select risk levels and ferocity of attack (or defense). Thus, a player can test the defenses of the opponent without committing your entire brigade. A highly recommended purchase from Strategic Simulations for $39.95.

'Voyage of the Valkyrie!' is a high-res space game. Your goal as the pilot of the attack ship Valkyrie is to conquer the island of FUGLOY. You are a Private in the Space Vikings and your future in the service will be determined by your performance in this campaign. If you succeed in defeating the island's air force and in capturing all ten of the island's castles, you will become the Prince Regent and rule the island with the support of the Viking Space Navy. You must, also, try to get as much of the island's gold as possible as you proceed. Your 'high score' is the amount of gold you have at the end. When you capture a castle, you are credited with all of its gold. The amount of gold is based upon the type of castle and number and quality of defenders. When you force a castle to surrender, you do not get any gold. Instead, you turn the castle into a refueling base. I have managed to crash into the mountains and am sending this report by carrier eagle. A recommended purchase for space and adventure fans. From Advance Operating Systems for $39.95.

USING A FIRMWARE CARD IN SLOT 4

by David Morganstein

Many Apple owners have upgraded their computer by adding 16K memory cards which reside in Slot 0. These cards have many uses (Pascal, CP/M, VisiCalc, etc.). In the process, firmware cards containing Integer Basic or Applesoft have been removed and shelved. After all, since you can load these languages into the memory card, who needs the firmware card. Well, like Mt. Everest, some of us may want to use the card "because it's there". Actually, several reasons justify this desire. First, it takes a while to load (and sometimes reload) the 50 sector FPBASIC or INTBASIC program into the language card. Second, some protected disk systems wipe out the language card contents and, in the process, may make the programs on the disk unusable to the purchaser ("Tawala's Last Redoubt" is such a protected disk.) However, having the needed language on a firmware card may make it possible to use such programs. Third, perhaps most importantly, you can now obtain programs which will move DOS onto a 16K board in slot zero. (Our new Vol 101 has such a program). Doing so wipes out the Basic normally located on the card. If you have a firmware card in some other slot, though, you can have the additional 10.2K of RAM for use in either Basic!!!

Given that you have both a memory card and a firmware card, how to proceed. My thanks to Bob Sander-Cederlof and his Apple Assembly Line magazine (a recommended newsletter if you want to learn about machine language...PO Box 280300, Dallas Texas, 75228) for the starting clue. The second tip is found in Beneath Apple DOS (another excellent book on the Apple's disk operating system). According to the July AAL, putting a C0 in A5B8 and a C1 in A5C0 (i.e. from the monitor type *A5B8:C0 then *A5C0:C1) will do the trick. "What is the trick?" you may ask.

To answer this, disassemble the DOS code starting at A5B2. This routine is described in Beneath Apple DOS as the subroutine CALLed to switch between versions of Basic. The routine, shown below is entered with either a 4C or a 20 in the accumulator, the former indicating that Applesoft is desired and the latter that INTEGER is the choice. Looking at the code we see that the accumulator is compared with $E000, the warm start location for either Basic. If the right Basic is active a branch to a subroutine return is made, RTS. If not, a store to $C080 is done, STA $C080. What does this store accomplish? It turns on the peripheral in slot zero, causing a change of Basics if either a memory card or firmware card is in the slot (Note, an LDA, load accumulator, would accomplish the same thing. Address $C080 is not a RAM location, just a switch). Aha, you say, what if we change the address to $C080+$40 to refer to slot 4 instead?!! Well, you would have it, because thats exactly what putting a $C0 (=$80+$40) in A5B8 does. Continuing on in the subroutine, if the correct Basic is still not in effect, a STA to $C081 is tried, this will turn off the peripheral card. Similarly, by changing the STA to refer to $C0C1, the subroutine will now function to switch between motherboard Basic and firmware card in slot 4.

```
A5B2- CMP $E000     ;compare A with current
                     Basic
A5B5- BEQ $A5C5     ;if desired Basic act-
                     ive, exit
A5B7- STA $C080     ;if not, turn on slot
                     zero device
A5BA- CMP $E000     ;see if desired Basic
                     now active
A5BD- BEQ $A5C5     ;if so, exit
A5BF- STA $C081     ;if not, turn off slot
                     zero card
A5C2- CMP $E000     ;compare again
A5C5- RTS           ;we've tried all we
                     can
```

Now, given that we need only change the indicated two bytes, what else is desired? Well, what if you don't always have the firmware card in place? What if you want to check to see if there is one and if so modify DOS accordingly? But if not, you want to load the appropriate binary file into your memory card. How to solve that one???

There was a proposed solution in the AAL, September issue. I tried it but found a fatal flaw. It appears to work fine if the firmware card is there, but didn't realize when it wasn't. My answer is given below. Essentially, the program determines what Basic is active, then tries to toggle any slot 4 device. If $E000 changes, then a firmware card is present, and DOS should be modified with the two bytes mentioned above (after which the firmware card must be retoggled to return to the original state.) If no change in $E000 is found, the retoggle is tried to invoke the motherboard language. Again $E000 is checked and the same logic follows.

```
0300- LDX $E000     ;check if basic changes
0303- LDA $C0C0
0306- CPX $E000
0309- BNE $0314
030B- LDA $C0C1
030E- CPX $E000
0311- BNE $0314     ;if not, return
0313- RTS
0314- LDA #$C0      ;if so, modify DOS
0316- STA $A5B8
0319- LDA #$C1
031B- STA $A5C0
031E- CPX $E000     ;determine if firmware
                     was active
0321- BNE $0324     ;if not change state
0323- RTS           ;if so return
0324- LDA $C0C0     ;turn on peripheral
```

contd. on pg. 19

QUESTIONS, QUESTIONS, QUESTIONS

by Mark L. Crosby

Q. I need a fast sort that I can use with Applesoft arrays. Do you know of a routine that will work quickly?

A. If you don't mind using machine language, there is a public domain program entitled "Ampersort" by Alan G. Hill. On our disks #8 and #37 you will find this routine and a demo of how it works. Typical sort times for 1,200 strings in an array average about 15-20 seconds.

Q. I entered a program from a magazine but ran into difficulty when I found a right-arrow included as part of a statement. What does it mean? How can I enter that character from the keyboard?

A. To answer your last question - you don't! This arrow means "raised to the power of...". On the APPLE you would substitute "^" where the right-arrow appeared. Another situation requiring this substitution would be the occurrence of two asterisks "**" or an "up-arrow" used as an operator on a numerical expression.

Q. I am somewhat confused by memory boards that are available for the APPLE. Some commercial programs use these boards to "extend" the amount of data that can be stored. Can Applesoft programs use the extra space to store strings?

A. Unfortunately, it is not as simple as that. The "space" available on one of these boards overlaps the same machine-address area as Applesoft (the language). A memory "switch" is flipped back and forth to select either this "space" or Applesoft. They cannot be utilized simultaneously. Machine language programs that do not use Applesoft do not have this problem (e.g., VisiCalc). Because Applesoft strings are handled dynamically (by Applesoft) it cannot use this memory "space" for that type of storage since this would require that Applesoft "turn itself off". This would leave things "hanging" since there would be nothing available that would turn Applesoft on again. The extra memory space is available for "static" storage - including using the space to store DOS to gain more dynamic memory that Applesoft can use. Usually, the documentation included with the boards explains how to use this area.

Q. While using "Super Copy", a commercial copy program, I have noticed that it reads files very quickly. Often, the copies contain I/O errors or are missing portions of files. Is there some bug in this program?

A. In earlier versions there were some problems related to file storage. In the latest version (3.6) there seems to be no bugs. In fact, this is one of the more reliable programs available. Because it reads files at a much faster speed, it is imperative that you adjust the disk drive speed to as close to 300 RPM as possible. If you use two drives to do the copying, both drives must be adjusted similarly. Small errors of speed (2-3 percent) may cause I/O errors on the copy. Forewarned is forearmed!

Q. I have heard about a graphics screen dump that will handle almost any printer with various sized dumps, etc. Can you tell me which it is?

A. Although there are several screen dumps available from various sources, you are probably referring to "GRAPHTRIX" by Data Transforms, Inc. It will dump either of the Hi-Resolution screens to the following printers: Anadex 9500/9000/9501/9001, all IDS "G" models, Epson MX-70/80/100, Centronics 739, MPI 88G and Silentype. It gives various options including magnification, vertical "cropping" marks to avoid dumping blank lines, centering, etc. It is approximately $64.95 locally.

Q. The On-Off switch on my APPLE went bad and it cost me quite a bundle to have it replaced. How come this is so expensive? Is there any alternative?

A. The Level I service centers used to have to do a wholesale exchange of the entire power supply because it is sealed with rivets. Recently they have been authorized to replace the switch by opening up the power supply. This cuts the cost in half from what it was formerly. This switch is a "weak link" in the APPLE and many people have experienced the frustration of having it replaced. To avoid this problem in the first place, buy a switched power strip (an extension cord with a switch and several sockets), plug your APPLE into it and use that switch instead.

Q. I have heard from some friends that they use both sides of their diskettes by cutting a write-protect notch on the opposite side of the diskette. Others tell me that it is dangerous to do this. Can you clear up the confusion?

A. Using both sides of a diskette on a single-sided drive has its risks and advantages. The advantages are fairly obvious - double the amount of storage using the same amount of disks you have now. The disadvantages are not as obvious. There is a pressure pad which presses down on the disk to keep it in contact with the read/write head. This pad is made of fibers that will slowly

contd.

COLUMBIA COMPUTER SYSTEMS

The discount house that takes a BIG bite out of hardware and software prices.

DESCRIPTION	LIST $	SALE $
Apple Computer		
Apple II or II+, 48K	1,530.00	**1,299.00**
Apple II or II+, 16K	1,345.00	**1,249.00**
Disk II with DOS 3.3 Controller	645.00	**555.00**
Disk II	525.00	**475.00**
Amdek		
Color 1, 13" Color Monitor	449.00	**386.00**
100G, 12" Green Phosphor Monitor	199.00	**169.95**
100, 12" b&w Monitor	169.00	**149.95**
Epson America		
MX-100, 132 Column Printer	995.00	**775.00**
MX-70, 80 Column Printer	449.00	**379.00**
MX-80, 80 Column Printer	645.00	**475.00**
MX-80G, MX-80 with Hi-Res Graphics	695.00	**549.95**
MX-80 F/T, MX-80 with Friction & Tractor Feed	745.00	**579.00**
MX-80G F/T, MX-80 with Hi-Res Graph & Friction & Tractor	795.00	**655.95**
8915, Graftrax-80	95.00	**80.00**
8131, Apple Parallel Interface	85.00	**69.95**
8141, Serial Interface	75.00	**65.00**
8161, IEEE 488 Interface	60.00	**52.50**
8220, TRS-80 Interface	35.00	**31.95**
8230, Apple Parallel Cable	25.00	**22.50**
8260, IEEE 488 Interface Cable	25.00	**22.50**
8150, Serial Interface w/2K Buffer	149.00	**129.95**
8151, Serial Interface w/2K Buffer & Current Loop	160.00	**138.95**
Mountain Computer, Inc.		
MHP-X003, The Clock	280.00	**250.00**
MHP-X004, 100,000 Day Clock w/Charger	375.00	**309.00**
MHP-X024, Expansion Chassis	750.00	**649.95**
MHP-X028, Mark Sense Card Reader	1,195.00	**1,046.00**
MHP-X031, CPS Multi-Function Card	239.00	**179.95**
CPS Cables		
MHP-X036, CPS to Centronics (or Epson)	24.95	**21.95**
MHP-XIDS, CPS to IDS Printers Printer	24.95	**21.95**
The Keyboard Company		
Numeric Keypad	149.95	**129.95**
Cursor III	49.95	**39.95**
Joystick II	49.95	**39.95**
Hand Controller	29.95	**22.95**
Ashton-Tate		
dBase	700.00	**649.95**

DESCRIPTION	LIST $	SALE $
Hayes Microcomputer		
MMII, Micromodem for the Apple	379.00	**307.95**
Datacomm, Pascal Communications Software	50.00	**40.00**
Smartmodem Direct Connect-RS232	279.00	**235.00**
NEC Spinwriters and Monitors		
5510, RO RS232 w/o Tractor	3,055.00	**2,595.00**
5515, RO RS232 w/o Tractor (Diablo Compatible)	3,135.00	**2,695.00**
5520, KSR RS232 w/o Tractor	3,415.00	**3,095.00**
5525, KSR RS232 w/o Tractor (Diablo Compatible)	3,455.00	**3,055.00**
5530, RO Cent I/F w/o Tractor	3,040.00	**2,585.00**
HOR, Horizontal Tractor (Max 4-part Forms)	155.00	**135.00**
VERT, Vertical Tractor (Heavy Duty)	230.00	**189.00**
JB-1201M(A), High Res. Green Monitor	285.00	**239.00**
JC-1202DH(A), High Res Color Monitor w/RGB Input	1,195.00	**1,025.00**
PC-80191A, RGB Cable for JC-1202DH(A)	19.95	**17.95**
LKJ Enterprises		
Letter Perfect Word Proc. Syst. (48M)	149.95	**124.95**
Edit 6502	49.95	**42.95**
Lower Case Generator	34.95	**28.95**
Personal Software		
VisiCalc 3.3, (48+)	199.95	**154.95**
VisiDex (48+)	199.95	**169.95**
VisiPlot (48M)	179.95	**152.95**
VisiTrend/VisiPlot (48M)	259.95	**220.95**
VisiTerm (32M)	149.95	**127.95**
Desktop Plan II (32M)	199.95	**169.95**
VisiFile ——*New*——	250.00	**229.95**
Software Publishing, Inc.		
PFS (Personal Filing System) (48M)	95.00	**76.95**
PFS: Report (48M)	95.00	**76.95**
Both PFS Programs ——*Special*—	190.00	**149.95**
Blank Diskettes		
Dysan, Pkg. of 10	50.00	**42.95**
Maxell, Pkg. of 10	55.00	**46.95**
3M, Pkg. of 10	45.50	**32.95**
Data Transform		
Graftrix	55.00	**42.95**
Addison-Wesley		
Micro-DSS/Finance	1,500.00	**1,399.95**

The NEC Spinwriter Model 3500Q has an average print speed of 30 CPS, handles forms up to 16" wide, and uses all existing NEC thimbles, with up to 128-character versatility.

The NEC Model 3500Q Spinwriter. It's here just when you need it most.

——Special Sale——

NEC Spinwriters		
New Spinwriter 3510 Letter Quality	2,295.00	**1,950.00**

- VISA, MC add 3% (include card no., expiration date, and MC. InterBank no.)
- Allow 3 weeks for personal check to clear.
- COD's require 10% deposit—all COD charges to be paid by customer.
- Prices subject to change without notice—all items subject to availability.
- Money order, certified check, cashier's check or bank wire deposit accepted.
- Include full telephone number on mail orders.
- Include $2.50 for postage and handling per software shipment (UPS)
- DC residents add 6% tax. If tax exempt, include number.

VISA

COLUMBIA COMPUTER SYSTEMS

P.O. Box 40813
Washington, DC 20016
Call (202) 364-0273 any time, every day

MasterCard

polish the disk thereby removing small particles of iron oxide from the diskette surface. Some of these particles remain trapped on the pad. They are abrasive so they tend to remove more and more of the surface of the disk as time goes by. This "vicious circle" of diskette destruction doesn't matter if you are using only one side of the disk since the unused side is the one that gets damaged. Flip the disk over, though, and you might have some trouble since the abrasion will slowly remove not only oxide but data as well. My rule-of-thumb is to use both sides of a disk if it is used infrequently (e.g. master library disks) and to use only one side if it is used often or is used for file storage (with attendant disk accesses). One other important thing - keep that pad clean by replacing it often.

Q. I recently tried to make a very large Pascal unit with a number of items in the interface and quite a few procedures not in the interface. I regularly got stack overflow errors until I made it in small pieces. But then the library and the linker wouldn't cooperate on putting the pieces together. Very frustrating. Any clues or do you need more information?

A. More information would help but here are some clues: Do you have the compiler's double swap, S++, Option on? If not, try it. Stack overflows in this situation are usually symptomatic of lack of space in the compiler symbol table. S++ makes maximum symbol table space available. If you must still split the unit you will have to pay attention to common declarations which appear in two or more unit INTERFACES. You may need to use nested units. Here I need more information.
(Thanks to Dr. Wo for supplying Pascal answers!).

Q. How about a patch for Applewriter that will allow control codes in text (e.g., to control character fonts on Epson MX-80)?

A. APPLEWRITER EXTENDED will do that and more. Edit text or Applesoft files; Create or edit EXEC files; Convert Text files to Applewriter files and vice versa. Sends documents to disk for mass mailings. Generates hex codes for printer control, etc. Requires a 48K APPLE with Applesoft in ROM, Applewriter, and DOS 3.2 or 3.3. $34.95. Contact Eastern Software Distributors, Inc., Baltimore Chamber of Commerce, 17 Commerce St., Room 6, Baltimore, MD 21202, (800) 638-7563 in MD (301) 539-5022.

Q. I would like to know the locations of the Hi-Res routines in the Applesoft ROMs if they exist.

A. Here is a list of the major routines:

ROUTINE NAME	HEX ADDRESS	DECIMAL ADDRESS
HGR2	F3D8	62424
Inits and clears page 2 Hires		
HGR	F3E2	62434
Inits and clears page 1 Hires		
HCLR	F3F2	62450
Clear screen to black		
BKGND	F3F6	62454
Clear screen to last plotted color or POKE 28,n: CALL 62454.		
HPOSN	F411	62481
Position Hires cursor without plotting. $E6 determines which page the cursor is pointed at.		
HPLOT	F457	62551
Call HPOSN then plot		
HLIN	F530	62768
Draws a line from last plotted point. On entry Horizontal=X,A; Vertical=Y.		
HFIND	F5CB	62923
Converts Hires cursor to X-Y coord. On exit: $E0=horizontal lsb, $E1=horizontal msb, $E2=vertical		
DRAW	F601	62977
Draw shape pointed to by Y,X using current color setting. On entry A=rotation factor		
XDRAW	F65D	63069
Draw shape exclusive-or. On entry A=rotation factor		
SETHCOL	F6EC	63212
Set color to X. X must be less than 8.		
SHLOAD	F775	63349
Loads shape from tape and sets up the pointer at $E8 and $E9.		

Other items of interest:

$E4 = current Hires plot color
$E6 = current Hires page
$20=page 1; $40=page 2
$E8 = Address of shape table low-byte
$E9 = Address of shape table high-byte

Y,X,A = 6502 Registers

Q. I have enclosed three short programs. The first produces an image on the Hi-Res screen. The other two are identical although one is written in Integer and the other in Applesoft BASIC. After running program #1, run #2. Then run #1 and #3. Notice there is a difference between the results of programs #2 and #3 although they should be identical. Can you explain why?

Program #1 <Applesoft>

```
10  TEXT : HOME : HGR : HCOLOR= 3
20  FOR X = 0 TO 150
30  HPLOT X,X TO X,X + 5
40  NEXT
```

contd.

Program #2 <Integer>

```
10   POKE 60,128
20   POKE 61,38
30   POKE 62,165
40   POKE 63,38
50   POKE 66,40
60   POKE 67,56
70   CALL -468
80   END
```

Program #3 <Applesoft>

(identical to Program #2)

A. Yes. Programs #2 and #3 simply call the Monitor ROM memory move subroutine to move one area of memory (on the Hi-Res screen) to another. They are absolutely identical in every respect except they are written in two different languages.

While using Integer BASIC, the 6502 Microprocessor is set to Binary Mode. In Applesoft, the Decimal Mode is usually set. The ADd with Carry (ADC) and SuBtract with Carry (SBC) Operation Codes (OP Codes) execute with differing results depending on the Mode. Since the Monitor memory move subroutine uses the SBC and ADC OP Codes, your two programs will move two different areas of memory depending on the two different settings of the Mode.

The solution is to set Binary Mode in your Applesoft program first. See program #4 for details. Note the first few POKEs are machine language which - when called - will set Binary Mode and then CALL the Monitor move subroutine.

Program #4 <Applesoft>

```
10   POKE 768,216:  POKE 769,160
20   POKE 770,0  :  POKE 771,76
30   POKE 772,44 :  POKE 773,254
40   POKE 60,128
50   POKE 61,38
60   POKE 62,165
70   POKE 63,38
80   POKE 66,40
90   POKE 67,56
100  CALL 768
110  END
```

Machine Language Disassembly:

```
0300-  D8          CLD
0301-  A0 00       LDY  #$00
0303-  4C 2C FE    JMP  $FE2C
```

USING A FIRMWARE contd. from pg. 14

```
0327- CPX $E000
032A- BNE $032D     ;is original Basic
                     back?
032C- RTS           ;yes, return
032D- LDA $C0C1     ;no, turn off card
0330- RTS
```

I modified the HELLO program to enter this routine and then CALL it prior to loading the language card (i.e. CALL 768). After the CALL, a PEEK to 42424 ($A5B8) will tell you if the DOS has been changed. If so, end the HELLO program before loading the language card.

A SIMPLE FULLSCREEN TEXT EDITOR by Walter Lee

This is a user guide to a simple fullscreen text editor. It is of use in the editing of small (less than 500 lines) text files such as EXECs, preparation of modem uploads, small word processing requirements and creation of various data files. It works with text files that consist of lines of characters (each line terminated by a carriage return) with an extra carriage return as the last line.

It is a fullscreen editor. That is, most editing operations consist of cursor movement within the text window and then typing over what you want changed. The text window is the part of the text buffer that is visible on the screen; it is a rectangle 39 characters by 20 lines. The editor can support merging of text files, allows you to execute DOS commands from within the editor, will work with disk drives in any slot or drive number, has the ability to print the buffer to any slot (i.e. a printer or modem) and is written entirely in Applesoft Basic. (There are two cases of machine dependency - CALLs to Monitor routines to clear to end-of-line and clear to end-of-page.)

The major limitation is that it is an in-memory editor. The size of file that can be edited is limited to 500 lines, the size of the text buffer (and to edit 500 lines, the lines would have to be fairly short).

The top line is a status display. On the left, it gives the current filename, slot and drive. On the right, it has status indications of the leftcolumn (Ln), count of lines in the text buffer (Cn), current-line pointer (Pn), and the default increment amount (In).

The second line is the command line. The following commands are accepted on this line:

ADD

Allows entry of new lines to the end of the text in the text buffer from the keyboard. Line entry is terminated by an empty line. Lines of any length less than 254 characters can be entered.

CP=

Sets the currentline pointer to the value supplied. The status display on line 1 will show the currentline pointer (Pn). The currentline pointer is the line at the top of the text area, or window (line 3).

D or -

Moves the window down by the amount given on the command line. A minus sign will also work. For example:

```
D23  moves down 23 lines
D    moves down default amount
-16  moves down 16 lines
```

DEL

Deletes lines from the currentline pointer downwards by the amount given on the command line. An "*" will delete all lines to the end of the text buffer. The default is one line. For example:

```
DEL* deletes all lines to end
DEL23 deletes 23 lines
```

DOS

Executes the DOS command supplied on the command line. For example:

```
DOSCATALOG,D2
```

This command will pause to allow you to view any results. Strike any key to return to editing.

END

Returns control back to Applesoft Basic. If you need to restart the Editor after using END, but would like to retain the current text buffer unchanged, then use "GOTO 1000" to restart.

FN=

Sets the filename. For example:

```
FN=NEWFILE
```

The new filename will show up on line 1.

FM=

Sets the filemode (slot and drive). The slot is entered as Sn and the drive as Dn. A comma must be between them. For example:

```
FM=S5,D2  (Slot 5, Drive 2)
```

GET

Gets text from the filename (set by previous FN= and FM=) and adds it to the end of the lines currently in the text buffer.

H

Help listing of commands and controls.

IN=

Sets the default increment (used in the U, D, Ctl-L and Ctl-O commands and controls). The current value of the increment is shown in the rightmost field on line 1 (In).

LS=

Sets the column to be put at the left of the screen window. For example:

```
LS=34
```

In this example, column 1 on the screen would be column 34 in the text buffer. The current LS value shows on line 1 as Ln.

PR#

Prints the text buffer to the slot given. On my machine, entering PR#1 will print the buffer to my printer.

PUT

Puts (writes) the text buffer out to

contd.

the file determined by the current filename and filemode. If an old file of that name exists, it is deleted and a new file written.

TOP
Sets the currentline pointer to line 1 of the text buffer.

U or +
Moves the window up by the amount given on the command line. A plus sign will also work. This is similar to the D command, but in the other direction.

Carriage Return
A carriage return will cause the fullscreen edit controls to be enabled. The cursor enters the home position on the text part of the display.

The next area of the screen, lines 3 through 22, is the text window area into the text buffer. Normally, up to 20 lines of 39 characters will be displayed. You can use the leftside (LS) command to control which column is put on the leftside of the screen in the window. This allows lines longer than 40 characters to be edited (albeit 39 characters at a time). The last line in this area will either be the message *** EOF *** for the end of the file, or the message *** MORE FOLLOWS *** to show that there is more text after the last line on the text window.

In fullscreen edit mode (which is initiated by a carriage return on the command line), there is a set of control characters that provide for cursor movement, line insertion, line deletion, window movement, and text entry. These controls are entered within the text window. The controls are:

Ctl-D
Deletes the character under the cursor and moves the rest of the line to the left.

Ctl-E
Sets markers in the text for deletion or copying. You must set two markers that determine the range. They will be labeled in the text as marker 1 and marker 2. The markers can be on the same line.

Ctl-F
Causes a copy operation to take place from the range specified by the markers to just before the current cursor place. You cannot copy into the marker range. To perform a move, first copy and then do a delete (Ctl-R). You must reset the markers before the delete.

Ctl-H
Backspaces cursor with end of line wraparound. The back arrow key will also work.

Ctl-I
Insertion. It works by copying the current line so that it can be typed over. This can also be used to make copies of a line.

Ctl-J
Advances cursor with wraparound.

Ctl-K
Tabs 5 spaces to the right with wraparound.

Ctl-L
Lowers the text window by the default amount (normally 5 lines but this can be altered with the IN= commmand).

Ctl-M
Carriage return motion with wraparound from bottom to top. Carriage return key will also work.

Ctl-N
Vertical downwards cursor movement with wraparound.

Ctl-O
Raises the window the default amount.

Ctl-P
Truncates the line at the point of the cursor. Be sure to truncate lines that might have data or characters off the screen to the right that are not visible (if you do not want them in the text buffer).

Ctl-R
Used with the text markers set by Ctl-E to remove lines from the buffer.

Ctl-U
Vertical upwards cursor movement with wraparound. The right arrow key will also work for this function.

Ctl-Y
Homes the cursor to the upper left.

Escape key
Exits fullscreen edit mode and returns to the command mode.

Any non-control key
Replaces text in window at the current cursor position and moves the cursor one space to the right. Note that lines longer than the window can be entered while inputting characters, as long as no control characters are hit, and as long as they do not exceed 253 characters.

Below is the listing of the Editor. The major sections of the listing include the following:

```
  10      Branch to initilization at 10000.
 90-150 Print of text area routine.
200-230 Disk input routine.
250-292 Keyboard input routine.
500-540 Horiz. and vert. cursor control.
600-699 Pointer value corrections.
1000-1060 Main input loop for commands.
1100-1999 Inputs for fullscreen mode.
2000-2999 Dispatch section for command.
3000-4999 Various command service rout.
3000-3050 ADD routine.
3100-3130 DEL routine.
3200-3220 U or + routine.
```

contd.

```
3300-3320 D or - routine.
3400-3450 GET routine.
3500-3560 PUT routine.
3600-3630 DOS routine.
3700-3740 PR# routine.
4000      First non-control character.
4010      Next characters.
4020-4060 Line replacement.
4100-4110 Ctl-I routine.
4200      Ctl-L routine.
4300      Ctl-O routine.
4400-4420 Ctl-P routine.
4500-4510 Ctl-E routine.
4600-4690 Ctl-R routine.
4700-4760 Ctl-F routine.
4800-4850 Ctl-D routine.
5000-5999 Help display.
10000-10999 Initializations.
```

To reiterate, this fullscreen editor is of use for many small text editing jobs. While not as sophisticated as some of the available word processors and the like, it doesn't cost as much either! To save you the trouble of keying in the Editor, it will be available shortly on the WAP ABBS.

```
]LIST

10  GOTO 10000
90  HTAB 1: VTAB 3
95  CALL  - 958
100 LP = CP + 20: IF LP > LC THEN
    LP = LC
110  IF CP = LP THEN  PRINT "***
    EOF ***": RETURN
115  GOSUB 120: IF LP = LC THEN  PRINT
    "*** EOF ***": RETURN
116  PRINT "*** MORE FOLLOWS ***"
    : RETURN
120  FOR I = CP TO LP - 1
125 J =  LEN (L$(I))
130  IF J < LS THEN  PRINT : NEXT
    I: RETURN
140  IF J < RS THEN  PRINT  MID$
    (L$(I),LS,J - LS + 1): NEXT
    I: RETURN
150  PRINT  MID$ (L$(I),LS,RS - L
    S): NEXT I: RETURN
200 I$ = ""
210  GET A$: PRINT A$;
220  IF A$ = C$ THEN  RETURN
230 I$ = I$ + A$: GOTO 210
250 I$ = ""
260  GET A$: PRINT A$;
270  IF A$ = C$ THEN  RETURN
280  IF A$ =  CHR$ (8) THEN 285
281 I$ = I$ + A$: GOTO 260
285  PRINT " "; CHR$ (8);
290  IF  LEN (I$) = 0 THEN  PRINT
    ">";: GOTO 250
291  IF  LEN (I$) = 1 THEN  GOTO
    250
292 I$ =  LEFT$ (I$, LEN (I$) - 1
    ): GOTO 260
500  IF H < 0 THEN H = RS - LS +
    1
510  IF H > RS - LS + 1 THEN H =
    0
520  IF V < 0 THEN V = LP - CP -
    1
530  IF V > LP - CP - 1 THEN V =
    0
540  VTAB 1: HTAB H + 1: VTAB 3 +
    V: RETURN
600  IF CP < 0 THEN CP = 0
610  IF LS < 1 THEN LS = 1
620 RS = LS + 39
630  IF CP > LC THEN CP = LC
699  RETURN
1000  REM
1006  GOSUB 1010: GOTO 1040
1010  HOME : GOSUB 600
1015  PRINT F$;FT$;"  ";: HTAB 20
    : PRINT "L";LS;" C";LC;" P";
    CP;" I";IN
1020  PRINT ">";
1030  GOSUB 90: RETURN
1040  VTAB 2: HTAB 2: GOSUB 250
1050 V = 0:H = 0
1060  IF  LEN (I$) > 0 THEN  GOSUB
    2000: GOTO 1000
1100 LP = CP + 20: IF LP > LC THEN
    LP = LC
1190 I$ = "":I1 = 1
1200  GET A$:A =  ASC (A$)
1210  IF A >  = 32 THEN  GOSUB 40
    00: GOTO 1200
1220  IF I1 = 0 THEN  GOSUB 4020
1230  IF A = 8 THEN H = H - 1: GOSUB
    500: GOTO 1200
1240  IF A = 10 THEN H = H + 1: GOSUB
    500: GOTO 1200
1250  IF A = 21 THEN V = V - 1: GOSUB
    500: GOTO 1200
1260  IF A = 14 THEN V = V + 1: GOSUB
    500: GOTO 1200
1270  IF A = 25 THEN H = 0:V = 0:
     GOSUB 500: GOTO 1200
1280  IF A = 13 THEN H = 0:V = V +
    1: GOSUB 500: GOTO 1200
1290  IF A = 11 THEN H = H + 5: GOSUB
    500: GOTO 1200
1300  IF A = 9 THEN 4100
1310  IF A = 12 THEN 4200
1320  IF A = 15 THEN 4300
1330  IF A = 16 THEN  GOSUB 4400:
     GOTO 1200
1340  IF A = 5 THEN 4500
1350  IF A = 18 THEN 4600
1360  IF A = 6 THEN 4700
1370  IF A = 4 THEN 4800
1900  IF A = 27 THEN 1000
```

contd.

```
1999 I$ = "BAD CONTROL CHAR": GOSUB
     2999: GOSUB 1010: GOSUB 500:
      GOTO 1200
2000 J =  LEN (I$):I$ = I$ + "
       "
2010  IF  LEFT$ (I$,3) = "FN=" THEN
     F$ =  MID$ (I$,4,J - 3): RETURN

2015  IF  LEFT$ (I$,3) = "FM=" THEN
     FT$ = "," +  MID$ (I$,4,J -
     3): RETURN
2020  IF  LEFT$ (I$,3) = "LS=" THEN
     LS =  VAL ( MID$ (I$,4)): RETURN

2035  IF  LEFT$ (I$,3) = "CP=" THEN
     CP =  VAL ( MID$ (I$,4)): RETURN

2036  IF  LEFT$ (I$,3) = "IN=" THEN
     IN =  VAL ( MID$ (I$,4)): RETURN

2040  IF  LEFT$ (I$,3) = "END" THEN
      HOME : END
2050  IF  LEFT$ (I$,3) = "ADD" THEN
     3000
2060  IF  LEFT$ (I$,3) = "TOP" THEN
     CP = 0: RETURN
2070  IF  LEFT$ (I$,3) = "DEL" THEN
     3100
2071  IF  LEFT$ (I$,3) = "DOS" THEN
     3600
2080  IF  LEFT$ (I$,1) = "U" OR  LEFT$
     (I$,1) = "+" THEN 3200
2090  IF  LEFT$ (I$,1) = "D" OR  LEFT$
     (I$,1) = "-" THEN 3300
2100  IF  LEFT$ (I$,3) = "GET" THEN
     3400
2110  IF  LEFT$ (I$,3) = "PUT" THEN
     3500
2120  IF  LEFT$ (I$,3) = "PR#" THEN
     3700
2997  IF  LEFT$ (I$,1) = "H" THEN
     5000
2999  VTAB 2: HTAB 2: PRINT "????
     "I$: FOR I = 0 TO 1000: NEXT
     I: RETURN
3000  VTAB 3: HTAB 1
3005 CP = LC
3010  CALL  - 958
3020  GOSUB 250
3030  IF  LEN (I$) = 0 THEN CP =
     LC - 19: GOSUB 600: RETURN
3040 L$(CP) = I$:CP = CP + 1:LC =
     LC + 1
3050  GOTO 3020
3100  IF  MID$ (I$,4,1) = "*" THEN
     LC = CP: RETURN
3110 J =  VAL ( MID$ (I$,4)): IF
     J = 0 THEN J = 1
3115 PC = CP
3120  FOR I = CP + J TO LC:L$(CP)
      = L$(I):CP = CP + 1: NEXT I
     :LC = LC - J
3130 CP = PC: RETURN
3200 J =  VAL ( MID$ (I$,2)): IF
     J = 0 THEN J = IN
3205 CP = CP + J
3210  IF CP > LC THEN CP = LC
3220  RETURN
3300 J =  VAL ( MID$ (I$,2)): IF
     J = 0 THEN J = IN
3305 CP = CP - J
3310  IF CP < 0 THEN CP = 0
3320  RETURN
3400  PRINT D$;"OPEN ";F$;FT$
3410  PRINT D$;"READ ";F$
3420  GOSUB 200:L$(CP) = I$
3425  IF  LEN (I$) = 0 THEN 3450
3426 CP = CP + 1
3430  IF CP > LC THEN LC = LC + 1

3431  GOTO 3420
3450  PRINT D$;"CLOSE ";F$:CP = 0
     : RETURN
3500  IF LC = 0 THEN  RETURN
3510  PRINT D$;"OPEN ";F$;FT$
3520  PRINT D$;"DELETE ";F$;FT$
3530  PRINT D$;"OPEN ";F$,FT$
3540  PRINT D$;"WRITE ";F$
3550  FOR I = 0 TO LC - 1: PRINT
     L$(I): NEXT I
3560  PRINT : PRINT D$;"CLOSE ";F
     $: RETURN
3600  HOME : PRINT : PRINT D$;"MO
     N C,I,O"
3610  PRINT D$; MID$ (I$,4)
3620  PRINT D$;"NOMON C,I,O"
3630  PRINT "STRIKE ANY KEY TO CO
     NTINUE";: GET A$: PRINT : RETURN

3700  IF LC = 0 THEN  RETURN
3710  PRINT D$;I$
3720  FOR I = 0 TO LC - 1: PRINT
     L$(I): NEXT I
3730  PRINT  CHR$ (12)
3740  PRINT D$;"PR#0": RETURN
4000  IF I1 THEN I$ = A$: PRINT A
     $;:I1 = 0: RETURN
4010 I$ = I$ + A$: PRINT A$;: RETURN

4020 I1 = 1:I2 = CP + V:I3 =  LEN
     (L$(I2)):I4 =  LEN (I$)
4030  IF I3 + 1 < LS + H THEN L$(
     I2) = L$(I2) +  LEFT$ (B$,LS
      + H - I3 - 1) + I$:H = H +
     I4: RETURN
4035  IF I3 + 1 = LS + H THEN L$(
     I2) = L$(I2) + I$:H = H + I4
     : RETURN
4040  IF I3 > LS + H - 1 AND I3 <
      = LS + H + I4 - 1 AND LS +
```

contd.

```
    H > 1 THEN L$(I2) =  LEFT$ (
    L$(I2),LS + H - 1) + I$:H =
    H + I4: RETURN
4050  IF LS + H = 1 AND I3 <  = I
    4 THEN L$(I2) = I$:H = H + I
    4: RETURN
4055  IF LS + H = 1 AND I3 > I4 THEN
    L$(I2) = I$ +  MID$ (L$(I2),
    LS + I4):H = H + I4: RETURN

4060 L$(I2) =  LEFT$ (L$(I2),LS +
    H - 1) + I$ +  MID$ (L$(I2),
    LS + H + I4):H = H + I4: RETURN

4070  PRINT L$(I2): PRINT I$: PRINT
    I3,LS,H,I4: END
4100 LC = LC + 1
4110  FOR I = LC TO CP + V + 1 STEP
     - 1:L$(I) = L$(I - 1): NEXT
    I: GOSUB 90: GOSUB 500: GOTO
    1200
4200 CP = CP - IN: GOSUB 1010: GOSUB
    500: GOTO 1200
4300 CP = CP + IN: GOSUB 1010: GOSUB
    500: GOTO 1200
4400  CALL  - 868
4410  IF LS + H = 1 THEN L$(CP +
    V) = " ": RETURN
4420 L$(CP + V) =  LEFT$ (L$(CP +
    V),LS + H - 1): RETURN
4500  IF I5 = 0 THEN K1 = CP + V:
    I5 = 1:K1$ = L$(K1):L$(K1) =
    "*** MARKER 1 ***": GOSUB 90
    : GOSUB 500: GOTO 1200
4510 I5 = 0:K2 = CP + V:K2$ = L$(
    K2):L$(K2) = "*** MARKER 2 *
    **": GOSUB 90: GOSUB 500: GOTO
    1200
4600  IF K2 < K1 THEN I = K1:K1 =
    K2:K2 = K1
4610  IF K2 >  = LC - 1 THEN LC =
    K1: GOTO 4690
4620  FOR I = K2 + 1 TO LC - 1:L$
    (K1) = L$(I):K1 = K1 + 1: NEXT
    I:LC = K1
4690  GOSUB 1010: GOSUB 500: GOTO
    1200
4700 L$(K1) = K1$:L$(K2) = K2$
4710  IF K2 < K1 THEN I = K1:K1 =
    K2:K2 = K1
4720 K3 = CP + V:K4 = K2 - K1 + 1

4730  IF K3 >  = K1 AND K3 <  = K
    2 THEN 4690
4740  IF K3 > K2 THEN 4760
4750 K5 = LC + K4: FOR I = LC TO
    K3 STEP  - 1:L$(K5) = L$(I):
    K5 = K5 - 1: NEXT I: FOR I =
    K3 TO K3 + K4 - 1:L$(I) = L$
    (K1 + K4):K1 = K1 + 1: NEXT
    I:LC = LC + K4: GOTO 4690
4760 K5 = LC + K4: FOR I = LC TO
    K3 STEP  - 1:L$(K5) = L$(I):
    K5 = K5 - 1: NEXT I: FOR I =
    K3 TO K3 + K4 - 1:L$(I) = L$
    (K1):K1 = K1 + 1: NEXT I:LC =
    LC + K4: GOTO 4690
4800  CALL  - 868
4810  IF LS + H >  LEN (L$(CP + V
    )) THEN 1200
4820  IF (LS + H = 1) AND ( LEN (
    L$(CP + V)) = 1) THEN L$(CP +
    V) = " ": GOTO 1200
4830  IF LS + H =  LEN (L$(CP + V
    )) THEN L$(CP + V) =  LEFT$
    (L$(CP + V),LS + H - 1): GOTO
    1200
4840  IF LS + H = 1 THEN L$(CP +
    V) =  MID$ (L$(CP + V),2)
4850  IF LS + H > 1 THEN L$(CP +
    V) =  LEFT$ (L$(CP + V),LS +
    H - 1) +  MID$ (L$(CP + V),L
    S + H + 1)
4860  GOSUB 90: GOSUB 500: GOTO 1
    200
5000  HOME : PRINT "HELP MENU": PRINT
    "  COMMANDS (LINE 2)"
5010  PRINT "ADD  ADD NEW LINES T
    O END"
5020  PRINT "CP=  SET CURRENTLINE
     POINTER TO VALUE"
5030  PRINT "D    MOVE DOWN INCRE
    MENT AMOUNT"
5040  PRINT "DNNN MOVE DOWN NNN A
    MOUNT"
5050  PRINT "DEL  DELETE LINES. *
     DEL'S ALL"
5060  PRINT "DOS  DOS COMAND"
5070  PRINT "END  RETURN TO BASIC
    "
5080  PRINT "FN=  SET FILENAME"
5090  PRINT "FM=  SET FILEMODE (S
    LOT,DRIVE)"
5100  PRINT "GET  GETS TEXT FROM
    FN TO END OF BUFFER"
5110  PRINT "H    LIST HELP MENU"

5120  PRINT "IN=  SET DEFAULT INC
    REMENT"
5130  PRINT "LS=  SET LEFT COLUMN
    "
5140  PRINT "PR#  PRINT TO DESIGN
    ATED SLOT"
5150  PRINT "PUT  PUT TEXT TO FN
    FROM BUFFER"
5160  PRINT "TOP  SET CP TO TOP"
5170  PRINT "U    MOVE UP INCREME
    NT AMOUNT"
5180  PRINT "UNNN MOVE UP NNN LIN
    ES"
```

contd.

```
5190  PRINT "+NNN MOVE DOWN NNN L
      INES"
5200  PRINT "-NNN MOVE UP NNN LIN
      ES"
5210  PRINT "CARRIAGE RETURN   EN
      TER EDIT MODE"
5300  PRINT "HIT ANY KEY TO CONTI
      NUE...";: GET A$: PRINT : HOME

5400  PRINT "FULLSCREEN EDIT COMM
      ANDS"
5405  PRINT "CTL-D  DELETE CHARAC
      TER"
5410  PRINT "CTL-E  SET MARKERS"
5420  PRINT "CTL-F  COPY FROM MAR
      KERS TO HERE"
5430  PRINT "CTL-H  BACKSPACE"
5440  PRINT "CTL-I  COPY CURRENT
      LINE (INSERT)"
5450  PRINT "CTL-J  FORWARD SPACE
      "
5460  PRINT "CTL-K  TAB 5 SPACES
      RIGHT"
5470  PRINT "CTL-L  LOWER WINDOW
      BY INCREMENT"
5480  PRINT "CTL-M  CARRIAGE RETU
      RN MOTION"
5490  PRINT "CTL-N  DOWNWARDS VER
      TICAL MOTION"
5500  PRINT "CTL-O  RAISE WINDOW
      INCREMENT AMOUNT"
5510  PRINT "CTL-P  TRUNCATE LINE
       AT THIS POINT"
5520  PRINT "CTL-R  REMOVE LINES
      BETWEEN MARKERS"
5530  PRINT "CTL-U  UPWARDS VERTI
      CAL MOTION"
5540  PRINT "CTL-Y  HOME CURSOR T
      O UPPER LEFT"
5550  PRINT "CARRIAGE RETURN   SA
      ME AS CTL-M"
5560  PRINT "ESCAPE  RETURN TO CO
      MMAND MODE"
5570  PRINT "ANY NON-CTL KEY   RE
      PLACE TEXT"
5600  PRINT "HIT ANY KEY TO CONTI
      NUE...";: GET A$: PRINT : RETURN

10000 D$ =  CHR$ (4)
10010 C$ =  CHR$ (13)
10015 ML = 500:LC = 0
10020  DIM L$(ML)
10030 FT$ = ",S6,D1"
10040 CP = 0
10050 LS = 1:RS = 38
10060 IN = 5
10070 B$ = "

      "
10080  PRINT "NAME OF FILE>";: GOSUB
       250:F$ = I$
10999  GOTO 1000
30000  REM **********************
30001  REM *WASHINGTON APPLE PI*
30002  REM *A SIMPLE FULLSCREEN*
30003  REM *   TEXT EDITOR      *
30004  REM *     10/10/81       *
30005  REM **********************
```

]

WASHINGTON APPLE PI AT THE COMPUTER SHOW

by Bernie Urban

Mixed feelings about the Mid-Atlantic Computer Show this year. Fewer exhibits than last year...but certainly a lot cooler. Different exhibitors...more minis, peripherals and vendors. There were perhaps fewer attendees, but they sure crowded around our booth. Our representatives who fielded questions and sold our wares all seemed to enjoy themselves, despite the hard work. All in all, we "faired" well. Will we do it again next year? Some questions on that. Perhaps not, unless we get advance notice and assurance that there will be more exhibitors and that it will be a better show.

SORT YOUR DIRECTORY

by Andy O'Brien

We are about to enter a realm that is visited by only a few foolhearty souls. Unlike those APPLE owners who are content to use disk drives through standard DOS commands, we will use the deadly RWTS, which will allow us to read or write anywhere on the disk. Boy, I'm beginning to scare myself!

Actually, what I am about to describe could be considered dangerous (to your disks) and if you are apprehensive about fooling around with your disk directory, please go on to the next article. For you brave souls who are still with me, this could be a rewarding and educational experience.

Some months ago, I had a hard time locating a file that I needed. It was right in front of me, but when I did a CATALOG, I missed it because the files weren't listed in alphabetical order. This prompted me to write SRTDIR, a program which will test your faith in using programs appearing in this newsletter. Just kidding! I have been using it for months with no ill effects. In fact, I use this program periodically on all of my disks and have been very pleased with its performance.

It is very simple to operate. Assuming SRTDIR has been BSAVE'd to disk, simply BLOAD SRTDIR. Then insert the disk whose directory is to be alphabetized and type CALL 28672. The disk will turn on and in a few seconds the directory on the disk will be sorted.

Please look through Listing 1 so that you have an understanding of what the program is doing. There are parts of the code that can be improved, like the sort itself. From experience, however, I have always been reluctant to change things once I am satisfied that they are working. I do suggest, however, that you use a test disk at first, whether you modify SRTDIR or not, just to prove to yourself that it works.

For those of you who are still too squeamish to use SRTDIR, look for my next program which will read the directory, sort it and then display it along with sectors used and sectors free. However, since it will be a read-only program, the directory itself will remain unchanged. Good luck!

Listing 1.

```
         ********************************************************
3        *                    SRTDIR.ASM                        *
4        *                       by                             *
5        *                   Andy O'Brien                       *
6        *                                                      *
7        *    This program reads in a disk catalog,             *
8        *    sorts it, and writes it back out again.           *
         ********************************************************
10       *
11       RECLEN     EQU   $23
12       DATATO     EQU   $7400
13       DATARW     EQU   $9600
14       TR         EQU   DATARW+$1
15       SC         EQU   DATARW+$2
16       *
17       TO         EQU   $10
18       FROM       EQU   TO+$2
19       NUMNAM     EQU   TO+$4
20       TEMP       EQU   TO+$5
21       TEMP1      EQU   TO+$6
22       ILOOP      EQU   TO+$8
23       TMPNAM     EQU   TO+$9
24       FINAL      EQU   TO+$B
25       TEST       EQU   TO+$D
26       *
27       RWTS       EQU   $3D9
28       IOB        EQU   $B7E8
29       EXPVOL     EQU   IOB+$3
30       TRACK      EQU   IOB+$4
31       SECTOR     EQU   IOB+$5
32       RWBUFF     EQU   IOB+$8
```

contd.

```
                        33    COMAND     EQU   IOB+$C
                        34    ACTVOL     EQU   IOB+$E
                        35    *
                        36               ORG   $7000
                        37               OBJ   $7000
                        38    *
                        39    *
                              ********************************
                        41    *  Main program
                              ********************************
                        43    *
7000:  D8               44               CLD
7001:  20 0E 70         45               JSR   INIT
7004:  20 25 70         46               JSR   GETDIR
7007:  20 66 70         47               JSR   SORT
700A:  20 E4 70         48               JSR   PUTDIR
700D:  60               49               RTS
                        50    *
                        51    *
                              ********************************
                        53    *  Initialize variables
                              ********************************
                        55    *
700E:  A9 00            56    INIT       LDA   #$0             Clear number of
7010:  85 14            57               STA   NUMNAM          files counter.
7012:  A9 00            58               LDA   #<DATAT0        Initialize
7014:  85 10            59               STA   T0              pointers to
7016:  A9 74            60               LDA   #>DATAT0        the sorting
7018:  85 11            61               STA   T0+$1           area.
                        62    *
701A:  A9 0F            63    INIT2      LDA   #15             Setup first sector
701C:  8D ED B7         64               STA   SECTOR          of the directory
701F:  A9 11            65               LDA   #17             for the call to
7021:  8D EC B7         66               STA   TRACK           RWTS.
7024:  60               67               RTS
                        68    *
                        69    *
                              ********************************
                        71    *  Get the directory
                              ********************************
                        73    *
7025:  20 40 71         74    GETDIR     JSR   READ            Read a sector and
7028:  A9 0B            75               LDA   #$B             point to the
702A:  85 12            76               STA   FROM            first file
702C:  A9 96            77               LDA   #>DATARW        entry in that
702E:  85 13            78               STA   FROM+$1         sector.
                        79    *
7030:  A0 03            80    CHKEND     LDY   #$3             Return when there are no
7032:  B1 12            81               LDA   (FROM),Y        more entries in the
7034:  C9 00            82               CMP   #0              directory.
7036:  D0 01            83               BNE   CHKDEL
7038:  60               84               RTS
                        85    *
7039:  A0 00            86    CHKDEL     LDY   #$0             If the file is deleted,
703B:  B1 12            87               LDA   (FROM),Y        go to NEXT.
703D:  C9 FF            88               CMP   #$FF
703F:  F0 16            89               BEQ   NEXT
7041:  E6 14            90               INC   NUMNAM          Increment the number of
7043:  B1 12            91    MOVE       LDA   (FROM),Y        entries counter and move
7045:  91 10            92               STA   (T0),Y          the current entry to the
```

contd.

```
7047:  C8        93              INY   .               sorting area.
7048:  C0 23     94              CPY   #RECLEN
704A:  D0 F7     95              BNE   MOVE
704C:  18        96              CLC
704D:  A5 10     97              LDA   TO              Update the pointer into the
704F:  69 23     98              ADC   #RECLEN         sorting area.
7051:  85 10     99              STA   TO
7053:  90 02     100             BCC   NEXT
7055:  E6 11     101             INC   TO+$1
                 102   *
7057:  18        103   NEXT      CLC
7058:  A5 12     104             LDA   FROM            Update the pointer into
705A:  69 23     105             ADC   #RECLEN         the buffer containing the
705C:  85 12     106             STA   FROM            current sector of the
705E:  90 D0     107             BCC   CHKEND          directory.
                 108   *
7060:  CE ED B7  109   NEWSCT    DEC   SECTOR          Decrement the sector number
7063:  4C 25 70  110             JMP   GETDIR          and get the next sector.
                 111   *
                 112   *
                       **********************************
                 114   *  Sort the directory
                       **********************************
                 116   *
7066:  E6 14     117   SORT      INC   NUMNAM          Setup
7068:  A9 00     118             LDA   #$0
706A:  85 18     119             STA   ILOOP           variables
706C:  A9 DD     120             LDA   #<DATATO-$23
706E:  85 10     121             STA   TO              for
7070:  A9 73     122             LDA   #>DATATO-$1
7072:  85 11     123             STA   TO+$1           sorting
7074:  A5 14     124             LDA   NUMNAM
7076:  85 15     125             STA   TEMP            routines.
                 126   *
7078:  E6 18     127   JLOOP     INC   ILOOP           Top of
707A:  A5 10     128             LDA   TO              sorting loop.
707C:  85 1D     129             STA   TEST
707E:  A5 11     130             LDA   TO+$1
7080:  85 1E     131             STA   TEST+$1
                 132   *
7082:  A6 18     133             LDX   ILOOP
7084:  18        134   TOP       CLC
7085:  A5 1D     135             LDA   TEST
7087:  69 23     136             ADC   #RECLEN         Update the
7089:  85 1D     137             STA   TEST            TEST pointer.
708B:  90 02     138             BCC   HERE1
708D:  E6 1E     139             INC   TEST+$1
708F:  CA        140   HERE1     DEX
7090:  D0 F2     141             BNE   TOP
                 142   *
7092:  A5 1D     143             LDA   TEST            Reset
7094:  85 1B     144             STA   FINAL           FINAL and
7096:  85 19     145             STA   TMPNAM          TMPNAM
7098:  A5 1E     146             LDA   TEST+$1         pointers.
709A:  85 1C     147             STA   FINAL+$1
709C:  85 1A     148             STA   TMPNAM+$1
                 149   *
709E:  A6 18     150             LDX   ILOOP
70A0:  A0 03     151   COMPAR    LDY   #$3             Compare one
70A2:  B1 19     152   COMP1     LDA   (TMPNAM),Y      directory
```

contd.

```
70A4:  D1 1D      153              CMP   (TEST),Y        entry against
70A6:  F0 05      154              BEQ   CONT            another.
70A8:  90 10      155              BCC   INCI
70AA:  4C B2 70   156              JMP   REPLACE
70AD:  C8         157   CONT       INY
70AE:  C0 20      158              CPY   #$20
70B0:  D0 F0      159              BNE   COMP1
                  160   *
70B2:  A5 1D      161   REPLACE    LDA   TEST            Exchange pointers.
70B4:  85 19      162              STA   TMPNAM
70B6:  A5 1E      163              LDA   TEST+$1
70B8:  85 1A      164              STA   TMPNAM+$1
70BA:  18         165   INCI       CLC
70BB:  A5 1D      166              LDA   TEST
70BD:  69 23      167              ADC   #RECLEN         Get next
70BF:  85 1D      168              STA   TEST            directory
70C1:  90 02      169              BCC   HERE2           entry.
70C3:  E6 1E      170              INC   TEST+$1
70C5:  E8         171   HERE2      INX
70C6:  E4 14      172              CPX   NUMNAM
70C8:  D0 D6      173              BNE   COMPAR
                  174   *
70CA:  A0 00      175              LDY   #$0             Move the entries
70CC:  B1 1B      176   SRT2       LDA   (FINAL),Y       around so that
70CE:  85 16      177              STA   TEMP1           they will be in
70D0:  B1 19      178              LDA   (TMPNAM),Y      alphabetical
70D2:  91 1B      179              STA   (FINAL),Y       order.
70D4:  A5 16      180              LDA   TEMP1
70D6:  91 19      181              STA   (TMPNAM),Y
70D8:  C8         182              INY
70D9:  C0 23      183              CPY   #RECLEN
70DB:  D0 EF      184              BNE   SRT2
                  185   *
70DD:  C6 15      186              DEC   TEMP            Are we done yet?
70DF:  D0 97      187              BNE   JLOOP           No.
70E1:  C6 14      188              DEC   NUMNAM          Yes.
70E3:  60         189              RTS
                  190   *
                  191   *
                        ********************************
                  193   *  Put the directory
                        ********************************
                  195   *
70E4:  A9 00      196   PUTDIR     LDA   #<DATAT0        Setup variables
70E6:  85 10      197              STA   T0
70E8:  A9 74      198              LDA   #>DATAT0        for writing
70EA:  85 11      199              STA   T0+$1
70EC:  20 1A 70   200              JSR   INIT2           the sorted
70EF:  A9 11      201              LDA   #$11
70F1:  8D 01 96   202              STA   TR              directory back
70F4:  A9 0F      203              LDA   #$F
70F6:  8D 02 96   204              STA   SC              out to disk.
                  205   *
70F9:  CE 02 96   206   PUT1       DEC   SC              When sector counter
70FC:  F0 41      207              BEQ   RETN            is 0, return.
70FE:  A9 00      208              LDA   #<DATARW
7100:  85 12      209              STA   FROM
7102:  A9 96      210              LDA   #>DATARW
7104:  85 13      211              STA   FROM+$1
7106:  A0 0B      212              LDY   #$B
7108:  A9 00      213              LDA   #$0
```

contd.

```
710A:  91 12     214   ZERO      STA   (FROM),Y       Zero out RWTS
710C:  C8        215             INY   .              buffer from byte
710D:  D0 FB     216             BNE   ZERO           $8 to byte $FF.
710F:  A9 0B     217             LDA   #$B
7111:  85 12     218             STA   FROM
                 219   *
7113:  C6 14     220   SCT       DEC   NUMNAM
7115:  30 1F     221             BMI   WRAPUP
7117:  A0 00     222             LDY   #$0            Put the sorted
7119:  B1 10     223   PUT       LDA   (TO),Y         directory
711B:  91 12     224             STA   (FROM),Y       in the RWTS
711D:  C8        225             INY   .              buffer.
711E:  C0 23     226             CPY   #RECLEN
7120:  D0 F7     227             BNE   PUT
7122:  18        228             CLC
7123:  A5 10     229             LDA   TO
7125:  69 23     230             ADC   #RECLEN
7127:  85 10     231             STA   TO
7129:  90 02     232             BCC   INCF
712B:  E6 11     233             INC   TO+$1
                 234   *
712D:  18        235   INCF      CLC
712E:  A5 12     236             LDA   FROM           Update the
7130:  69 23     237             ADC   #RECLEN        FROM pointer.
7132:  85 12     238             STA   FROM
7134:  90 DD     239             BCC   SCT
                 240   *
7136:  20 5C 71  241   WRAPUP    JSR   WRITE          We filed a sector,
7139:  CE ED B7  242             DEC   SECTOR         write it out.
713C:  4C F9 70  243             JMP   PUT1
                 244   *
713F:  60        245   RETN      RTS
                 246   *
                       **********************************
                 248   *  Disk I/O routines
                       **********************************
                 250   *
7140:  A9 96     251   READ      LDA   #>DATARW
7142:  8D F1 B7  252             STA   RWBUFF+$1
7145:  A9 00     253             LDA   #<DATARW
7147:  8D F0 B7  254             STA   RWBUFF
714A:  A9 00     255             LDA   #$0
714C:  8D EB B7  256             STA   EXPVOL
714F:  A9 01     257             LDA   #1
7151:  8D F4 B7  258             STA   COMAND
7154:  A9 B7     259             LDA   #$B7
7156:  A0 E8     260             LDY   #$E8
7158:  20 D9 03  261             JSR   RWTS
715B:  60        262             RTS
                 263   *
715C:  A9 96     264   WRITE     LDA   #>DATARW
715E:  8D F1 B7  265             STA   RWBUFF+$1      --- END ASSEMBLY ---
7161:  A9 00     266             LDA   #<DATARW
7163:  8D F0 B7  267             STA   RWBUFF         TOTAL ERRORS: $00
7166:  A9 00     268             LDA   #$0
7168:  8D EB B7  269             STA   EXPVOL         CODE STARTS AT $7000
716B:  A9 02     270             LDA   #2                    ENDS AT $7177
716D:  8D F4 B7  271             STA   COMAND              RANGE =  $0178
7170:  A9 B7     272             LDA   #$B7
7172:  A0 E8     273             LDY   #$E8           $0178 BYTES OF OBJECT CODE GENERATED.
7174:  20 D9 03  274             JSR   RWTS
7177:  60        275             RTS                                        ®
```

APPLE AND APPLE COMPATIBLE PRODUCTS

Model No	Description	Cat	Retail	Discount
A2S1048	Apple II PLUS, 48K RAM	CM	1530.00	1095.00
A2B0001	Prototyping Card	B	24.00	21.00
A2B0002	Parallel Card	B	180.00	154.00
EPS-00P	Epson Parallel Card	B	85.00	79.95
GRAPPLER	Graphics Parallel Intfc with software & cable	B	165.00	159.95
A2B0003	Communications Card	B	225.00	193.00
A2B0005	Serial Card	B	195.00	167.00
A2B0006	Pascal Language System	SB	495.00	419.00
A2B0007	Centronics Printer Card	B	225.00	193.00
A2B0009	Applesoft II ROM Card	B	200.00	169.00
A2B0010	Integer BASIC Card	B	200.00	169.00
A2B0015	IEEE-488	B	450.00	389.00
A2B0043	Language Card	B	195.00	175.00
A2M0024	Clock/Calendar	B	280.00	249.00
EPS-00C	Epson Printer Cable	O	25.00	22.50
A2M0044	Disk II w/Intfc/DOS 3.3	DB	645.00	530.00
A2M0003	Disk II Drive	D	525.00	450.00
CCC-Z1	12" Zenith B/W Monitor	T	150.00	142.50
A2M0029	Graphics Tablet	G	795.00	660.00
A2M0007	Hand Controls	O	29.95	26.00
A2M0009	Vinyl Carrying Case	O	30.00	27.00
A2M0019	Programmers Aid #1	O	50.00	46.00
A2M0027	Auto-Start ROM	O	65.00	59.00
A2G0001	Vinyl Diskette Pages	O	8.50	7.95
A2G0002	Software Binder	O	9.50	8.95
A2G0003	Apple II Extended Warranty	O	225.00	225.00
A2M0036	Silentype Printer	P	395.00	345.00
EMX80	Epson MX80 Printer	P	645.00	499.00

Clinton COMPUTER

9020-C Woodyard Road , Clinton, Md.

868-0005

Route 5 South from Beltway. Right on Woodyard Road (Route 223).

apple computer

Authorized Dealer & Service Center

To order: Please send cashier's check, money order, or personal check (allow 2 wks to clear). VISA or Master card add 2%. Insurance, Shipping and Handling Charge add 1% ($3.00 min). Maryland residents or items picked up at store add 5% Md sales tax. Equipment subject to price change and availability.

ON THE OVERALL STRUCTURE OF APPLESOFT by C.K. Mesztenyi

INTRODUCTION.

This article attempts to describe the overall structure of Applesoft in the ROM space $D000-F7FF. It may be considered as a preceding chapter to [2] which gives descriptions of many subroutines and Zero page usage.

Before going into details, I must define certain terms for the sake of this article which may be very confusing in the Applesoft Manual [1]. These terms are the 'statement', 'command', 'instruction', 'line number' and 'line'. The first three of these are used somewhat interchangeably in the Manual. It refers to REM and Assignment or LET statements in Chapter 1, lists them as Commands together with ABS in Appendix Q, and assumes them to be instructions in Chapter 2 and Appendix N. I do not intend to clear all these confusions and errors in the syntactic definition and subsequently used terminology. Instead, the following syntactic definitions will be used here with the hope that I will not confuse the issue further. These definitions are as follows:

```
statement           := end-st / for-st/
                       / ...  / new-st
let-st              := assign-st / LET
                       assign-st
compound-statement  := statement CR/
                       statement :
                       compound-statement
labeled-statement   := linenumber
                       compound-statement
```

I.e. I define a 'statement' as any of the 64 statements with the keyword 'end', 'for',... as listed in the keyword column of the Statement Type Entry Table; the syntactic rules of these individual statements are given in the Manual under their descriptions. The compound-statement is a list of (simple) statements separated by ':', while the labeled-statement is a line number followed by the compound-statement which the Manual defined as 'line'. CR stands for carriage return.

With these definitions, one can state that a compound-statement is a program in immediate mode, while a labeled-statement is a program part in deferred mode.

1. DATA STRUCTURE.

The data areas used by Applesoft reside:

1. Flags and temporaries on Zero page.
2. Five Tables in memory $D000-D364.
3. Scattered (locally used) data interspersed in the program area $D365-F7FF.
4. Zero page load data in memory $F10B-F126.
5. Stored program normally from memory address $0801.
6. Variable areas.

1.1 Zero Page.

The Zero page use is described in [1], pp. 140-141. Further information may be found in [2], [3] and [4].

1.2 Tables.

The five tables residing in $D000-D364 are are follows:

```
$D000-D07F = Statement Type Entry Table.
$D080-D0B1 = Function Entry Table.
$D0B2-D0CF = Operator Tag and Entry
             Table.
$D0D0-D25F = Keyword Token Table.
$D260-D364 = ASCII Messages.
```

The Statement Type Entry Table is used to recognize statements and to obtain the proper entry points in the program area. It consists of 64 2-byte entries containing the entry point low-high addresses minus one. The order of the 64 entries correspond to the tokens, 128 to 191, assigned to the keywords END to NEW, as given in [1] p. 121. Table 1 summarizes these data, giving the actual entry point addresses.

The Function Entry Table is used during expression evaluation to obtain entry points to the function subroutines in the program area. It consists of 25 2-byte entries with low-high addresses. The order of the entries corresponds to the tokens 210 to 234 assigned to the keywords SGN to MID$ as given in [1], p. 121. Table 2 gives the summary. The description of the function subroutines with their entry points are given in [2].

The Operator Tag and Entry Table is used during expression evaluation. It consists of 10 3-byte entries corresponding to the tokens 200 to 209 assigned to the keywords + to < as given in [1], p. 121. Of these 3 bytes, the first byte contains the Tag which also serves as a precedence number. The next two bytes contain the low-high addresses minus one of the entry points in the program area. Table 3 shows the Tag values and actual entry point addresses.

The Keyword Token Table is used by the Tokenizer routine which replaces keywords by appropriate tokens. It consists of the 107 keywords (from END to MID$) concatenated such that each byte is an ASCII character with high bit set to zero, unless the character is the last one of a keyword, in which case it is set to 1. E.g. it contains

E N D F O R N E X T . . .

where the '-' over the character indicates that the high bit is one.

The ASCII Message Table contains ASCII characters where the individual message

contd.

(e.g. the error message part 'SYNTAX ERROR') is separated either by having the high bit set to its last character byte, or followed by a zero byte.

1.3 Scattered Data.

Scattered data may occur in many places; some of them are the floating point constants (see [2] and [4]), short table for high resolution graphics (see [3]), etc.

1.4 Zero Page Load Data.

The memory area $F10B-F126 is the CHRGET/CHRGOT routine followed by an initial random number which gets loaded into the Zero page $B1-CC during initialization.

1.5 Stored Progam Area.

Zero page locations $67-68 contain the address (low-high) of the beginning of the stored program, usually $0801. From this address, the memory contains the tokenized label-statements ordered by their line numbers. The format of a tokenized label-statement is as follows:

2-byte pointer (low-high address) to the next tokenized statement
2-byte binary value (low-high) of the line number bytes of the tokenized compound-statement
1-byte containing zero

The last tokenized labeled-statement is followed by two extra bytes containing zero. Thus the stored program has a chain of pointers starting with the contents of $67-68, and ending with a zero value. Each pointer indicates the beginning of a labeled-statement, while a byte containing zero indicates its end; and three zero bytes indicate the end of the stored program.

1.6 Variable Areas.

These areas and corresponding pointers are adequately described in [1], with further explanations in [5].

2. CHRGET/CHRGOT SUBROUTINE.

The most important subroutine in Applesoft is the CHRGET/CHRGOT subroutine residing on the Zero page $B1-C8 with the TXTPTR imbedded at $B8-$B9. It has been described in [2] but it is repeated here because of its importance.

The CHRGOT entry ($B7) loads the register A with the contents of the memory whose address is in the TXTPTR ($B8-B9, low-high). CHRGET entry (B1) does the same except it increments the TXTPTR prior to loading. If the obtained byte is equal to the ASCII space ($20) then the control goes back to CHRGET, i.e. spaces (blanks) are skipped. Otherwise the flag Z is set if A=$3A or $00, i.e. ASCII colon (:) or null; flag C is set if A is not an ASCII number 0 to 9, i.e. A<$30 or A>$39; finally the control goes back to the calling routine.

The importance of this routine comes into light if one compares it to an instruction fetch cycle in a computer with the TXTPTR as a counter register. The instruction code is returned in register A, flags Z and C, ready to be executed (interpreted). The ASCII space code behaves like a no-op, and is automatically skipped. This feature is realized in the implementation of gosub- and return-statements by placing the TXTPTR value together with line-number and tag $B0 on the stack in the gosub- statement, resetting them in the return-statement. Unfortunately, the call-statement has been implemented differently by not saving the above data in the stack. It would have been simple to implement in the same way as the gosub-statement, and the return-statement could have served as a return address from the machine language subroutine. This would have allowed a call of the Applesoft routine at $D43C with a call-statement from a stored program with request for input of a compound-statement ending with RETURN ready to be executed in immediate mode, where the RETURN causes the return to the stored program.

3. PROGRAM STRUCTURE.

The overall program structure of Applesoft can be illustrated by the following semantic program:

3.1. Initialization
3.2. Request and receive input from the keyboard.
3.3. Tokenize the input
3.4. If the first character of the input is an ASCII number then store the input as part of the stored program, and goto 3.2.
3.5. If the first character of the input is not an ASCII number then execute the input as a program, after which goto 3.2.

3.1 Initialization.

The Initialization (starting at $F128) sets up the Zero page and various other pointers.

3.2 Input.

The input request starts at $D43C. It uses the subroutine at $D52E to display the prompt symbol and through the Monitor GETLN, to receive the input line into the input buffer at $0200. It sets the high bits of the input data to zero, places a zero byte after the last input character, and initializes the TXTPTR to the input buffer address minus one.

3.3 Tokenization.

The Tokenization Subroutine ($D559-D619, with entry at $D559) replaces the keywords with the appropriate tokens in the input buffer. It also removes blanks with the result still in the input buffer. It places two extra zero bytes at the end of the line. No syntax checking is performed by this routine.

Following the Tokenization, the first character in the input buffer decides whether 3.4 or 3.5 is to be executed.

contd.

3.4 Stored Program.

If the first character in the input buffer is an ASCII number then Applesoft assumes it to be the first character of a line-number of a labeled-statement and either inserts it or replaces an old labeled-statement with the same line-number in the stored program with the help of the routine starting at $D46A.

3.5 Execution.

If the first character of the input is not an ASCII number then Applesoft assumes the input to be a compound-statement ready to be executed. It sets the TXTPTR to the beginning of the input buffer and enters into an execution loop at $D805. At this stage, TXTPTR really behaves like a program counter. The execution of a statement advances or changes TXTPTR, e.g. to the stored program. Finally, the control returns to 3.2 requesting new input under the following conditions:

(i) Execution of an end- or stop-statement
(ii) Encountering 3 consecutive zero bytes
(iii) Detecting syntax error without an onerr-statement.

Individual statements are recognized by their first, possibly tokenized, byte. If this is between $80 and $BF then it is assumed to be a token, and the statement is executed by jumping to the appropriate entry point listed in Table 1. Otherwise it is assumed to be a let-statement without the word LET. These statement execution routines are called as subroutines, but not all of them return.

The execution loop in $D805-D848, and its preceding section in $D7D2-D804, is fairly complex. It is listed below with appropriate remarks.

```
        Statement Handler Routine
          $D7D2 - $D804

NEWSTT  TSX             save
        STX   $F8          stackpointer
        JSR   $D858     check for control-C
        LDA   $B8       get
        LDY   $B9          TXTPTR
        LDX   $76       check if immediate mode
        INX                   ($FF in current line #)
        BEQ   N1
        STA   $79       no, thus put TXTPTR into
        STY   $7A          old TXTPTR
N1      LDY   #$00      check byte at TXTPTR
        LDA   ($B8),Y
        BNE   COLON     if non-zero then it should be ':'
        LDY   #$02      if zero then end of compound-st.
        LDA   ($B8),Y   check for end of program
        CLC                zero pointer 2 bytes further
        BEQ   PREND
        INY             it is a new labeled-statement
        LDA   ($B8),Y      get and store new
        STA   $75          current line #
        INY
        LDA   ($B8),Y
        STA   $76
        TYA                update TXTPTR
        ADC   $B8
        STA   $B8
        BCC   EXECUTE
        INC   $B9

EXECUTE BIT   $F2       check for the trace bit
        BPL   L1        notrace if positive
        LDX   $76       trace is on, check
        INX             for mode
        BEQ   L1        no print in immediate mode
        LDA   #$23      print out line #
        JSR   $DB5C     as trace information
        LDX   $75
        LDA   $76
        JSR   $ED24
        JSR   $DB57
L1      JSR   CHRGET    get first byte of statement
        JSR   STYPE     use JSR to get return address
                        in stack for
STTRET  JMP   NEWSTT    <-- statement execution subroutine
                            returns here
PREND   BEQ   $D88A     end of program

STYPE   BEQ   $D857     statement type check
                              on its first byte
        SBC   #$80
        BCC   ASGST        <$80 then assign statement
        CMP   #$40
        BCS   $D846        >$BF then error
        ASL                otherwise get
        TAY                entry point
        LDA   $D001,Y      from the 2-byte
        PHA                statement-type table
        LDA   $D000,Y      and put it into stack
        PHA                as return address of CHRGET
        JMP   CHRGET       and go to there
ASGST   JMP   $DA46     go to LET-st. routine

COLON   CMP   #$3A      check for colon
        BEQ   EXECUTE      yes, go to execute
        JMP   $DEC9        otherwise error
```

Addresses:				
	NEWSST	$D7D2	N1	$D7E5
	EXECUTE	$D805	L1	$D81D
	STTRET	$D823	PREND	$D826
	STYPE	$D828	ASGST	$D83F
	COLON	$D842		

CONCLUSION.

With the knowledge of the Data Structure, one may trace the internal workings of Applesoft based on the five point (3.1 to 3.5) Program Structure, and on the 64 statement interpreter subroutines with the given entry points. There are two difficult parts which need further documentation.

1. The expression evaluation routine, called by FRMEVL in [2], which is used by many statement routines. I think that part of the complication is because Applesoft has been implemented before its syntactic rules were (correctly?) established.

2. The other difficulty lies in the multiple use of the stack. Beside the statement subroutines (gosub-, return-, call-, for- and next-statement), FRMEVL uses it, and also the internal programs in Applesoft (JSR, RTS instructions).

contd.

Dealer's Corner

HLA Inc. has just opened in Beltsville and is offering a special service to WAP members. They will sell your used computer equipment, software, etc. on consignment at a charge of 5 percent of the actual sale price.

Appendix. Tables.

Statement Type Entry Table from $D000-D07F

HEX TOKEN	KEY-WORD	ENTRY-POINT	HEX TOKEN	KEY-WORD	ENTRY POINT
$80	END	$D870	$A0	COLOR=	$F24F
$81	FOR	$D766	$A1	POP	$D96B
$82	NEXT	$DCF9	$A2	VTAB	$F256
$83	DATA	$D995	$A3	HIMEM:	$F286
$84	INPUT	$DBB2	$A4	LOMEM:	$F2A6
$85	DEL	$F331	$A5	ONERR	$F2CB
$86	DIM	$DFD9	$A6	RESUME	$F318
$87	READ	$DBE2	$A7	RECALL	$F3BC
$88	GR	$F390	$A8	STORE	$F39F
$89	TEXT	$F399	$A9	SPEED=	$F262
$8A	PR#	$F1E5	$AA	LET	$DA46
$8B	IN#	$F1DE	$AB	GOTO	$D93E
$8C	CALL	$F1D5	$AC	RUN	$D912
$8D	PLOT	$F225	$AD	IF	$D9C9
$8E	HLIN	$F232	$AE	RESTORE	$D849
$8F	VLIN	$F241	$AF	&	$03F5
$90	HGR2	$F3D8	$B0	GOSUB	$D921
$91	HGR	$F3E2	$B1	RETURN	$D96B
$92	HCOLOR=	$F6E9	$B2	REM	$D9DC
$93	HPLOT	$F6FE	$B3	STOP	$D86E
$94	DRAW	$F769	$B4	ON	$D9EC
$95	XDRAW	$F76F	$B5	WAIT	$E784
$96	HTAB	$F7E7	$B6	LOAD	$D8C9
$97	HOME	$FC58	$B7	SAVE	$D8B0
$98	ROT=	$F721	$B8	DEF	$E313
$99	SCALE=	$F727	$B9	POKE	$E77B
$9A	SHLOAD	$F775	$BA	PRINT	$DAD5
$9B	TRACE	$F26D	$BB	CONT	$D896
$9C	NOTRACE	$F26F	$BC	LIST	$D6A5
$9D	NORMAL	$F273	$BD	CLEAR	$D66A
$9E	INVERSE	$F277	$BE	GET	$DBA0
$9F	FLASH	$F280	$BF	NEW	$D649

Function Entry Table from $D080-D031

HEX TOKEN	KEY-WORD	ENTRY-POINT	HEX TOKEN	KEY-WORD	ENTRY POINT
$D2	SGN	$EB90	$DF	SIN	$EFF1
$D3	INT	$EC23	$E0	TAN	$F03A
$D4	ABS	$EBAF	$E1	ATN	$F09E
$D5	USR	$000A	$E2	PEEK	$E764
$D6	FRE	$E2DE	$E3	LEN	$E6D6
$D7	SCRN(	$D412	$E4	STR$	$E3C5
$D8	PDL	$DFCD	$E5	VAL	$E707
$D9	POS	$E2FF	$E6	ASC	$E6E5
$DA	SQR	$EE8D	$E7	CHR$	$E646
$DB	RND	$EFAE	$E8	LEFT$	$E65A
$DC	LOG	$E941	$E9	RIGHT$	$E686
$DD	EXP	$EF09	$EA	MID$	$E691
$DE	COS	$EFEA			

Operator Tag and Entry Table from $D0B2-D0CF

HEX TOKEN	KEY-WORD	HEX TAG	ENTRY POINT
$C8	+	$79	$E7C1
$C9	-	$79	$E7AA
$CA	*	$7B	$E982
$CB	/	$7B	$EA69
$CC	^	$7D	$EE97
$CD	AND	$50	$DF55
$CE	OR	$46	$DF4F
$CF	>	$7F	$EED0
$D0	=	$7F	$DE98
$D1	<	$64	$DF65

References:

[1] Applesoft, Basic Programming Reference Manual.
[2]-[5] are all available in "Call-A.P.P.L.E in Depth" No. 1. Apple Puget Sound Program Library Exchange. 1981.
[2] John Crossley: Applesoft Internals.
[3] C.K. Mesztenyi: Notes on Hi-Res Graphics Routines.
[4] David A. Lingwood: Amplifying Applesoft.
[5] Val J. Golding: Applesoft from Bottom to Top.

FORMATTED OUTPUT OF FLOATING POINT NUMBERS

by J. Philip Childress

I joined Washington Apple Pi recently at the DC Armory Computer Show, and would like to contribute the attached subroutine for publication and to the library. It allows formatted output of floating point numbers as strings, with dollar signs, commas in the integer part, etc., plus rounding to a designated number of decimal places. It will either "fix" or "float" the minus sign and dollar sign to the left side of the field or to the left of the most significant digit or decimal point. It is rather slow, so I sometimes modify it to return quickly if a switch is set, putting an IF test after line 10180.

Some explantion of what it does:

The FM(k) vector is the "format statement":

- FM(1) = field width (e.g. 12 spaces).
- FM(2) = No. of decimal places (e.g. 3). If negative, no decimal point.
- FM(3) = 0 or 1; if 1, then a $ sign is printed.
- FM(4) = 0 or 1; if 1, then the position of $ and - signs are fixed to leftmost spaces.
- FM(5) = 0 or 1; if 1, then commas are added to the integer part to separate thousands, millions, etc.

N is the number to be transformed. D$ is the equivalent string to be printed after the GOSUB. It is good practice to put the FM(k) initialization in a one-line subroutine so that individual formats may be changed in the PRINT statement and the "standard" format reinitialized afterwards by a GOSUB.

A few comments on how it works:

Line 10120 Performs the rounding. IP is the integer part of the number, on the next line.

Line 10150 Calls subroutine 10500 to add commas to the integer part if FM(5)=1. The new IP string returns in T$(5), and is transferred to T$(3). The S/R checks for E format - a potential problem.

Line 10160 Is complicated. R(2) is the decimal part plus 10^k, where k is the desired number of decimal places. If R(5) were not then added, the decimal part would not have zeroes padded in the right side if FM(2) is greater than N's decimal part.

Line 10190 If FM(2) is negative, no decimal point or fraction is printed.

Line 10210 Puts "*" in all places if the field is too small, and returns.

Lines 10230 - 10490 Entry points to fix or float a minus sign and dollar sign.

Other Notes:

I have also used this program to print tables with "N/A" entered where data are not available, by setting up the table entry data with -99999 entries, testing around statement 10190 and returning D$ = "N/A" if the trigger number is encountered.

It would be good to program this in assembly language, but the APPLE documentation is relatively scarce concerning the actual treatment of floating point numbers. Perhaps it could be compiled with one of the new compilers to speed it up.

```
]LIST
10000  REM    SUBROUTINE TO RIGHT JUSTIFY,ROUND,ADD COMMAS & FIX OR FLOAT
10010  REM   A "-" AND "$" SIGN. IF NUMBER OVERFLOWS FIELD,THEN "****"
10020  REM  IS PRINTED.INPUTS N,THE NUMBER.OUTPUTS D$, A STRING EQUIVALENT
10030  REM   FM(K) ARE THE CONTROLS-PUT THEM IN MAIN PROGRAM.
10040  REM   BY PHIL CHILDRESS,WASHINGTON,D.C.AUGUST 31,81.
10050  REM   (IT'S SLOW,BUT IT WORKS)
10060  REM  FM(X) IS FORMAT-1=FIELD,2=DEC PLCS,3=$IF 1,4=FIX IF 1,5=, IF 1.
10070 D$ = ""
10080 T$(1) = " "
10090 T$(2) = ""
10100 R(6) =  SGN (N) + .0005
10110 R(5) = 10 ^ FM(2)
10120 R(4) =  ABS (N) + 0.5 / R(5)
10130 IP =  INT (R(4))
10140 T$(3) =  STR$ (IP)
10150  IF FM(5) = 1 THEN  GOSUB 10500:T$(3) = T$(5)
10160 R(2) =  INT (R(5) * (R(4) - IP)) + R(5)
10170  IF FM(2) > 0 THEN T$(2) =  RIGHT$ ( STR$ (R(2)),FM(2))
```

contd.

Riverbank Software Inc.

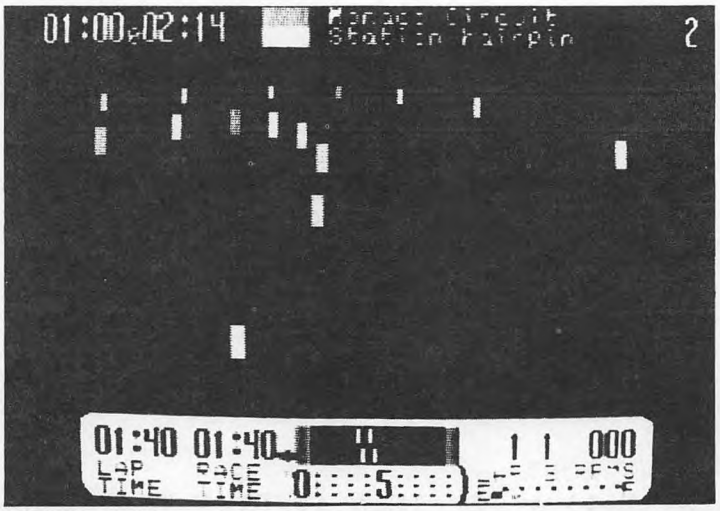
PROUDLY INTRODUCES

INTERNATIONAL GRAN PRIX

an arcade-like race car simulation
by RICHARD ORBAN
author of THREE MILE ISLAND*

REQUIRES
APPLE II★ ★ OR
APPLE II PLUS★ ★
48K, 13 OR 16
SECTOR DISK,
PADDLE CONTROL

VISA/MASTERCHARGE
MONEY ORDERS/COD
DEALER/DISTRIBUTOR
INQUIRIES INVITED



$30.00 PER DISK
MD. RESIDENTS ADD 5%
U.S. CURRENCY ONLY

See your local dealer
or call Toll-free
24 hours for
ORDERS ONLY:
1-800-327-9191 ex. 244
(Outside Florida)
INQUIRIES AND FLORIDA
ORDERS: 301-479-1312
SMITH'S LANDING ROAD
POST OFFICE BOX 128
DENTON, MD. 21629

● SPECIAL FEATURES ●

Five GRAND PRIX-style road circuits, including: Oulton Park, Warwick Farm, Karlskoga, and Monaco ● Five speed manual or automatic transmission (with or without cruise control) ● Eight levels of difficulty.

● ADDITIONAL FEATURES ●

Speeds to 198 MPH ● controlled skids ● spinouts ● spectacular crashes ● hair pin turns ● narrow corners ● obstacles ● identified circuit features ● number of laps selection ● flashing last lap indicator ● 'Christmas tree' controlled start ● switch for silent operation ● blue post marks 300' intervals ● best lap/best race times posted ● fully instrumented control panel: lap timer ● race timer ● indicator lights ● edge detectors ● position indicator ● steering indicator ● moving speed tape ● lap counter ● gear and RPM indicators ● operating fuel gauge

RIVERBANK WILL REPLACE DAMAGED DISKS WITHIN 1 YEAR OF PURCHASE. RETURN DISK WITH PROOF OF PURCHASE PLUS FIVE DOLLARS POSTAGE AND HANDLING FOR IMMEDIATE REPLACEMENT (TEN DOLLARS OVERSEAS)

*TRADEMARK MUSE CO., BALT. MD. ★ ★ APPLE COMPUTER, INC., CUPERTINO, CA.

```
10180 D$ = T$(3) + "." + T$(2)
10190  IF FM(2) < 0 THEN D$ = T$(3)
10200 LL =  LEN (D$)
10210  IF LL + 1 > FM(1) THEN D$ = "": FOR WW = 1 TO FM(1):D$ = D$ + "*": NEXT
     WW: RETURN
10220 R(1) = FM(1) - LL - 2
10230  IF FM(3) <  > 1 THEN 10380
10240  IF FM(4) <  > 1 THEN 10310
10250  REM      -$bbbb52.5843  FIXED -$.
10260  REM     R(1)IS NO OF BLANKS TO PAD OUT WITH
10270  IF R(1) < 1 THEN 10290
10280  FOR WW = 1 TO R(1):D$ = T$(1) + D$: NEXT
10290  IF R(6) > 0 THEN D$ = " $" + D$: RETURN
10300 D$ = "-$" + D$: RETURN
10310  REM      FLOAT -$ THE LEFT OF DIGITS
10320  IF R(6) > 0 THEN D$ = " $" + D$
10330  IF R(6) < 0 THEN D$ = "-$" + D$
10340  IF R(1) < 1 THEN  RETURN
10350  FOR WW = 1 TO R(1):D$ = T$(1) + D$
10360  NEXT
10370  RETURN
10380 R(1) = R(1) + 1
10390  IF FM(4) <  > 1 THEN 10450
10400  REM      -bbbb52.5843 FIXED --NO $
10410  IF R(1) < 1 THEN 10430
10420  FOR WW = 1 TO R(1):D$ = T$(1) + D$: NEXT
10430  IF R(6) > 0 THEN D$ = T$(1) + D$: RETURN
10440 D$ = "-" + D$: RETURN
10450  REM       FLOAT A MINUS SIGN
10460  IF R(6) > 0 THEN D$ = T$(1) + D$
10470  IF R(6) < 0 THEN D$ = "-" + D$
10480  IF R(1) < 1 THEN  RETURN
10490  FOR WW = 1 TO R(1):D$ = T$(1) + D$: NEXT : RETURN
10500  REM   SUB TO PUT IN COMMAS
10510 LL =  LEN (T$(3))
10520  REM   IF E FORMAT PUNT
10530  IF  MID$ (T$(3),2,1) = "E" THEN T$(5) = T$(3): RETURN
10540  IF LL < 4 THEN T$(5) = T$(3): RETURN
10550  FOR WW = 1 TO  INT ((LL - 1) / 3)
10560 T$(4) =  MID$ (T$(3),LL - 3 * WW + 1,3)
10570  IF WW = 1 THEN T$(5) = T$(4): GOTO 10590
10580 T$(5) = T$(4) + "," + T$(5)
10590  NEXT WW
10600 T$(6) =  STR$ ( INT ((IP + .00001) * .001 ^ (WW - 1)))
10610 T$(5) = T$(6) + "," + T$(5)
10620  RETURN
]
```

##########

"INSIDE WASHINGTON APPLE PI" is published! See Order Form on page 45.

##########

HIT PARADE

by John Alden

WELCOME to the continuing series of game surveys. This article is a result of a survey of SIGAMES members during the July meeting. During the follwing months, other games will be surveyed at the SIGAMES meetings.

The groupings include adventure games, simulations, war games, arcade games, board games, and puzzle games. Each grouping (except arcade and board games) will be subgrouped into text or graphical games.

Each category or subcategory will be surveyed at the beginning of the SIGAMES meeting. The results will be presented at the next SIGAMES meeting and published in the following issue of the club newsletter.

Anyone who cannot attend the meetings and would like to contribute to the surveys should contact Al Gass (703) 371-3560 or John Alden (202) 686-1656.

New contributions will be incorporated into the appropriate survey and it will be republished as soon as possible.

Here is the scedule for conducting the surveys:

MEETING	CATEGORY SURVEYED
November	Space games (graphic)
December	Space games (text)
January	Sport games (all)
February	Puzzle games (all)
March	Board games
April	Adventure (graphic)
May	War games (all)
June	Adventure (text)

The following surveys have been conducted:

June	Adventure (graphic)
July	War games (all)
August	Adventure (text)
September	Simulation (all)
October	Arcade

CATEGORIES

The surveys are based on the following seven categories:

ACTION: The action is the series of events which form the plot or theme of any game. It is the pace of the game. It should move along rapidly, yet the player should not have to defend his or her life when entering each new room or area. The game should not grind to a halt because the player cannot locate the tool or object necessary to advance past an obstacle.

REALISM: In a fantasy game????? Absolutely!!!!! In the August issue of "Creative Computing," Robert Plamondon stated that realism "means that none of the events breaks the character's 'willing suspension of disbelief.' Players can accept magic anddraoonsas part of the background of the fantasy world. They cannot accept worlds that turn upside down at night, outdoor human colonies on the sun, or personal clues displayed on billboards."

RULES: Traditionally, the worst aspect of most games has been the rules. If they were complete, they were written for cryptographers. If the rules were at all understandable, they were incomplete.

BALANCE: How quickly were you killed the last time you tried an adventure game? Balance refers to the capability of the program to act as a referee and as your opponent and still present you with a fair chance to win.

STRATEGY: What is the overall game plan? Is it to get the golden apple, rescue the princess, destroy the asteroid, or escape the island? Strategy is the planning and developing of game goals.

TACTICS: Hand-to-hand combat!!! you versus the computer dragon!!! Tactics is the step-by-step process for a successful strategy.

contd.

BUY..SELL..SWAP BUY..SELL..SWAP

COMPUTER

F L E A M A R K E T

Looking for a bargain? Have a piece of hardware or software that's just sitting around gathering dust? Come to the First Annual

W A S H I N G T O N A P P L E P I / N O V A P P L E

C O M P U T E R F L E A M A R K E T

To be held at the November meeting of the Washington Apple Pi. November 21 at 9:30 AM, George Washington University. Limited table space for sellers will be available on a first come, first served basis. No admission. No fees. We only ask that sellers observe two rules:

1. All copyrighted software for sale MUST be on original disks or tapes and be accompanied by original documentation.
2. Absolutely no commercial sales.

So come and check out the bargains!

WASHINGTON APPLE PI and NOVAPPLE cannot assume responsibility for any item offered for sale or trade.

BUY..SELL..SWAP BUY..SELL..SWAP

CUMM: The cumulative score for each of the games surveyed.

LIST $: The manufacturer's suggested list price

TEXT ADVENTURES:	ACTION	COMPNT	REALISM	RULES	BALANCE	STRATEGY	TACTICS	CUMM:	LIST $
ZORK PART I	8.50	8.00	9.50	9.00	9.00	9.00	7.50	8.64	$39.95
OO-TOPOS	8.00	7.00	8.00	8.00	8.00	8.00	8.00	7.86	$29.95
CLASSIC ADVENTURE	5.67	8.00	9.33	8.67	9.00	7.00	5.67	7.62	$29.95
STRANGE ODYSSEY	8.00	5.00	9.00	5.00	8.00	8.00	8.00	7.29	$19.95
PYRAMID OF DOOM	5.00	7.00	8.50	7.00	9.00	8.00	6.50	7.29	$19.95
VOODOO CASTLE	7.00	5.00	9.00	5.00	8.00	8.00	8.00	7.14	$19.95
THE COUNT	7.00	5.00	9.00	5.00	8.00	8.00	8.00	7.14	$19.95
MISSION: IMPOSSIBL	7.00	5.00	8.00	5.00	7.00	7.00	8.00	6.71	$19.95
GHOST TOWN	6.50	7.00	7.50	5.50	7.50	7.00	5.00	6.57	$19.95
ADVENTURELAND	7.00	4.50	6.00	4.00	5.50	5.50	4.50	5.29	$19.95
PIRATE'S ADVENTURE	6.00	4.00	5.50	4.00	5.00	5.00	4.50	4.86	$19.95

INTERNATIONAL APPLE CORE
P. O. BOX 976, DALY CITY, CALIFORNIA 94017 USA

ISSUE

42

NOVEMBER 1981

WASHINGTON APPLE PI

OPEN FORUM
ON
SOFTWARE COPY PROTECTION
AND
NIBBLE COPY PROGRAMS

MAY 2, 1981

INTRODUCTION:

This report summarizes views aired during an open forum on copy protection and nibble copiers. The International Apple Core is providing this summary because of the strong interest expressed in the subject by our members and sponsors.

The forum was held on May 2, 1981, prior to the IAC Annual Meeting. Before opening the meeting, the IAC President asked members of the audience to come forward to briefly state their views on the subject. Attendees ranged from representatives of user groups to representatives of software and hardware manufacturers. Each speaker was allowed two minutes. After thirty minutes the forum was ended to allow for the Annual Meeting.

A rough tape was made of the proceedings, from which this summary has been made. An attempt has been made to be as faithful to the original intent of the speakers as possible. It was not always possible to discern the speakers' names correctly. Their rough background was always clear, however, and is indicated with the comments.

The International Apple Core recognizes that unauthorized reproduction of copyright material for distribution is clearly illegal. The following comments represent the views of the individuals expressing them. They do not in any way represent the views or policies of the International Apple Core.

Joe Budge,
Secretary,
International Apple Core

FORUM SUMMARY:

SPEAKER 1: SIDNEY BROOKHILL
CHAMPAIGN-URBANA APPLE USERS GROUP

It is not appropriate or legal for the IAC to try to stop the sale of bit-copy programs. Advertisers putting pressure on the magazines seems like trying to outlaw Xerox machines. It is impossible, and unjustifiable, once a person has purchased a program, to try to stop him from manipulating his own data. Resale of copied programs, however, is definitely wrong. If anyone is caught doing that, that person can and should be brought to court by the software vendors. All of us need to exercise constraint. We should not diffuse software around the clubs. My experience has been that swapping is confined to software that the swappers wouldn't have purchased anyway.

SPEAKER 2: DAVID LINGWOOD
SECRETARY OF A.P.P.L.E.
(a user's group, not affiliated with Apple Computer)
SOFTWARE AUTHOR AND PUBLISHER

Our Board made a decision not to advertise bit copiers prior to hearing from any advertisers on the subject. Unfortunately there is a good reason for bit copy programs to exist. That is the abysmal back up policies that many software companies have. Paying $30 and waiting three weeks for a backup copy of a program you use every day is terrible. The club magazine is trying to pressure its advertisers to make available good backup policies. That doesn't make bit copy programs more palatable though. There should be a plague on both houses.

SPEAKER 3: KEN WILOPE
PAMLICO WISCONSIN APPLE USERS

The law says that an individual can make copies of purchased software. If I have an important piece of software, I want to make three backup copies of it. If it's locked, I feel I need to make even more before I feel comfortable. Locked copies only create lots of anxiety in the end user. I like C/PM software, which is not locked.

SPEAKER 4: TERRY TAYLOR
PUBLIC DOMAIN SOFTWARE LIBRARIAN

As a librarian, lots of folks send me programs they've copied from elsewhere. Some even go so far as to put their own name on the stuff. This is clearly ridiculous. I bought and used Locksmith to back up our library's data base, and haven't used it for anything else. My personal experience as I've crossed the nation a couple of times is that most people copy a program and then never use the stuff. They put it on the shelf and never would have bought it anyway. There seems to be a fad around where you have to have a copy of everything. On the other hand, as a user I don't like paying $200 for Visicalc with no backup. I won't pay $30 for a backup either, as I expect a complete product when I buy it. I paid $15 for the Vcopy program instead, and made one backup. There is no easy answer, but I don't think the answer is people ganging up on one product.

page 2 contd.

SPEAKER 5: CRAIG VAUGHAN
SOURCE APPLE USERS GROUP
PRESIDENT, SOFTWARE SORCERY, SOFTWARE PUBLISHERS

I don't protect any of my software. My products are the type that are in daily use by my customers. Not to let them make backups would be sending them up a creek without a paddle. In addition, I haven't seen a program on the market yet that was perfect for everybody. If something is protected, you can't get in there to make it fit your needs. Locked software has kind of limited usefulness because I can't get it onto my hard disk. I can't recommend Visicalc to anyone with a hard disk because it won't work. I think that publishers of high value software should let their software be open for people to backup and modify as they see fit. Piracy is a bad problem though. All we need is a good test case to really nail somebody to the wall. That will stop it. Copy protection itself generates a lot of interest in copying, simply as a challenge to break the scheme and make a copy.

SPEAKER 6: BARRY THAYER
MEMBER OF A.P.P.L.E.

I challenge A.P.P.L.E. to take a mail vote of all their membership about advertising Locksmith. On December 12, 1980 President Carter signed the Copyright Protection Act which modified the law and made it very clear that software is protected under the act and that individuals have the legal right to make backup copies for their own use. This is something that eveyrone who is against the bit copiers should keep in mind. There has been a big financial attack on bit copiers. The editorial director of Creative Computing magazine told me that advertisers had threatened to withdraw $10,000 per month in advertising if they advertised a bit copier. I'm very unhappy about that.

SPEAKER 7: KEN ROSE
MEMBER OF NORTHERN ILLINOIS APPLE USERS GROUP

There have always been, and will be, people capable of copying anything. Programs like Locksmith allow you to make a backup but don't make it any easier to make a copyable disk than it ever was. This isn't anything new, it's just spreading the knowledge and making it available on a more individual basis.

SPEAKER 8: JOHN MCMULLEN
PRESIDENT, BIG APPLE USER GROUP
PRESIDENT, MCMULLEN & MCMULLEN, SOFTWARE PUBLISHERS

There is tremendous ongoing paranoia about copying. The nibble copy programs generate attention and just feed the paranoia. Locked programs cause hardships for those who want to modify any systems. A guy who wants to integrate his data base with an accounting system is a good example. Anyway, it's impossible to stop software from being public if you sell it.

SPEAKER 9: GEORGE MEPHIS
MEMBER, NORTHER ILLINOIS APPLE USERS GROUP
SOFTWARE WRITER FOR MICROMUSIC, INC.

Libraries always buy more than one copy of a book, which they certainly can't back up. People are still allowed to make 1 copy of part of the work for occasional use. Musicians have a system to recieve royalties for life on their works via ASCAP. Maybe that would work. There were lots of illegal copies of video tapes until the police started arresting anyone who resold them. I would like to urge software vendors to sell backup copies at cost. I think a person has a right to make a backup copy for himself. There are lots of grey areas, so we have to be careful.

SPEAKER 10: LEE MEADOR
FORTH WORTH APPLE USERS GROUP
PUBLISHER OF VCOPY

The authors and companies that do not copy protect their software should be commended. There should be an organization that does nothing but provide backup copies to individuals. Software producers can lock users in without copy protection by providing high-level support. A good newsletter is an example which works sometimes.

SPEAKER 11: JIM HOYT
APPLE COMPUTER

At this point there is no real solution to the problem. There will always be some software vendors that suffer from paranoia. This session will provide recommendations to vendors who continue to copy protect their software. What I'm hearing is that most people use bit copiers to make archival backups. If that solution is provided by the vendors it really becomes a moot point whether the bit copier is there or not.

SPEAKER 12: DAVE GORDON
FOUNDER OF PROGRAMMA INTERNATIONAL
(SOFTWARE PUBLISHERS)
IAC VICE PRESIDENT
MEMBER OF ORIGINAL APPLE CORPS

I probably have one of the largest software libraries in the Apple world. As a user I love nibble copiers. We lie to ourselves when we say we use them only for archival purposes. Everyone who has a nibble copier at some point has copied something that was copyrighted. The purpose of the IAC is to convey ideas to vendors. Programma never protected any software. I think the pressure on the magazines is O.K., I can advertise anywhere I want. I had to take a stand. Programma is now thinking of protecting its software just because of the nibble copier. They spend more time cracking nibble copiers than they do writing software. It's a game. Copiers that can't copy themselves are ridiculous. The IAC should provide archival copies of everything as the representatives of the end users rather than allow the out and out copying that now goes on to continue. By the way, every publisher I talked to unilaterally decided not to publish the bit copier ads, before they recieved any threats.

SPEAKER 13: FRED WILKINSON
PRESIDENT, SAN FRANCISCO APPLE CORE
DIRECTOR, IAC

contd.

At a club meeting we asked the members what to do about a nibble copy program that a member had given the club to put on a club disk. The vote was like 52 to 48 to put it on the disk. Then there were lots of comments from the floor and we had a re-vote. The revote was 48 to 52, so it switched right around dead zero. I don't like all the pressure that was brought on the member and his family to say that the information shouldn't be distributed. If software vendors and magazines are putting pressure on the nibble copiers, not letting them advertise, then at least the magazines should be responsible enough to pressure vendors to provide reasonable backup by not letting them advertise until they do. The same pressure should be brought to both sides.

SPEAKER 14: NEAL LIPSON
PRESIDENT, PHILADELPHIA APPLE USERS GROUP
PARTNER, PROGRESSIVE SOFTWARE, SOFTWARE PUBLISHERS

As a user, I can't condem bit copiers. I recommend that the IAC take a strong position that piracy is counterproductive. There should be a strong moral atmosphere in which individuals are neither encouraged nor allowed to copy copyrighted material. If there is this atmosphere, I think lots of the publishers would drop protection. You can't teach morality, but you can encourage it and take steps to enforce it with peer pressure.

SPEAKER 15: LOU MILRAD:
PRESIDENT, APPLE-CAN MICROCOMPUTER USERS GROUP

In my view it is inappropriate for the IAC or a club to take a stand on the issue. It's a matter of concience for the individual user. Piracy is not the correct term for what's going on. It denotes one is lifting the property and selling it for profit, as happpened with cassettes and 8-tracks. The Apple Orchard should not accept ads if they are illegal or immoral, and they should if the ad's are neither. It is inappropriate for the IAC to set morality standards.

WAP TUTORIAL REGISTRATION

The second WAP tutorial will be held on two consecutive Saturdays, February 13 and 20, 1982, from 10:00 AM to 1:00 PM at USUHS on Jones Bridge Road (on the campus of the National Naval Medical Center) in Bethesda, MD. Check the ABBS and club phone for any changes in details. An outline of the two sessions is shown below.

February 13, 1982

9:00 - 11:30 Introduction
- A. Binary/hex number systems
- B. Bits, bytes, and nibbles
- C. RAM, ROM and devices

11:30 - 1:00 Internals
- A. Memory Map: What's really in there
- B. The Monitor: Examine, disassemble
- C. The mini-assembler, step and trace

February 20, 1982

9:00 - 11:30 Applesoft
- A. Basic programming
- B. Commands and applications
- C. Memory usage; HIMEM, LOMEM and variable space

11:30 - 1:00 DOS
- A. The Catalog and VTOC
- B. Reading and writing files
  1. sequential
  2. random access

WAP is requiring a nominal fee for this tutorial to assure the interest of attendees and to gauge the number of interested participants. A maximum of 40 people will be included, half with their own APPLE and half without.

NAME _____ _____

Daytime Phone_____ Evening Phone _____

Check One:
I will bring an APPLE (fee $25)_____
I will not bring an APPLE (fee $40) _____

Please return application and check made out to "Washington Apple Pi" to:
David Morganstein
13329 Woodruff Court
Germantown, MD 20874

INSIDE APPLE PI
MAIL ORDER FORM

"Inside Apple Pi" is published and ready for distribution. It is a bound compendium of articles from the Washington Apple Pi Newsletters of 1979 and 1980. We do not plan to print any more of the original 1979 and 1980 newsletters, so this will be the only form in which they will be available for purchase. You may order by mail using the form below. Copies will also be available for purchase at our regular monthly meeting (at a slightly lower price than that quoted below because no postage necessary.)

Make your check payable to "Washington Apple Pi". Foreign orders must be in U.S. dollars and payable on a U.S. bank.

( ) $ 7.50 U.S. Canada and Mexico
( ) $ 7.50 Foreign surface mail
( ) $12.50 Foreign air

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

Send order to: Washington Apple Pi
Attn. Inside Apple Pi
P.O. Box 34511
Bethesda, MD 20817

Amount Enclosed $ -------

ADVERTISING RATES

The following table shows our current advertising rates. Our newsletter distribution is about 1350 copies, with approximately 200 of these going around the country. If you would like to advertise please send camera ready ad copy (half tones are permitted, no bleeds) by the 10th of the month to:

Washington Apple Pi
Attention: Editor
P.O. Box 34511
Bethesda, MD 20817

	RATES			
	FULL PAGE	HALF PAGE	QTR PAGE	8TH PAGE
Single issue (or split series)	$ 40	$ 25	$ 15	$ 8
3-months series (or more)	35	20	10	5
Full-year contract	30	15	10	5

WASHINGTON APPLE PI
MAIL ORDER FORM

Washington Apple Pi has a program library, and disks are available for purchase by anyone. The price to members is $5.00 per disk and $8.00 to non-members. These disks are chock full of exceptional programs - the utilities are especially useful. The games are some of the best - not just simple and uninteresting ones. You may pick them up at any meeting or have them mailed for $2.00 per disk additional. (If you order five or more the additional charge will be $10.00 total.) They will come in a protective foam diskette mailer.

PROGRAM DISKETTES

Members: $5.00 picked up at meeting
$7.00 mailed to you (for the first five, remainder at $5.00)

Non-members: $8.00 per disk picked up at meeting
$10.00 mailed to you (for the first five, remainder at $8.00)

Volume 1	Utilities I	( )	Volume 36	Games XII	( )
Volume 2	Utilities II	( )	Volume 37	Utilities IX	( )
Volume 3	Games I	( )	Volume 38	Games XIII	( )
Volume 4	Games II	( )	Volume 39	IAC VII	( )
Volume 5	Games III	( )	Volume 40	IAC VIII	( )
Volume 6	Games IV	( )			
Volume 7	Games V	( )	VOLUME 100	DOS 3.3 UTILITIES A	( )
Volume 8	Utilities III	( )	VOLUME 101	DOS 3.3 UTILITIES B	( )
Volume 9	Educational I	( )	VOLUME 102	DOS 3.3 GAMES A	( )
Volume 10	Math/Science	( )	VOLUME 103	DOS 3.3 MERRY CHRISTMAS	( )
Volume 11	Graphics I	( )			
Volume 12	Games VI	( )			
Volume 13	Games	( )			
Volume 14	IAC Utilities IV	( )	*Volume 180	Dungeon Designer	( )
Volume 15	Games VII	( )	*Volume 181	Beginner's Cave	( )
Volume 16	Utilities V	( )	*Volume 182	Lair of Minotaur	( )
Volume 17	Graphics II	( )	*Volume 183	Cave of the Mind	( )
Volume 18	Educational II	( )	*Volume 184	Zyphur Riverventure	( )
Volume 19	Commumications	( )	*Volume 185	Castle of Doom	( )
Volume 20	Music	( )	*Volume 186	Death Star	( )
Volume 21	Apple Orchard	( )	*Volume 187	Devil's Tomb	( )
Volume 22	Utilities VI	( )	*Volume 188	Caves of Treas. Isl.	( )
Volume 23	Games VIII	( )	*Volume 189	Furioso	
Volume 24	Games IX	( )	*Volume 190	The Magic Kingdom	( )
Volume 25	Utilities VII	( )	*Volume 191	The Tomb of Molinar	( )
Volume 26	Stocks/Investments	( )	*Volume 192	Lost Island of Apple	( )
Volume 27	Math	( )	Pascal:		
Volume 28	Planetfinder	( )	Volume 300	ATTACH BIOS	( )
Volume 29	Utilities VIII	( )	Volume 301	PIG1:	( )
Volume 30	Games X	( )	Volume 302	PIG2:	( )
Volume 31	Plot Utilities	( )	Volume 303	PIG3:	( )
Volume 32	Games XI	( )	Volume 304	PIG4:	( )
Volume 33	Accounting	( )	Volume 305	PIG5:	( )
Volume 34	Solar Tutor	( )	Volume 306	PIG6:	( )
Volume 35	Garden Management	( )	*Vol. 181 required with these disks.		

TOTAL ORDER = $ ------

Check here if you want these shipped---

NOTE: PLEASE ALLOW 6 - 8 WEEKS FOR MAILING.

NAME ----------

ADDRESS ----------

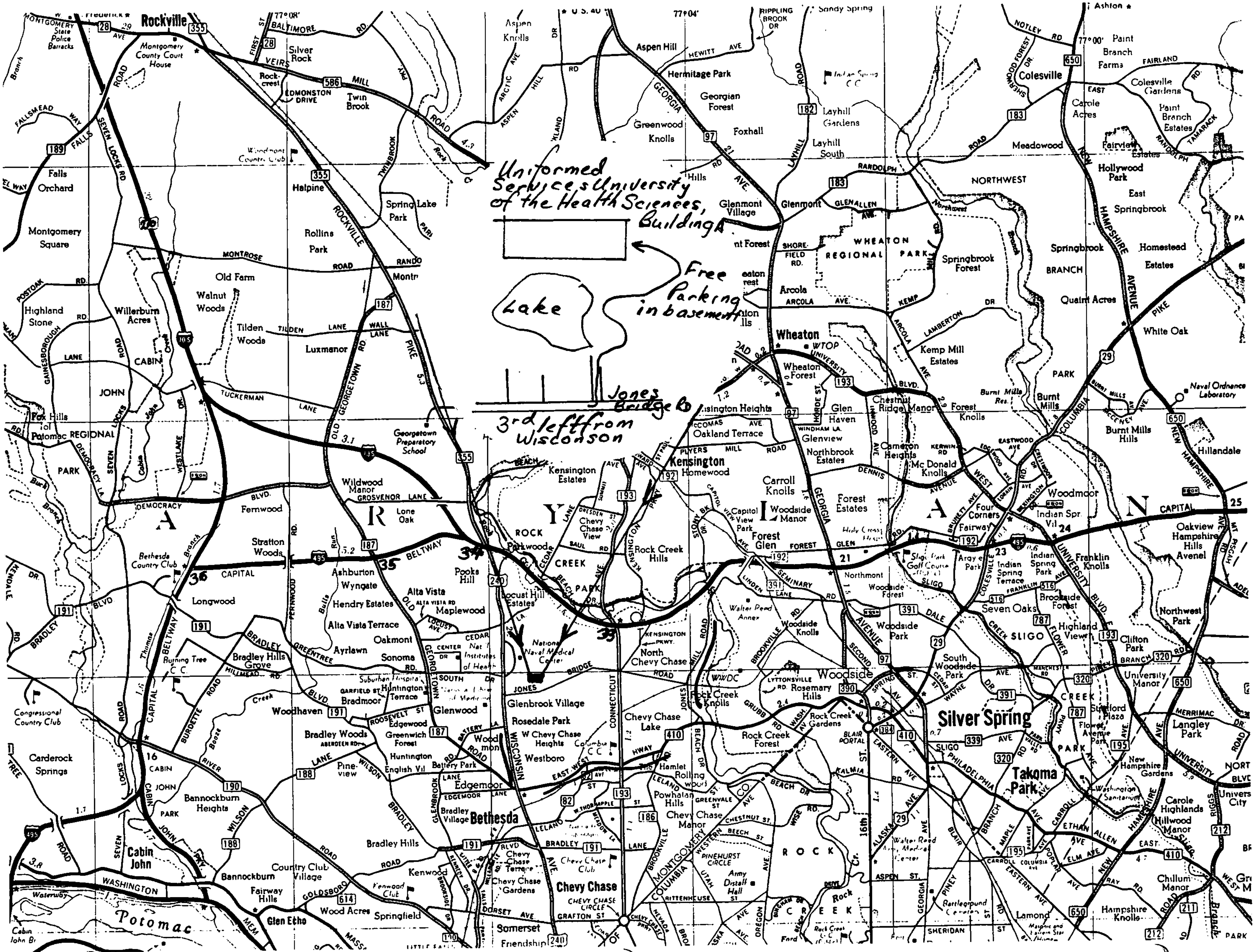
CITY, STATE, ZIP ----------

TELEPHONE ----------

Membership No.(1st three digits after WAP on mailing label) --------

Make checks payable to "Washington Apple Pi"

Send order to: Washington Apple Pi- ATTN: Librarian
PO Box 34511
Bethesda, MD 20817


Uniformed Services University of the Health Sciences, Building A
Free Parking in basement
Lake
Jones Bridge Rd
3rd left from Wisconsin
Rockville
Aspen Hill
Colesville
Wheaton
Kensington
Silver Spring
Takoma Park
Bethesda
Chevy Chase
Cabin John
Glen Echo
Potomac
National Naval Medical Center
Walter Reed Army Medical Center
Naval Ordnance Laboratory
Capital Beltway
Montgomery County Court House
Georgetown Preparatory School
Rock Creek Park
Wisconsin
Connecticut
Georgia Ave.
New Hampshire Avenue
Old Georgetown Rd
Rockville Pike
Seven Locks Rd
Cabin John Pkwy
University Blvd.
Colesville Rd
East West Hwy
Northwest Branch
Wheaton Regional Park
Forest Glen
Woodside
Glenmont
Four Corners
White Oak
Burnt Mills
Hillandale
Langley Park
Chillum Manor
Friendship
Somerset
Kenwood