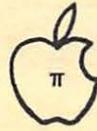


# Washington Apple Pi



Volume 2      November 1980      Number 11

## Highlights

### HARDWARE CORNER

a sound generator for the APPLE II

IDS-460 Impact Printer

Apple FORTRAN

Computerized Garden Planner

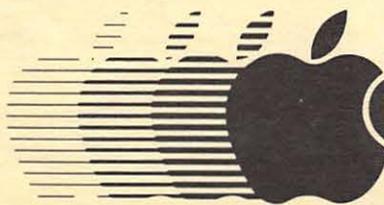
## In This Issue

	Page
MEMBERSHIP INFORMATION, EVENT QUEUE, EDITORIAL	1
MINUTES, SIG-NEWS, CLASSIFIEDS	2
WASHINGTON APPLE DIGEST	3
DOS 3.3 AND THE LANGUAGE SYSTEM - HOWARD LEFKOWITZ	4
NYBBLES	4
QUESTIONS, QUESTIONS, QUESTIONS - MARK L. CROSBY	5
MEMORY MAP IN APPLESOFT - MARK L. CROSBY	6
GREENAPPLES: A REVIEW OF LIBRARY DISK 30 - ERIC AND GREG URBAN	7
HARDWARE CORNER: A SOUND GENERATOR FOR THE APPLE II - SUSAN ZAKAR	8
IDS-460 IMPACT PRINTER: A REVIEW - MARK L. CROSBY	12
PRETTYLISTING - PAUL A. SAND	13
SIG FOR EDUCATION - CHARLES C. PHILIPP	15
APPLE FORTRAN: A REVIEW - PAUL A. SAND	16
COMPUTERIZED GARDEN PLANNER - FRED P. SHARP	19
WISHING WELL CODER - HOWIE MITCHELL	23
AIDS TO FORMATTING REM STATEMENTS - HOWIE MITCHELL	25
THE DISAPPEARING ACT - JOSE M. SANCHEZ	28
INTELLIGENT 80-COLUMN VIDEO INTERFACE (SMARTERM)-HOWARD LEFKOWITZ	28
LETTER TO THE EDITOR - STEPHEN BACH	29
REPLY TO LETTER TO THE EDITOR - SANDY GREENFARB	29
INTERFACE WITH YOUR OWN LITTLE COMPUTER: A REPRINT	30
FLAVORS: LITTLE TIDBITS - BURTON S. CHAMBERS, III	32
TO B** OR NOT TO B** (TESTING THE 6502 COMPARE INSTRUCTIONS) - SANDY GREENFARB	33

# ComputerLand<sup>®</sup>

We know small computers.



 **apple<sup>®</sup> computer**  
Sales and Service

ComputerLand/Tysons Corner  
8411 Old Courthouse Road at Rt. 123  
893-0424

# EDITORIAL

## OFFICERS & STAFF

President -Bernard Urban (301) 229-3458  
Vice President -Rich Wasserstrom (703) 893-9147  
Treasurer -Bob Peck (301) 468-2305  
Secretary -Dana Schwartz (301) 621-2894  
Members-at-Large-Mark Crosby (202) 488-1980  
-Sandy Greenfarb (301) 674-5982  
-Hersch Pilloff (301) 292-3100  
Editor -Bernard Urban (above)  
Associate Editor-Rich Wasserstrom (above)  
-Genevie Urban (301) 229-3458  
Librarian -David Morganstein(301) 972-4263

Washington Apple Pi  
P. O. Box 34511  
Washington, D.C. 20034  
(301) 468-2305

ABBS (301) 983-9317

Membership dues for Washington Apple Pi will be changing for the calendar year beginning January 1981. The Executive Board plans to recommend an increase, but this is subject to approval by the membership. Renewal information will be included in the December Newsletter. It is suggested that new applicants start their membership with the new year. A revised membership form will be included in the December Newsletter. You may also request a membership form by calling our club phone and leaving a message with your name and address or by writing us at the above address.

## 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, NW. (To be sure call the club phone during meeting week.) The November meeting will be on the 22nd, and the program will be given by Steve Milner and Chuck Philipp on educational applications.

Looking ahead, the December meeting will feature a program on the Z80 Card, given by Dave Neumann. Since the 4th Saturday of December falls within the holidays, the date of the meeting will be December 20 (the 3rd Saturday.)

Looking even further ahead, the plans for the January meeting include a program by Tom Woteki (Dr. Wo).

The Executive Board meets on the 2nd Wednesday evening of each month. All members are welcome to attend. Call the President (229-3458) for specifics.

NOVAPPLE meets on the 2nd Saturday of the month at 1:00 PM at Kings Park Library on Burke Lake Road in Fairfax County. The December meeting will be on December 13.

Folks, please think on this. We have reached a point where some hard choices need to be made. I am using the editorial column to convey these matters to our entire membership and I shall be requesting membership approval at our November or December meeting.

1. I have long felt it desirable to have one Apple Users Club for the greater Washington D.C. and Baltimore areas (or affiliated chapters of a Ma's Apple Pi). However, I have come to the conclusion that this has been and continues to be too difficult to accomplish and that such an organization would be faced with too many logistical, political and administrative problems. Therefore, I am proposing that beginning in January Washington Apple Pi would be an Apple Users Group whose members are individuals, families and corporate entities with currently paid up membership dues. These members would be entitled to full benefits, present and future. As such I propose that all other organizations or individuals be treated as non-members for purposes of purchasing the newsletters, software or documentation, or gaining equal access to such services as computer courses or group purchases. This does not preclude possibilities for our co-sponsoring workshops, conferences, or ABBS systems with other groups in the area or elsewhere. Nor does this preclude attendance by non-members at our monthly meetings at GWU. We would cooperate to the fullest extent with the principals of each Users Group to establish equitable exchange procedures and reciprocal agreements.

2. It is my understanding that many of our members currently own modems. I therefore propose that we purchase an APPLE II with appropriate peripherals, including an originate and answer modem, for installation in a current member's home together with a dedicated phone line for use as WAP's official ABBS. Whereas John Moon has assumed this responsibility to date, I propose that he be appointed to run this service as prescribed by rules to be established by the Board.

3. I propose that the Constitution and By-laws be changed to clarify and refine those portions which pertain to certain of the financial dealings of the group, e.g., purchase vs acquisition. Further, I propose that we include a section which prohibits the President of WAP from serving more than two consecutive terms.

4. Pursuant to the consensus of the members at the October 25 meeting, the Board has established a finance and organization committee. This committee will develop recommendations on the appropriate legal form of club organization, revised Constitution and By-laws provisions, and a proposed 1981 budget. This budget will enable careful board review of the dues increase that has been discussed at previous general meetings. All of these matters will be presented to the membership at the November or December meeting for approval or revision.

In addition to the foregoing, I propose that you consider the following items:

- a. A Greenapples SIG for our youngsters.
- b. An official and active WAP hotline to augment the ABBS.
- c. An orientation workshop for newcomers to WAP (to be given to 10 or more new members at a time).
- d. A library of exchange newsletters, magazines, APnotes, information on hardware and software, etc.
- e. Creation of other SIGs as interest becomes apparent, e.g. EDSIG, MEDSIG.
- f. Courses on the APPLE II, both through self-instruction software rented to members for use in the home and at the University of Maryland and elsewhere.
- g. Better software documentation!
- h. The Best of Washington Apple Pi.

I have asked the finance committee to propose budgets on these activities. What do you think? ☺

## MINUTES

### Executive Board Meeting

The Executive Board meeting of October 8, 1980 was held at the home of the President, and was called to order at 7:10 PM. The President announced the formation of a Finance Committee to draw up the 1981 budget and to oversee the Club's financial policies. Hersch Pilloff was directed to draw up a Constitutional Amendment to be presented to the membership which will clarify the Club's financial reporting obligations.

It was decided not to pursue at this meeting the concept of an area-wide integration of the local users groups. Some time was spent in reviewing proposals for the future direction and priorities of the Club.

A motion was passed to allocate \$15 per month for printer services (repairs and maintenance) for the newsletter Editor.

The meeting was adjourned at 10:25 PM.

### Monthly Membership Meeting

The Washington Apple Pi meeting of October 25, 1980 was called to order at 9:35 AM by the President, with approximately 95 members in attendance.

After some discussion of the Club's past and future policies, and after further discussion of the Club's financial practices, a motion was passed referring the matter to the Executive Board. The Treasurer's report was presented, followed by an exchange of ideas on disk drive problems and fixes.

The Pascal SIG announced that their meetings will be held on the third Thursday of each month. The Membership Chairman exhibited samples of the proposed membership directory and requested comments.

The meeting was then turned over to Dr. Lipson of the Apple Education Foundation, associated with the National Science Foundation, for a talk on obtaining grants

for microcomputer research. He may be contacted at:

SEDR W-638

NSF

Washington, DC 20550

Following a question and answer session, the meeting was adjourned to SIG meetings and viewing a video tape of Steve Wozniak and Steve Jobs. Dana J. Schwartz, Secretary

## SIG-NEWS

SIGAMES is a group of people sharing a mutual interest in developing and using games for the APPLE and what they contribute to the learning process. The general meeting of this group is held at a location announced at and following the Washington Apple Pi monthly meeting.

The November meeting will cover two areas. One, organizing a group interested in developing peripherals (e.g. joy sticks, switch closures, or game I/O port expander) for the APPLE II. Two, software product reviews will be given by John Epperson on SARGON; Thomas Steck on North Atlantic Raider; and Jim Eatherly on Bill Budge's Space Games. ☺

## CLASSIFIEDS

FOR SALE: Applesoft Card w/Autostart ROM, \$135. D.C. Hayes Micromodem, \$300. NEED: 80-column board and second disk drive. Call Steve Sondag (703) 281-5392.

FOR SALE: IDS 440G. Bought about Feb 1980; perfect condition; \$750. Sherman Bloom (301) 229-7514 after 6:00PM.

### HELP WANTED:

Apple Pi member to put together documentation/articles/software on cassette-oriented systems.

Organization/member(s) to duplicate disks under contract to WAP.

Volunteer(s) to maintain supply of back issues of WAP and APnotes.

Volunteer(s) to be responsible for delivery of newsletters to computer store(s).

Volunteer(s) to test contributed software prior to publication.

Volunteer to put together hints on running WAP Software on differently configured APPLES.

### \*\*\*REMINDER\*\*\*

With the return of "static electricity" weather, members are reminded to discharge the electricity before touching computer equipment. We have had reports of some damage caused by static electricity.

\*\*\*\*\*

# Washington Apple Digest

---

CREATIVE COMPUTING, September 1980

This issue is almost entirely devoted to the subject of Computers in Education and Computer-Assisted Instruction. While there are a number of articles that could be of interest to Apple Computer users, they are not reviewed in this column because the reviewer has not read them well enough to give a summary and evaluation.

Heapsort, P136. Description of the method and an algorithm to do it.

CREATIVE COMPUTING, October 1980

Music system for the Apple, P26. A review of six systems for creating music. Informative for the uninitiated, and technical enough for the serious comparative shopper.

Tuesday Night Football, P32. A review of this game software. It's a strategic game as opposed to a paddle-controlled screen-action game. The reviewer likes it a lot. \$14 tape or \$18 disk.

A Stellar Trek, P32. Same reviewer, same enthusiasm. High resolution display, full color, and animation. 22 pages of instructions. Price not given.

Easywriter, P34. A review of this word processing package. After telling us his philosophy on high technology and cheap hotel rooms, the author describes some of the useful and some of the clumsy features of the package. \$100.

WP6502, P54. A review of a word processing package for the Ohio Scientific line, but coming out in versions for the Apple, Pet, and Atari. It runs in 8K of RAM, costs \$75, and sacrifices many standard word processing features for simplicity of operation.

Educational software, P56. Part II of a review of software packages for the Apple and others. Biology, math, test creation, grammar, physics.

Actor languages, P61. A symposium on the subject. People who are trying to master the BASIC language should skip this one. I didn't understand any parts of any of the articles, but byte-niks and philosophers might.

Sort methods, P88. Continuation of a series of articles on various sort methods. Linked lists and merge sorting. Technical discussion.

INFOWORLD, August 18, 1980

High Technology sues Apple. High Technology, a former distributor of Apple computer products, has filed a \$70 million breach-of-contract and antitrust suit against Apple Computer, Inc., (alleging) that Apple terminated its distributorship after illegally attempting to restrict its marketing activities to a six-state area. Apple, Inc. was not available for comment on the story.

Apple III on time. Apple III's are being shipped to dealers now, and should be completed by the end of September.

Stock-Market Strategy Machine. Software review, intermediate level discussion. ANAL, by Galaxy, and Stock Market Systems, by RTR Software, Inc. are reviewed in detail, as aids to investors. They are both graphically interesting and they each have a substantial data structure. The reviewer likes each for what it does, but neither does everything one might like. \$100 each.

Word processing software. Elementary review. A one-page check-list of features for five packages is presented: Word Processing System (Programma), Super-Text (Muse), Write-on II (Rainbow), Apple Writer (Apple), and Easy Writer (Information Unlimited). The Pascal editor was also included for reference. A useful survey.

Apple Sues IT&T. Apple Computer, Inc. has filed suit against IT&T Consumer Products (UK) Ltd. claiming that the U.K. subsidiary of IT&T has copied, without permission, Apple's Disk II analog board, controller board, system firmware and software, and the user's manual of the Apple II floppy disk subsystem. IT&T has European rights to distribute some Apple products, but apparently there is disagreement as to just what.

INFOWORLD, September 1, 1980

Paddle-Graphics. Software review, elementary level. In addition to the utilities necessary for loading and saving high-resolution images on the disk, Paddle-Graphics provides four different methods of creating graphics images. The package has programs to plot text in several sizes on the high-resolution screen. Four separate drawing modes permit users to outline images on the screen by manipulating the paddles. Users can fill outlined figures with any of 21 different colors. Finally the package contains tools for defining and manipulating images stored as shape tables. The author seems to like the package, and it is a fair review. Worth noting if you

contd.

need hi-res graphics.

INFOWORLD, September 15, 1980

Bell & Howell to service Apples. Having over 600 service representatives with local parts inventories, the company will do warranty work as normal service on Apple computers.

Elementary Math. Software review of this Muse product. A good package for educators who want to present drill-and-practice material in a more enlightened manner. This program by Muse Software is a precursor of the innovative way that microcomputer technology can be brought into the home and classroom. It is easy to use, hard to improperly bomb, and clever in its use of graphics, color, and sound:

Business BASIC. Intermediate software review. A floating-point BASIC designed for business applications, it has three chief improvements over other BASIC's. The internal data format has a 10-decimal-digit mantissa so you can work with nine- or ten-digit figures, the print formatter (using PRINT USING) can print numbers and strings in fixed fields as are found in most business reports, and I/O commands are part of the language. For the experienced programmer who can deal with the documentation and who wants to write business software. The review is good, but a bit picky at times. I would be even more enthusiastic about the product than the reviewer is.

Data desk. Product announcement. A company is selling a desk that is designed to hold an Apple computer, a monitor, 2 disks (cooled area) and various miscellany. For \$400 it seems expensive, but the concept is useful.

INFO WORLD, September 29, 1980

Apple goes public. Apple, Inc. will become a publicly held corporation in future. A

preliminary prospectus is due in 30 days.

To copy or not to copy. Copying copyrighted programs is not a clearcut issue of good or bad. Most of the time it ought not be done, since it violates a condition of purchase of the software and, more importantly, discourages software developers. On the other hand, software developers and distributors who overburden the user with restrictions and poor documentation are just letting themselves in for trouble. The author's preferred solution is for everybody to act as reasonably as possible, for users to stick to their guns and not permit copying of copyrighted programs, and for manufacturers to eliminate unenforceable contract conditions, provide better documentation and other features that will let the user feel that value was received for value given, and finally to educate the user about the need for honesty and fair dealing.

Super-Text product review. A word processing package that runs on a 48K Apple II using Applesoft. The reviewer says it is good, but because it is a complex processor, learning all its features will take time, and the special functions could cause problems for the non-programmers. The evaluation is good. People who have used dedicated word processors could probably get into Super-Text more easily than could others.

-----  
Editor's Note

Thanks to Dave Efron and Mike Leavitt for these abstracts.

We've had some positive feedback on the usefulness of these abstracts. We would like to expand our coverage of literature. Please call Dave Efron at 251-0225 if you would like to become a contributing editor.

## DOS 3.3 AND THE LANGUAGE SYSTEM

by Howard Lefkowitz

If you have a DOS 3.3 diskette dated August 24, 1980, you have the bug-free version. However, for you Language System owners there is still a change you should make. In the September, 1980 Apple Pi Newsletter a patch to track 0, sector 9, byte \$CC was suggested. Do it, you will like it. The current byte is 81 and you should change it to 10. This will allow you to reboot any 3.3 disk without wiping out your current language which is resident in the language card. This change (I have been assured by CALL-A.P.P.L.E.) does not affect anything else. A PR#6 will reboot any of your DOS 3.3 diskettes and run the resident boot program without the necessity of using your master 3.3 disk and going through two boots. ☺

## NYBBLES

We have received a complimentary copy of PEELINGS II, a new magazine which reviews APPLE software. The issue which we received is VOL 1 NUMBER 3, Sep-Oct 1980. Reviews are presented by category, e.g. games, utilities, simulations/educational, languages, technical, and in this issue a special data base management section. The reviews appear to be well-written and thorough. The magazine is attractively presented. We will have this copy available at the November meeting for your viewing. Subscription rates are \$15.00 per year (6 issues). Back issues are \$4.00 each postpaid. Subscriptions will be mailed first class if started before January 1, 1981. After that they will be mailed second class unless extra postage is paid by the subscriber. The address is:

PEELINGS II  
945 Brook Circle  
Las Cruces, New Mexico 88001

# Questions, Questions, Questions

edited by Mark L. Crosby

This new column will be very reader-dependent. Please pass on any questions and answers that come up in SIC's or from other sources so we can publish them for all to see. Call the club telephone (301) 468-2305 or if you have a modem you may leave questions on the club ABBS (301) 983-9317 addressed to me. In either case, be sure to leave your name, complete telephone number and clearly state your question. Messages will be retrieved Mon-Fri during the day. Answers will either be telephoned to individuals or left on the ABBS depending on the source. Long distance voice calls will be returned collect.

Q. I have Applesoft ROM and I want to be able to clear the HI-RES screen to a color fast! Is there a machine language program available to do this?

A. Applesoft already has this built in - in machine language of course. First execute an HGR or HGR2. Then POKE 28,C and CALL 62454. C can be any number from 0-255. Representative colors are as follows: BLACK=0 or 128, GREEN=42, VIOLET=85, WHITE=127 OR 255, ORANGE=170, BLUE=213. Colors may vary depending on your television or monitor.

Q. After using a "GET" statement in Applesoft, disk commands don't always work. What's wrong?

A. All disk commands must begin by printing the CONTROL-D in column #1 of any particular line. Disk commands are just like other print statements except for the CONTROL-D. The Disk Operating System (D.O.S.) looks for CONTROL-D's only in column #1 (HTAB 1). This is done to save time. Otherwise D.O.S. would have to look at each and every character of every print statement!

**EXAMPLES:**

```
100 PRINT "HELLO";
200 PRINT CHR$(4)"CATALOG"
```

```
100 GET A$
200 PRINT CHR$(4)"CATALOG"
```

In both of the examples above the disk commands will not function. In the first example the CHR\$(4) (a CONTROL-D) is printed in the next position following the HELLO or column #8 (because of the semicolon). In the second example it is printed in column #2 because the GET leaves you "hanging" after column #1 "gets" the string.

The solution to both problems is to insert one line:

```
150 PRINT -or-
150 HTAB 1
```

Q. I need to transfer some binary files from disk to disk but don't know where they start or how long they are. Can you tell me how to do this?

A. Using D.O.S. 3.2 with 48k, first "BLOAD" the file then in Applesoft type:

```
PRINT PEEK(43674) + PEEK(43675) * 256
(CR)
PRINT PEEK(43616) + PEEK(43617) * 256
(CR)
```

The first number is the starting address and the second the length. Save your file using the standard format:

```
BSAVE (NAME),A(STARTING ADDRESS),
L(LENGTH),D(DRIVE #)
```

Q. How can I defeat the stop list at full page for the CATALOG command?

A. In a 48K system with D.O.S. 3.2 you can change D.O.S. by typing:

```
POKE 44601,234
POKE 44602,234
POKE 44603,234
```

Q. I frequently use POKE -16300,0 and POKE -16299,0 to switch from HI-RES page 1 to page 2. When I try to draw or plot, however, the plotting goes to the page that was initialized (HGR OR HGR2). How can I direct the plotting to the other page without using HGR or HGR2?

A. In APPLESOFT, Plotting or drawing is directed to the appropriate page by the following poke:

```
POKE 230,n (ROM) or
POKE 806,n (RAM)
```

Where "n" is 32 for page 1 and 64 for page 2. In this manner you may display one page while plotting on another.

Q. How can I convert Integer programs to APPLESOFT?

A. You will need your disk drive active and D.O.S. booted. Then Load your Integer program and add line # 0 as follows:

```
0 D$ = "(CTRL-D)": PRINT D$;"OPEN
TEXT": PRINT D$;"WRITE TEXT": POKE
33,33: LIST 1,32767: PRINT
D$;"CLOSE": END
```

Then type "RUN" (CR). A text file with the name "TEXT" will be created on your diskette containing a copy of your program.

Now switch to APPLESOFT and type:

```
NEW (CR)
EXEC TEXT (CR)
```

You now have your program in memory in APPLESOFT. you will probably have to go through this version to make syntax changes before it will operate properly - particularly with regard to the handling of strings and arrays.

After changes are made just save the

program as usual. The text file need not be retained except as a backup.

Q. How can I find out how large my APPLESOFT programs are? Also where they start, where the free space is etc.?

A. First, load your program and RUN it until you are certain all strings have been assigned and all arrays dimensioned and filled, etc. Then execute a CONTROL-C to stop or hit RESET on an APPLE II PLUS. You need not run the program if you are interested only in program length.

Then PRINT the following PEEKS:

PROGRAM START = PEEK (103) + PEEK (104) \* 256

PROGRAM END = PEEK (175) + PEEK (176) \* 256

(Subtract START from END for program length)

LOMEM: = PEEK (105) + PEEK (106) \* 256

START OF SIMPLE VARIABLES = PEEK (105) + PEEK (106) \* 256

START OF NUMERIC AND STRING POINTER ARRAYS = PEEK (107) + PEEK (108) \* 256

START OF FREE SPACE = PEEK (109) + PEEK (110) \* 256

START OF STRINGS (Actually the bottom of string space which starts at HIMEM: and works down) = PEEK (111) + PEEK (112) \* 256

HIMEM: = PEEK (115) + PEEK (116) \* 256

Many thanks to Call-A.P.P.L.E. for permitting extracts as space permits. ☺

# MEMORY MAP IN APPLESOFT

by Mark L. Crosby

I often find myself in the middle of a program wanting to know how long the program is, how much variable space has been used, how much free space there is, etc.

While it is possible to just sit there and type PRINT PEEK (103) + PEEK (104) \* 256 for the program start and do that for each of the other memory map segments there is an easier and faster way.

First type in the program exactly as shown in APPLESOFT. The question mark is shorthand for "PRINT" if you prefer it and the CHR\$(34) will ultimately produce a quotation mark. Then save the program on a disk and then run it.

Running it will produce a text file named MEM which, when EXEC'd will print all the information you need about memory usage.

You can be in the middle of a program and break (either a CONTROL-C or a RESET for the APPLE II PLUS) with all variables and strings intact - then EXEC MEM will produce the memory map. If you wish you may then CONTINUE the program execution as though nothing had taken place (except the clearing of the text screen).

I have found this little file invaluable during the debugging process and I hope you find it as useful.

!!LIST

```

0 REM MEMORY MAP BY MARK L. CROSBY
1 REM PRODUCES A TEXT FILE WHEN RUN
2 REM TYPE EXEC MEM TO PRODUCE THE MAP
10 D$ = CHR$(4)
20 PRINT D$"OPEN MEM"
30 PRINT D$"WRITE MEM"
40 PRINT "TEXT:HOME"
50 PRINT "VTAB1:?" CHR$(34)"PROGRAM START" = " CHR$(34)"PEEK(103)+PEEK(104)*256"
60 PRINT "VTAB2:?" CHR$(34)"PROGRAM END" = " CHR$(34)"PEEK(175)+PEEK(176)*256"
70 PRINT "VTAB3:?" CHR$(34)"PROGRAM LENGTH" = " CHR$(34)"(PEEK(175)+PEEK(176)*256)-(PEEK(103)+PEEK(104)*256)"
80 PRINT "VTAB4:?" CHR$(34)"SIMPLE VARIABLES START=" CHR$(34)"PEEK(105)+PEEK(106)*256"
90 PRINT "VTAB5:?" CHR$(34)"SIMPLE VARIABLES SPACE=" CHR$(34)"(PEEK(107)+PEEK(108)*256)-(PEEK(105)+PEEK(106)*256)"
100 PRINT "VTAB6:?" CHR$(34)"ARRAY VARIABLES START =" CHR$(34)"PEEK(107)+PEEK(108)*256"
110 PRINT "VTAB7:?" CHR$(34)"ARRAY VARIABLES END =" CHR$(34)"PEEK(109)+PEEK(110)*256"
120 PRINT "VTAB8:?" CHR$(34)"ARRAY VARIABLES SPACE =" CHR$(34)"(PEEK(109)+PEEK(110)*256)-(PEEK(107)+PEEK(108)*256)"
130 PRINT "VTAB9:?" CHR$(34)"FREE SPACE START =" CHR$(34)"PEEK(109)+PEEK(110)*256"
140 PRINT "VTAB10:?" CHR$(34)"FREE SPACE END =" CHR$(34)"PEEK(111)+PEEK(112)*256"
150 PRINT "VTAB11:?" CHR$(34)"FREE SPACE TOTAL =" CHR$(34)"(PEEK(111)+PEEK(112)*256)-(PEEK(109)+PEEK(110)*256)"
160 PRINT "VTAB12:?" CHR$(34)"BOTTOM OF STRINGS =" CHR$(34)"PEEK(111)+PEEK(112)*256"
170 PRINT "VTAB13:?" CHR$(34)"TOP OF STRINGS =" CHR$(34)"PEEK(115)+PEEK(116)*256"
180 PRINT "VTAB14:?" CHR$(34)"STRING SPACE TOTAL =" CHR$(34)"(PEEK(115)+PEEK(116)*256)-(PEEK(111)+PEEK(112)*256)"
190 PRINT "VTAB15:?" CHR$(34)"HIMEM:" = " CHR$(34)"PEEK(115)+PEEK(116)*256"
200 PRINT D$"CLOSE"

```



# HARDWARE CORNER

## a sound generator for the APPLE II

by Susan Zakar

We are including in this bulletin, a schematic for construction of a sound generator for the APPLE II, along with some comments on the design and the software needed to drive it. The board was designed by Joe Zakar and has been built and is running in his (and mine--his wife) APPLE II at home. The hardware can be completed in an evening or two. Cost runs about \$20 plus the prototype board (also about \$20), for a 3-voice + noise-generator configuration.

Only 5 chips are required for this device! An AY3-8910 Programmable Sound Generator, a quad 2 input NOR-gate, and 3 quad bistable latches. The sound generator chip is a very powerful LSI design chip which can be used to create a great variety of sounds, from race cars to concerts. Music is of fair quality, sufficient for most program needs. All sound is produced under software control, but in contrast to the APPLE II's own capability to produce sound, the AY3-8910 can be set to continue producing a given waveform while the APPLE II continues with another part of the program.

Software controls the tone period for each of 3 tone generators, the noise period for noise, the amplitude for each of the 3 tone channels, and sets the envelope period, shape and cycle. All the programming is done by addressing internal registers and storing appropriate values in them. Each tone period, and the envelope period have a six-bit resolution. A four bit value may be used to effect a coarse tune. If all of this sounds rather complicated, it is. It is a VERY good idea to order the 60-page programming manual for this chip (Cost: about \$3).

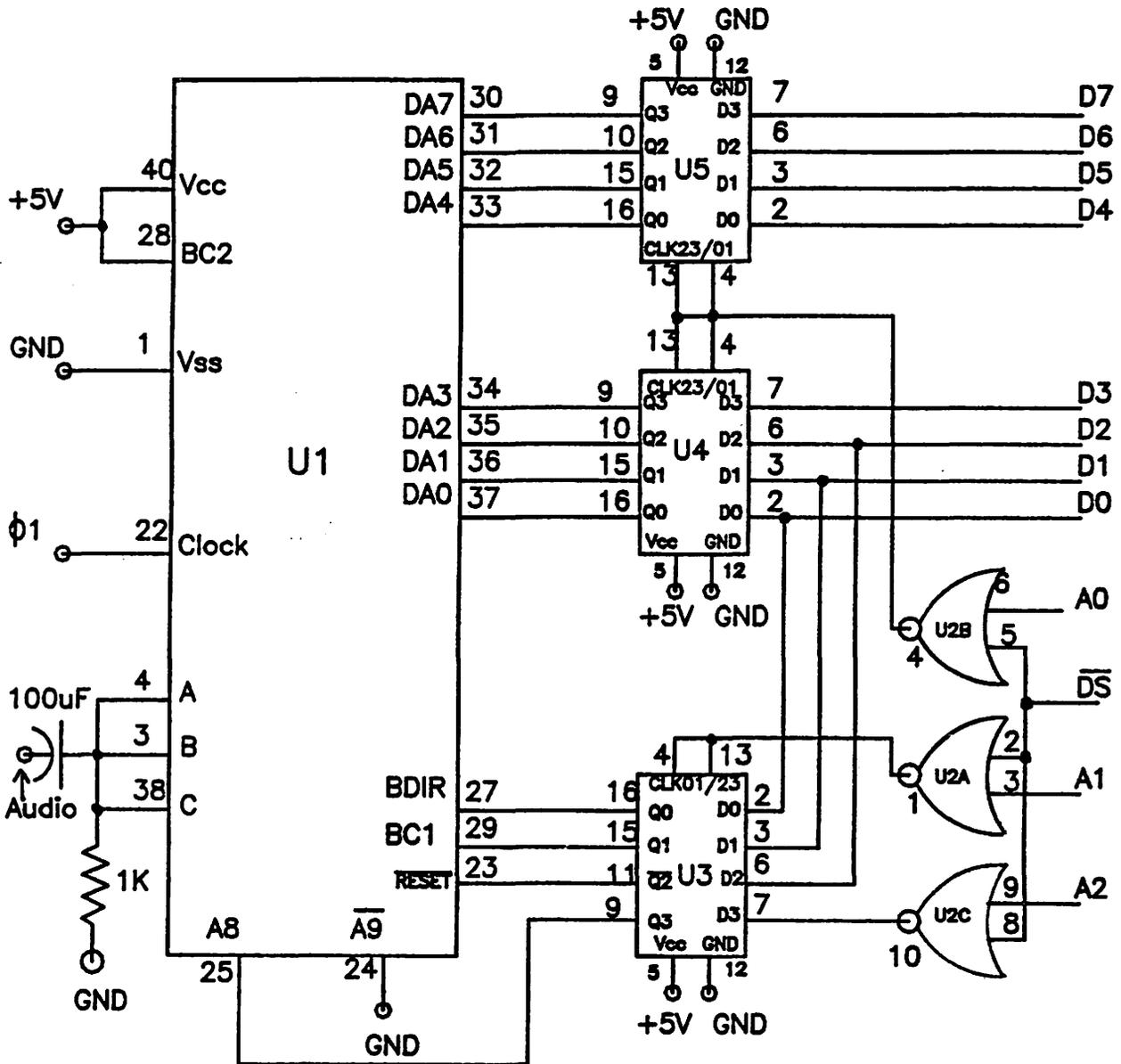
One of the assumptions in the manual is that the chip will be provided with a clock rate of 1.78977Mhz. Since the design here uses the APPLE II's 1.023Mhz clock from Pin 38 of the APPLE II port, values must be recalculated for the examples in the manual. All the necessary formulas are provided, so it is not difficult to do. Outline examples are given in the manual for race cars, sirens, lasers, bombs, etc., along with a chart of musical frequencies for 5 octaves.

Perhaps the most aggravating factor in designing a board for the APPLE II was that the AY3-8910 is a rather slow chip, requiring interface bus signals to be held for 250 to 500 ns. or more. As a result, the APPLE II's own bus signals cannot be used directly, (they are far too fast), and latches must be used to hold the data stable long enough for the sound generator chip to grab it. This sad fact also means extra instructions in the driver routine, but can be handled pretty straightforwardly by the use of the two subroutines provided below.

This design can easily be expanded to accommodate two AY3-8910 chips, thus providing six voices and 2 channels of noise (in stereo, if you want). The output can be directed to an audio amplifier or the input jack of a tape recorder (We even used the mike input on a tape recorder and plugged a speaker into the monitor jack, with fair results). This is a good project for a beginner, since it only has

five chips. It also can be a very educational experiment, teaching a lot about how a device uses internal registers, and how they must be addressed under adverse circumstances, like the incompatible timings requirements, between the chip and the APPLE II.

Hopefully, those readers who do build this board will let the APPLE PI know, so we can exchange software listings and hints with one another.



U1 AY3-8910 PROGRAMMABLE SOUND GENERATOR.  
 U2 74LS02 QUAD 2-INPUT NOR GATE  
 U3-U4-U5 74LS75 QUAD BISTABLE LATCH

:ASM

# 'EXPLOSION'

```

1  *THE FOLLOWING PROGRAM IS AN *
2  *EXAMPLE OF HOW TO USE THE *
3  * AY3-8910 BOARD FOR THE APPLE *
4  * THE PROGRAM IS BASED ON AN *
5  * EXAMPLE CONTAINED IN THE BOOK*
6  * 'AY3-8910/8912 PROGRAMMABLE *
7  * SOUND GENERATOR DATA MANUAL'.*
8  * THE PROGRAM WAS WRITTEN BY *
9  * JOE ZAKAR WITH ADDITIONAL *
10 * COMMENTS BY SUSAN ZAKAR *
11 *****
12 *
13 * THE FOLLOWING IS A GENERAL *
14 * ROUTINE (LA: LATCH ADDRESS & *
15 * WD: WRITE DATA. HC: HOLD *
16 * COMMAND IS USED AS THE TAIL *
17 * TO BOTH ROUTINES. WHEN LA IS*
18 * CALLED, THE CHIP IS SET UP TO*
19 * RECEIVE THE ADDRESS (NUMBER) *
20 * OF ONE OF ITS INTERNAL REGIS-*
21 * TERS. (IN REALITY LA PUTS THE*
22 * REGISTER NUMBER INTO CHIPS U5*
23 * & U6, AND INSTRUCTIONS AT HC *
24 * RAISE AND LOWER THE CONTROL *
25 * LINES ON THE AY3-8910 TO LOAD*
26 * THE SOUND GENERATOR CHIP.) *
27 *****
28 *
29         ORG    $800
30         OBJ    $7000
31     SLOT    EQU    $C0B2
32     SLOT1   EQU    $C0B1
33 *
34 *LA IS ENTERED WITH THE REGISTER
35 *
36 * ADDRESS IN THE ACCUMULATOR.
37 * HOLD REGISTER CONTENTS IN
38 * INTERMEDIATE STORAGE AND SET
39 * 'LATCH ADDRESS COMMAND'
40 *
0800: 8D B2 C0 41     LA        STA    SLOT
0803: A9 03         42             LDA    #$03
0805: D0 05         43             BNE    HC
44 *
45 * HOLD REGISTER CONTENTS IN
46 * INTERMEDIATE STORAGE AND
47 * SET 'WRITE DATA' COMMAND.
48 *
0807: 8D B2 C0 49     WD        STA    SLOT
080A: A9 01         50             LDA    #$01
51 *
52 * HOLD COMMAND IN INTERMEDIATE
53 * STORAGE, THEN LOWER CONTROL
54 * BUS LINES TO 0 TO COMPLETE.
55 *
080C: 8D B1 C0 56     HC        STA    SLOT1
080F: A9 00         57             LDA    #$00
0811: 8D B1 C0 58             STA    SLOT1
0814: 60           59             RTS
60 *

```

contd.

```

61 * HERE IS WHERE WE INITIALIZE THE
62 * SOUND GENERATOR. RESET REMAINS
63 * IN EFFECT UNTIL THE NEXT REGI-
64 * STER ADDRESS COMMAND IS GIVEN.
65 *
0815: A9 04 66 RST LDA ##04
0817: 8D B1 C0 67 STA SLOT1
68 *
69 *ALL REGISTERS RESET
70 *
71 * THIS IS WHERE WE SET UP SOUND.
72 *
081A: A9 06 73 LDA ##06 ;SET NOISE PERIOD
081C: 20 00 08 74 JSR LA ;(REGISTER 6)
081F: A9 00 75 LDA ##00 ;TO MAX VALUE
0821: 20 07 08 76 JSR WD ;FINISH JOB
0824: A9 07 77 LDA ##07 ;SET NOISE ENABLE
0826: 20 00 08 78 JSR LA
0829: A9 07 79 LDA ##07 ;TO ENABLE NOISE
082B: 20 07 08 80 JSR WD ;ON ALL 3 CHANNELS
082E: A9 08 81 LDA ##08 ;SET AMPLITUDE
0830: 20 00 08 82 JSR LA ;RANGE: R9,CHAN A
0833: A9 10 83 LDA ##10 ;TO VARY UNDER
0835: 20 07 08 84 JSR WD ;ENVELOPE CONTROL
0838: A9 09 85 LDA ##09 ;SAME FOR CHAN B
083A: 20 00 08 86 JSR LA
083D: A9 10 87 LDA ##10
083F: 20 07 08 88 JSR WD
0842: A9 0A 89 LDA ##0A ;AND FOR CHAN C
0844: 20 00 08 90 JSR LA
0847: A9 10 91 LDA ##10
0849: 20 07 08 92 JSR WD
084C: A9 0C 93 LDA ##0C ;SET ENVELOPE
084E: 20 00 08 94 JSR LA ;PERIOD: R12
0851: A9 38 95 LDA ##38 ;TO 2.05 SECS
0853: 20 07 08 96 JSR WD
0856: A9 0D 97 LDA ##0D ;SET ENVELOPE
0858: 20 00 08 98 JSR LA ;CONTROL,R13 TO
085B: A9 00 99 LDA ##00 ;DECAY ENVELOPE
085D: 20 07 08 100 JSR WD ;AFTER 1 CYCLE.
0860: 00 101 END BRK

```

--- END ASSEMBLY ---

TOTAL ERRORS: 0

97 BYTES GENERATED THIS ASSEMBLY

6

Classified ads accepted from members  
50 words or less at no charge provided  
the material is obviously non-commercial

Submit your classified at least 30 days  
in advance attention CLASSIFIED ADS, PO  
BOX 34511, Washington, D.C. 20034

# IDS-460 Impact Printer

## a review by Mark L. Crosby

Before I begin with a review of the 460, you should know that I have worked with the IDS-440 for over a year now. The problems experienced with the 440 in a production environment are many. Two print heads have broken their pins, changing spools of inked ribbon and their inked rollers has rendered two pairs of pants permanently smudged with black ink, ribbon tension and paths cause foldover and jams, paper feed is not always positive and the unit is just not designed to operate in 16.5 CPI because tolerances are too loose.

Lest you get the mistaken opinion that I dislike the 440 let me correct you now. For home use the 440 will stand up nicely and I plan to continue using it for certain types of work.

The 460, however, is almost a different kind of "animal" than the 440. Here is an overview of its special features:

### \*\*\* FEATURES \*\*\*

- \* MICROPROCESSOR CONTROLLED
- \* FULL ASCII CHARACTER SET WITH TRUE DESCENDERS
- \* 24 x 9 DOT MATRIX WITH OVERLAP
- \* 10, 12, 16.8 CPI AND ENHANCED
- \* CARTRIDGE RIBBON MOBIUS LOOP WITH BIASED ROUTING
- \* BALLISTIC-TYPE PRINT HEAD
- \* OPTIONAL HI-RES GRAPHICS WITH 84 x 84 DOTS PER INCH
- \* PROGRAMMABLE CONTROL FUNCTIONS:
  - LINE FEED
  - FORM FEED
  - VERTICAL TAB
  - HORIZONTAL TAB
  - CARRIAGE RETURN
  - ENHANCED/NORMAL PRINT
  - GRAPHICS/TEXT
  - CHARACTER DENSITY
  - AUTO-JUSTIFICATION
  - PROPORTIONAL SPACING
  - ALTERNATE CHARACTER SETS
- \* 300, 1200, 2400, 9600 BAUD
- \* AUTO PERFORATION SKIP

and the list goes on from there!

The most notable difference is the print style. Characters are formed under software control with overlapping dots using a 9-wire staggered print head. This produces characters that closely resemble those of a normal typewriter. The dots are visible if you look closely but one is not conscious of them while reading. The new ballistic type print head runs cool even at maximum throughput. Its rugged construction should give it a much longer life. Another departure from the 440 is the print height. To permit true lower case descenders, the height of capital letters has been reduced to .09 inch. While this is very noticeable when printing in the enhanced mode, I quickly got used to the shorter characters. The folks at IDS also dropped the 8.3 CPI mode and

moved tolerances toward the 16.8 CPI mode. This is no loss at all since 8.3 CPI is very difficult to read anyway. The new cartridge ribbon is easy to change and will last longer. It has biased (slanted) routing and is a mobius loop. This permits printing evenly over the entire surface top to bottom and both sides of the ribbon. (Actually, being a mobius it has only one surface!).

There is a new "TOP-OF-FORM" switch that permits setting the form feed top without having to turn off the power. In this mode the paper may be moved either up OR down to the desired point.

If you need proportional spacing (between each character) a single ASCII code will call that feature. Also programmable under software control are line feed increment size, both right and left margins, vertical and horizontal tabs, forms length and perforation skip size, absolute vertical position in lines or inches, and intercharacter spacing. With the appropriate word-processing software or under your program control, the 460 is quite easily capable of printing like the pros do it. Open any textbook and you'll see what I mean. (p.s. Because of intrinsic differences between Easy Writer and the 460's text justification, this paragraph could not be right- and left-justified while printing proportionally).

I have not yet received a software package designed to dump APPLE hi-res graphics to the 460. When I do you will be the first to see the results. Since the inter-dot spacing is much smaller, we should be able to get practically dot-less graphics.

Because the 460 is essentially built around the 440 but seemingly lacking all of the latter's mechanical problems I will not go into any detail on the remaining, standard features of the 460.

Are there any problems? To be fair, yes there seems to be one. The PROM software sometimes permits dropping a single character in a print line. This happens about once every 5000 characters and randomly. There is a cure for this coming from IDS in the form of new PROMS. Be sure you have PROMS labeled RELEASE 5.1 or higher and you should have no problems. The local distributor from which I purchased the 460 will receive and distribute them to purchasers at no charge.

I can heartily recommend this excellent printer to anyone and I congratulate IDS for doing a fine job of correcting past mistakes.

INTEGRAL DATA SYSTEM, INC.  
MILFORD, NH 03035

IDS 460 PAPER TIGER \$1295  
W/GRAPHICS \$1394

# MESA, INC.

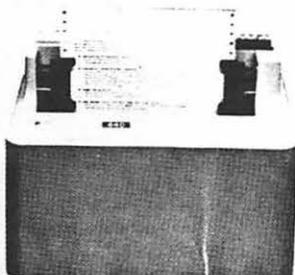
## OFFERS EXCLUSIVE OEM DISCOUNT PRICING TO APPLE PI MEMBERS!

BECAUSE OF YOUR QUALIFIED MEMBERSHIP  
IN APPLE PI, YOU CAN OBTAIN SIGNIFICANT  
DISCOUNTS ON THE PURCHASE OF:

### INTEGRAL DATA SYSTEM'S "PAPER TIGER" PRINTERS

- \* MESA, INC. IS AN AUTHORIZED  
DISTRIBUTOR FOR IDS
- \* SERVICE - AUTHORIZED DEPOT  
SERVICE CENTER - SPARE PARTS &  
FACTORY TRAINED STAFF

\*\*TO PLACE AN ORDER, SIMPLY CALL  
MS. LESLIE HARE AT 301/948-4350  
WITH APPROPRIATE PROOF OF MEMBERSHIP



Integral Data Systems, Inc.

MESA, INC.

16021 INDUSTRIAL DRIVE  
GAITHERSBURG, MD 20760  
(301) 948-4350

# Prettylisting

by Paul A. Sand

A program listing in any language should be "pretty" - it should encourage people to read and understand it. Although prettiness is a subjective quality, there are a number of formatting guidelines which most agree will make a program more readable. I will only be discussing a couple of purely cosmetic techniques here; anyone who is interested in the subject should read "The Elements of Programming Style" by Brian Kernighan and P. J. Plauger, an excellent book covering all aspects of good programming practice.

One way to make a program easier to understand is to use indentation to set aside program statements that form a logical block, such as the bodies of FOR - NEXT loops and statements controlled by an IF - THEN. The indentation should clearly show which statements are to be executed together and the controlling statement for the block. This allows the person reading the program to immediately get a general idea of the control structure of the program by just looking at the listing, rather than painfully tracing the logic step by step. Unfortunately, the standard listing of a Basic program on the Apple doesn't allow such indentation directly.

Another good formatting technique is to use only one statement per program line. Again, this is suggested strictly for readability; a long skinny program is easier to decipher than a short, fat one. Applesoft programmers soon find out, however, that multiple-statement lines are almost irresistible. A program that uses them can take up considerably less memory than one limited to one statement per line. And IF - THEN statements can often be most conveniently written with multiple statements after the THEN.

For these reasons, a Basic program can often resemble a impenetrable mass of code which can discourage even the program's original author from attempting to understand it, let alone anyone else. But this unreadability is due to our reliance on Applesoft's LIST command. LIST simply calls a machine language routine that translates the bytes in memory into a printed source text. There's nothing sacred about LIST.

The program presented here, Prettylist, is an alternative to the LIST command. It does indentation of program blocks as described above and prints only one Basic statement per line. It is written in Applesoft and must be appended to whatever Applesoft program is to be listed. (This can be done either with Apple's Renumber/Append program or by saving Prettylist on disk as a text file and EXECing it into memory.)

Prettylist is executed by the command "RUN 63000". It will list on the current output device all program lines with line numbers less than 63000. It first loads a string array with the Applesoft tokens pulled from ROM. A token is a language element such as "FOR", "DIM", or "HGR2". Instead of being stored character by character, tokens are represented by

a single byte each; a list of Applesoft tokens and their byte representations is given in Appendix F in the Applesoft Reference Manual.

After the array is filled (which takes about 8 seconds) the program lines are scanned and listed. Applesoft programs are stored starting at the location pointed at by addresses \$67 and \$68. For ROM Applesoft, this is usually \$801. Each line has the following format: the first two bytes contain the address of the beginning of the next line in memory. Two zero bytes here indicate the end of the program. The next two bytes contain the line number. The text of the line follows, a series of bytes representing either tokens or ASCII characters. If a byte is less than 128 then it's a character, otherwise it's a token. Finally, a zero byte indicates the line's end. (Obviously, this information could be used to write other useful utility programs.)

Here is the Prettylist routine:

```

63000 nt = 106: dim tk$(nt): ad = 53456: sp$ = chr$(32): for i
= 1 to 5: b5$ = b5$ + sp$: next i: for i = 0 to nt
63010 c = peek(ad): if c < 128 then tk$(i) = tk$(i) + chr$(c):
ad = ad + 1: goto 63010
63020 tk$(i) = tk$(i) + chr$(c - 128): ad = ad + 1: next i: na
= peek(103) + 256 * peek(104)
63030 nl = peek(na) + 256 * peek(na + 1): if nl = 0 then end
63040 ln = peek(na + 2) + 256 * peek(na + 3): if ln >= 63000
then end
63050 i = 4
63060 c = peek(na + i): if c = 0 then ni = 0: gosub 63160: na =
nl: goto 63030
63070 if c < 128 then cd$ = cd$ + chr$(c): goto 63100
63080 if cd$ <> "" and right$(cd$,1) <> sp$ then cd$ = cd$ +
sp$
63090 cd$ = cd$ + tk$(c - 128) + sp$
63100 if c = 34 then iq = not(iq): goto 63150
63110 if (c = 58 and not iq) or c = 196 then gosub 63160: goto
63150
63120 if c = 173 then ni = ni + 1: goto 63150
63130 if c = 129 then nf = nf + 1: goto 63150
63140 if c = 130 then ns = 1: nf = nf - 1
63150 i = i + 1: goto 63060
63160 if ln < 0 then print b5$;: goto 63180
63170 print right$(b5$ + str$(ln),5);: ln = -1
63180 if ns then in = in - 3: ns = 0
63190 for j = 1 to in: print sp$;: next j: in = 1 + 3 * (nf +
ni): print cd$: cd$ = "": return

```

Here is a listing of a program fragment generated by the LIST command:

```

200 DT = 1: FOR K = 1 TO N: IF A(K,K) < > 0 THEN GOTO 250
210 J = K + 1
220 IF J > N THEN DT = 0: GOTO 270
230 IF A(K,J) = 0 THEN J = J + 1: GOTO 220
240 FOR I = K TO N: SV = A(I,J): A(I,J) = A(I,K): A(I,K) = SV:
NEXT I: DT = - DT
250 DT = DT * A(K,K): IF K < N THEN FOR I = K + 1 TO N: FOR J
= K + 1 TO N: A(I,J) = A(I,J) - A(I,K) * A(K,J) / A(K,K): NEXT
J: NEXT I
260 NEXT K
270 RETURN

```

Here is the listing of the same program generated by  
Prettylistter:

```
200 DT = 1:
    FOR K = 1 TO N:
        IF A(K,K) < > 0 THEN
            GOTO 250
210 J = K + 1
220 IF J > N THEN
    DT = 0:
    GOTO 270
230 IF A(K,J) = 0 THEN
    J = J + 1:
    GOTO 220
240 FOR I = K TO N:
    SV = A(I,J):
    A(I,J) = A(I,K):
    A(I,K) = SV:
    NEXT I:
    DT = - DT
250 DT = DT * A(K,K):
    IF K < N THEN
        FOR I = K + 1 TO N:
            FOR J = K + 1 TO N:
                A(I,J) = A(I,J) - A(I,K) * A(K,J) / A(K,K):
            NEXT J:
        NEXT I
260 NEXT K
270 RETURN
```

Prettylistter was inspired by an article in the Dallas user group's newsletter (AppleGram) of June 21, 1980. The article contained a nicely-formatted program listing, but not the formatting program itself. The idea of such a program was so appealing to me that I decided to write my own version. (In the computer field, this practice is widely referred to as "reinventing the wheel.")

☺

---

## SIG For Education

Washington Apple Pi is rapidly becoming a major community resource for helping people learn about computing. Our membership is now well over 300 and includes a growing number of educators. I believe that now is the time to consider forming a special interest group (SIG) for educators and other persons interested in sharing ideas and developing ways to promote learning by using APPLE microcomputers.

For the last year, Washington Apple Pi has focused mostly on the technical aspects of using the APPLE. There have been helpful discussions about how to use hardware and software. Different members have shared their knowledge of programming languages and have helped others analyze and evaluate software. This kind of dialogue ought to continue. In fact, it would be a required resource for the success of a SIG for education.

## by Charles C. Philipp

In addition to the above, Washington Apple Pi has a rapidly expanding library. The library is more than a collection of programs; it is a cost-effective way of sharing information. It is also a required resource for education.

I believe that a SIG for education should place high priority on designing ways to teach young children about computing. Washington Apple Pi has the resources to develop and disseminate models that demonstrate the unique learning opportunities provided by this new technology. The focus should be on teaching children how to use and control microcomputers for various purposes. I believe that the APPLE is ideally suited for this task.

For this SIG to be a success, we need a group of members who have more than a casual interest in this topic and who like to work with children. We also need the commitment of occasional help from our hardware and software experts. If you wish to explore this idea further, contact me at the November meeting of WAP or write to EDSIG, Washington Apple Pi, P.O. Box 34511, Washington, DC 20034.

☺

# Apple FORTRAN a review

by Paul A. Sand

Fortran is one of the oldest and most popular computer languages. It was developed in the 1950's to allow easier programming of scientific and engineering number-crunching programs that computers are so good at, although it has been used in just about every application you could think of. For example, text processors, compilers for other languages, and even an early version of "Adventure" have been written in Fortran. Just about every computer has a Fortran compiler available for it. Microcomputers are no exception - Microsoft has provided a compiler for 8080 and Z-80 based computers for a few years. (I'll be referring to this compiler as "Microsoft Fortran" in the following; note that it is available to Apple users who have Microsoft's Z-80 Softcard.) Now, thanks to the good folks in Cupertino, the Apple is now also programmable in Fortran.

The hardware requirements for Apple Fortran are:

- Apple II or II Plus with 48K RAM
- Apple Disk II
- Apple Language System

The recommended list price for Fortran is \$200.

One disk drive is a minimum for running Fortran. For serious work, two drives are strongly recommended. Otherwise, the user will spend more time removing and inserting diskettes than he will programming.

Apple Fortran essentially adds on the Fortran language to the UCSD Pascal Operating System. Only three different files exist on the diskettes supplied with the package. These are SYSTEM.COMPIILER, SYSTEM.LIBRARY, and FORTLIB.CODE.

SYSTEM.COMPIILER is the Fortran compiler. It accepts as input a Fortran source file and translates it to F-code. For those unfamiliar with that term, F-code is a machine-like code developed at UCSD for a (then) mythical processor. The purpose of generating F-code instead of native machine code is to increase the portability of the UCSD software between computers using different processors. Since the UCSD Pascal compiler also generates F-code, the user can, within limitations, call Fortran subroutines from Pascal programs, and vice-versa. In comparison, Microsoft's compiler generates 8080 machine code.

SYSTEM.COMPIILER is copy-protected - it cannot be copied using the Filer program, the normal method for backing up diskettes. Apple supplies a backup diskette with the system. If one or both copies of the compiler become damaged within the 90-day warranty period, your friendly Level 1 Service Center will provide you with a new diskette. After the warranty expires this service will cost \$15.00.

contd.

SYSTEM.LIBRARY contains routines necessary to generate a runnable program that aren't produced by the compiler. As in Pascal, it includes routines to do high-resolution graphics, calculate transcendental functions, access the game I/O port, and so on. Unlike Pascal, it contains a run-time unit containing routines specific to Fortran programs. This is a sizeable hunk of code, and it must be added on to even the smallest Fortran program. As a result, the runnable code files produced after linking the compiled program to the library are relatively large, especially when compared to the equivalent Pascal program.

FORTLIB.CODE is the program necessary to put independently compiled program units into a Fortran-compatible library file. LIBRARY.CODE, the equivalent program provided with the Pascal system won't work because the formats of Pascal and Fortran libraries are slightly different.

The other utilities necessary to write and run programs are supplied from the Pascal system. The UCSD Editor, Filer, Linker, Assembler, and the other programs operate identically whether you're using Fortran or Pascal.

The documentation included with the package is up to the usual high Apple standard. The manual includes a complete Fortran reference guide, instructions on setting up the system on either a one-disk or multi-disk system, how to mix Fortran, Pascal, and Assembly Language code, how to do overlays - in short, everything needed to allow the user to write, compile, and run Fortran programs. Instead of duplicating the documentation for the part of the operating system shared by Fortran and Pascal, the user is referred to the appropriate section in the Pascal Reference Manual. I wouldn't recommend the manual as a sole reference to the computer novice wishing to learn Fortran; it isn't meant to be a tutorial. But it is complete and well-written, and the user who knows his way around computers will have little or no problems with the manual. Assuming he reads it.

The version of Fortran supported on the Apple is the ANSI Standard Subset of Fortran 77. (ANSI Standard X3.9-1978) "Subset" means that some features of the full language are not supported. Fortran 77 itself, however, is a new version of Fortran superseding the earlier standard. The earlier version, generally known as Fortran IV (ANSI Standard X3.9-1966), is the one offered by Microsoft. Compared to Fortran IV, Fortran 77 has a number of additional features that make the programmer's life easier. (If you aren't acquainted with any Fortran past or present, you may skip over this section.)

First the bad news: double precision is not supported in Apple Fortran. This is regrettable, in my opinion. Although many Fortran users tend to overestimate their need for precision in their calculations and use double precision variables where single precision would be more than adequate, there is a valid need for extended precision calculations in many Fortran applications. Most Fortran implementations (including Microsoft's) have double precision.

contd.

The good news is the extensions to Fortran IV present in Fortran 77 and Apple Fortran. The greatest boon to Fortran programmers is the structured if-then-else construct. Fortran IV programmers have to code an if-then-else using two labels and two GOTO statements:

```
          if (hours.le.40.0) goto 100
          regpay = 40.0 * rate
          ovtpay = 1.5 * rate * (hours - 40.0)
          goto 110
100       regpay = rate * hours
          ovtpay = 0.0
110       continue
```

In Fortran 77, it can be done more naturally:

```
          if (hours.gt.40.0) then
            regpay = 40.0 * rate
            ovtpay = 1.5 * rate * (hours - 40.0)
          else
            regpay = rate * hours
            ovtpay = 0.0
          end if
```

The ELSE IF statement is supplied to enable the choice of exactly one of a number of alternatives:

```
          if (sales.lt.50.00) then
            comm = 0.00
          else if (sales.lt.100.00) then
            comm = 0.02 * sales
          else
            comm = 0.03 * sales
          end if
```

These constructs may be nested to an arbitrary depth.

Fortran 77 also supports a CHARACTER data type, which makes non-numerical data processing much easier than it was with Fortran IV. The declaration statement:

```
character*10 string(5)
```

defines an array of five strings of 10 characters maximum length each. Character data may be assigned values, compared with each other, printed out, and so on, much like Basic.

Do you need Fortran? That depends very much on your previous experience with computer languages and your current applications. In its promotional literature, Apple stresses that it does not recommend Fortran for the beginner. Most smaller applications can be more easily accomplished using Basic, simply because of the ease of using an interactive system for entering, debugging, and running a program. Basic is also by far the easiest language for the beginning programmer to learn. However, as a program increases in size, Basic's disadvantages (lack of good control structures, line number bookkeeping, awkwardness in passing parameters to subroutines, and relative sluggishness to name four.) begin to show up. Both Pascal and Fortran have numerous advantages over Basic. Pascal, however, is usually considered to be a "better"

language than Fortran because of its more flexible control and data structures.

Why then should anyone consider Fortran? If Fortran is one of the languages you already know and you don't have any desire to learn Pascal, then Fortran may be an excellent choice. The primary advantage of Fortran is its wide use in many environments, which makes it the nearest thing there is to a universal computer language. It is relatively easy to write Fortran programs that will run without modification on just about any computer. So the Apple can be used to run many existing Fortran programs developed on other systems, and programs developed on the Apple can be run on other systems.

In summary, Apple Fortran is another high-quality software package from Apple. If you like Fortran (and can live with copy-protected software) then it is an excellent buy. ③

---

# Computerized Garden Planner

by Fred P. Sharp

The program described below will be available on one of Washington Apple Pi's library disks. It is written in Applesoft II for a 32K memory, with printer optional.

## BACKGROUND

Anyone comparing price tags on a packet of seeds and a basket of vegetables can understand why home gardening is becoming the "in" thing. The mild-mannered expert on tomatoes and planting dates is suddenly the center of attention at social gatherings. Conversation nowadays often dwells on such things as zucchini yields and the pH factor. Actually, gardening is much simpler than the experts would like to admit. With a little work and normal luck, anyone can coax an abundance of good eating from a fairly small plot of ground. The advantages of gardening are threefold. One, the vegetables are fresher and taste better. Two, not only are the vegetables cheaper, but one does not have to pay taxes on the value of the vegetables grown. Also, along those lines, one's real income increases by the amount of what you would have spent for vegetables had you not had a garden. Third, the combination of fresh air and exercise makes the home gardener healthier.

## PROBLEM

The 13% annual inflation rate coupled with a 6% annual average salary increase and six growing appetites to satisfy compelled me to make the most of my home garden. With this increased interest in vegetable gardening, and food in general, I realized I would have to establish detailed records of crop varieties, ideal planting dates, harvest dates, depth to plant seeds, distance between plants, distance between rows, expense items, plus a myriad of other details. This all seemed too complicated to even imagine. But, if the home garden was to produce at its peak, it would have to be done.

## SOLUTION

Eureka! this was an example of a marriage made in heaven - my home garden and my APPLE II home computer. The APPLE II could act as a repository of information. It could store, sort, compare, evaluate and report on this information in any format I desired.

## PROGRAM OUTLINE

The Computerized Garden Planner consists of a group of low-overhead programs, each satisfying a specific requirement.

1. Title Page - This program provides a visual animated display depicting gardening on the TV screen while an audio musical representation of "Old McDonald Had a Farm" comes from the APPLE's speaker. (This element can be selected or by-passed depending on one's mood.)
2. Garden Planner - This program provides a means to store and retrieve vital gardening information. The following information is stored in text files for each crop: planting dates, harvest dates, feet of row per person, seeds or plants per 100 feet, distance between rows, and distance between plants. When it's time to plant, the APPLE II provides a listing by planting date of what to plant, how to plant and how much to plant for the family.
3. Garden Accounting - This program provides a means to keep track of twelve different gardening expenses and up to thirty different crops. For each crop the unit of issue, unit value, and total harvest value is recorded in text files. Several reports are provided to include an Expense Listing, a Crop Listing, and a Financial Summary.

GARDEN PLANNER MENUS:

Initial Menu:

YOU HAVE FOUR INITIAL OPTIONS:

1. READ DATA FROM DISK
  2. MAKE INPUT
  3. READ OUTPUT
  4. RETURN TO MENU
- WHAT IS YOUR PLEASURE?

Menu of Reports:

COMPUTERIZED GARDEN PLANNER

- =====
- PRINTER IS OFF
1. CROP LISTING
  2. GARDEN PLANTING INFORMATION
  3. GARDEN HARVEST COMPARISON REPORT
  4. CRT GARDEN PLANTING & HARVEST REPORT
  5. CRT SPRING PLANTING REPORT
  6. CRT HARVEST REPORT
  7. PLANTING REQ. FOR FAMILY SIZE
  8. TURN PRINTER ON
  9. TURN PRINTER OFF
  10. RETURN TO MAIN MENU
- WHAT IS YOUR PLEASURE?

< GARDEN PLANTING REPORT >

CROP NAME	PLANTING DATES		PLANTING DISTANCE BETWEEN ROW	DEPTH TO PLANT	SEEDS REQ.
	SPRING MO DA	FALL MO DA			
BEEET DETROIT	04 01	06 15	15	2	.5 1 OZ
BROCCOLI DECIDO	04 01	05 01	24	12	.5 30 TP
CARROT NANTES	04 01	05 01	15	2	.25 .25 OZ
LETTUCE LEAF	04 01	07 15	12	4	.25 .25 PK
LETTUCE HEAD	04 01	08 01	18	4	.25 12 TP
ONION RED/YEL	04 01	00 00	15	3	1 .48 LB
STRAWBERRIES	04 15	00 00	30	12	5 15 TP
HONEY	04 01	00 00	30	30	2 .04 HI

<=GARDEN HARVEST COMPARISON REPORT=>

CROP NAME	UI	GARDEN REQUIRED	GARDEN HARVEST PROJECTED	DIFFERENCE (+ OR -)
ASPARAGUS	LB	32	LB 16	LB -16
BEAN SNAP	LB	50	LB 50	LB 0
BEAN POLE LIMA	LB	48	LB 18	LB -30
BEEET DETROIT	LB	32	LB 40	LB 8
BROCCOLI DECICO	LB	20	LB 25	LB 5
CANTALOUPE HALE	EA	20	EA 24	EA 4
CARROT NANTES	LB	96	LB 40	LB -56
CORN SWEET	EA	300	EA 225	EA -75
CUCUMBER	LB	32	LB 50	LB 19
EGGPLANT BLACK	EA	24	EA 18	EA -5
LETTUCE LEAF	BU	40	BU 50	BU 10
LETTUCE HEAD	BU	20	BU 12	BU -8
ONION RE/YEL	LB	48	LB 28.8	LB -20
PEPPER CA	LB	19.2	LB 19.2	LB 0
PEAS	LB	24	LB 42	LB 18
POTATOES	LB	120	LB 30	LB -90
RADISH RED/ICE	BU	12	BU 7.2	BU -5
SPINACH BLOOMSD	BU	8	BU 4.8	BU -4
SPINACH NEWSEL	BU	8	BU 4.8	BU -4
SQUASH ACORN	LB	36	LB 45	LB 9
SQUASH COCOZELL	LB	32	LB 30	LB -1
SQUASH SUMMER	LB	12	LB 15	LB 3
STRAWBERRIES	PT	24	PT 90	PT 67
TOMATO BIG HY	LB	80	LB 200	LB 120
TOMATO ROMA	LB	64	LB 48	LB -16
TURNIP	LB	16	LB 12	LB -3
HONEY	PT	2.4	PT .6	PT -2

COMPUTERIZED GARDEN PLANNER

CROP NAME	PLANTING REQ. FT OF ROW	REQUIRED	FOR FAMILY OF SEEDS REQ.	ROW DIS
ASPARAGUS	80	80	CR	30
BEAN SNAP	100	.75	LB	24
BEAN POLE LIMA	80	.6	LB	30
BEEET DETROIT	40	.8	OZ	15
BROCCOLI DECICO	40	24	TP	24
CANTALOUPE HALE	20	.1	OZ	48
CARROT NANTES	120	.6	OZ	15
CORN SWEET	200	20	OZ	30
CUCUMBER	16	.08	OZ	48
EGGPLANT BLACK	8	2.4	TP	30
LETTUCE LEAF	20	.2	PK	12
LETTUCE HEAD	20	20	TP	18
ONION RED/YEL	40	.8	LB	15
PEPPER CA	24	12	TP	30
PEAS	40	.2	LB	18
POTATOES	160	12.8	LB	30
RADISH RED/ICE	20	.2	OZ	12
SPINACH BLOOMSD	20	.2	OZ	12
SPINACH NEWSEL	20	.2	OZ	30
SQUASH ACORN	20	.2	OZ	60
SQUASH COCOZELL	16	.16	OZ	48
SQUASH SUMMER	12	.12	OZ	48
STRAWBERRIES	8	4	TP	30
TOMATO BIG HY	20	4.4	TP	36
TOMATO ROMA	20	4.4	TP	30
TURNIP	16	.08	OZ	18
HONEY	8	.16	HI	30

GARDEN ACCOUNTANT:

Initial Menu:

COMPUTERIZED GARDEN ACCOUNTANT

=====

YOU HAVE FIVE INITIAL OPTIONS:

1. READ DATA FROM DISK
  2. MAKE INPUT
  3. READ OUTPUT
  4. SAVE DATA TO DISK
  5. RETURN TO MENU
- WHAT IS YOUR PLEASURE?

Menu of Reports:

COMPUTERIZED GARDEN ACCOUNTANT

- =====
- OUTPUT MENU
1. CROP LISTING
  2. EXPENSE LISTING
  3. FINANCIAL SUMMARY
- WHAT IS YOUR PLEASURE?

LIST

1. Crop listing:

COMPUTERIZED GARDEN ACCOUNTANT				
CROP NAME	UI	U.\$	QUANT	T.\$
ASPARAGUS	LB	1.00	8	8.00
BEAN GREEN	LB	.80	25	20.00
BEAN POLE LIMA	LB	.80	5	4.00
BEEF DETROIT	LB	.50	20	10.00
BROCCOLI DECICO	LB	.70	14	9.80
CANTALOUPE HALE	EA	1.00	13	13.00
CARROT NANTES	LB	.50	0	0.00
CUCUMBER SPACE	LB	.25	5	1.25
CUCUMBER NAT P	LB	.25	151	37.75
EGGPLANT BLACK	EA	.80	0	0.00
LETTUCE BLACK	BU	.30	8	2.40
LETTUCE BUTTER	BU	.30	8	2.40
LETTUCE RUBY	BU	.30	5	1.50
ONION RED/YEL	LB	.25	22	5.50
PEPPER CA	EA	.15	11	1.65
PEAS DWARF GRAY	LB	1.00	12	12.00
PEAS WONDO	LB	1.00	0	0.00
POTATOES	LB	.30	21	6.30
RADISH RED/ICE	BU	.20	13	2.60
SPINACH BLOOMSD	BU	.50	6	3.00
SQUASH ACORN	EA	.70	44	30.80
SQUASH COCOZELL	LB	.30	67	20.10
RASPBERRY	PT	3.20	14	44.80
STRAWBERRIES	PT	.50	6	3.00
TOMATO BIG HY	LB	.50	2	1.00
TOMATO ROMA	LB	.50	40	20.00
TOMATO VF	LB	.50	88	44.00
CORN	EA	.12	75	9.00
TURNIP	LB	.25	8	2.00
HONEY	QT	3.00	31	93.00
GRAND VALUE				408.85

2. Expense List:

COMPUTERIZED GARDEN ACCOUNTANT	
EXPENSE ITEM	TOTAL COST
SEEDS	7.75
PLANTS	0.00
LIME	5.05
FERTILIZER	7.87
CHEMICALS	0.00
BEE SUPPLIES	30.00
GRAND TOTAL 50.67	

3. Financial Summary:

COMPUTERIZED GARDEN ACCOUNTANT	
FINANCIAL SUMMARY	
VALUE OF VEGETABLES HARVESTED	408.85
TOTAL GARDEN EXPENSES	50.67
NET RETURN ON GARDEN	358.18



```

1 GOSUB 32000
10 W = 150
15 H = 120
20 Q = 100
25 E = 50
97 GOTO 4000
98 FOR I = 1 TO 2
99 GOSUB 1000: GOSUB 10000
100 GOSUB 1000: GOSUB 10000
101 GOSUB 1000: GOSUB 10000
102 GOSUB 1015: GOSUB 10000
103 GOSUB 1010: GOSUB 10000
104 GOSUB 1010: GOSUB 10000
105 GOSUB 2015: GOSUB 10000
106 GOSUB 1025: GOSUB 10000
107 GOSUB 1025: GOSUB 10000
108 GOSUB 1030: GOSUB 10000
109 GOSUB 1030: GOSUB 10000
110 GOSUB 2035: GOSUB 10000
115 FOR N = 1 TO 200
116 NEXT N
118 NEXT I
120 GOSUB 1035: GOSUB 10000
121 GOSUB 1035: GOSUB 10000
122 GOSUB 2035: GOSUB 10000
123 FOR I = 1 TO 200: NEXT I
124 GOSUB 1000: GOSUB 10000
125 GOSUB 1000: GOSUB 10000
126 GOSUB 2000: GOSUB 10000
127 FOR I = 1 TO 400: NEXT I
128 GOSUB 1035: GOSUB 10000
129 GOSUB 1035: GOSUB 10000
130 FOR I = 1 TO 200: NEXT I
131 GOSUB 1035: GOSUB 10000
132 GOSUB 1035: GOSUB 10000
133 FOR I = 1 TO 200: NEXT I
134 GOSUB 1000: GOSUB 10000
135 GOSUB 1000: GOSUB 10000
137 GOSUB 1000: GOSUB 10000
138 GOSUB 1000: GOSUB 10000
139 FOR I = 1 TO 400: NEXT I
140 GOSUB 1000: GOSUB 10000
142 GOSUB 1000: GOSUB 10000
143 GOSUB 1000: GOSUB 10000
144 GOSUB 1015: GOSUB 10000
145 GOSUB 1010: GOSUB 10000
146 GOSUB 1010: GOSUB 10000
147 GOSUB 2015: GOSUB 10000
148 GOSUB 1025: GOSUB 10000
149 GOSUB 1025: GOSUB 10000
150 GOSUB 1030: GOSUB 10000
151 GOSUB 1030: GOSUB 10000
152 GOSUB 2035: GOSUB 10000
199 TEXT : GOTO 3000
200 REM DRAW THE PICTURE
210 GR
215 POKE - 16302,0
220 COLOR= 2
230 FOR I = 0 TO 47
240 HLIN 0,39 AT I
241 NEXT I
242 COLOR= 8
243 FOR I = 38 TO 47
244 HLIN 0,39 AT I
245 NEXT I
249 FOR I = 14 TO 15
250 COLOR= 13
252 FOR I = 35 TO 38
254 VLIN 12,24 AT I
256 NEXT I
258 FOR I = 36 TO 37
260 VLIN 10,37 AT I
262 NEXT I
264 VLIN 36,37 AT 35
266 COLOR= 11
268 FOR I = 5 TO 9
270 HLIN 35,38 AT I
271 NEXT I
    
```

```

274 PLOT 34,7
276 COLOR= 0: PLOT 36,6: PLOT 35
281 COLOR= 13
282 FOR I = 14 TO 16
284 HLIN 32,34 AT I
286 NEXT I
287 COLOR= 0
288 VLIN 13,31 AT 31
290 COLOR= 1
292 HLIN 29,33 AT 32
294 HLIN 29,33 AT 33
296 HLIN 30,32 AT 34
298 HLIN 30,32 AT 35
300 VLIN 36,37 AT 31
302 HLIN 6,18 AT 23: HLIN 7,17 AT
24: HLIN 7,17 AT 25
304 HLIN 8,16 AT 26: HLIN 8,16 AT
27: HLIN 9,15 AT 28
306 HLIN 9,15 AT 29: HLIN 10,14 AT
30
308 COLOR= 0
310 HLIN 6,22 AT 22: PLOT 12,29:
HLIN 11,13 AT 30
312 VLIN 12,31 AT 31
314 HLIN 10,14 AT 31
316 FOR I = 32 TO 33
318 HLIN 9,15 AT I
320 NEXT I
322 FOR I = 34 TO 35
324 HLIN 10,14 AT I
326 NEXT I
328 HLIN 11,13 AT 36: PLOT 12,37

330 COLOR= 1: PLOT 5,6: HLIN 5,2
AT 5: VLIN 5,11 AT 2
332 HLIN 2,5 AT 11: VLIN 11,9 AT
5: HLIN 5,4 AT 9
334 VLIN 6,11 AT 7: VLIN 6,11 AT
10: HLIN 8,9 AT 5: HLIN 8,9 AT
8
336 VLIN 5,11 AT 12: VLIN 5,8 AT
15: HLIN 13,14 AT 5: HLIN 13
,14 AT 8: PLOT 13,9: PLOT 14
,10: PLOT 15,11
338 VLIN 5,11 AT 17: VLIN 6,10 AT
20: HLIN 18,19 AT 5: HLIN 18
,19 AT 11: PLOT 19,6: PLOT 1
9,10
339 VLIN 5,11 AT 22: HLIN 22,25 AT
5: HLIN 23,24 AT 8: HLIN 22,
25 AT 11
340 VLIN 5,11 AT 27: VLIN 5,11 AT
30: VLIN 6,8 AT 28: VLIN 8,1
0 AT 29
349 FOR I = 14 TO 15
350 HLIN 3,29 AT I
351 NEXT I
400 GOTO 98
500 PRINT "RUN GARDEN PLANNER"
600 PRINT "RUN GARDEN ACCOUNTING
"
999 END
1000 P = 97:D = H'
1001 RETURN
1005 P = 103:D = H'
1010 P = 115:D = H'
1011 RETURN
1015 P = 129:D = H'
1016 RETURN
1020 P = 145:D = H'
1021 RETURN
1025 P = 153:D = H'
1026 RETURN
1030 P = 172:D = H'
1031 RETURN
1035 P = 193:D = H'
1036 RETURN
1040 P = 206:D = H'
1135 GOSUB 1000: GOSUB 10000
2000 P = 97:D = W
2001 RETURN

```

```

2005 P = 103:D = W
2006 RETURN
2010 P = 115:D = W
2011 RETURN
2015 P = 129:D = W
2016 RETURN
2020 P = 145:D = W
2021 RETURN
2025 P = 153:D = W
2026 RETURN
2030 P = 172:D = W
2031 RETURN
2035 P = 193:D = W
2036 RETURN
2040 P = 206:D = W
2041 RETURN
3000 REM MENU
3010 HOME
3015 VTAB 4
3020 PRINT TAB( 16)"M E N U"
3025 PRINT TAB( 16)"====="
3030 PRINT : PRINT "1. MUSICAL
ANIMATED INTRODUCTION"
3040 PRINT : PRINT "2. GARDEN P
LANNER"
3050 PRINT : PRINT "3. GARDEN A
CCOUNTING"
3055 PRINT : PRINT "4. TERMINAT
E WORK"
3060 PRINT : PRINT : INPUT "WHIC
H? ";N
3070 IF N < 1 GOTO 3060
3075 IF N > 4 GOTO 3060
3090 ON N GOTO 200,500,600,999
4000 REM TITLE PAGE
4010 HOME
4015 VTAB 8
4020 PRINT : PRINT TAB( 8)"GARD
EN MANAGEMENT SYSTEM"
4030 PRINT : PRINT : PRINT TAB(
18)"BY"
4040 PRINT : PRINT : PRINT TAB(
13)"FRED P SHARP"
4050 FOR I = 1 TO 2500
4060 NEXT I
4070 GOTO 3000
10000 POKE 768,P: POKE 769,D: CALL
770: RETURN
32000 POKE 770,173: POKE 771,48:
POKE 772,192
32010 POKE 773,136: POKE 774,208
: POKE 775,5
32020 POKE 776,206: POKE 777,1: POKE
778,3
32030 POKE 779,240: POKE 780,9: POKE
781,202
32040 POKE 782,208: POKE 783,245
: POKE 784,174
32050 POKE 785,0: POKE 786,3: POKE
787,76
32060 POKE 788,2: POKE 789,3: POKE
790,96: RETURN

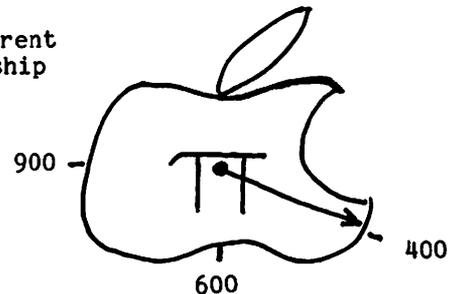
```

]

*Listing  
incomplete - Ed.*

6

Our Current  
Membership



# Wishing Well Coder

by Howie Mitchell

COUNT THE NUMBER OF LETTERS IN YOUR FIRST NAME. IF 6 OR MORE, SUBTRACT 4. IF 5 OR LESS, ADD 3. THIS WILL GIVE YOU A CODE NUMBER. CIRCLE ALL THE LETTERS APPEARING UNDER YOUR CODE NUMBER FOR A MESSAGE FOR YOU !

NOTE: WHEN YOU REACH A PERCENT MARK (%), THAT'S THE END OF YOUR MESSAGE !

(THIS PROGRAM DESIGNED AFTER THE DELIGHTFUL WISHING WELL GAME IN THE RICHMOND TIMES DISPATCH.)

## WISHING WELL

```

3 5 2 8 7 3 5 6 2 7 6 5 5 6 5 7 2 5 7 3
V O G M E I N S I A L E   A M S R A Y R

5 3 6 4 2 3 3 4 7 7 5 5 6 2 5 4 7 2 7 5
N T N T L U O H   T   T D S E E O   L

5 8 3 8 3 7 3 5 8 7 6 2 2 3 2 7 5 6 3 6
L A U N S R   S   U E M A T R N   R E

8 8 3 3 3 2 8 6 6 5 3 6 3 8 7 2 8 5 7 8
M U N   Y R S C A A E N A T   Y   D B

5 6 2 7 6 5 4 8 4 2 2 2 4 3 4 3 8 4 3 7
L N   D O I   E L T O   I R T S   T   W

3 2 4 5 5 4 7 5 7 6 6 4 6 3 3 5 3 7 8 5
S P L E - E N D   T   D T I O L H S Z

2 6 4 7 5 4 2 5 7 2 4 6 7 4 2 5 3 6 8 7
L E L I E E E N L A A S L R S S L T H -

5 3 4 3 3 5 7 7 2 8 5 7 8 2 2 5 6 2 4 5
   N N O W M U E A I C R   P L R A E L

7 8 6 4 5 2 2 6 4 6 8 8 3 4 8 4 4 4 4 5
H P O D   R E Y   E N T M E A N   I R

4 4 7 4 8 5 6 3 2 8 6 4 5 2 8 2 7 5 6 5
S   U D E A   N   N S P T O S P E   A

7 2 3 5 5 2 3 3 3 2 8 2 5 7 2 2 4 5 8 8
U - E T   W N O U I N D I F O W E T   M

4 8 8 7 4 5 5 3 7 6 2 8 4 8 7 7 3 3 8 6
F A N F U   A G I H S   L L N G H - I O

3 2 4 5 8 2 4 2 6 3 3 5 8 8 4 7 2 7 4 6
E   - S K T W O N V I   E   H   T A E

4 2 6 6 7 5 3 3 2 4 7 7 7 7 7 8 5 3 7 3
T P S T O T L   L   R U N   K H O U N
    
```

```

8 4 7 8 3 4 7 7 2 6 3 4 7 7 5 2 8 3 6 3
N U P I E S . % E   E % % E A F D M A

4 8 4 2 3 2 7 2 8 8 4 5 7 4 2 8 7 3 8 4
E T S Y E %   O H   % E T N %

4 2 2 3 3 2 5 5 6 3 7 8 7 4 6 6 3 7 7 5
G H E T O M T R A O % S % R N -   % % U

5 6 4 2 8 4 3 2 3 4 8 6 4 3 5 4 2 7 3 7
T W E S T A M E U T O H   C H B L % H %

5 2 6 2 7 7 2 7 2 2 8 3 3 6 7 7 6 7 7 6
. V E E % % S % . % N   A N % %   % % T

3 2 7 6 6 3 6 7 2 7 4 8 7 7 5 8 6 6 2 3
L % % H E R   % % % I E % % % . F L % E

8 6 8 4 2 2 5 2 6 7 7 8 2 6 3 2 6 7 8 7
% O % G % % % % O % % % % D A %   % % %

6 5 7 4 8 5 7 7 3 2 8 6 6 3 3 2 2 4 7 8
R % %   % % % % D % % E C Y . % % B % %

5 7 6 2 3 5 3 2 8 6 4 8 5 3 2 5 5 3 5 5
% % E % % % % % % D L % % % % % % % %

8 2 8 5 4 3 8 8 3 4 2 3 3 8 3 5 5 5 3 2
% % % % O % % % % C % % % % % % % % %

3 8 3 6 6 2 2 4 8 8 8 8 5 5 8 4 6 8 5 8
% % % E S % % K % % % % % % H   % % %

8 5 5 6 5 8 8 6 4 6 8 5 5 6 6 6 6 4 8 8
% % % T % % % H E E % % %   R O C A % %

6 4 6 4 4 6 8 4 6 6 4 6 6 6 6 6 6 4 4 4
K D   ? % I % % S   % T H E R E . % % %

4 4 4 4 4 4
% % % % % %
    
```

## LIST

```

5 REM *****
  *                               *
  * WISHING WELL CODER.         *
  *                               *
  * *****

10 TEXT : HOME
20 HTAB 2
30 PRINT "*** 'WISHING WELL' MES
   SAGE CODER ***
   -----": PRINT
    
```

```

40 PRINT " THIS PROGRAM WILL AC
   CEPT (AND WILL RE-QUIRE) EXA
   CTLY SEVEN DIFFERENT MESSAGE
   S."
50 PRINT " IT WILL THEN PRINT O
   UT THE MESSAGES INCODED FORM
   , AS IN THE 'WISHING WELL'
   GAME SEEN IN THE RICHMOND
   TIMES DIS- PATCH."
60 PRINT : PRINT " DO YOU WANT
   TO PREPARE MESSAGES FOR PA
   RTICULAR PEOPLE (Y/N) ? "; GET
   ANS$: PRINT ANS$: IF ANS$ <
   ) "Y" THEN PRINT : GOTO 12
   0
70 PRINT
80 FOR SPECIAL = 1 TO 10: PRINT
   "NAME #";SPECIAL;: INPUT " ?
   ";NAME$:L = LEN (NA$):L =
   L - 7 * (L > 7):NAME$(L) = N
   AME$(L) + "," + NAME$: IF NA
   ME$ = "" THEN 100
90 NEXT SPECIAL
100 FOR N = 1 TO 10: IF NA$(N) <
   ) "" THEN NA$(N) = RIGHT$
   (NAME$(N), LEN (NAME$(N)) -
   1)
110 NEXT N
120 FOR N = 1 TO 7: PRINT : PRINT
   "MESSAGE #";N;
130 IF NA$(N) < ) "" THEN PRINT
   ": (";NA$(N);)";
140 PRINT
150 INPUT MESSAGE$(N): IF BIG <
   LEN (MESSAGE$(N)) THEN BIG =
   LEN (MESS$(N))
160 NEXT N
170 PRINT : PRINT " THE MESSAGE
   S I HAVE ARE AS FOLLOWS:" : PRINT
180 FOR N = 1 TO 7: PRINT N;": "
   ;MESS$(N): NEXT
190 PRINT : PRINT " ARE THEY AL
   L O.K. ? ";: GET ANS$: PRINT
   ANS$: PRINT
200 IF ANS$ < ) "N" THEN 220
210 INPUT "TYPE IN MESSAGE #, CO
   MMA, AND CORRECT MESSAGE:
   ";N,MESSAGE$(N): HOME : GOTO
   170
220 HOME
230 PRINT : PRINT " - ONE MOMEN
   T, PLEASE -": PRINT
240 Y = PEEK (37) + 1
250 FOR N = 1 TO 7:L = LEN (MES
   S$(N)): IF (L < BIG AND L >
   0) THEN FOR ADD = L TO BIG -
   1:MESS$(N) = MESS$(N) + "Z":
   NEXT ADD
260 NEXT N

```

```

270 FOR N = 1 TO 7:SUM = SUM + LEN
   (MESSAGE$(N)): NEXT N: REM :
   FIND TOTAL LENGTH OF ALL 7
   MESSAGES.
280 L = SUM
290 IF L = 0 THEN PRINT CHR$ (
   7): PRINT "HECK. THERE AREN'
   T ANY MESSAGES.": PRINT : PRINT
   "(PROGRAM ENDED.)": END
300 FOR N = 1 TO SUM:A = RND (1
   ): NEXT N
310 DIM IN(SUM),OUT(SUM)
320 SELECT = INT ( RND (SUM + 5)
   * 7 + 1): REM : GET A RANDO
   M NUMBER FOR SELECTING A LET
   TER FROM MESSAGE$(SELECT).
330 VTAB Y: PRINT "COUNTDOWN: ";
   SUM;" "
340 IF SUM > 0 THEN GOSUB 550: GOTO
   320
350 PR# 1: PRINT CHR$ (31); CHR$
   (14):A$ = "WISHING WELL": HTAB
   (32 - LEN (A$)) / 2: PRINT
   A$: PRINT CHR$ (15); CHR$ (
   30)
360 FOR N = 1 TO L STEP 20
370 FOR S = 0 TO 19: IF (S + N) <
   (L + 1) THEN PRINT IN(S + N
   );" ";
380 NEXT S: PRINT
390 FOR S = 0 TO 19: IF (S + N) <
   (L + 1) THEN PRINT CHR$ (O
   UT(S + N));" ";
400 NEXT S: PRINT
410 PRINT : NEXT N: PRINT : PRINT
420 PRINT " COUNT THE NUMBER OF
   LETTERS IN YOUR FIRST NAM
   E. IF 6 OR MORE, SUBTRACT 4.
   IF 5 OR LESS, ADD 3. THI
   S WILL GIVE YOUA CODE NUMBER
   . CIRCLE ALL THE LETTERS A
   PPEARING UNDER YOUR CODE NUM
   BER FOR A MESSAGE FOR YOU !
   "
430 PRINT : PRINT "NOTE: WHEN YO
   U REACH A PERCENT MARK (%),T
   HAT'S THE END OF YOUR MESSAG
   E !"
440 IF SPECIAL = 0 THEN 470
450 PRINT : PRINT "MESSAGES FOR:
   ": PRINT : FOR N = 1 TO 7: IF
   NAME$(N) < ) "" THEN PRINT
   NAME$(N)
460 NEXT N
470 B$ = CHR$ (14):L$ = CHR$ (1
   5): REM : CONTROL CHARACTERS
   FOR BIG AND LITTLE LETTERS
   FROM THE PRINTER.

```

contd.

```

480 W$ = B$ + "W" + L$:R$ = B$ +
    "R" + L$:T$ = B$ + "T" + L$:
    D$ = B$ + "D" + L$
490 PRINT : PRINT "(THIS PROGRAM
    DESIGNED AFTER THE"
500 PRINT "DELIGHTFUL ";W$;"ISHI
    NG ";W$;"ELL GAME"
510 PRINT "IN THE ";R$;"ICHMOND
    ";T$;"IMES ";D$;"ISPATCH.)"
520 PR# 0
530 PRINT : PRINT "DO YOU WANT A
    NOTHER COPY ? ";: GET ANS$: PRINT
    : IF ANS$ = "Y" THEN 350
540 END
550 REM *****
    *
    * BUILD UP STRINGS OF NUM-
    * BERS AND CORRESPONDING *
    * MESSAGE LETTERS, RAND- *
    * DOMLY SELECTED. *
    *
    *****

560 COUNT(SELECT) = COUNT(SELECT)
    + 1
570 CODE = SELECT + 3 * (SELECT =
    < 5) - 4 * (SELECT = > 6)
580 IF MID$(MESSAGE$(SEL),COUN
    T(SEL),1) < > "" THEN X = X
    + 1:IN(X) = CODE:OUT(X) = ASC
    (MID$(MESSAGE$(SEL),COUNT
    (SEL),1)):SUM = SUM - 1
590 RETURN
26000 REM *****
    *
    * HOWIE MITCHELL *
    * 408 JACKSON AVE. *
    * LEXINGTON, VA. 24450 *
    * OCTOBER, 1980 *
    *
    *****

26010 REM *****
    *
    * ORIGINALLY PROGRAMMED TO *
    * OPERATE THE EATON LRC *
    * 7000+ PRINTER IN SLOT #1 *
    * (SEE LINE #350). *
    *
    *****

```

## IN INTEGER BASIC=

1. SET THE RIGHT-HAND TEXT WINDOW (POKE 33) TO A VALUE WHICH IS 33 + THE NUMBER OF DIGITS IN THE LINE NUMBER OF THE COMMENT.
2. USE THE AUTO COMMAND.
3. TYPE IN THE COMMENT LINE, ENDING BEFORE THE CURSOR 'DROPS' TO THE NEXT LINE.

### EXAMPLE:

If the comments are, say, on lines #100-#999, then use poke 33,36.

## IN FP BASIC=

Here, the DESCRIPTION is a bit complicated, but the actual DOING of it is easy as apple pie.

1. SET TEXT WINDOW TO POKE 33,28.
2. TYPE IN THE LINE NUMBER, THE WORD 'REM', AND THEN SPACE OVER 1 SPACE USING ESCAPE A.
3. BEGINNING AT THE 'M' IN 'REM', COUNT OVER TO THE LEFT A NUMBER OF SPACES EQUAL TO THE NUMBER OF DIGITS IN THE LINE NUMBER.

THIS WILL GET YOU TO THE LOCATION OF THE LEFT EDGE OF THE REM COMMENT WHEN YOU'VE RETURNED TO THE TEXT MODE (WINDOW NOW AT POKE 33,28).

This will give a rather curious wrap-around appearance as shown in fig. 1 below.

FIG. 1 (at POKE 33,28)

```

J250 REM *****
*****
** HERE IS WHAT AN FP BA
SIC ** REM COMMENT LOOKS LIK
E ** AT POKE 33,28 OR NORM
AL. **
*****
*****

```

FIG. 2 (back to TEXT mode)

```

250 REM *****
*
* HERE IS WHAT AN FP BASIC *
* REM COMMENT LOOKS LIKE *
* AT POKE 33,28 OR NORMAL. *
*
*****

```

# Aids to Formatting REM Statements by Howie Mitchell

This is a short note on the subject of Documentation - not how to Do it, but rather a couple of ideas as to how to make REM comments neatly, efficiently, and in a pleasing manner.

\*\*\* 'PERM-REM' PROGRAM \*\*\*

THIS PROGRAM FORMATS NEAT REM-COMMENT TEXT FILES, WHICH CAN BE EXEC'D INTO YOUR PROGRAMS - QUICK-AS-A-FLASH !

JREM: HERE IS AN EXAMPLE OF THE USE OF THE PERM-REM PROGRAM:

JRUN  
\*\*\* 'PERMANENT' REM-CREDIT WRITER \*\*\*

DO YOU KNOW HOW THIS PROGRAM WORKS ? N

YOU ENTER YOUR NAME, ADDRESS, CITY-STATE-ZIP DATA IN 'DATA' STATEMENTS.

EXAMPLE:

J500 DATA HOWIE MITCHELL,7823 SW. 55TH PLACE,"GAINESVILLE, FLA. 32601"

(NOTE: USE QUOTES TO ENCLOSE DATA CONTAINING , OR : !)

ENTER EXACTLY THREE SETS OF DATA PER CREDIT LINE! -----

USE LINE NUMBERS #500-#600.

(PRESS ANY KEY.)

499 REM \*\*\*\*\*  
\* \* \* \* \*  
\* USE LINES #500-#600 \*  
\* FOR DATA STATEMENTS. \*  
\* ENCLOSE PHRASES CONTAINING : OR , IN " 'S ! \*  
\* \* \* \* \*  
\*\*\*\*\*

601 DATA (END OF DATA.)

ENTER YOUR DATA NOW, THEN RUN THE PROGRAM AGAIN.

J500DATAHILLORY GILLOCK,123 PELEYE AVENUE,"LORTON, KANSAS 22367"

JRUN  
\*\*\* 'PERMANENT' REM-CREDIT WRITER \*\*\*

DO YOU KNOW HOW THIS PROGRAM WORKS ? Y  
I HAVE DATA FOR THE FOLLOWING:

#1: HILLORY GILLOCK  
#2: (END OF DATA.)

WHICH ARE YOU ? 1

O.K., NOW WHAT IS THE DATE ? "SEPTEMBER, 1980"

-AND WHAT LINE NUMBER FOR THE CREDIT LINE ? 5555

5555REM\*\*\*\*\*

\*\*\*  
\*\* HILLORY GILLOCK  
\*\* 123 PELEYE AVENUE  
\*\* LORTON, KANSAS 22367  
\*\* SEPTEMBER, 1980  
\*\*  
\*\*\*\*\*  
\*\*

IS THE ABOVE DATA CORRECT ? Y  
NOW, I WILL MAKE AN EXEC FILE FOR YOU, WHILE YOU WATCH.

OPEN CR:HILLORY GILLOCK:SEP '80  
DELETE CR:HILLORY GILLOCK:SEP '80  
OPEN CR:HILLORY GILLOCK:SEP '80  
WRITE CR:HILLORY GILLOCK:SEP '80  
5555REM\*\*\*\*\*  
\*\* HILLORY GILLOCK  
\*\* 123 PELEYE AVENUE \*\* LOR  
TON, KANSAS 22367 \*\* SEPTEMBER, 1980  
\*\*  
\*\*\*\*\*  
CLOSE CR:HILLORY GILLOCK:SEP '80

SAVE PERM-REM

NOW, ALL YOU NEED TO DO IS LOAD YOUR PROGRAM, AND TYPE THE FOLLOWING:

EXEC CR:HILLORY GILLOCK:SEP '80

RESULT:

THE CREDIT REM LINE WILL BE INSERTED INTO YOUR PROGRAM !

J EXEC CR:HILLORY GILLOCK:SEP '80

J5555REM\*\*\*\*\*  
\*\* HILLORY GILLOCK  
\*\* 123 PELEYE AVENUE \*\* LO  
RTON, KANSAS 22367 \*\* SEPTEMBER, 1980  
0 \*\*  
\*\*\*\*\*  
JLIST

5555 REM \*\*\*\*\*  
\* \* \* \* \*  
\* HILLORY GILLOCK \*  
\* 123 PELEYE AVENUE \*  
\* LORTON, KANSAS 22367 \*  
\* SEPTEMBER, 1980 \*  
\* \* \* \* \*

contd.

```

5 TEXT : HOME
6 Q$ = CHR$ ( ASC ("Q") - 1): REM : Q$=QUOTE MARK.
10 PRINT "*** 'PERMANENT' REM-CREDIT WRITER ***"
15 PRINT : PRINT " DO YOU KNOW HOW THIS PROGRAM WORKS ?": GET
    ANS$
16 PRINT
20 IF ANS$ ( ) = "N" THEN 100
25 PRINT : PRINT " YOU ENTER YOUR NAME, ADDRESS, CITY- ST
    ATE-ZIP DATA IN 'DATA' STATEMENTS."
30 PRINT : PRINT "EXAMPLE:" PRINT "-----": PRINT
35 PRINT "3500 DATA HOWIE MITCHELL,7823 SW. 55TH PLACE,";Q$;
    "GAINESVILLE, FLA. 32601";Q$
37 PRINT : PRINT " (NOTE: USE QUOTES TO ENCLOSE DATA
    CONTAINING , OR : !)":PRINT
38 PRINT " ENTER EXACTLY THREE SETS OF DATA PER CREDIT L
    INE! -----"
40 PRINT : PRINT " USE LINE NUMBERS #500-#600."
42 PRINT : PRINT "(PRESS ANY KEY.)": GET A$: HOME : LIST 49
    ? - 601
45 PRINT : PRINT " ENTER YOUR DATA NOW, THEN
    RUN THE PROGRAM AGAIN.": END
100 HOME : PRINT " I HAVE DATA FOR THE FOLLOWING:" PRINT
103 RESTORE
105 FOR N = 1 TO 10
110 READ NAME$(N)
112 PRINT "#N": NAME$(N)
115 IF NAME$(N) = "(END OF DATA.)" THEN 150
120 READ ST$(N),CI$(N)
130 NEXT
150 PRINT : PRINT " WHICH ARE YOU ?": PRINT : GET L
155 IF L = N THEN 45
157 IF L > N THEN 100
160 NAME$ = NAME$(L):ST$ = ST$(L):CI$ = CI$(L)
165 PRINT : INPUT " O.K., NOW WHAT IS THE DATE ? ";DA$
170 PRINT : INPUT " -AND WHAT LINE NUMBER FOR THE CREDIT L
    INE ? ";LINE$
175 CFILE$ = "CR:" + LEFT$(NAME$(L),18) + ":" + LEFT$(DAY
    TE$,3) + " " + RIGHT$(DAYTE$,2)
180 RIM$ = "REH*****"
185 BLANK$ = " "
187 BOTTM$ = "*****"
190 LFT$ = " "
195 RTE$ = " "

```

```

197 REM *****
    *
    * RIM$ HAS 36 STARS
    * BLANK$ & BOTTM$: 28 CHAR.*
    * LF$ & RT$: 3 & 27, RESP. *
    *
    *****

```

```

200 FIRST$ = LINE$ + LEFT$(RIM$,30 - LEN(LINE$))
205 NAME$ = LFT$ + NAME$ + RIGHT$(RTE$,25 - LEN(NAME$))
210 IF ST$( ) "" THEN ST$ = LFT$ + ST$ + RIGHT$(RTE$,25 -
    LEN(ST$))
220 IF CI$( ) "" THEN CI$ = LFT$ + CI$ + RIGHT$(RTE$,25 -
    LEN(CI$))

```

```

230 IF DA$( ) "" THEN DA$ = LFT$ + DA$ + RIGHT$(RTE$,25 -
    LEN(DA$))
240 CREDIT$ = FIRST$ + BLANK$ + NAME$ + STREET$ + CITY$ + DAY
    TE$ + BLANK$ + BOTTM$
245 HOME
250 POKE 33,28
255 PRINT CREDIT$
260 TEXT : VTAB 15: PRINT " IS THE ABOVE DATA CORRECT ? ": GET

```

```

    ANS$
270 PRINT : IF ANS$ = "N" THEN RUN
280 HOME
290 PRINT " NOW, I WILL MAKE AN EXEC FILE FOR YOU, WHILE Y
    OU WATCH.": PRINT
300 D$ = CHR$(4)
310 PRINT D$"MON O,I,C"
320 PRINT D$"OPEN ";CFILE$
330 PRINT D$"DELETE ";CF$
340 PRINT D$"OPEN ";CFILE$
350 PRINT D$"WRITE ";CFILE$
360 PRINT CREDIT$
370 PRINT D$"CLOSE ";CFILE$
372 PRINT : PRINT D$"SAVE PERM-REM"
380 PRINT : PRINT " NOW, ALL YOU NEED TO DO IS LOAD YOUR
    PROGRAM, AND TYPE THE FOLLOWING:" PRINT
390 PRINT " EXEC ";CFILE$
395 PRINT
400 PRINT " RESULT:" PRINT
410 PRINT " THE CREDIT REN LINE WILL BE INSERTED INTO YO
    UR PROGRAM !"
420 NEW
499 REM *****
    * USE LINES #500-#600 ** FOR DATA STATEMENTS. *
    * ENCLOSE PHRASES CON- ** TAINING : OR , IN "S ! *
    * *****

```

```

601 DATA (END OF DATA.)
25000 REM *****
    * HOWIE MITCHELL ** 7823 SW. 55TH PLACE *
    * GAINESVILLE, FLA. 32601 ** MAY, 1980 *
    * *****

```

Lee Meador's

# VCOPY

- Make backup copies of your VISICALC disk.
- Uses a single disk drive.
- For either 48K Apple II or Apple II Plus.
- NOT ASSOCIATED WITH PERSONAL SOFTWARE OR SOFTWARE ARTS OF WHOM VISICALC IS COPYRIGHTED PROPERTY.
- Price: \$20.00
- Contact: Lee Meador  
Box 3261  
Arlington, TX 76010  
(817) 469-6019

# The Disappearing Act

by Jose M. Sanchez

Quite frequently a user unfamiliar with a particular set of program files will not realize that individual units constitute portions of a large program which need to be chained together in order to function properly. In cases such as these it would be better to hide the names of certain files in a disk's directory. One of the easiest ways of accomplishing this would be to imbed a string of ctrl-H's (backspaces) in the filename. However, APPLE's Monitor prevents us from doing this.

When the APPLE encounters a ctrl-H from the keyboard, it interprets the ASCII character as a direct command and proceeds to 'POP' the last character from the input buffer. However, a ctrl-H will be interpreted by DOS as a normal character and therefore may be used as part of a filename. Normally, DOS marks the end of the filename with a special character and then proceeds to write on the following line when displaying the catalog. If a filename includes enough backspaces, the invisible cursor will be moved to just after the 'end of filename' mark. DOS will then print out any subsequent filename on the following line, thereby overwriting the original file's name.

This then would prove an excellent means of hiding program names except that the Monitor does not allow you to enter the backspace into the program name. The solution to this problem is to concatenate the ctrl-H into a character string which contains the actual program name. To do so from within a program, enter the following line:

```
60000 FOR I=1 TO LEN(A$)+19:A$ = A$+
      CHR$(8):NEXT I:RETURN
(Where A$="programname")
```

By assigning the program's name to the variable A\$ and then executing a GOSUB to line 60000, the resulting string variable will contain backspaces. The statement 'FOR I=1 TO LEN(A\$)+19' produces one backspace for every character in the program's name and then adds nineteen more. These additional backspaces move the cursor to just after the end of the last printed filename.

If you wish to open an 'invisible' file, use the following steps:

```
100 A$="FILE":GOSUB 60000
110 PRINT CHR$(4);"OPEN ";A$
120 PRINT CHR$(4);"WRITE ";A$
130 REM ***** PROGRAM LINES WHICH WRITE
    TO FILE GO HERE
800 PRINT CHR$(4);"CLOSE ";A$
etc.
```

Note that the above will work with DOS commands: RUN, READ, WRITE, OPEN, CLOSE, BRUN, BLOAD, LOAD, RENAME, AND DELETE. There are, however, a couple of restrictions when using this trick. The first restriction is that the filename must not be longer than 10 characters in length, since you are appending at least 19 additional backspaces plus one for each letter in the name. The second point to

remember is to be sure that the filename will never be the last or the first to appear in the catalog, for obvious reasons.

(Editor's Note: With a printer on line the full catalog will be printed including the 'invisible files', and the backspace will be performed by those printers equipped to do so.)

## Intelligent 80-Column Video Interface (SMARTERM™)

by Howard Lefkowitz

SMARTERM is a new interface card from Advanced Logic Systems, Inc. being distributed by Apple, Inc. A partial list of features include:

- o Plugs directly into APPLE slot (#3 if you have Pascal).
- o Standard video output is connected to the card which allows software switching between 80 columns, 24 lines or the standard APPLE II video output. (Both outputs can be used simultaneously with two monitors.) No other connections or alterations are required.
- o Upper and lower case compatible with both Basic's and Pascal (no reprogramming required).
- o Provides three graphics modes:
  1. APPLE II standard lo-res (40x40)
  2. APPLE II standard hi-res (280x192)
  3. SMARTERM medium-res (160x72)
- o Provides a complete set of primitives for plotting points, lines, or any of 64 different graphic characters.
- o Supports all Basic tab commands (TAB, VTAB, HTAB)
- o For Pascal, provides eight software selectable cursor control definitions, GOTOXY absolute cursor addressing, rapid horizontal cursor movement (8 columns) and rapid vertical cursor movement (4 rows).
- o Lots of other features.
- o Package (Order No. D2B0010) includes SMARTERM card, video connector cable.

SMARTERM should be available in late November. Thanks are due to Computerland of Tysons Corner for providing information. This is one I have been waiting for. See you on the big screen.

LETTER TO THE EDITOR

Dear Editor,

First, compliments on the new "shape" of the Washington Apple Pi Newsletter, specifically the cover. A great improvement. Much easier to use. Keep up the good work.

I am writing mainly in response to DOS 3.3 and Sandy Greenfarb's article on it in the October 1980 WAP. Perhaps I should say I am rambling, since that is what these thoughts are.

My initial response to hearing that Apple was releasing another version of DOS was one of great interest and excitement. "What goodies does Apple have in store for us now?" I was eager to obtain it and would have, had it not been for the 'no mail orders' on the WAP group purchase. (Since I live 125 miles southwest of D.C. I cannot attend meetings.) Well, then comes the September 1980 WAP, with the note on Page 1 about bugs in DOS 3.3. Whew! Relief that I didn't purchase DOS 3.3 after all.

On reading Sandy Greenfarb's article I ask myself, "Why buy DOS 3.3?" The only solid reason is that it enables 23% more space on disks. At least, that is the only reason that I see. Sandy says that the File Developer program is almost itself worth the price of DOS 3.3, but from his description of the program I believe that most of the features it offers presently exist in already available public domain - or at least club domain - software. Further, he does not mention the bugs in DOS 3.3!

So I have to say to Apple Computer, "What is it that you are presenting us with?" What are the claims for DOS 3.3? And although it is not uncommon for such software to be released with bugs in it, the fact that it took so little time for the bugs to be discovered forces one to the conclusion that had Apple released it later, thereby giving themselves more time in which to test it, everyone would have been happier. So to Apple Computer I say, "Shame, shame. How can you release such a shabby product? Where is the excellence and dependability brandished in Softalk and in your ads?"

I think we users will get such products if we put up with them. I hope the notes about the bugs in DOS 3.3 in WAP, in CALL-A.P.P.L.E. and elsewhere, will deter people from buying DOS 3.3. We will get good systems software only when we refuse to buy shabby products.

Meanwhile, and sadly, those who have already purchased DOS 3.3 are paying Apple to do its own field testing.

Sincerely,

Stephen Bach

REPLY TO LETTER TO THE EDITOR

Dear Stephen,

To use a phrase of my teenage sons, "I see where you are coming from." In honesty, I cannot argue with your view. FID, the utility I extolled so highly as a DOS 3.3 feature, works equally well under 3.2. I've put what I consider a lot of unnecessary effort into converting to 3.3, including the conversion of some "uncopyable" diskettes. While I still begrudge this essentially non-productive effort of conversion, I still adhere to my pro-conversion attitude.

When Apple went to Pascal, they were forced to come up with the 16-sector format for the general Pascal standard. As some firmware change was required, I feel that the rest was inevitable. What could be more natural than having a system where all diskettes would boot under the same PROMs? In my personal opinion, 3.2 was doomed to obsolescence while Pascal was in the planning stages.

As to the bugs, I am very prejudiced. I have been programming for over twenty years. Not only have I learned to program, but how not to program. Over the years I have learned the myriad traps and pitfalls that lead to occasional error. When I deliver a finished product to a user, I am personally certain that it has been tested in every possible way and that this program or series of programs will work to perfection. However, Murphy's Law generally prevails. My empathy is for Apple's programming staff and no amount of argument could convince me that programming professionals would deliberately allow a product such as 3.3 to be released without equally thorough testing. If anything, comparing product releases of Apple with IBM would show that Apple is doing very well indeed. In fairness to the big guy however, Apple's products are a lot smaller and a lot less complicated. Before I get a law suit, let me add that this example is only for the purpose of showing that noone is perfect.

To reiterate what I said in my article, I feel the major benefit of 3.3 is that it is the current state-of-the-art. I've recently noticed that some software ads request a choice of 3.2 or 3.3 diskette and I have already seen two ads that say 3.3 only. Over a period of time, I would reasonably expect 3.2 products to disappear from the market. DOS 3.2 will go the way of the 78 RPM records.

At the October meeting I noted to the group that there are two releases of 3.3 on the market, one which boots up with an April 1980 date on the screen and one which boots with an August 1980 date. I used the April release for two months with no problem. Upon learning of the August version, I went to my dealer who allowed me to update my master. I would expect that all dealers will do the same.

In order to get a quick reply to meet the WAP newsletter deadline, your letter was read to me over the phone. I hope that I have covered all your points.

Sandy Greenfarb



Editor's Note: The following article is reprinted from the July 1980 issue of PLANNING, a journal of the American Planning Association.

# Interface with your own little computer

By Jeffrey K. Finn

It was during a late night work session a little over two years ago that I realized that personal computers would change the way we approach the planning process. There on my living room television screen was a color map of Seattle's 121 census tracts. The map had just been

generated by my personal computer. The colors represented the average selling price of the homes in each tract. I looked at the pattern for a while and then typed out a new set of breakpoints for the data. In seconds, my computer had a new display for me.

Since that night, I have become convinced that personal computers will make it possible for one- and two-person planning offices to develop alternative solutions to even the most complex planning problems. By using standard computer programs, modified to reflect a particular situation, planners can take advantage of the world's best thinking and the most advanced analytical techniques.

To give you an idea of its power, I've listed the uses I have made of my personal computer. (See box.) The last use—making a remote terminal map display of census tract data—is possible because anyone who wants Seattle census data and has an 80-column terminal can call my personal computer via regular telephone lines and generate an 80-column map with that information. Other uses are just now becoming feasible, as new accessories are developed. These accessories include digitizer tablets, light pens, and electrical circuit controllers (which can be programmed up to a year in advance). Other possibilities are automatic timekeeping and two-way spoken communications.

## How much?

A computer system that can do most of this costs less than \$5,000. The computer on which I do my census tract mapping cost me about \$1,200, because I use my own color television set. For this low price, one receives a computer more powerful than those that cost \$1.5 million

The author at work.



Terry Good

about 15 years ago. Although the current generation of microcomputers does not have the capacity of the large time-sharers, tiny personal computers now on the drawing boards will give users the capacity to deal with all but the largest data bases and the most complex scientific problems. However, it may not be economically feasible for a small office to own or develop the programs and data bases needed for all its jobs.

It's likely that small planning offices will take a hybrid approach, combining in-house capability with time-sharing on a large computer system—an approach known in the computer field as distributed data processing. The typical small planning office will complete many of the day-to-day activities on its own computer. When a specialized data base or data analysis is needed, the office computer will be connected by telephone to a larger computer that has the special capabilities required for that portion of the job. The information generated in this way will be displayed on the firm's own computer in a manner reflecting the special character and approach of that firm.

#### Free advice

If what I've presented so far has aroused your interest, let me caution you not to rush out to buy your own computer. It's not yet an easy task, since microcomputer systems and programs have not yet been geared specifically to the unique needs of planners. But, if you're eager to get started, here are some pointers to guide you along the uncharted path of a new technology:

- As with any project, determine the goals and objectives you expect to accomplish by using the computer. Given the relatively low price of personal computers, a valid objective—particularly for those with no previous computer experience—would be simply to learn about them and have fun playing around with them. I know of several professional offices that have acquired personal computers for the sole purpose of having their employees become familiar with the general principles of data processing.

## What the hardware can do

### Geographic-based data display

- Demographic data by census tract
- County assessor's property tax valuation data by half-acre tracts
- Land uses by half-acre tracts
- Satellite land coverage classifications by 1.2-acre tracts

### Planning data analysis

- Real estate economic feasibility analysis
- Financing costs
- Interactive statistical analysis—frequencies, histograms, correlation, means, linear regression, multiple regression, data transformations

- Critical path method (CPM)
- Animated times series, variable relationship graphs
- Perspective study drawings

### Office operations

- Word processing
- Accounting
- Mailing lists, labels
- Resource directories

### Telecommunications

- Data bank retrieval
- Electronic community bulletin board
- Remote terminal map display of census tract data

- Spend some time reviewing the various brands and models available. Buying a personal computer system is a little like buying components for a hi-fi system. All the parts must work together. Unfortu-

nately, there are fewer interface standards for computers than for hi-fi equipment. Make sure the computer you buy has an RS 232C serial interface, which—used with special phone equipment—allows your

## ... with the right software

At the heart of any microcomputer system are the programs—the sets of instructions that tell the computer what to do. The right programs are so important that computer experts now recommend finding the programs first and then looking for the computer systems on which they can operate.

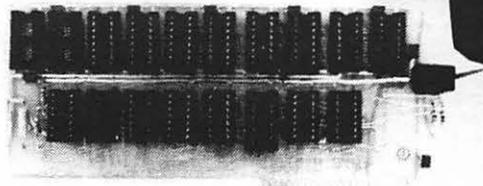
While there are hundreds of companies selling general-purpose business and statistical programs, very little is advertised in the way of planning-oriented graphic display programs. The shortage of these valuable planning tools is due to: 1) the relatively small market for them; 2) the fact that even general-purpose map display programs require some custom programming to make them applicable to a specific geographic area; and 3) the difficulty of modifying mapping programs designed for large

computers to run on the micros. However, there are professionals who are now in the process of making useful planning programs available commercially. Also, some of the general-purpose programs are easily applied to planning office work.

Be cautious when you buy an existing program. If possible, operate it before you buy it. A program written with the user in mind will feel comfortable to you. Finally, look for understandable and complete instructions on its use.

A good source of more information on the general subject of microcomputers is *An Introduction to Personal and Business Computing* by Rodnay Zaks. (Sybex, Inc., 2020 Milvia St., Berkeley, CA 94704; 1978; \$6.95). It is available in many of the retail stores that sell microcomputers.

**Now Available  
for the Apple II**



# 16 K Ram Expansion Board

-  expand your 48 K Apple to 64 K of programmable memory
-  allows you to run Apple's new Fortran package and other languages
-  plugs in slot 0
-  eliminates the need for an Applesoft™ or Integer Basic ROM card
-  switch selection of RAM or mother board ROM language
-  includes instruction manual

introductory offer **\$195<sup>00</sup>** Includes Shipping

VISA



Allow 4-6 Weeks  
Apple II and Applesoft  
are trademarks of  
Apple Computer, Inc.

A product of   
Andromeda  
Computer Systems.

P.O. Box 94, Hauppauge, N.Y. 11787

(516) 360-0988

FLAVORS: little tidbits

by BURTON S. CHAMBERS, III

(The flavors chosen for each tidbit are not necessarily an indication of content.)

LEMON: Apple Fortran

Elsewhere in this issue, Paul Sand briefly introduces Apple Fortran to Washington Apple Pi members. I have used it and recommend it to the membership, although if your choice is between Fortran and Pascal, I recommend the latter. Unfortunately, a serious compiler error exists in the current version of Apple Fortran. DATA statements are supposed to be non-executable statements, which provide a useful way of initializing variables. The error in the current version is that the variables are "initialized" each time the subroutine, containing the DATA statement, is invoked. Alas, this substantially reduces the utility of DATA statements. Furthermore, more care must be exercised when modifying existing Fortran routines that run on other machines. Apple Computer, Inc. has been informed of this, and I'm sure we will be hearing from them in the near future.

GRAPE: Datacomm by D. C. Hayes:

Eleven months ago, I heard that the good folks that bring you the Micromodem II for the Apple II were developing a set of routines, written in Pascal, that would allow quite powerful applications for those fortunate enough to have the Apple Language System. This set of routines is now available (reference Paul Sand), and I believe comes with the source. I have used these routines, and recommend them to those with Pascal and the Micromodem II.

APPLE: Pascal File Names:

Almost one year ago, I decided to begin the task of learning Pascal. Now I am glad I did, however there were times when I doubted I would ever become comfortable with the language. During this past year I have learned some things that Apple hasn't bothered to pass along about file names. I am suspicious that they either want to keep us out of trouble, or that they realize it is somewhat of a game people play, i.e. figuring out how it works. In any event, try the following file names (You can make them easily in the FILER with the Make command. The name "SOMETHING" can be replaced with whatever you like):

- SOMETHING.BACK
- SOMETHING.CODE
- SOMETHING.DATA
- SOMETHING.FOTO
- SOMETHING.GRAF
- SOMETHING.INFO
- SOMETHING.TEXT

I have included the familiar ones as well.

PEAR: Apple Pascal Directories

If you want to know how to access the directory on Pascal diskettes, buy Datacomm (if it indeed is offered with the source code).

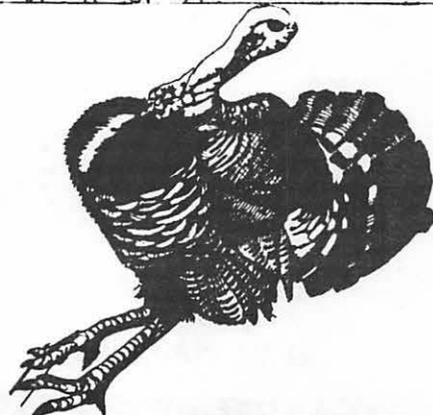
BANANA: Usins Serial Card with Datacomm

You can use a printer hanging off of an Apple Serial card IF you can drive your printer fast enough. The serial is too slow to allow the Datacomm program to both receive characters at 300 baud with your Micromodem II and keep the printer fed. However if your printer can accept characters at 9600 baud, it will work. If one of the members needs to know how much one could back off of this requirement and still work, I would be willing to experiment.

ORANGE: The following is a smaller version of Bruce Field's neat Double Graphics for the Apple silentspe. It is small enough to be loaded in without interferring with Apple Plot, for those of you that use Apple Plot. A portion of Bruce's coding has been made a subroutine. Also I took out the check on the left margin, so BEWARE!!! (Please review Bruce's fine article in the October edition of WAF.)

0300-	A2	0D	A9	FF	20	A8	FC	CA
0308-	D0	FA	AD	FF	CF	AD	00	C1
0310-	20	02	CD	A0	00	84	2A	A9
0318-	BF	85	2B	20	62	03	29	0F
0320-	AA	BD	94	03	20	56	03	F0
0328-	05	C6	2B	4C	1B	03	20	8B
0330-	03	A9	BF	85	2B	20	62	03
0338-	29	78	4A	4A	4A	AA	BD	A4
0340-	03	20	56	03	F0	05	C6	2B
0348-	4C	35	03	20	8B	03	A4	2A
0350-	C8	C0	28	90	C0	60	8D	2B
0358-	CF	20	0B	CB	20	0B	CB	A5
0360-	2B	60	A5	2B	A4	2A	48	29
0368-	C0	85	1A	4A	4A	05	1A	85
0370-	1A	68	85	1F	0A	0A	0A	26
0378-	1B	0A	26	1B	0A	66	1A	A5
0380-	1B	29	1F	0D	13	CF	85	1B
0388-	E1	1A	60	A9	04	20	AB	CC
0390-	20	02	CD	60	00	60	18	78
0398-	06	66	1E	7E	01	61	19	79
03A0-	07	67	1F	7F	00	40	30	70
03A8-	0C	4C	3C	7C	03	43	33	73
03B0-	0F	4F	3F	7F				

\*



TO B\*\* OR NOT TO B\*\* (TESTING THE 6502 COMPARE INSTRUCTIONS)

Sandy Greenfarb

The 6502 provides three compare instructions: CMP (compare Accumulator and memory), CPX (compare Index X and memory) and CPY (compare Index Y and memory). In technical terms, a compare operation tests the result of subtracting the operand from the specified register and sets the negative, zero, and carry flags accordingly. Assuming that the novice machine language programmer has not reached the stage of understanding the mechanics of the operation, this article attempts to explain, in simple terms, how to evaluate the results of a compare.

In order to compare two values on the 6502, one value must be in the specified register (A, X or Y) and the other must be in memory or immediate mode (part of the compare instruction). The results of a compare (with some explanation to follow) are summarized in the following table:

	Unsigned Values		Signed Values	
	<u>C</u>	<u>Z</u>	<u>Z</u>	<u>N</u>
Register < Memory	Reset	Reset	Reset	Set
Register = Memory	Set	Set	Set	Reset
Register > Memory	Set	Reset	Reset	Reset

Keep in mind that the above chart is merely a guide as to how to interpret the results of a compare, and not an explanation as to how the flags are set.

Unsigned values are simply those actual values as represented by one byte (0-255). Signed values are those in which the high order bit is considered to be a sign bit. These range from -128 to +127. In general, when testing unsigned values, use the C and Z tests; when testing signed values, use the Z and N tests. For these tests, 6502 provides six instructions:

- BCC - Branch on Carry Clear (Carry reset)
- BCS - Branch on Carry Set
- BEQ - Branch on Result Zero (Zero set)
- BNE - Branch on Result Not Zero (Zero reset)
- BMI - Branch on Result Minus (Negative set)
- BPL - Branch on Result Plus (Negative reset)

This article is simply to get a new programmer "over the hump". For more detail, read a 6502 text or software manual or read Randall Hyde's article "Converting Integer Basic Programs to Assembly Language" in APPLE ORCHARD No. 1.

PERSONAL COMPUTING SEMINAR - RECORD KEEPING, PROBLEM SOLVING & INCOME TAX

WHERE: Talbott Shopping Center(1000 block) Rockville Pike, Rockville, MD

WHEN: 9 AM - 4:30 PM, Friday, Dec.19 (or) Monday, December 22, 1980

AGENDA:

COMPUTERS

- Applications and users
- Sizes & user impact
- Costs and key factors
- Hardware, firmware & software

APPLICATIONS (with

- demonstrations)
- Financial Planning
- Records/data mgmt
- Text Processing
- Accounting & control
- Income Tax

PROGRAMMING

- Purchase costs and types
- Programming languages
- 'How to' & trade offs

'HANDS ON' experience

COST: \$25 including lunch and handouts. For maximum participation each seminar is limited to 18 people.

TO REGISTER: Mail the following form, with with payment to:

Charles H. Reinbrecht, Jr., 8400 Fox Run, Potomac, MD 20854

Please enroll me in the PERSONAL COMPUTING SEMINAR for (check one):  Friday, December 19, 1980  Monday, December 22, 1980

Name: \_\_\_\_\_ Telephone: \_\_\_\_\_

Address: \_\_\_\_\_ ZIP: \_\_\_\_\_

If date requested is full, I will be contacted. Payment will be refunded if no alternate is suitable, or if cancellation is received by 12/12/80.

Signed: \_\_\_\_\_ Date: \_\_\_\_\_

-----  
 WASHINGTON APPLE PI  
 MAIL ORDER FORM  
 -----

Washington Apple Pi now 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. They will come in a protective foam diskette mailer.

PROGRAM DISKETTES

Members: \$5.00 picked up at meeting  
 \$7.00 mailed to you...

Non-members: \$8.00 per disk picked up at meeting  
 \$10.00 mailed to you...

Volume 1	Utilities I	( )	Volume 22	Utilities VI	( )
Volume 2	Utilities II	( )	Volume 23	Games VIII	( )
Volume 3	Games I	( )	Volume 24	Games IX	( )
Volume 4	Games II	( )	Volume 25	Utilities	( )
Volume 5	Games III	( )	Volume 26	Stocks/Investments	( )
Volume 6	Games IV	( )	Volume 27	Math	( )
Volume 7	Games V	( )	Volume 28	Planetfinder	( )
Volume 8	Utilities III	( )	Volume 29	Utilities	( )
Volume 9	Educational I	( )	Volume 30	Games	( )
Volume 10	Math/Science	( )	Volume 31	Plot Utilities	( )
Volume 11	Graphics I	( )	Volume 180	Dungeon Designer	( )
Volume 12	Games VI	( )	Volume 181	Beginner's Cave	( )
Volume 13	Games	( )	*Volume 182	Lair of Minotaur	( )
Volume 14	IAC Utilities IV	( )	*Volume 183	Cave of the Mind	( )
Volume 15	Games VII	( )	*Volume 184	Zyphur Riverventure	( )
Volume 16	Utilities V	( )	*Volume 185	Castle of Doom	( )
Volume 17	Graphics II	( )	*Volume 186	Death Star	( )
Volume 18	Educational II	( )	*Volume 187	Devil's Tomb	( )
Volume 19	Communications	( )	*Vol. 181	reUired with these	( )
Volume 20	Music	( )	disks.		
Volume 21	Apple Orchard	( )			

-----  
 TOTAL ORDER = \$ -----

Check here if you want these shipped---

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  
 Washington, DC 20034

## **DID YOU KNOW THERE IS AN APPLE STORE IN WHEATON MARYLAND? COMPUTER CRAFTERS OF WHEATON**

- APPLE Computers are our specialty**
- Most desirable items always in stock**
- We represent over 40 manufacturers of Apple and Apple related products.**
- Large Assortment of Books and Magazines**
- Friendly, Knowledgeable Atmosphere**
- Family Run Business**

*Stop in and say hello to Gary, Ken or Betty*

<b>M-W</b>	<b>11:00 to 6:00 PM</b>
<b>TH-F</b>	<b>11:00 to 9:00 PM</b>
<b>Sat</b>	<b>11:00 to 6:00 PM</b>

Washington Apple Pi  
P.O. Box 34511  
Washington, D.C. 20034



First Class

