

\$1.50

Washington Apple Pi



Volume 2 October 1980 Number 10

Special **Highlights** Harvest Issue

BLAISE AWAY!

COMMUTER

**DOUBLE-SIZE GRAPHICS
FOR THE SILENTYPE**

In This Issue

	Page
EVENT QUEUE	1
CLASSIFIEDS	1
EDITORIAL	1
MINUTES	1
SIGNEWS	2
ABBS	2
NOVAPPLE	3
SOUTHEASTERN SOFTWARE	3
DOS 3.3 (FIRST IMPRESSIONS) - SANDY GREENFARB	6
HOW TO CONVERT FROM DOS 3.3 TO 3.2 - SANDY GREENFARB	7
PROPOSED MEMBERSHIP DIRECTORY - TOM JONES	7
TUESDAY NIGHT FOOTBALL: A REVIEW - BILL SCHULTHEIS	8
AIR TRAFFIC CONTROLLER: A REVIEW - BILL SCHULTHEIS	9
A REVIEW OF LIBRARY DISK VOLUME 9: EDUCATIONAL I - LES STUBBS	10
THE PEOPLE I SAW AT THE COMPUTER SHOW - GENEVIE URBAN	11
SAVING AN ADVENTURE TO DISK AND MUCH MORE - MICHAEL THOMAS	12
BLAISE AWAY! FOOLING THE COMPILER OR HOW TO KNOW WHERE YOU'RE POINTING - DR. WO	13
BUGS AND WORMS	20
COMMUTER - DANA SCHWARTZ	20
THE PROGRAMMA WORD PROCESSING SYSTEM: A REVIEW BY A NEW USER - SUE ZAKAR	25
SPACED FILECABINET II - DENNIS J. REEDER	25
TINY LETTER WRITING PROGRAM - HOWIE MITCHELL	26
TAKE THAT, WORD PROCESSOR: A REPRINT	30
DOUBLE-SIZE GRAPHICS FOR THE SILENTYPE - BRUCE F. FIELD	31

ComputerLand®

We know small computers.



 **apple® computer**
Sales and Service

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

OFFICERS

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) ~~985-1298~~ 983-9317

Membership dues for Washington Apple Pi are \$12.00 per calendar year. If you are interested in joining our club, call our number and leave your name and address. An application form will be mailed to you. Or if you prefer, write us at the above PO Box.

EVENT QUEUE

Washington Apple Pi meets on the 4th Saturday of each month at 9:30 AM at George Washington University, Building C, on G at 23rd Street, NW. The October meeting will be on the 25th. Call the club telephone during the week of the meeting for the agenda or any changes.

The Executive Board meets on the 2nd Wednesday evening of each month. Members are welcome to attend. Call the President at 229-3458 if you plan to come.

NOVAPPLE meets on the 2nd Wednesday at 7:30 PM at Computers Plus in Franconia, and on the 4th Thursday at 7:30 PM at Computerland of Tysons Corner. See the NOVAPPLE minutes for details of The October meetings.

CLASSIFIEDS

FOR SALE: Anadex DP-8000-AP line printer with Apple parallel interface card. All cables and manuals included. Was bought brand new in Feb. 1980. Hardware valued at \$1040, asking \$900. Brian Dormer (301) 831-8799 after 5:00PM.

ADVERTISERS PLEASE NOTE: Washington Apple Pi now has over 350 members. Our newsletter circulation is approximately 800. Advertising rates are: Full Page, \$30; Half Page, \$15; Quarter Page, \$10; Eighth page, \$6. If you wish to advertise, please send your ad copy in black and white (no halftones) by the first of the month to the Editor, Washington Apple Pi, PO Box 34511, Washington, DC 20034.

EDITORIAL

I'll keep this short. The Mid-Atlantic Computer Show was lots of fun. Let's hope for bigger and better next time. Our booth was a great success. I suspect we may gain as many as 100 new members as a result of our efforts. Thanks again, gang, for helping me person the booth. A particular thanks to Mark Crosby for showing up often and early, and for that great DEMO!!!

The video cassettes of Wozniak and Jobs proved to be popular and will be available for your viewing at our next meeting after the session planned by Sue Zakar.

We thought that since we are now into the fall season we should have a harvest issue. Enjoy. Keep the contributions, letters and articles coming!

MINUTES

EXECUTIVE BOARD MEETING

The Executive Board meeting of September 17, 1980 was held at the home of the President and was called to order at 7:30PM.

A report was submitted from the Membership Chairman with a design for a membership card (at a cost of 3 cents per member), and a proposal for the format of the Membership Directory. Both will be published in the newsletter before implementation.

In other business Al Weiner of the New Users SIG asked for volunteer speakers. It was decided that the Secretary would collect and catalog the mass of printed material being received by the club. The President displayed copies of "WHO WE ARE", a descriptive handout about the club to be distributed at the Mid-Atlantic Computer Show on September 18-21.

A motion was passed to continue group purchases from an established list of sources, to be approved by the Board. Several members have volunteered to help coordinate these purchases.

The remainder of the meeting was spent with representatives of other nearby users' groups discussing proposals for mutually beneficial consolidation of resources.

GENERAL MONTHLY MEETING

The Washington Apple Pi meeting of September 17, 1980 was called to order at 9:30AM by the President. The membership was informed of the proposal currently being considered by the Executive Board for the merger of the local APPLE clubs. A lengthy discussion of the issue followed, providing much useful input for future Board discussions.

A motion was again passed directing the Treasurer to provide a verbal report at each monthly meeting. The new coordinators of group purchases, Howard Lefkowitz and Ira Cotton, were introduced to the membership and they asked for specific items to acquire under this program. Joe Lipson of the Apple Education Foundation informed the club of

contd.

his group's purpose. Two University of Maryland students requested assistance in a survey of personal computer owners. Peter Hirshberg of the Washington Children's Museum gave a slide presentation of their plans for a Communications exhibit and requested help from the membership.

The meeting was adjourned to special interest group sessions at 10:45AM.
Dana Schwartz, Secretary

SIG-NEWS

The special interest group on games, SIGAMES, will hold its meeting at a location announced at and immediately following the Washington Apple Pi monthly meeting. Tom Lucas will discuss "Writing Interactive Literature". Also to be discussed are new ideas for future projects.
(Al, Gass Chairman)

The New Users SIG meets on the third Saturday of every month at 11:00 AM at Twinbrook Library (corner of Twinbrook Parkway and Viers Mill Road in Rockville). Call 279-1980 for directions.

Minutes of New Users SIG Meeting

The New Users Group met at Twinbrook Library on August 16, 1980.

Topics discussed at the meeting included modems. Although acoustic couplers are less expensive, the micromodems (DC Hayes, direct connection modem) can also provide automatic dialing and answering, and thus in contrast to an acoustic coupler is independent of your presence. Either should be capable of accessing the Club's proposed Bulletin Board.

Programmer's Aid (a chip available for about \$50) can provide music, do memory check, append, renumber, etc. This is provided in Integer Basic. Tool-kit (a package of utilities from Apple) for about \$65 will provide considerable versatility for Applesoft.

One of the utility disks from the Club allows for checking free space available on a disk at any time. Another nice utility provides each item in the catalog with a letter of the alphabet. One then need only type in the designated letter to load or run the program desired.

There does not seem to be a substitute for the word PRINT in Integer Basic as there is in Applesoft, which uses the ? as a typing shortcut for the word PRINT.

The commercially available programs, Supertext and Supercheckbook, are used and liked by some new members.

The members present expressed interest in having the programs on purchased library disks explained since most are not documented. Bob Peck offered and suggested that other club members might offer to share their knowledge on how to use some of the particular disks at future meetings. A demonstration of printers was also suggested as a future topic. Bob Peck announced that Apple Pi is considering several proposals for conducting mini-courses in Basic, Machine Language, PASCAL, etc.

The members adjourned to Al Weiner's home to view his system. He demonstrated use of his micromodem and various utility disks plus his adventure game.

(Sara L. LaVilla)

ASMSIG (Assembly Language Special Interest Group) gathered after Apple Pi's regular September meeting to welcome another eight courageous folk to the growing throng of assembly language users.

The news at our gathering was meager only in quantity - not in importance. A new version (792?) of TED II+ is available from: Apple Puget Sound Program Library Exchange, 304 Main Ave., South, Suite 300, Renton, WA 98055. The cost is less than \$20 and the documentation is superlative!

Assembly language access to DOS commands was discussed, with examples and code, in the first issue of SOFTALK. Many (but not all) members received this issue free. Can someone bring copies to the next meeting?

Stimulated by an idea from the NEWSIG group, we propose to implement a Question and Answer system somewhat differently. Members with assembly language questions are encouraged to submit these in writing. We'll auction these at our regular group meetings, and hopefully most will be answered.

(Ever wonder what really happens as a result of CMP, CPY and CPX? Sandy Greenfarb found the answer. Look for it in the newsletter!)

We found it difficult to discuss our common interests in the post-meeting environment. We all have interests in other SIG's, joint purchases and new disk volumes for sale. We consequently agreed (almost unanimously) to meet regularly

:on the 3rd Wednesday evening of the month - 7:30 Oct. 22.

:at the University of Maryland, Computer and Space Sciences Bldg. (room 3340)

Our October gathering promises at least two features

:RWTS Peeled, and a discussion of the difference between DOS 3.3 and 3.2 - Sandy Greenfarb

:TED II for the APPLE II PLUS: relocating the Sweet Sixteen - Mike Hartman

ASMSIG members will receive a reminder, map and a new membership roster. But all are welcome and encouraged to join our investigations into this arcane corner of the APPLE computer.

(Jim Rose)

ABBS 

The Washington Apple Pi ABBS is available to members. Use a Bell 103 compatible modem (i.e., D.C. HAYES, NOVATION, etc.), full duplex, ASCII. A password is required and can be obtained by sending a postcard with your name, address, phone no. and WAP membership no. to: WAP ABBS, 9800 River Road, Potomac, MD 20854. If you had a password assigned previously, it is still valid.

NOVAPPLE

Minutes for September 10, 1980

The meeting was opened at 7:30 PM by the President. It was announced that anyone not paying dues by the last meeting in September would be dropped from the rolls and would no longer receive the Apple Pi. The next order of business was the nomination of new officers for 1981. These officers will begin their duties in November. The following persons were nominated at this meeting:

Secretary-----No Nominations
Vice President--Steve Plusch
President-----Nick Cirillo
-----Theron Fuller

No further nominations were made at this meeting.

The program was presented by Mark Spahn. He presented the second part of a two-part program on hi-res graphics. Before he could get into the graphics portion of his talk a discussion took place on the differences between memory locations for Applesoft and Integer. Everyone participated in the discussion and learned a great deal about the internal memory. Mark then showed how simple graphics could be generated and demonstrated a program which used an equation to generate a spiral type picture.

Minutes for September 25, 1980

The meeting was opened at 7:42 PM by the President. The first order of business was to reopen the nominations for the officers. In addition to the officer positions for which persons were nominated at the previous meeting an additional office of Treasurer was opened for nomination. Nick Cirillo volunteered to remain as Treasurer if he was not elected President, a post for which he was nominated previously. Everyone present agreed to change the previous nomination from President to Treasurer. Although requests were made for additional nominations none were forthcoming. A motion was made to change the present check signature system from a three signature system to a one signature system. The motion was seconded but modified during the discussion to require the Treasurer to look into the matter including by-laws and make a recommendation back to the members. A discussion was held on member responsibility which promoted further discussion on the purpose of the club and the need to examine where we are going in the near future. Some of the suggestions are listed below:

1. Members should wear a name tag so we can identify each other at meetings.
2. A list of members should be generated which gives enough information to let members know who has similar interest but does not have enough data to help criminals, should it fall into the wrong hands.
3. A software library should be set up for NOVAPPLE members. Mike Thomas and

Robert Steele volunteered to be keepers of the library.

4. It was the concensus of the people present that the club has grown large enough so that we need to look at a new meeting place and examine the desirability of changing the meeting day as well. A straw poll indicated that about half of the members would like to change the meeting to Saturday if a suitable place can be found. A committee will look into what can be done.

5. The members feel that a question and answer session should take place for 15 minutes at each meeting to answer questions of both new and old members as they arise.

Due to the length of the discussion the Adventurefest which was scheduled was canceled. It will be scheduled at a later date.

At the next meeting which will be at Computers Plus on October 8 we will have a discussion and demonstration of the Z-80 card by Phil Eastman. On October 23, at Computerland of Tysons, we will have a session devoted to Word Processors. If you are interested in either of these topics please come out and join us. Don't forget we will begin the election of your new officers as well. Be sure to bring your nominations or volunteer your services. It is the only way our club can remain strong.

Gerald Eskelund, Secretary

Southeastern Software
6414 Derbyshire Drive
New Orleans, LA 70126

504/246-8438 504/246-7937
September 16, 1980

Bernard Urban
Washington Apple Pie
P.O. Box 34511
Washington, DC 20034

Dear Mr. Urban,

I enjoy reading your publication. Here are 2 routines you might be interested in running for your members.

The first is for those without a disk utility program to modify 3.3 DOS to prevent zeroing of \$E000. This is needed only by Language Card owners and will allow booting of 3.3 without the need to reload either Integer or Applesoft II into the card.

NOTE

The modified DOS should only be used for booting after Integer or Applesoft II has been loaded into the card. The user should also have made no changes to any area of the Language Card RAM.

The program was written with the S-C Assembler II and constructed in a way to be easily understood by the novice. Each step is explained and no indirect indexing is used into the IOB table. A shorter method would be to write the overwritten byte back to the Language Card after a second boot. That isn't as much fun.

The second program is for the Z-80 Softcard owner. No method was provided for returning to DOS or Pascoal without turning off the Apple. This program will do a cold boot. It was written by Dave Hughes the day he received his Softcard. Dave also is the author of LCMOD for Pascal.

If the user types BOOT and changes his mind he can hit RESET and remain in CP/M. Pressing any key will boot DOS or Pascal.

Hope you find the programs of use.

Regards,



George McClelland

```

1000 * SOUTHEASTERN SOFTWARE
1010 *
1020 *   3.3 DOS ONLY !!!!!!!!!!!!!!!
1030 *   48 K LANGUAGE SYSTEM ONLY
1040 *
1050 * NO ERROR CHECKING ON READ OR WRITE !!!!!!!!!!!!!!!
1060 *
1070 *** *** *** *** *** *** *** ***
1080 *
1090 * PROGRAM TO MODIFY 3.3 DOS
1100 * TO PREVENT ZEROING OF $E000
1110 * SO THAT AS II OR IB WILL NOT
1120 * HAVE TO BE BLOADED WHEN BOOTING
1130 * WITH A MODIFIED DISKETTE
1140 *
1150 *** *** *** *** *** *** *** ***
1160 *
1170 RWTS      .EQ $3D9
1180 VOL       .EQ $B7EB
1190 TRACK    .EQ $B7EC
1200 SECTOR   .EQ $B7ED
1210 ERR      .EQ $B7F5
1220 BUFFLO   .EQ $B7F0
1230 IOCMD    .EQ $B7F4
1240 BUFBYT   .EQ $96D3      THIS IS FIRST BYTE IN BUFFER TO BE CHANGED TO A NOP
1250 *
1260 *
1270          .OR $300
1280 *
1290 * SET UP BUFFER AT $9600
1300 *
0300- A9 00 1310          LDA #0          LO BYTE OF BUFFER
0302- 8D F0 B7 1320          STA BUFFLO
0305- A9 96 1330          LDA #$96          HI BYTE OF BUFFER
0307- 8D F1 B7 1340          STA BUFFLO+1
1350 *
1360 * SET VOL. # TO WILD CARD (ZERO)
1370 *
030A- A9 00 1380          LDA #0
030C- 8D EB B7 1390          STA VOL
1400 *
1410 * CLEAR ERROR FLAG
1420 *
030F- A9 00 1430          LDA #0
0311- 8D F5 B7 1440          STA ERR
1450 *
1460 * SELECT TRACK, SECTOR - IN THIS CASE TRACK 0, SECTOR 9
1470 *
0314- A9 00 1480          LDA #0          TRACK ZERO
0316- 8D EC B7 1490          STA TRACK
0319- A9 09 1500          LDA #9          SECTOR 9
031B- 8D ED B7 1510          STA SECTOR
1520 *
1530 * SET IO COMMAND TO DO A READ
1540 *
031E- A9 01 1550          LDA #1          READ COMMAND
0320- 8D F4 B7 1560          STA IOCMD
1570 *
1580 * EXECUTE THE READ
1590 *

```

```

0323- A9 B7 1600 LDA #B7 HI BYTE OF IOB TABLE
0325- A0 E8 1610 LDY #E8 LO BYTE OF IOB TABLE
0327- 20 D9 03 1620 JSR RWTS DO IT
1630 *
1640 * MODIFY THE BYTES IN THE BUFFER TO NOP'S
1650 *
032A- A9 EA 1660 LDA #EA NOP
032C- 8D D3 96 1670 STA BUFBYT
032F- 8D D4 96 1680 STA BUFBYT+1
0332- 8D D5 96 1690 STA BUFBYT+2
1700 *
1710 * SET IO COMMAND TO EXECUTE A WRITE
1720 *
0335- A9 02 1730 LDA #2 WRITE COMMAND
0337- 8D F4 B7 1740 STA IOCMD
1750 *
1760 * WRITE THE BUFFER BACK TO THE DISK WITH THE CHANGES TO DOS TRACK 0, SECTOR 9
1770 *
033A- A9 B7 1780 LDA #B7 HI BYTE OF IOB TABLE
033C- A0 E8 1790 LDY #E8 LO BYTE OF IOB TABLE
033E- 20 D9 03 1800 JSR RWTS DO IT
0341- 60 1810 RTS DONE

```

SYMBOL TABLE

```

RWTS 03D9 VOL B7EB TRACK B7EC
SECTOR B7ED ERR B7F5 BUFFLO B7F0
IOCMD B7F4 BUFBYT 96D3

```

```

1: ;
2: ;BOOT PROGRAM
3: ;
4: ;written by:
5: ; DAVID HUGHES
6: ;
7: ;
8: a$vec equ 0f3d0h 27: ;
9: z$cpu equ 0f3deh 28: ;
10: boot equ 0faa6h 29: ; Start of main program
11: bdos equ 0005h 30: ;
12: cr equ 0dh 31: main call home ; a message all Mothers appreciate
13: lf equ 0ah 32: lxi d,msg
14: org 100h 33: mvi c,9
15: ; 34: call bdos ; Print message
16: ; 35: mvi c,1
17: jmp main 36: call bdos ; Get Console character
18: ; HOME routine 37: lxi h,boot
19: ; 38: shld a$vec
20: home mvi e,1bh 39: lhld z$cpu
21: mvi c,2 40: mov m,a ; Jump to 6502
22: call bdos 41: msg db cr,lf,lf,lf,lf,' Southeastern Software Cold Boot'
23: mvi e,2ah 42: db cr,lf,lf,'Insert a Pascal, BASICS, or DOS'
24: mvi c,2 43: db cr,lf,'system diskette in the boot dirve'
25: call bdos 44: db cr,lf,'and press any key to boot it'
26: ret 45: db cr,lf,lf,lf,lf,' ..... > $'

```

DOS 3.3 (FIRST IMPRESSIONS)

by SANDY GREENFARB

When I opened my DOS 3.3 pack, I found two diskettes, two ROMs, a ROM extractor, a sheet of '16 sector' labels (little red apples with 16 in each center), and the various and sundry papers that always accompany a new purchase. First I reviewed the enclosed manual, naturally entitled "The DOS Manual". Owners of the DOS 3.2 manual will observe that with the exception of four new appendices, changing all "13 sector" references to "16 sector" references and some other minor updates and changes, that the two manuals are virtually identical, page by page.

Needless to say, the first new appendix, Appendix H, describes how to update the system. I took the extractor and removed the specified PROMs from my disk controller card and proceeded to install the new ones. The pins on one PROM were spread too wide and I had to (very carefully) bend them into place. After a little heart failure when I almost bent a pin, the PROMs were installed. Next I returned the controller card to the APPLE and turned on the power.

Well, with the Autostart feature, I forgot the drive would self-start and, lo and behold, I learned that I could no longer boot up the system with a 13-sector diskette. Oh well, back to the manual. (Note: There are several pages on PASCAL systems that I will not describe, as I do not have the Language Board and can not properly evaluate them. However, from reading them, my "uneducated" opinion is that PASCAL owners will want to upgrade to 3.3.) From now on, it would take a DOS 3.3 diskette to boot DOS.

So what about all my 13-sector diskettes? As I said, the system came with two diskettes. The first was a System Master, containing the usual group of demonstrations and utilities, including some new ones which I will describe later. The second, DOS 3.3 Basics, is described in Appendix I. Boot the system with this diskette and in about five seconds the screen will say "INSERT YOUR 13-SECTOR DISKETTE AND PRESS RETURN". I tried this with a couple of 13-sector diskettes and relieved a lot of anxieties; they would all still work on my system. This included both my standard DOS diskettes as well as my non-standard ones, the type that contain their own version of DOS so that they can't be copied. From now on, one new step was necessary before booting a 13-sector diskette. I would have to boot the DOS 3.3 Basics diskette first. Once in either 13- or 16-sector mode, the system would expect the same sector configuration until it was rebooted. On a boot from "PR#6" it always would expect a 16-sector configuration. (When I examined the System Master diskette, I decided to experiment with one program called BOOT13. BRUN BOOT 13 and your system will also be ready to boot a 13-sector diskette. This was not covered in the manual and I guess that it was expected to be obvious to users.)

A side note for Language Board owners: according to the manual, the DOS 3.3 System Master loads both Basics into your system upon boot. Depending on which is in ROM, the other will be loaded into RAM on your Language Card. A procedure is described to set any blank 3.3 diskette to do the same.

At this point, please note that the rest of the article deals exclusively with 16-sector diskettes unless 13-sector is specifically mentioned.

The new utilities are excellent. A diskette copy program is provided in two flavors, Integer and Applesoft. It appears to be designed to use single-drive, dual-drive, or dual-controller as designated by the user. I can't speak for the latter two options, but it works fine on my single-drive system.

FID (File Developer) is the subject of Appendix J. Six pages are devoted to this fantastic utility. Not only does it allow you to easily catalog, copy, delete, lock and unlock all types of DOS files, it also lets you copy all types of files from one diskette to another on a single or multi-drive configuration, all with simple keystrokes. The "wild card" option provides a capability to specify file names that meet certain characteristics. Specifying a file name of "=", a single equal sign, means that all files on the source diskette qualify for what you are doing; "G.=" would mean all files that began with "G.". These are just a sample of some of the possibilities. This utility alone is almost worth the price of the system.

The final new appendix, Appendix K is entitled "Using the MUFFIN Program". No mention is made how Apple, Inc. came up with this name/acronym, but this is the program that allows the conversion or copying from 13-sector to 16-sector diskettes. BRUN this program and you will be asked to specify your slots and drives for your source diskette, 13-SECTOR, and your destination diskette, 16-SECTOR. Then it will ask for your filename. You may specify actual file names or use the "wild card" options as described in FID. If you specify a "wild card" option, you will also be asked if you desire prompting. This gives the option of saying yes or no to each file name as it is presented on the screen. Unlike the old "UPDATE 3.2" program which updated diskettes from DOS 3.0 or 3.1, this update to 16-sector does require both a source and destination diskette. Of course, a 13-sector diskette can be re-initialized to 16-sector, but any information on the diskette would be lost. I honestly do not know if it is possible to update a diskette to itself without losing the information on it. I do know that Apple did not provide any utility to do this "thing" on the System Master. I have already converted ten diskettes with no problem whatsoever. However, I have found this process very tedious on a single-drive system and intend to borrow another drive from a friend before doing any more. The constant switching of diskettes, one in and out and then the other in and out for each program, becomes very boring. Beg, borrow or steal a drive, but most definitely use two drives contd.

for your 13 to 16 conversion. Incidentally, INIT now takes about 30 seconds instead of the old several minutes.

Except for the difference in number of sectors, DOS 3.3 appears to function identically to 3.2; in fact, even the page-3 hooks are identical. I'm sure some differences must exist, but I haven't observed any yet. Dos 3.3 also occupies exactly the same amount of memory as 3.2. Programs that internally use DOS may or may not work properly. I've found that if they use the page-3 hooks, they stand a good chance of working. The main conversion problem that appears to occur is in the number of sectors. Those programs that look for a max sector number \$0C (decimal 12) should be changed to \$0F (decimal 15). I am only partially into my conversions, but I still have a pessimistic attitude as to what will happen when I test any programs that bypass the page-3 hooks and go directly to DOS.

Knowing what I know now, I would still willingly purchase the DOS 3.3 package. However, I do have mixed feelings. I know that some of my program conversions will be very frustrating, and I do not like the extra step that has become necessary for booting "non-standard" diskettes. By the time I convert my relatively large diskette library, I will surely be grateful for the increased diskette space. But does that justify the purchase? I think not. I believe my gain will be realized some time in the future when I go to buy some new programs and find that they are only sold on 16-sector diskettes. DOS 3.3 is the new "state-of-the-art" for APPLE II and I can only find it logical that software suppliers will come to the same realization.

HOW TO CONVERT FROM DOS 3.3 TO 3.2

by SANDY GREENFARB

Granted that what I am describing is backwards, but I found myself in the situation where I needed a program on a 13-sector diskette that was only on a DOS 3.3 16-sector diskette. Not having an unlimited supply of diskettes, I had already initialized a diskette for 16-sector that had contained a program that I needed to use with other 13-sector programs. Thus I was placed in a situation where I needed to convert backwards. What I came up with is "extreme brute force", but it does work.

1. Boot the system into DOS 3.2 mode.
2. BSAVE the DOS to a 13-sector diskette.
3. Boot the system into DOS 3.3 mode.
4. Using the MUFFIN program, transfer DOS to a 16-sector diskette.
5. While in 3.3 mode, load the program you wish to convert to 3.2.
6. Now load the DOS 3.2 image, taking care not to overwrite your program nor DOS 3.3.
7. Execute the Monitor move utility, moving the DOS 3.2 image where it belongs.

At this stage, it is just as though you had booted with 3.2, and you can save your program on a 3.2 diskette. There may be other solutions, but I don't expect to do this over often, and it works.

(Editor's Note: NIFFUM, a backwards MUFFIN that runs on DOS 3.2 with new ROMs is available on a WAP Library Disk.)

PROPOSED MEMBERSHIP DIRECTORY

TOM JONES, Membership Chairman

On the fourth Saturday of each month, 100 or so APPLE enthusiasts gather at George Washington University for the Washington Apple Pi club meeting. At the meeting, I look forward to picking up the club's current newsletter, the newest release of the club's program library disk(s), and listening to the meeting's speakers. But most of all, I enjoy the communication with other APPLE owners, talking with them about my accomplishments, frustrations and problems, and finding that many of them have solutions to my problems and ways to improve my programs.

We share our knowledge and everyone benefits. Our club is built around communicating with fellow members, whether through our newsletter, the speakers at our meetings, or just chatting with other meeting attendees. What can we do to improve our club communications? I feel that an effective way is to create a Membership Directory.

As Membership Chairman, I have taken on the task of producing the first Directory. Below is the print format that I propose for the Directory. I also foresee that it would contain two listings of our members, one by last name, the other by zipcode. Of course, any member who has not given permission for his name to be released will not appear in the Directory. Your comments are welcome.

WASHINGTON APPLE PI MEMBERSHIP DIRECTORY

---LAST NAME---	---FIRST NAME---	-----CITY-----	ST	-ZIP-	----PHONE----
JONES	THOMAS E.	ROCKVILLE	MD	20853	(301)450-8773
PECK	ROBERT	ROCKVILLE	MD	20852	(301)468-2305
SCHWARTZ	DANA J.	LAUREL	MD	20810	(301)725-6281
URBAN	BERNARD	BETHESDA	MD	20016	(301)229-3458
WASSERSTROM	RICHARD S.	MCLEAN	VA	22101	(703)893-9147

TUESDAY NIGHT FOOTBALL - A REVIEW

BY BILL SCHULTHEIS

THE STEELERS BREAK FROM THE HUDDLE
BRADSHAW BARKS THE SIGNALS
AND HERE'S THE PLAY...

BRADSHAW IS FADING BACK
HE'S GETTING A HARD RUSH
DELAYED HAND OFF TO FRANCO
FRANCO'S ACROSS THE 25

AT THE 30
CREAMED AT THE 33
TACKLE BY BRAZILE

FIRST DOWN PITTSBURGH

Is it Monday Night Football on ABC television? No, it's Tuesday Night Football on your APPLE! If you are a pro football fan, you will love Tuesday Night Football. The program captures the color and feel of pro football with sound, text, and graphics in a way that makes other computer football games seem pale in comparison. Item. The kicking game is simulated in 3-D perspective low-res graphics. Item. You see both a diagram of the playing field and get a play-by-play description of the action in text like the sample above. Item. Clever sound effects. Item. The game simulates momentum which can change after a big play. Item. The game has injuries and substitutions. And would you believe a halftime show featuring the University of Houston Marching Dots?

Only a dedicated pro football fanatic could have packed so much into one Applesoft program. The program is published by Shoestrings Software in Houston, Texas. So it should be no surprise that you call the plays for the Houston Oilers playing against the hated Pittsburgh Steelers. (The instruction book tells you how to change the names of the players so you can coach the Redskins against the Cowboys or any other set of players which captures your fancy.)

The game comes on disk. Actually the disk includes both ROM and RAM versions of two programs, KICKING PRACTICE and TUESDAY NIGHT FOOTBALL. It even includes a copy of Applesoft. The Kicking Practice game lets you practice your punting and place kicking (I'm not kidding) before you start the real game. During practice the field is green and the stands are grey. When the game starts... I won't spoil the surprise by telling you.

As with most computer football games, you can set the length of the quarters. If you chose 15 minute quarters, the game will last just about as long as a real pro game on the tube. You cannot save a game and come back to it, but you can have up to three time outs per half to control the clock, answer the phone, or get a brew.

Depending on whether you have the ball, you pick the offensive or defensive plays from a list. According to the documentation, the computer promises not to peek at your selection. The scoreboard on the screen has a 30 second timer which counts down while you are pondering your selection. Take too long and it's a 5 yard penalty for delay of game. The list of offensive plays includes a good variety of running and passing plays. The defensive list is shorter and assembles the usual ones (STANDARD, GOAL LINE, NICKEL, and BLITZ with unusual choices like GAMBLING (try to force a fumble) and BUTKUS (try to dismember the quarterback.)

And there are penalties, often at the worst possible time. If the penalty is against the opposition, the computer explains your options in text and lets you accept or decline. If the penalty is against you, the computer describes your infraction, names the guilty player and lets Pittsburgh make the choice.

At the end of each quarter the program displays the stats in the usual format with yards rushing, yards passing, time of possession, etc. for each side. The Steelers are pretty good, so you will be behind a lot. But every so often Stabler seems to get a hot hand, and Houston leads both in points and statistics. You might even win!

Tuesday Night Football is available at The Program Store which is at Van Ness and Wisconsin in Northwest D.C. The Program Store specializes in software for the TRS-80, but is building up a decent stock of programs for the Apple.

Periodical Reviews

BY BILL SCHULTHEIS

BYTE Vol. V No. 9 (SEP 80)

The theme of the issue is "homebrewing". It is almost exclusively aimed at hardware construction enthusiasts. Only a few items of possible Apple interest, and those are for assembler and Pascal programmers. Short items on pages 98 and 114 describe techniques for making indirect subroutine calls in 6502 assembly language. A short item on page 324 (yes it's a thick issue) describes a UCSD Pascal compilation unit which provides PEEK and POKE functions. Other articles include:

P-270 EXPLORING BALLISTICS WITH YOUR COMPUTER by Robert W. Jenks

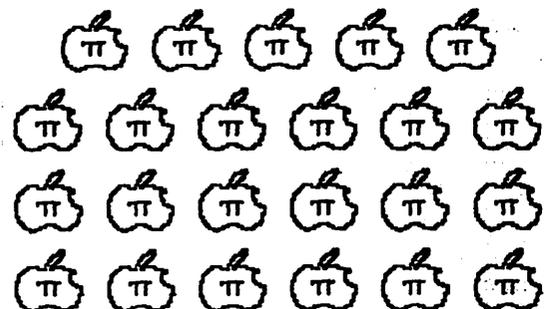
Describes a program which calculates the trajectory of a bullet given cartridge description, atmospheric parameters, muzzle velocity, etc. Might be useful to gun nuts. In North Star BASIC and would need some rewriting to run on the Apple.

P-76 DISSECTING THE TI SPEAK & SPELL by Michael A. Rissby

Did you ever have the urge to dismember an electronic toy and figure out how it works? If so you might find this article interesting. The author was able to get the insides out of the TI "Speak & Spell" toy. He found some interesting IC's but was not able to figure out how they work together.

P-106 MACHINE PROBLEM SOLVING, Part 1: Trial and Error Search, A Mechanical Plan to Save the Missionaries by Peter W. Frey

Describes how to solve problems by searching and backtracking using the familiar missionaries and cannibals problem as an example. The example program is in TRS-80 BASIC.



A REVIEW OF LIBRARY DISK VOLUME 9 - EDUCATIONAL I

by LES STUBBS

The review of Library Disk 15 by Brian Dormer (August 1980 issue) was informative and well presented. I keyed in his review into a new HELLO program which now presents a brief description of each program when I boot the disk. Now, I have a nice record of what each program does right on the diskette. I found this so helpful that I started writing descriptions for other Library Disks that I have, the first one being Volume 9 - Educational I. I have tried to follow the same format in which Brian presented his review of Volume 15. As in Brian's review, the rating scale is as follows:

***** Superb
**** Better than average
*** Good
** OK
* Forget it.

COLOR MATH - Lo-res multiple colored math problems are presented. Multiplication, division, addition and subtraction problems are included. A "HAPPY" face is shown if the answer is correct. A "SAD" face is shown for incorrect answers. Faults: Problem is repeated when an incorrect answer is given and this continues until the correct answer is given. This could be frustrating for a child who does not know the correct answer. Rating: ***

CONVENTIONS - Shows the programming conventions used on APPLE CORE submitted programs. Some good programming techniques exist here. A must for beginner programmers to review. Rating: ***

ECHOCARDIOGRAPH - Calculates factors of blood flow using TEICHHOLZ'S modifications to Dodge's formula. Text oriented. Good, if you know something about medicine. Rating: **

FLASH CARD - Allows you to practice your addition, subtraction, multiplication skills. Answers must be given in a specific time period. Time period decreases automatically as you begin answering correctly. Text oriented. Rating: ***

FLASH CODE - A Morse Code training aid. Uses speakers and flashes the letter being sounded at the same time. You can change the speed. Faults: Only random letters are presented. Rating: **

INTEGER INSTRUCTION SET - Displays the tokens for Integer Basic. This is nice to have but can be found in many publications. Rating: *

MATH TUTOR - Practice your addition, subtraction, multiplication and division using this program. Text oriented. Also has nice messages when you give answers. I've seen better, though. Rating: **

MORSE CW - Excellent Morse Code learning

tool. Three ways to learn: one letter at a time, a complete message, or random letters. Speaker sound is best I've heard. Rating: ****

MORSE TRAINER - Another Morse Code training aid. Advantage of this one is that it will not display the characters until test is completed. Has three options: a message, random letters (both of which display characters as they are sounded), and a test of random letters (which are not displayed until the end). Rating: ***

NAME STATES - Practice naming the 50 states (spelled correctly). Text oriented. Keeps a count of the number you have named, and allows you to see the names you have correctly given. Rating: ***

NORTHERN CONSTELLATIONS - A good program that explains the stars briefly and also shows the Northern Constellations based on your input of a month, day, and hour. Display uses hi-res graphics. Rating: ****

QUIZBUILD - This program allows you to build a text file of data to be used later by a program called "QUIZ" (which I do not have). Instructions on how to build these text files via input from keyboard are good. Maybe someone has the "QUIZ" program. Rating: **
(Editor's Note: I have "QUIZ" on Vol. 5B of the SF APPLE CORE library.)

SIMULATION-6502 - Want to learn 6502 Assembler? This program may help. It allows you to graphically see the data bus, registers, accumulators and the program counter. It lets you enter instructions and then watch them being executed one at a time. Rating: *****

STATES/CAPITALS - Do you know the capital of each state? This program can help anyone who is learning them. Faults: If answer is wrong, it keeps asking you for the correct answer with no way to get out. Rating: ***

TITRATION - Chemistry, anyone? The object of this program is to normalize an acid by neutralizing it with an amount of base. Low-res. Rating: ***

TOP-DOWN PROGRAMMING - Basically the same conventions are presented here as in the program "CONVENTIONS". This one does present a bit more. You should review this program, at least once. Rating: **

TYPING PRACTICE: Allows you to practice your typing. Has four options: Single letters, 2-letter words, 3-letter words, and words of random length. The speaker sounds different tones for correct and incorrect replies. A score is given at the end. Rating: ***

Well, that's it. I hope to see more such reviews from other members.

(Editor's Note: Les is a resident of Harbor City, California, and a member of Washington Apple Pi. Thanks, Les, for a great review!)

THE PEOPLE I SAW AT THE COMPUTER SHOW

by GENEVIE URBAN

The Mid-Atlantic Computer Show at the DC Armory last month was full of new and interesting technical displays, but the PEOPLE who attended the show (at least, those that came by our club booth) were just as interesting to me as the technology. Here are some of the PEOPLE I saw:

There were WAP members who stopped by to say hello, order library disks, or just to attach a face to the mailing label...There were people who were in the market for a personal computer but could not decide which one to buy (guess which direction I sent them in)...There were people who wanted to "blow off steam", not about computers or Washington Apple Pi, but about The American Medical Association...There were kids, tons of them, all seeming to feel perfectly at ease in this wonderful world of computers...There were people who didn't know what an APPLE or an APPLE PI is...There was an amiable Armory police guard who stopped by each day for one of our apples (eating) who had never heard of an APPLE computer and who said, when I showed him the computer, "Yea, that's the typewriter but where's the rest of it?"...There were former WAP members from the "old days" when we were just getting started who had drifted away and wanted to return...There were girls on roller skates wearing short shorts...There were "paper collectors" who took any free literature and chucked it in their plastic Computer Store bag...There were even a few people who took "freebies" that really weren't free...There were people who were astonished to see that The Woz is not much more than a teenager...There were people who thought it was just great that we were giving away delicious apples...There were people who just wanted to talk at great length about programming the APPLE...There were people who weren't people at all, but robots walking around or playing rock music...There were teachers looking for people to help them teach about the APPLE...There were people from the Children's Museum who wanted assistance on the APPLE unit in their Communications Exhibit...There were people looking for an APPLE III...There were people from all over who thought our DEMO program (produced by our own Mark Crosby) was "the greatest"...

There were these and many more, and all in all the PEOPLE I saw at the Show lightened my heart and tickled my fancy. And I came away from the show feeling good that I had "womaned" our booth for a few hours.

STOCK MARKET UTILITIES

3 STOCK MARKET PROGRAMS ON DISK

STK.1 (37 Sectors) provides complete utilities for manual entry of stock data.

FEATURES: names stored alphabetically by exchange, easy addition and deletion of names, automatic prompting and extensive error trapping for data entry (date, volume, price), numerous entry points for data correction, all data displayed prior to updating stock files with further option for data correction, option for inputting historical data to a single data file, display individual stock files from disk, option to reduce files to last 260 entries for high-res graphics.

DATA CORRECTER (14 Sectors) used to correct and rewrite stock data files.

FEATURES: option for general data correction - correct any entry, option for stock splits - all prices and volumes prior to split scaled by split ratio to provide continuous momentum and price curves.

EVAL (20 Sectors) provides comparative evaluation of stock performance.

FEATURES: synchronizes NYSE index ave with first stock entry, option to evaluate all stocks automatically or just one, simultaneous high-res display of momentum, price, and price relative to NYSE index, auto scaling graphics, numerical figure of merit for performance relative to NYSE index ave.

Much more! Programs written by H. S. PILLOFF

Requires Apple II, ROM Applesoft, 48K and Disk

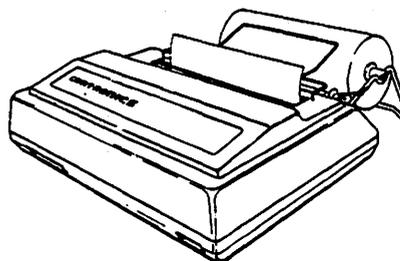
Master Charge

Price \$39.95
Md. residents add 5%

Visa

H&H SCIENTIFIC
13507 PENDLETON ST.
Oxon Hill, MD. 20022

TEL (301) 292-3100



CENTRONICS MODEL 730-1 PRINTER

Used - \$450

- Parallel interface
- 96 character ASCII
- 7x7 dot matrix
- Uses plain paper
- Friction&pin feed
- 2 character sizes

90 day warranty



12503 Kings Lake Dr.
Reston, VA 22091
(703) 620-2994

Maryland area call
Ira Cotton 468-2266

SAVING AN ADVENTURE TO DISK AND MORE

IF GVP THEN 4511: GOTO 2260

by MICHAEL THOMAS, NOVAPPLE

I have had the "ADVENTURE" program from Conflict Control since February of this year. It is an excellent game with one bad fault. It lacks the ability to halt an adventure and return to it later. This is very inconvenient, since whenever I have to leave for some reason I either have to end my adventure early or I have to leave the APPLE with the power on and my adventure at the mercy of the all too frequent power outages. I do not like to do either of these. An adventure in progress also precludes the use of the APPLE for anything else. Again, this is inconvenient when I need to use the computer for a school project, or if I just want to play "SPACE INVADERS". As a solution to these problems I wrote the following routines to allow an adventure to be saved to disk and later restarted.

There are three parts to this. First, are the modifications to the main adventure program, "ADVM"; second, is the program "RESUME.WRITE", which writes the text file "RESUME". After this file has been written on the adventure diskette, delete "RESUME.WRITE". Third, is the program "HELLO", which replaces the current "HELLO" program. After these have been entered and saved, it is necessary to delete the following programs: "EDTEXT", "LISTEXT" and "ADVENTURE". This is to provide disk space for the variables and tables of the halted adventure. These programs are not needed for running "ADVENTURE", but they do have their uses, so save them on another diskette. It is not necessary to delete these programs if you have the DOS 3.3.

When an adventure is first resumed, the prompt will come up; type "LOOK" to find out where in your adventure you are.

I also had the problem of not knowing which of the 600,000 words in the English language are the 300 recognized by "ADVENTURE". The program "ADV WORDS" prints a list of these words.

Finally, the program "LISTEXT", which came on the adventure diskette, is supposed to list the "ATEXT" file (which contains the adventure messages), but it does not do this. The modifications to make it work also follow.

MODIFICIATIONS TO ADVM

Add:

```
15 IF QUIT THEN 2260
4511 M1=0:M2=594:M3=0:PRINT "DO YOU WISH
    TO RESUME YOUR ADVENTURE AT A LATER
    DATE?":PRINT:GOSUB YES
4512 IF NOT (YEA) THEN GOTO 7000:QUIT=1
4513 PRINT D$:"BSAVE PNTRS,A$CC,L2"
4514 PRINT D$:"BSAVE ADVIP,A$800,L";
    (PEEK(204)+PEEK(205)*256-2047)
4515 PRINT "COME BACK SOON"
4516 END
```

Change:

```
4510 M1=516:M2=0:M3=594:GOSUB YES:GVP=YEA:
```

RESUME.WRITE

```
10 D$=""
20 PRINT D$:"OPEN RESUME"
30 PRINT D$:"WRITE RESUME"
40 PRINT "NOMON C,I,O"
50 PRINT "LOMEM:9000"
60 PRINT "BLOAD PNTRS"
70 PRINT "BLOAD ADVIP"
80 PRINT "LOAD ADVM"
90 PRINT "CALL -9060"
100 PRINT D$
110 END
```

HELLO

```
10 REM K. BEASELY-TOPLIFFE WROTE THIS.
    COPYRIGHT 1979 NOW HELD BY CONFLICT
    CONTROL.
20 REM MODIFICATION ENABLING CONTINU-
    ATION OF AN ADVENTURE AT A LATER
    DATE BY MICHAEL THOMAS NOVAPPLE
30 D$=""
40 CALL -936
50 PRINT:PRINT:PRINT:PRINT "1) START
    A NEW ADVENTURE":PRINT "2) RESUME
    AN OLD ADVENTURE"
60 PRINT:PRINT:INPUT "1/2",I
70 CALL -936:PRINT "EXCUSE ME WHILE I
    INITIALIZE."
80 GOTO 90+(I-1)*20
90 PRINT D$:"EXEC ADVX"
100 END
110 PRINT D$:"EXEC RESUME"
120 END
```

ADV WORDS - APPLESOFT

```
10 D$=""
20 PRINT D$:"BLOAD TABLES"
30 PRINT "WORDS RECOGNIZED BY ADVENTURE":
    PRINT
40 FOR I=20097 TO 21594 STEP 20
50 FOR R= 0 TO 3
60 FOR J = 0 TO 4
70 PRINT CHR$(PEEK (I+J+(R*30)));
80 NEXT J:PRINT " ";
90 NEXT R:PRINT
100 NEXT I
110 END
```

For a single column, change:

```
40 FOR I=20097 TO 21594 STEP 5
80 NEXT J:PRINT
And delete line nos. 50 and 90
```

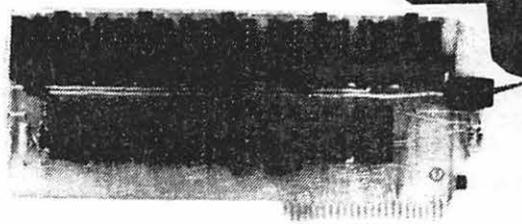
MODIFICATION TO LISTEXT

Change:

```
10 DIM A$(40)
70 FOR I=1 TO 1181
Delete line nos. 40, 50 and 60
```

Many thanks to Hal Weinstock for making his printer available to the Editor this month. Our printer is suffering from a mild case of downtime.

**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 **\$195⁰⁰** Includes Shipping



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

BLAISE AWAY!

Fooling the Compiler

-or-

How to Know Where Your'e Pointing

by

DR. UO

In this month's column we continue our discussion of pointers. It's a mixed bag of rather loosely related ideas which includes improvements to the program given in the July column (please refer to that issue) and a discussion of how to "fool" the compiler into pointing to absolute locations in memory.

Why Use Pointers?

=====

Some of you asked me what the July program "does". The main thing it does is illustrate three pedagogical points: declarations for pointers, dynamic memory allocation using the procedure "new", and traversal of a linked list in the procedure "printlist". Beyond that the program simply dynamically creates a list of records ("personrec"s) then prints the list on the console-- admittedly not a very exciting program, but keep in mind the teaching points.

Some of you also asked me why use pointers. Indeed, the major alternative data structure, an ARRAY OF personrec (a static data structure), seems appealing so why not use it? There are three general reasons, the importance of which will increase with the size and complexity of your application program. One is that we will likely not know at compile time how many entries will be in the list or array. Therefore, we may find ourselves recompiling to accommodate an unexpectedly large number of entries. The second reason is that the use of linked lists introduces the possibility of dynamic memory management using the procedures "mark" and "release". These built in procedures can be used to perform the inverse of "new", namely to deallocate memory so that lists larger than RAM storage can (sometimes) be processed. We will postpone discussion of such memory management 'till another time.

The third reason for using dynamic lists has to do with organization and sorting. In some applications you will want to have ready access to a sorted list and you will want to insert and delete entries at will, all the while maintaining the order of the list. Such exercises are cumbersome with an array: deleting an entry leaves a "hole" which must be dealt with, and inserting an entry means that the surrounding entries have to be pushed around to make room. With pointers the job is simpler: we merely unlink an entry to be deleted, or form a new link for an entry to be added. The overhead of finding the position for a new entry is the same in both the array set-up and the linked list.

Adding a Dynamic Sort to "first_in_last_out"

=====

Let us now consider the problem of inserting a new record in connection with our example program. Suppose we wish to create our list of names and telephone numbers alphabetized by name. Given the list is in some state,

contd.

including empty, how do we insert a new entry in its appropriate position? A straightforward solution is to traverse the list until we find the position of the new entry, then adjust pointers to reflect the updated situation. We declare a variable "traveller" of TYPE `personrec` which we use to traverse the list.

Suppose the situation is as pictured in Figure 1: we have traversed the list and located the position for "newperson"; it is to be inserted between "traveller" and the entry preceding "traveller". Can we now insert "newperson"? No! The trouble, if you will, is that we have a singly linked list; our pointers only point down the list and our program can't know what is behind "traveller" unless it keeps track of that. Therefore, we introduce a variable "companion" which follows "traveller" down the list.

With the declaration of "companion" the situation will be as in Figure two. The following code fragment will then link "newperson" into its proper place:

```
newperson.next:=traveller; (Fig. 3a)
companion.next:=newperson; (Fig. 3b)
```

A little thought should convince you this works and that a little extra care needs to be taken if "newperson" belongs at the head of the list.

The modifications to our original program are all contained in the procedure "crestelist". (Note the advantage of modular design in this regard.) Only the modifications are presented below. The task of obtaining a new record has been isolated in the procedure "setperson" and the task of inserting the new record is isolated in "insertperson". For clarity, "insertperson" passes "firstperson" as a parameter since we may have to insert the new record at the top of the list, thus requiring adjustment of "firstperson".

Putting Blinders on the Compiler =====

As I mentioned last time, one does not normally know, or care, what location a pointer is pointing to. However, there is a technique to relieve us of such ignorance which can be used to good advantage in a variety of situations. The technique, a certain use of variant records, contravenes the spirit of Pascal since it permits us to dynamically "type" a variable (thus avoiding the usually strong typing requirements) and it may not work for all implementations of Pascal. The technique relies on the fact that pointers are stored as signed integers in the Apple's memory.

Consider the following:

```
TYPE freeunion=RECORD CASE BOOLEAN OF
    TRUE:(addr:INTEGER);
    FALSE:(contents:↑INTEGER);
END;
VAR memory:freeunion;
```

"memory" is an undiscriminated variant record, or a variant record without a tag field (Groszono, pp 175-176). This means that the location in core reserved for "memory" can be referred to by two names at execution time. When referred to as "memory.addr" it will be considered an integer, and when referred to as "memory.contents" it will be considered a pointer to an integer. (Groszono, pp 239-240)

Refer to Figure 4 and consider the following fragment:

```
memory.addr:= -16368; (Fig. 4a)
memory.contents:=0; (Fig. 4b)
```

The first line deposits the value -16368, which is the signed integer address of the keyboard strobe, into the location reserved for "memory". The second line says: interpreting the information stored in "memory" as a pointer to an integer, store 0 in the location pointed to. The effect is to clear the keyboard strobe! The modular procedures "peek" and "poke" listed below use these techniques to duplicate the functions of their BASIC (ugh!) counterparts.

Further Applications: Saving and Recalling Graphics =====

To further illustrate the versatility of this technique let's develop a pair of modular procedures to transfer hi-res graphics images to disk and back again. Consider:

```
CONST screenstart=8192;
      screensize=16;
TYPE foto=PACKED ARRAY(0..8191) OF 0..255;
      fotounion=RECORD CASE BOOLEAN OF
          TRUE:(addr:INTEGER);
          FALSE:(contents:foto);
      END;
VAR picture:fotounion;
      blockstransferred:INTEGER;
      fotofile:FILE;
```

With these declarations the fragment

```
picture.addr:=screenstart;
blockstransferred:=blockwrite(fotofile,picture.contents,screensize,0);
```

sets "picture" to point to the start of the hi-res screen and then writes the array pointed to, the screen, to the diskfile currently associated with "fotofile".

Procedures for saving and fetching graphics, "savefoto" and "fetchfoto", are declared below. Neither provides any I-O security in that "fetchfoto" expects a file of 16 blocks corresponding to a graphics image to be on disk, and "savefoto" expects there to be room for a file of 16 blocks. You may want to improve them in these respects.

SIG News =====

Thanks to Dick Hodder of the Pascal SIG, and the Uniformed Services University Health Science Center, the SIG has an outstanding location for its monthly meetings. We will be meeting the third Thursday of each month, 7:30 pm, at the Center, and we will have use of the Apples, including language systems, that reside there. Our first meeting will be Oct. 16, 7:30 pm. The Center is located near the intersection of Wisconsin Ave./Rockville Pike and Jones Bridge Rd. in the neighborhood of the National Institutes of Health. Please call me for additional information.

There is another Pascal group around town which meets the fourth Wednesday of the month at Computers Plus in Franconia, Va. The group, all

contd.

Apple owners, is congenial and knowledgeable, but the meetings are somewhat disorganized-- last time our meeting room was pre-empted by another group and we spent the evening drinking beer and swapping stories.

Examples
=====

```
PROCEDURE createlist(VAR firstperson,lastperson:link);
VAR newperson :link;
    personcount:INTEGER;
    answr:CHAR;
```

```
PROCEDURE getperson(VAR newperson:link;VAR personcount:INTEGER);
BEGIN
    new(newperson);
    WITH newpersonf DO
        BEGIN
            writeln(personcount,' records entered. ');
            writeln;
            writeln('Enter the record of the next person: ');
            writeln;
            write('Name: ');
            readln(name);
            write('Phone: ');
            readln(phone);
            personcount:=personcount+1;
        END;
    END;
```

```
PROCEDURE insertperson(VAR newperson,firstperson:link);
VAR traveller,companion:link;
BEGIN
    (find the position of newperson and insert in the list.
     Start at the top of the list.)
    traveller:=firstperson;

    (Traverse the list until the alphabetical position of newperson is
     found or we reach the end of the list.)
    WHILE (newpersonf.name>travellerf.name) AND (traveller<>lastperson) DO
        BEGIN
            (set companion to point where traveller does)
            companion:=traveller;
            (move companion to point to the next record)
            traveller:=travellerf.next;
        END;

    IF traveller=firstperson
    THEN BEGIN
        newpersonf.next:=firstperson;
        firstperson:=newperson;
    END
    ELSE BEGIN
        newpersonf.next:=traveller;
        companionf.next:=newperson;
    END;
END;
```

```

BEGIN
  personcount:=0;
  new(lastperson);
  firstperson:=lastperson;

  REPEAT
    page(output);
    gotoxy(0,5);
    getperson(newperson,personcount);
    insertperson(newperson,firstperson);
    writeln;
    write('Enter another record? ');
    read(answer);
    writeln;
  UNTIL answer IN ['n','N'];

END;

```

```

-----
FUNCTION peek(location:INTEGER):INTEGER;
VAR memory:RECORD CASE BOOLEAN OF
  TRUE:(addr:INTEGER);
  FALSE:(contents:↑INTEGER);
END;

```

```

BEGIN
  memory.addr:=location;
  peek:=memory.contents↑;
END;

```

```

PROCEDURE poke(value,location:INTEGER);
VAR memory:RECORD CASE BOOLEAN OF
  TRUE:(addr:INTEGER);
  FALSE:(contents:↑INTEGER);
END;

```

```

BEGIN
  memory.addr:=location;
  memory.contents↑:=value;
END;

```

```

-----
PROCEDURE fetchfoto(filename:STRING;VAR blockstransferred:INTEGER);
CONST screenstart=8192;
  screensize=16;
TYPE foto=PACKED ARRAY[0..8191] OF 0..255;
  fotounion=RECORD CASE BOOLEAN OF
    TRUE:(addr:INTEGER);
    FALSE:(contents:↑foto);
  END;

```

```

VAR picture:fotounion;
  blockstransferred:INTEGER;
  fotofile:FILE;

```

```

BEGIN
  picture.addr:=screenstart;
  reset(fotofile,filename);
  blockstransferred:=blockread(fotofile,picture.contents↑,screensize,0);
  close(fotofile,lock);
END;

```

```

PROCEDURE savefoto(filename:STRING;VAR blockstransferred:INTEGER);
CONST screenstart=8192;
  screensize=16;

```

```

TYPE foto=PACKED ARRAY[0..8191] OF 0..255;
fotounion=RECORD CASE BOOLEAN OF
  TRUE:(addr:INTEGER);
  FALSE:(contents:foto);
END;
VAR picture:fotounion;
    blockstransferred:INTEGER;
    fotofile:FILE;
BEGIN
  picture.addr:=screenstart;
  rewrite(fotofile,filename);
  blockstransferred:=blockwrite(fotofile,picture.contents,screenstart,0);
  close(fotofile,lock);
END;

```

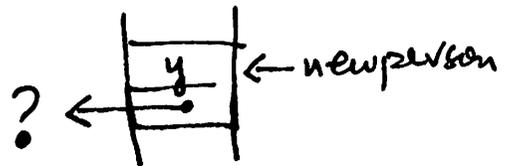
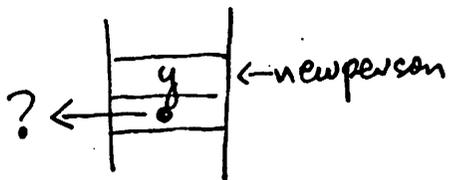
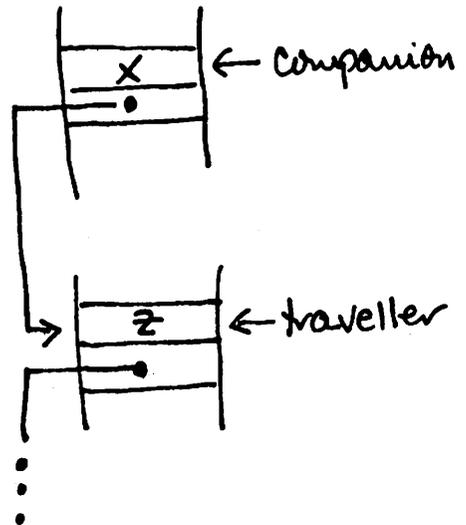
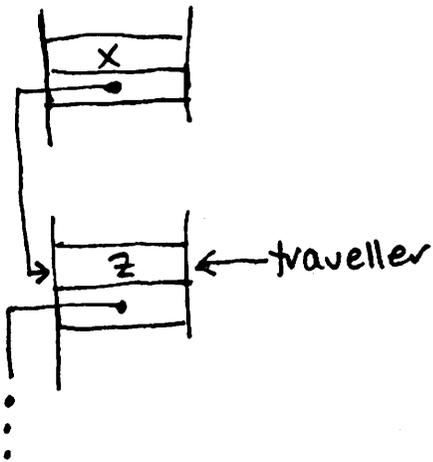
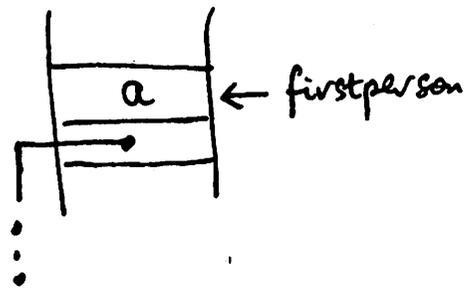
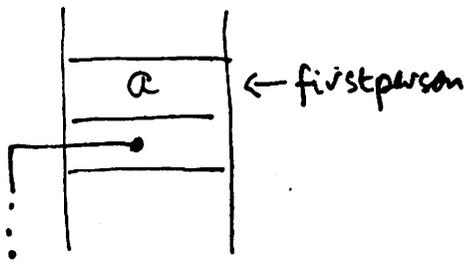


Fig. 1

Fig 2

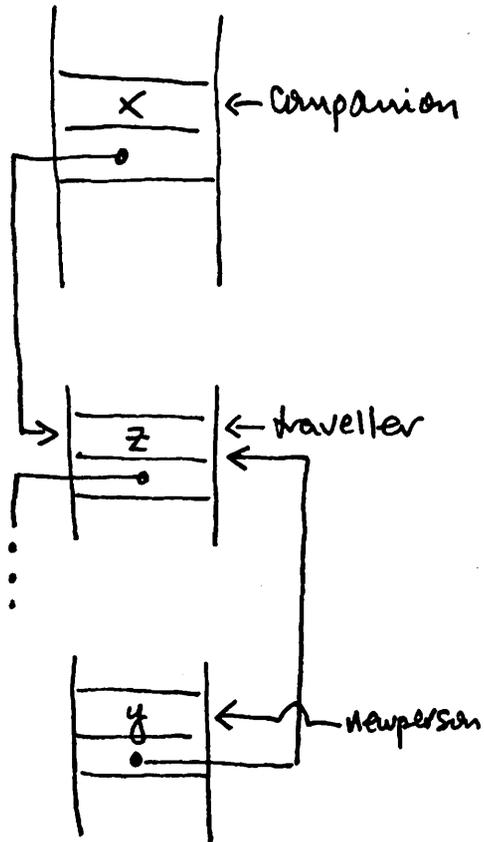


Fig 3a

newperson ↑.next := traveller

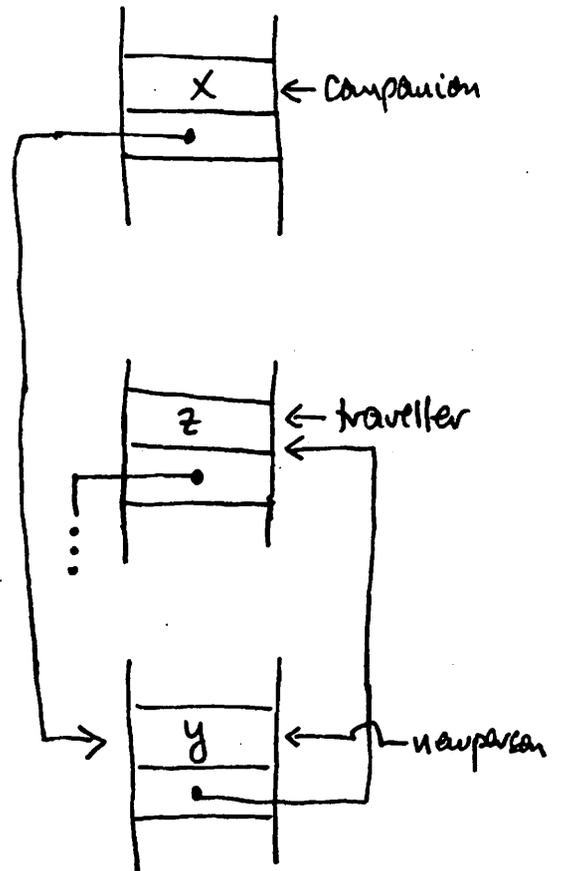


Fig 3b

Companion ↑.next := newperson

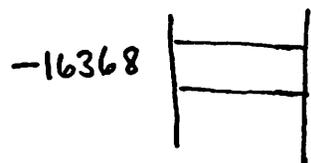
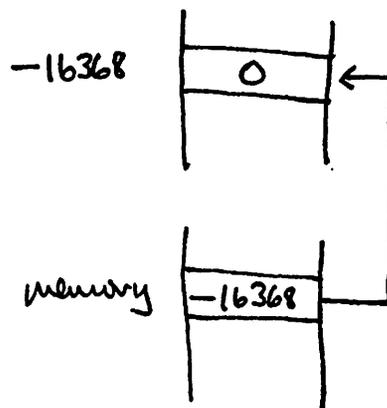


Fig 4a

memory.addr := -16368



memory ↑.contents := 0

Fig 4b

BUGS & WORMS

The following corrections/program notes have been reported. Users are encouraged to report any similar discoveries on programs either from Washington Apple Pi or other widely distributed programs to the Editor, Washington Apple Pi, PO Box 34511, Washington, DC 20034.

HELLO Program, Volume 6, Games IV: Lines 50 and 60 serve no functional purpose and can be deleted. Presently "SPACE ADVENTURE" will not run from the menu program. To fix this, change line 700 to the following:

```
700 B$="": IF (PEEK (P+3)) # 2 THEN 750:  
    B$="B":L$="RUN"
```

The change is underlined.

To convert the same program to DOS 3.3, replace line 50 as follows:

```
50 POKE 2370,15
```

DISK ACCESS UTILITY (as published in CALL -A.P.P.L.E. March 1979 and corrected April 1979): To convert the program so that it will work properly with DOS 3.3, make the following changes:

```
1420 TRACK=0:SECT=0:TRACK1=34:SECT1=15  
1520 SECT=SECT+1:IF SECT < 16 THEN 1460:  
    SECT=0:TRACK=TRACK+1
```

TUBS, The Ultimate BSTAT (originally from San Francisco Apple Core): To convert to DOS 3.3, change line 110 as follows:

```
110 T=17:S=15:FL=LEN(N$)
```

COMMUTER

BY Dana J. Schwartz

As an Integer Basic program is being developed, it sometimes becomes convenient (or necessary) to move a block of statements, say a subroutine, to a different position relative to the rest of the program. For example, you may want to move a frequently-called subroutine to the beginning of a program to improve execution speed, or an appended block may not be desired in the relative position at which it is necessary to bring it in.

Formerly, I had moved blocks of code by listing them and then tracing over them with the cursor, while making the needed updates to their line numbers. However, this soon became tedious and error-prone, and I decided to come up with a better method for transferring Integer Basic statements.

The COMMUTER routine is an extension of the Renumber/Append routines distributed by CALL-A.P.P.L.E. of Seattle, Washington in their "WOZPAK" (WAP Disk#8). The user supplies the limits of the block to be

moved (from, to) and the new destination address and the line number increment (start, step), exactly as in using RENUMBER. When COMMUTER is called, it checks for valid arguments, does the necessary memory moves, and then jumps to RENUMBER to actually change all the line number references. The Renumber and Append functions may still be used as described in the WOZPAK manual, and together with COMMUTER make a useful Integer Basic program development tool.

The program file (containing the COMMUTER subroutine) is created by poking the code from the listing below into memory from \$900 to \$9E0, loading the WOZPAK Renumber/Append routine from \$800 to \$8FF, and then saving the entire package with >BSAVE COMMUTER, A\$800, L\$1E1 to disk, or *800.9E0W to tape.

Operation:

- 1) Set LOMEM: 2560 (or greater)
- 2) a. Disk: >BLOAD COMMUTER , or
b. Tape: (RESET)
*800.9E0R and start tape
*(CTRL-C) (RTN)
- 3) >CLR
>START=(New starting line number for resulting block)
>STEP=(Line number increment for resulting block)
>FROM=(Starting line number of block to be moved)
>TO=(Ending line number of block to be moved)
>CALL 2304
- 4) If the parameters are valid, the old and new line number pairs will be listed (see example). If the parameters are invalid (overlapping ranges - FROM > TO, etc.) then the message ***RANGE ERROR* will be displayed and there will be no change to the program.

EXAMPLE:

```
>LIST  
10 GOSUB 500  
20 END  
500 FOR I=1 TO 1000  
510 GOSUB 1000  
520 NEXT I  
530 RETURN  
1000 PRINT I  
1010 RETURN
```

```
>LOMEM:2560  
>BLOAD COMMUTER  
>CLR  
>START=200  
>STEP=10  
>FROM=1000  
>TO=1010  
>CALL 2304  
1000->200  
1010->210
```

```
>LIST  
10 GOSUB 500  
20 END  
200 PRINT I  
210 RETURN  
500 FOR I=1 TO 1000  
510 GOSUB 200  
520 NEXT I  
530 RETURN
```

```

1 *****
2 *
3 * COMMUTER          6/25/80 *
4 *
5 *   BY DANA J SCHWARTZ *
6 *
7 *   INT BASIC SEGMENT MOVE *
8 *
9 *****
10 *
11 *   NEEDS WOZPAK RENUMB/APPEND
12 *
13 *           ORG   $0900
14 *
15 *   SWEET16 REGS
16 R1      EQU   $1      FR ADDR
17 R2      EQU   $2      TO ADDR
18 R3      EQU   $3      ST ADDR
19 R4      EQU   $4      LENGTH
20 R5      EQU   $5      TEMP PF
21 R6      EQU   $6      PF
22 R7      EQU   $7      MOVE-FR ; WORK
23 R8      EQU   $8      MOVE-TO
24 R9      EQU   $9      MOVE-ST ; WORK
25 R10     EQU   $A      MOVE-LEN; WORK
26 R11     EQU   $B      WORK
27 R12     EQU   $C      STACK PTR
28 *
29 *   PAGE 0 LOCS
30 FRL     EQU   $56     FR LNO
31 NEWTO   EQU   $58     TO LNO
32 STL     EQU   $5A     ST LNO
33 TMPX    EQU   $5C     SAVE X
34 *
35 LOMEM    EQU   $4A     LOMEM:
36 HIMEM    EQU   $4C     HIMEM:
37 PF       EQU   $CA     PROGRAM PTR
38 RENUM    EQU   $800    RENUM/APPEND
39 GBA      EQU   $E56D   GET BASIC ADDR
40 RNGER    EQU   $EE68   RANGE ERROR
41 SW16     EQU   $F689   SWEET16
42 *
0900: A0 21 43 CMUT      LDY   #$21      GET TO+1
0902: B1 4A 44          LDA   (LOMEM),Y
0904: C8    45          INY
0905: 18    46          CLC
0906: 69 01 47          ADC   #$01
0908: 85 58 48          STA   NEWTO
090A: B1 4A 49          LDA   (LOMEM),Y
090C: 69 00 50          ADC   #$00
090E: 85 59 51          STA   NEWTO+1
52 *
0910: A0 08 53          LDY   #$08      GET ST LNO
0912: B1 4A 54          LDA   (LOMEM),Y
0914: 85 5A 55          STA   STL
0916: 85 0E 56          STA   $E          ALSO TO R7
0918: C8    57          INY
0919: B1 4A 58          LDA   (LOMEM),Y
091B: 85 5B 59          STA   STL+1

```

091D:	85	0F	60		STA	\$F	
091F:	A0	11	61		LDY	#\$11	GET STEP
0921:	B1	4A	62		LDA	(LOMEM),Y	
0923:	85	14	63		STA	\$14	TO R10
0925:	C8		64		INY		
0926:	B1	4A	65		LDA	(LOMEM),Y	
0928:	85	15	66		STA	\$15	
092A:	A0	1A	67		LDY	#\$1A	GET FR LNO
092C:	B1	4A	68		LDA	(LOMEM),Y	
092E:	85	56	69		STA	FRL	
0930:	C8		70		INY		
0931:	B1	4A	71		LDA	(LOMEM),Y	
0933:	85	57	72		STA	FRL+1	
			73	*			
0935:	A2	04	74		LDX	#\$4	GET ADDRESSES
0937:	B5	56	75	ALP	LDA	FRL,X	LNO
0939:	85	CE	76		STA	\$CE	
093B:	B5	57	77		LDA	FRL+1,X	
093D:	85	CF	78		STA	\$CF	
093F:	86	5C	79		STX	TMPX	
0941:	20	6D	E5	80	JSR	GBA	GET BASIC ADDR
0944:	A6	5C		81	LDX	TMPX	
0946:	A5	E4		82	LDA	\$E4	PUT ADDR
0948:	95	02		83	STA	\$2,X	INTO SW16 REG
094A:	A5	E5		84	LDA	\$E5	
094C:	95	03		85	STA	\$3,X	
094E:	CA			86	DEX		
094F:	CA			87	DEX		
0950:	10	E5		88	BPL	ALP	
			89	*			
0952:	20	89	F6	90	JSR	SW16	
0955:	1C	E1	09	91	SET	R12,ST	SETUP STACK
			92	*			
0958:	23			93	LD	R3	CHK OVERLAP
0959:	D1			94	CPR	R1	
095A:	02	03		95	BNC	OKAY	START<FROM
095C:	D2			96	CPR	R2	
095D:	02	7E		97	BNC	RNG	START<TO
			98	*			
095F:	22			99	LD	R2	
0960:	B1			100	SUB	R1	
0961:	05	7A		101	BM	RNG	ERROR
0963:	34			102	ST	R4	LEN=TO-FROM
			103	*			
0964:	1B	4C	00	104	SET	R11,HIMEM	LNO OVERLAP
0967:	6B			105	LDD	@R11	
0968:	D3			106	CPR	R3	
0969:	06	17		107	BZ	NCHK	START=HIMEM
096B:	21			108	LD	R1	FR ADDR
096C:	39			109	ST	R9	
096D:	49			110	LD	@R9	LENGTH
096E:	F9			111	DCR	R9	
096F:	A9			112	ADD	R9	NEXT ADDR
0970:	D2			113	CPR	R2	TO ADDR
0971:	03	06		114	BC	LAST	NEXT >= TO
0973:	39			115	ST	R9	
0974:	27			116	LD	R7	LNO CTR
0975:	AA			117	ADD	R10	STEP
0976:	37			118	ST	R7	
0977:	01	F4		119	BR	CLP	

0979:	23		120	LAST	LD	R3	START+1 ADDR
097A:	3B		121		ST	R11	
097B:	EB		122		INR	R11	
097C:	6B		123		LDD	@R11	START+1 LNO
097D:	D7		124		CPR	R7	
097E:	06	5D	125		BZ	RNG	ERROR IF =
0980:	02	5B	126		BNC	RNG	ERROR IF <
			127	*			
0982:	1B	CA 00	128	NCHK	SET	R11,PP	
0985:	6B		129		LDD	@R11	
0986:	36		130		ST	R6	SAVE PP
0987:	B4		131		SUB	R4	
0988:	35		132		ST	R5	TPP=PP-LEN
			133	*			
0989:	26		134		LD	R6	FROM=PP
098A:	37		135		ST	R7	
098B:	23		136		LD	R3	TO=ST
098C:	38		137		ST	R8	
098D:	25		138		LD	R5	START=TPP
098E:	39		139		ST	R9	
098F:	0C	34	140		BS	MOVE	EXPAND
			141	*			
0991:	22		142		LD	R2	CHK IF UPDATE
0992:	D3		143		CPR	R3	
0993:	03	06	144		BC	CPY	NO IF ST=<TO
			145	*			
0995:	21		146		LD	R1	UPDATE
0996:	B4		147		SUB	R4	
0997:	31		148		ST	R1	FR=FR-LEN
0998:	22		149		LD	R2	
0999:	B4		150		SUB	R4	
099A:	32		151		ST	R2	TO=TO-LEN
			152	*			
099B:	21		153	CPY	LD	R1	FROM=FR
099C:	37		154		ST	R7	
099D:	22		155		LD	R2	TO=TO
099E:	38		156		ST	R8	
099F:	23		157		LD	R3	START=ST-LEN
09A0:	B4		158		SUB	R4	
09A1:	39		159		ST	R9	
09A2:	0C	21	160		BS	MOVE	COPY IN
			161	*			
09A4:	25		162		LD	R5	FROM=TPP
09A5:	37		163		ST	R7	
09A6:	21		164		LD	R1	TO=FR
09A7:	38		165		ST	R8	
09A8:	26		166		LD	R6	START=PP
09A9:	39		167		ST	R9	
09AA:	0C	19	168		BS	MOVE	CONTRACT
			169	*			
09AC:	00		170		RTN		
			171	*			
09AD:	A9	EF	172		LDA	#\$EF	NO RNG CHK
09AF:	8D	4B 08	173		STA	RENUM+\$4B	
09B2:	A9	EA	174		LDA	#\$EA	
09B4:	8D	50 08	175		STA	RENUM+\$50	
			176	*			
09B7:	20	08 08	177		JSR	RENUM+\$08	RENUMBER
			178	*			
09BA:	A9	2A	179		LDA	#\$2A	RESTORE

```

09BC: 8D 4B 08 180 STA RENUM+4B
09BF: A9 30 181 LDA ##30
09C1: 8D 50 08 182 STA RENUM+50
09C4: 60 183 RTS
184 *
185 **** SUBROUTINES
186 *
09C5: 28 187 MOVE LD R8
09C6: B7 188 SUB R7
09C7: 3A 189 ST R10 LENGTH=TO-FROM
09C8: 06 0C 190 BZ NOMV
09CA: 27 191 LD R7
09CB: D9 192 CPR R9 UP OR DOWN?
09CC: 03 09 193 BC DOWN
09CE: 29 194 LD R9 UP
09CF: AA 195 ADD R10
09D0: 39 196 ST R9
09D1: 88 197 ULP POP @R8
09D2: 99 198 STP @R9
09D3: FA 199 DCR R10
09D4: 07 FB 200 BNZ ULP
09D6: 0B 201 NOMV RS
09D7: 47 202 DOWN LD @R7 DOWN
09D8: 59 203 ST @R9
09D9: FA 204 DCR R10
09DA: 07 FB 205 BNZ DOWN
09DC: 0B 206 RS
207 *
09DD: 00 208 RNG RTN
09DE: 4C 68 EE 209 JMP RNGER RANGE ERROR
210 *
211 ST DS 2 SW16 STK
212 *

```

--- END ASSEMBLY ---

TOTAL ERRORS: 0

227 BYTES GENERATED THIS ASSEMBLY

```

0900- A0 21 B1 4A C8 18 69 01 *900.9E0
0908- 85 58 B1 4A 69 00 85 59
0910- A0 08 B1 4A 85 5A 85 0E
0918- C8 B1 4A 85 5B 85 0F A0
0920- 11 B1 4A 85 14 C8 B1 4A
0928- 85 15 A0 1A B1 4A 85 56
0930- C8 B1 4A 85 57 A2 04 B5
0938- 56 85 CE B5 57 85 CF 86
0940- 5C 20 6D E5 A6 5C A5 E4
0948- 95 02 A5 E5 95 03 CA CA
0950- 10 E5 20 89 F6 1C E1 09
0958- 23 D1 02 03 D2 02 7E 22
0960- B1 05 7A 34 1B 4C 00 6B

```

```

0968- D3 06 17 21 39 49 F9 A9
0970- D2 03 06 39 27 AA 37 01
0978- F4 23 3B EB 6B D7 06 5D
0980- 02 5B 1B CA 00 6B 36 B4
0988- 35 26 37 23 38 25 39 0C
0990- 34 22 D3 03 06 21 B4 31
0998- 22 B4 32 21 37 22 38 23
09A0- B4 39 0C 21 25 37 21 38
09A8- 26 39 0C 19 00 A9 EF 8D
09B0- 4B 08 A9 EA 8D 50 08 20
09B8- 08 08 A9 2A 8D 4B 08 A9
09C0- 30 8D 50 08 60 28 B7 3A
09C8- 06 0C 27 D9 03 09 29 AA
09D0- 39 88 99 FA 07 FB 0B 47
09D8- 59 FA 07 FB 0B 00 4C 68
09E0- EE

```

THE PROGRAMMA WORD PROCESSING SYSTEM - A REVIEW FROM A NEW USER

by SUSAN ZAKAR

PROGRAMMA, INC. has introduced a new word processing system which incorporates both a powerful editor and a word processing capability. The system is a very much improved version of two earlier programs. I received my copy of the system only about a week ago, so admittedly am no expert yet, but I will pass on my observations to date.

The PROGRAMMA Word Processing System is based on a UNIX designed system, known as ROFF. I became convinced of the magnitude of the similarity when someone in our office began to use PROGRAMMA's documentation to figure out how to operate a newly installed UNIX-based system. Those of you familiar with a RUNOFF type system should find the PROGRAMMA System very easy to use.

The system uses control keys to invoke editing functions, such as delete, scroll, insert, and so forth. In addition, the escape (ARG) key may be used to specify values associated with the particular function. Thus, to insert 5 lines in the text would take only 3 Key-strokes. Among the functions supported by the editor are the following:

- * Forward and reverse search
- * Global search and replace
- * Wraparound
- * Push and Pop buffers
- * Settable Tabs
- * Scrollings
- * Upper and Lower case
- * Supports Paymar Chip
- * Shift and Shift-lock
- * Insert control characters
- * GOTO any line
- * Output file to a printer
- * Many I/O Utilities

The above functions are by no means an exhaustive list. The selection of functions seems to be very complete and extremely well documented. The documentation (in a simulated leather binding) covers well over 100 pages just for the editor part of the system and another 40-plus for the word processing sections. Explanations and examples are by and large easy to follow, and comprehensive. PROGRAMMA is to be commended on their efforts.

The PROGRAMMA Word Processing System is easily adapted to use for form-letters, mailing lists, and other formats requiring input files. The System can even be used in editing BASIC programs and sequential text files. I have actually done the latter, and am depending on the accuracy of the documentation to support the former.

The part of the system known as FORMAT allows justification, indenting, underlining, head and foot titles, page numbering, print controls, paragraphing, and much more. Instructions are input using "dot" commands, (such as ".PL 80" to set the page length to 80). These commands are mnemonics and can be learned quickly. The range of supported commands is too wide to cover in such a short review as this. Suffice it to say that they are more than adequate for most word-processing requirements.

Provisions have been made to tailor the system to the User's environment. I have mine running on the IDS-440 Printer, off the same I/O. I had to do some jury-rigging to get it to work, but once you understand what has to be done, the process works easily and well. I can pass on specifics to anyone interested.

I hope this review is of some value to those of you still considering a word-processing program for your APPLE II. So far, I would have absolutely NO qualms about recommending the system. Cost is about \$80, well worth it, in my opinion.

SPACED FILE CABINET II

by DENNIS J. REEDER

Have you ever had the need to adjust the output of FILECABINET II (WAP Volume 2) to give spaces between output lines? My wife suggested that it would be helpful to have the capability, for editing purposes, of calling for some spacing in the output of her index data base. I found the following to work:

```
1073 DIM SPA$(1)
```

```
3494 IF SPA=0 THEN GOTO 3500  
3495 FOR LINE = 1 TO SPA  
3496 PRINT " "  
3497 NEXT LINE
```

```
5142 PRINT "HOW MANY SPACES  
BETWEEN LINES(0-5):INPUT  
SPA$:SPA=VAL(SPA$):IF  
SPA < 0 OR SPA > 5  
THEN 5142
```

Simply enter the above into the program, and when it is time to turn on the printer, the query will show up after asking for the print line. It makes the output more readable for editing purposes.



TINY LETTER WRITING PROGRAM

Max M. Methusela
662 Terracotta Trail
Entropy, Kansas
12345
September 14, 1980

Washington Apple Pi
Page 3
September 14, 1980

Gentlemen:

I wissh to demonstrayte to vew thisse moste remarkable inventionne, whiche is a machine and sette of instructiones which printtes my wordes in a fyne and miniature form.

From 'The Maxims of Methusela' Chapter IV

Yea, as fascinating as a loose tooth is a secret to a young maid. For she knoweth not whether to spit it out or keep it safe; yet she cannot forget it.

Catnip pleaseth the kitten; and the readins of her palm rejoiceth the damsel. Even as one who fitteth a **BOUCET** costume to a debutante, so is he who clotheth a woman's vanity with pleasant prophecies.

She soeth to the sorcerer and the fortune-teller and **she returneth with a marvel** alway. Yea, though she believeth not, yet doth she believe, and her lips are filled with wonders.

Washington Apple Pi
Page 2
September 14, 1980

Behold, a damsel said unto me:
How well thou understandest me;
yet I knew not what she spake, for she ended not her sentences. But I held my tongue, and forbore questionins; therefore was I clad in wisdom.

He who spilleth ice-cream upon her front breadth shall be forgiven; but whoso mentioneth last night's indiscretion shall be despised.

Better are two left-hand gloves, than a man in the moon-light **with the wrong woman;** and a maiden alone by the seashore is as a hat without a hat pin - she breedeth wild thoughts.

As a cushion which sheddeth its feathers, as a moulting dog which leapeth upon thee, so is a woman who saith continually **WHY** desireth thou to kiss me?

To be two years a widow exceedeth a college education; and a woman **without brothers** hath a hard time.

A teasing woman is as a squeaking shoe, or as when one walketh on spilt sugar.

A wise maiden scenteth trouble afar and avoideth a scene; but the foolish damsel exclaimeth: **Don't !**

A good woman would rather be the mother of a genius than the wife of a hero.

Not by their **STRENGTH** do men prevail over women to have their way, but by **OBSTINACY** and **PERSISTENCY**. For any man in time can win any woman.

It is naught, it is naught, saith the maiden; but when he has gone his way she hurrieth to the mirror and rejoiceth at her beauty.

```

LIST
Sincerely,
5  REM *****
   * TINY LETTER WRITER PRGM. **
   *****
10  REM ***** 100-199: INTRODUCTION *
   * 200-299: DATA FORMAT ** 300-399: READ/TEST DATA *
   * 400-599: WRITE LETTER ** 600-699: MAKE LETTERFILE*
   * 1000-???: DATA LINES *****
15  TEXT : NONE :
17  HTAB 7
20  PRINT "*** TINY LETTER WRITER ***
    -----"
25  PRINT : PRINT " MENU:" : PRINT : PRINT " 1. INTRODUCTION
   ": PRINT : PRINT " 2. DATA FORMAT (INSTRUCTIONS)": PRINT
   : PRINT " 3. DATA FORMAT": PRINT : PRINT " 4. READ & T
   EST TEXT": PRINT : PRINT " 5. PRINT OUT LETTER": PRINT
27  PRINT " 6. SAVE LETTER TEXT": PRINT : PRINT " 7. END": PRI
   NT
30  PRINT " WHICH ? "; GET A$:A = VAL (A$)
35  IF A < 1 THEN END
40  ON A GOTO 100,200,245,300,400,600
99  END
100 REM *****
   * INTRODUCTION **
   *****
105 TEXT : NONE
107 HTAB 7
110 PRINT "*** TINY LETTER WRITER ***
    -----"
115 PRINT : PRINT " THIS PROGRAM IS DESIGNED TO CONTROL T
   HE EATON LRC 7000+ PRINTER, FORMATTING AND PRINTING OUT
   A MINIATURE 'TYPED' LETTER - HAVING UPPER AND LOWER-C
   ASE CHARACTERS."
120 PRINT : PRINT " THE PRINTER SHOULD BE CAPABLE OF OUT-P
   UTTING 64 CHARACTERS/LINE."
125 PRINT : PRINT " 'NORMAL' CHARACTERS WILL BE LOWER CASE%
   CAPITAL LETTERS ARE OBTAINED BY TYPING IN 'CTRL.B', JUST
   BEFORE EACH LETTER YOUWANT TO BE CAPITALIZED."
130 PRINT : PRINT " SINCE THE LETTER COULD BE QUITE LONG, T
   HIS PROGRAM WILL STORE SENTENCES IN THEFORM OF DATA LINE
   S."

```

```

135 PRINT : INPUT "(PRESS RETURN)";HOLD$: HOME
140 PRINT " FOR SIMPLICITY, THE DATA LINES WILL BEFORMATTED
FOR YOU. YOU SIMPLY COPY OVER THE DATA LINE #, THE QUOT
E MARK FOLLOW- ING IT, AND THEN TYPE IN WHATEVER FITS."
142 PRINT : INPUT "(PRESS RETURN)";HOLD$: PRINT
145 PRINT : PRINT " FOR EXAMPLE:"; PRINT : FOR T = 1 TO 200
0: NEXT T
150 Q$ = CHR$( ASC ("Q") - 1); FOR N = 4040 TO 4050 STEP 5:
PRINT "J"FN;"DATA"Q$;: FOR DASH = 1 TO 39: PRINT CHR$(
(95));: NEXT DASH: PRINT : PRINT : NEXT N
152 S = - 16336
155 DEM$(1) = "J4040DATA" + Q$ + " NOW IS THE TIME FOR ALL G
OOD MEN TO"
157 FOR T = 1 TO 3000: NEXT T
160 VTAB PEEK (37) - 8: FOR PLACE = 1 TO LEN (DEM$(1)): INVER
SE
: PRINT MID$( DEM$(1),PLACE,1);:
162 SOUND = PEEK (S) - PEEK (S); IF PLACE ) 10 THEN FOR TI
ME = 1 TO RND (1) * 100 + 1000 * (PLACE = 11): NEXT TIM
E
163 REM *****
* NOTE LOGIC STATEMENTS IN ** LINE #162, WHICH INTRO- *
* DUCE A "LONG" DELAY ON ** THE 11TH CHARACTER, AND *
* RANDOM DELAY AFTER 10TH. *****
165 NORMAL = VTAB PEEK (37) + 1 - (PLACE = 40): HTAB PEEK
(36) + 40 * (PLACE = 40): PRINT MID$( DEM$(1),PLACE,1);
166 REM ***** SIMULATING THE MOVING CUR*
* SOR WAS A BIT TRICKY. ** WHEN AN INVERSE CHARACTER*
* IS 'TYPED' AT HTAB 40, ** THE HORIZONTAL CURSOR PO-*
* SITION (PEEK(36)) RESETS ** TO 0. THE 'NORMAL' CHAR- *
167 REM ** ACTER MUST ALSO BE TYPED *
* AT HTAB 40. (THAT'S THE ** PURPOSE OF THE: *
* HTAB PEEK(36)+40*(PL=40))*****
168 IF FLAG = 1 THEN RETURN
170 NEXT PLACE: PRINT
175 DEM$(1) = "J4045DATA" + Q$ + "COME TO THE AID OF THE PART
Y, WHICH"
180 PRINT :FLAG = 1: FOR PLACE = 1 TO LEN (DEM$(1)): INVERSE
: PRINT MID$( DEM$(1),PLACE,1);: GOSUB 162: NEXT PLACE
185 DEM$(1) = "J4050DATA" + Q$ + "AUNT AGATHA CONSIDERS TO BE
A NUISANCE."
190 PRINT : PRINT :FLAG = 1: FOR PLACE = 1 TO LEN (DEM$(1))
: INVERSE : PRINT MID$( DEM$(1),PLACE,1);: GOSUB 162: NEXT
PLACE
195 PRINT : PRINT : PRINT "WOULD YOU LIKE TO SEE THI
S AGAIN (Y/N)?";: GET A$: IF A$ = "Y" THEN CLEAR : HOME
: GOTO 140
200 REM *****
* DATA FORMATTER **
*****
205 REM *****
* INTRO.TO DATA FORMATTER **
*****
207 TEXT : HOME
210 PRINT "*** INSTRUCTIONS FOR DATA FORMATTER *** -----
-----"

```

```

215 PRINT : PRINT " AS WAS MENTIONED, YOUR LETTER S TEXT W
ILL BE ENTERED ON DATA LINES .THAT IS: NUMBERED LINES BE
GINNING WITH THE WORD 'DATA', AND FOLLOWED BY A QUOTE M
ARK).".
216 Q$ = CHR$( ASC ("Q") - 1)
217 PRINT : PRINT ")))--- J7125DATA"Q$;" NOW IS THE TIME .
....:ETC.)....."
225 PRINT : PRINT " IF YOU WANT TO PUT IN A SPACE (BE- T
WEEN LINES IN YOUR LETTER), SIMPLY COPY A DATA LINE TO JU
ST BEYOND THE QUOTE. ----- : PRINT
227 INPUT " (PRESS RETURN.)";HOLD$
230 HOME : PRINT " THIS PRINTER WILL ENABLE YOU TO SELECTAN
Y OF FOUR CHARACTER SIZES, CONTROLLED BY CONTROL CHARACT
ERS ENTERED AT THE BE-GINNING OF A LINE:"; PRINT
232 PRINT " WOULD YOU LIKE A PRINTED COPY OF THIS INFORMATI
ON AT THIS TIME ? ";: GET A$: PRINT : PRINT : IF A$ = "Y
" THEN PR# 1: PRINT CHR$( 30), CHR$( 15)
233 PRINT "*** CONTROL CHARACTERS FOR PRINTER *** -----
-----": PRINT
234 PRINT "1. 64 CHAR/LINE: (NO CTRL.CHAR.NEEDED)";: PRINT : PRI
NT
"2. 40 CHAR/LINE: (CTRL. ^ (SHIFT N))";: PRINT : PRINT "3
. 32 CHAR/LINE: (CTRL. N)";: PRINT : PRINT "4. 20 CHAR/LI
NE: (CTRL.N,CTRL.^)";: PRINT
235 PRINT "5. ALL CAPITALS: (CTRL.A)";: PRINT : PRINT "6.
CANCEL ALL CAPS: (CTRL.A)";: PRINT : PRINT " (ENTER AT A
NY POINT IN DATA LINE.)";: PR# 0
244 PRINT :: INPUT " (PRESS RETURN.)";HOLD$
245 DAYTA = 1000: REM : INITIAL DATA LINE NUMBER.
250 REM *****
* FORMAT ROUTINE: **
*****
252 Q$ = CHR$( ASC ("Q") - 1);: REM : Q$=QUOTE MARK.
255 HOME : PRINT " *** LETTER TEXT ***"
260 FOR DAYTA = DAYTA TO DAYTA + 25 STEP 5: PRINT "J"Q$DAYTA;
"DATA"Q$;: FOR DESH = 1 TO 16: PRINT CHR$( 95);: NEXT
DESH: PRINT "20/";
261 FOR DESH = 20 TO 28: PRINT CHR$( 95);: NEXT DESH: PRINT
"32/";: FOR DESH = 32 TO 36: PRINT CHR$( 95);: NEXT DES
H: PRINT "40/";: FOR DESH = 40 TO 60: PRINT CHR$( 95);:
NEXT DESH: PRINT "64/"
265 PRINT : NEXT DAYTA
270 PRINT " DAYTA="Q$DAYTA;"GOTO 250 (FOR MORE LINES.)"
275 PRINT " RUN 300 (CHECK LETTER TEXT.)"
280 VTAB 1
299 END
300 REM *****
* READ & TEST DATA: ** THIS ROUTINE WILL *
* PRINT OUT THE LETTER ** DATA, WITH CAPS IN *
* FLASHING VIDEO. *****
301 REM *****
* AT THE END OF ERROR- ** CHECKING, ERROR LINES *
* WILL BE PRINTED OUT ** FOR CORRECTION. *
* *****
305 TEXT : HOME : RESTORE :LINE = 1000: DIM ERR(30)
307 PRINT "*** CHECK LETTER TEXT ROUTINE: *** -----
-----": PRINT
310 READ LINE$: IF LINE$ = "(END OF DATA)" THEN EOF = 1: PRINT
: PRINT LINE$: POKE 37,19: GOTO 345

```

```

312 IF LAST > LINE THEN LINE = LINE + 5: GOTO 310
315 Q% = CHR% ( ASC ("Q") - 1): PRINT "Q";LINE;"DATA";Q%;
320 IF LINE% = "" THEN PRINT " - SPACE -": GOTO 340
322 CAP = - 1
325 FOR CHAR = 1 TO LEN (LINE%)
330 CHAR% = MID% (LINE%,CHAR,1): IF CHAR% = CHR% (2) THEN FLA
SH
: NEXT CHAR
332 IF CHAR% = CHR% (1) THEN CAP = CAP * - 1: REM :CONTROL
CHAR. FOR ALL CAPS/TURN OFF ALL CAPS.
334 IF CAP = 1 THEN FLASH
335 PRINT CHAR%: NORMAL : NEXT CHAR: IF OUT = 1 THEN RETURN
: REM :FLAG FROM PRINT-LETTER.
337 PRINT :ERR% = "E"
340 LINE = LINE + 5
345 IF PEEK (37) > 18 THEN PRINT : INPUT ")))--" ERROR LIN
E # ? (E='ENUFF') :ERR%: IF VAL (ERR%) > 0 THEN ERR =
ERR + 1:ERR(ERR) = VAL (ERR%): GOTO 345
350 IF ERR% < "E" THEN NONE : IF EOF = 1 OR ERR% = "E" THEN
360
355 GOTO 310
360 REM *****
* ERROR-CORRECT PRINTOUT **
*****
365 PRINT " *** THE FOLLOWING LINES HAVE ERRORS *** -----"
367 IF ERR(1) = 0 THEN PRINT : HTAB 6: PRINT "(NONE INDICAT
ED.)": END
370 RESTORE :LINE = 1000:EOF = 0: FOR N = 1 TO 30:BAD = BAD +
(ERR(N) > 0): NEXT
375 READ A$: IF A$ = "(END OF DATA)" THEN EOF = 1
380 FOR CHECK = 1 TO 30: IF ERR(CHECK) = LINE THEN PRINT "Q
";LINE;"DATA";Q%;A$: PRINT :BAD = BAD - 1: IF BAD > 0 THEN
IF ERR(CHECK) = 0 THEN 375
385 NEXT CHECK: IF EOF = 0 AND BAD > 0 THEN LINE = LINE + 5:
GOTO 375
390 VTAB 1
399 END
400 REM *****
* WRITE LETTER: **
* (GET SENDER, RECEIVER ** ADDRESSES, & DATE.) *
* *****
405 TEXT : NONE
407 HTAB 5
410 PRINT " *** LETTER-WRITING ROUTINE *** -----"
412 READ A$: IF A$ = "(END OF DATA)" THEN PRINT CHR% (7): PRI
NT
" SORRY. I HAVE NO LETTER TEXT.": PRINT : PRINT " (PRO
GRAM ENDED.)": END
413 RESTORE
415 PRINT : PRINT " WE ARE NOW READY TO GET YOUR NAME AND A
DRESS, AND THE NAME AND ADDRESS OF THEPERSON YOU'RE WRI
TING TO."
420 PRINT : PRINT " REMEMBER:": PRINT : PRINT " USE "; FLASH
: PRINT "CTRL.B": NORMAL : PRINT " TO MAKE CAPITAL LETT
ERS."
422 PRINT " BEGIN EACH INPUT LINE WITH A "; FLASH : PRINT
"QUOTE.": NORMAL : PRINT
425 PRINT : FOR N = 1 TO 4

```

```

430 PRINT "YOUR NM/ADR.LINE #";N: INPUT SADR%(N): IF LEN (
SADR%(N)) > BIG THEN BIG = LEN (SADR%(N))
435 IF SADR%(N) = "" THEN 445
440 NEXT N
445 INPUT "LETTER'S DATE ?";DAYTE%: IF LEN (DAYTE%) >
BIG THEN BIG = LEN (DAYTE%)
450 PRINT : PRINT " NOW, THE NAME AND ADDRESS OF THE PER- S
ON YOU'RE WRITING TO:": PRINT
460 INPUT " RECIPIENT'S NAME ? ";RECR%(1)
465 FIRST% = LEFT% (RECR%(1),7)
467 PRINT
470 FOR N = 2 TO 4
475 PRINT FIRST%: LINE #";N: INPUT RECR%(N)
477 IF RECR%(N) = "" THEN 485
480 NEXT
485 INPUT " LETTER SALUTATION ? ";SAL%
490 IF RIGHT% (SAL%,1) ( ) = "," THEN SAL% = SAL% + ","
495 INPUT " COMPLIMENTARY CLOSE ? ";COMP%: IF RIGHT% (COMP
%,1) ( ) = "," THEN COMP% = COMP% + ","
497 INPUT " SIGNATURE ? ";SIG%
500 REM *****
* WRITE LETTER, COND. **
*****
501 DIM LINE%(12)
502 OUT = 1: TEXT : NONE : PRINT " HERE IS THE DATA I HAVE:"
: PRINT
504 FOR N = 1 TO 4:LI%(N) = SADR%(N): NEXT N:LINE%(5) = DAYT
E%: FOR N = 6 TO 9:LINE%(N) = RECR%(N - 5): NEXT N
506 LI%(10) = SAL%:LI%(11) = COMP%:LI%(12) = SIG%
508 FOR N = 1 TO 12: PRINT "LINE #";N: " ";LINE% = LINE%(N)
: GOSUB 325: PRINT
510 IF N = 4 OR N = 5 OR N > 8 THEN PRINT
512 NEXT N: PRINT : INPUT "IS THE ABOVE DATA CORRECT (Y/N):"
:ANS%: IF LEFT% (ANS%,1) = "N" THEN INPUT "TYPE LINE #
, CORRECT DATA:";LINE,LINE%(LINE): NONE : GOTO 508
513 FOR N = 1 TO 4:SADR%(N) = LINE%(N): NEXT N:DAYTE% = LINE
%(5): FOR N = 6 TO 9:RECR%(N - 5) = LINE%(N): NEXT N:SAL%
= LINE%(10):COMP% = LINE%(11):SIG% = LINE%(12)
515 PR# 1: PRINT : GOSUB 550
517 PAGE = 1
520 FOR N = 1 TO 4: IF SADR%(N) ( ) = "" THEN A$ = SADR%(N): PRI
NT
TAB# 63 - BIG): GOSUB 560: GOSUB 800
522 NEXT N
525 IF DAYTE% ( ) = "" THEN A$ = DAYTE%: PRINT TAB# 63 - BIG
): GOSUB 560: GOSUB 800
527 PRINT : GOSUB 800
530 A$ = SAL%: GOSUB 560: PRINT :COUNT = COUNT + 2
535 READ A$: IF A$ = "(END OF DATA)" THEN 580: REM END LETT
ER
537 IF A$ < " " THEN GOSUB 560: GOSUB 800
540 IF A$ = "" THEN PRINT : GOSUB 800
545 GOTO 535
550 REM *****
* PRINT PAGE DELINEATOR **
*****
551 PRINT CHR% (15); CHR% (31): FOR DASH = 1 TO 63: PRINT
"-": NEXT DASH: PRINT : PRINT :COUNT = 0: RETURN
560 REM *****
* PRINT TEXT ** (UPPER,LOWER CASE.) *
*****

```

```

561 K = 32:CAP = - 1: PRINT CHR$ (31); CHR$ (15);: FOR CHAR
    = 1 TO LEN (A$):CHAR$ = MID$ (A$,CHAR,1):ASKI = ASC
    (CHAR$): IF ASKI < 65 THEN K = 0: IF ASKI = 2 THEN NEXT
    CHAR
562 IF ASKI = 1 THEN CAP = CAP * - 1: NEXT CHAR
563 PRINT CHR$ (ASKI + K - K * (CAP = 1)):K = 32
564 NEXT CHAR
568 PRINT : RETURN
570 REM *****
    * DRAW PAGE DELINEATOR, ** PRINT NEXT-PAGE HDNG., *
    * AND CONTINUE. ** *
    *****
572 GOSUB 550:PAGE = PAGE + 1
574 A$ = REC$(1): GOSUB 560:A$ = CHR$ (2) + "PAGE " + STR$
    (PAGE): GOSUB 560
575 A$ = DAYTE$: GOSUB 560
576 PRINT :COUNT = 4
578 GOTO 535
580 REM *****
    * CLOSE LETTER. ** *
    *****
582 PRINT : PRINT TAB(40);A$ = COMP$: GOSUB 560: PRINT : PRI
    NT
    TAB(40);A$ = SIG$: GOSUB 560:COUNT = COUNT + 4
584 FOR N = 1 TO 30 - COUNT: PRINT : NEXT N
586 GOSUB 550
588 GOTO 700: REM : PRINT ADDRESS LABEL.
590 END
600 REM *****
    * MAKE LETTER FILE ** *
    *****
602 RESTORE
605 TEXT : HOME :D$ = CHR$ (4)
606 HTAB 7
607 PRINT "### SAVE LETTER TEXT ###"
    -----
608 RESTORE : READ A$: IF A$ = "(END OF DATA)" THEN PRINT CHR
    $
    (7);A$: PRINT : PRINT " (SORRY. THERE'S NO TEXT TO SAVE
    .)": END
610 VTAB 5: INPUT "WHAT NAME FOR YOUR DATA ? ":FI$: IF FI$ =
    "" THEN END
615 FI$ = "MICRO:" + FI$
620 HOME : POKE 33,33
625 VTAB 5: PRINT "- ONE MOMENT, PLEASE.-"
630 PRINT D$"OPEN "FI$
635 PRINT D$"DELETE "FI$
640 PRINT D$"OPEN "FI$
645 PRINT D$"WRITE "FI$
647 PRINT "DEL 1000,5000"
650 LIST 1000 - 5000
655 PRINT D$"CLOSE"
660 TEXT : HOME
665 PRINT D$"CATALOG"
670 PRINT CHR$ (7): PRINT "### DONE! ###"
699 END
700 REM *****
    * ADDRESS LABEL PRINTER ** *
    *****
705 REM *****
    * THE LETTER HAS JUST BEEN ** PRINTED, AND HERE WE MAKE$
    * AN ADDRESS LABEL, USING ** SNDR$(N) & RECR$(N). *
    *****

```

```

710 IF RECR$(2) = "" THEN END : REM : NO ADDRESS GIVEN, SO
    QUIT.
715 FOR N = 1 TO 4
720 IF SNDR$(N) < ) "" THEN PRINT SNDR$(N)
725 NEXT N
730 FOR N = 1 TO 4: PRINT : NEXT N: REM : SKIP DOWN ON LABEL
    .
735 RECR$(1) = "TO: " + RECR$(1)
740 FOR N = 1 TO 4:BIG(N) = LEN (RECR$(N)): NEXT N: REM : GE
    T LENGTH OF ADDRESS LINES.
742 BIG = 0
745 FOR N = 1 TO 4: IF BIG(N) > BIG THEN BIG = BIG(N): REM :
    FIND LONGEST ADDRESS LINE FOR GOOD LEFT-FORMATTING OF RE
    CEIVER'S ADDRESS.
750 NEXT N: FOR N = 1 TO 4: IF BIG = BIG(N) THEN 755
751 NEXT N
755 PRINT CHR$ (30): REM :SHIFT TO LARGER CHARACTERS.
760 HTAB 36 - BIG: PRINT RECR$(1)
765 FOR N = 2 TO 4
770 IF RECR$(N) < ) "" THEN HTAB 40 - BIG: PRINT RECR$(N)
775 NEXT N
780 PRINT CHR$ (31): REM : BACK TO 64 CHAR/LINE.
785 FOR N = 1 TO 4: PRINT : NEXT N
790 FOR DASH = 1 TO 64: PRINT "-": NEXT DASH: PRINT
795 PR# 0: REM : VOILA !!
799 END
800 REM *****
    * LETTER LINE-COUNTER AND ** PAGE-LENGTH CONTROLLER. *
    *****
810 COUNT = COUNT + 1: REM : COUNT A PRINTED LINE.
820 IF COUNT > 28 THEN PRINT : GOTO 570: REM : PRINT NEW PA
    GE HEADINGS; BEGIN NEW PAGE.
830 RETURN
999 REM *****
    * DATA LINES FOLLOW ** *
    *****
24999 DATA (END OF DATA)
25000 REM *****
    * HOWIE MITCHELL ** 7923 SW. 55TH PLACE *
    * GAINESVILLE, FLA. 32601 ** JULY, 1980 *
    *****

```

MAX M. METHUSELA
 662 TERRACOTTA TRAIL
 ENTROPY, KANSAS
 12345

TO: WASHINGTON APPLE PI
 P.O. BOX 34511
 WASHINGTON D.C.
 20034

Speaking Up

Take That, Word Processor!

By Peggy Eastman

Special to the Journal

You sit before a silent television screen which has a typewriter keyboard attached to it. You take a small, thin disk gingerly between thumb and forefinger and insert it into the machine. A message, "Close disk door," flashes on the machine.

Obediently, you close the flap covering the disk slot. You stare at the empty screen, trying frantically to remember what you wanted to say. Tentatively, you press the return key. A little white blip called a cursor jumps around on the screen like an

animated flea.

The machine is a word processor. In a quiet but frighteningly ubiquitous way, word processing machines are taking over law offices, newspapers, consulting firms, government agencies. Press the "print" button and they spew forth appellant briefs, murder and fire stories, government reports on subjects as diverse as energy conservation and biomedical research.

The word processor long since has shelved that old manual

typewriter that helped you think when you banged away on it, and it's sent the sleek IBM Selectric back to the design room.

Word processors are entering private homes, demanding an honored place beside the piano. A consultant who works out of his Kensington home says his word processor saves him days of time and secretarial effort. "I can give a draft of a report to my secretary, see a typed version in a day and make changes in another day," he says. "It used to take a week."

Word processors can store reams of material on tiny record-like disks, allow you to change whole chunks of your manuscript without retyping, give you an instant replay with the press of a finger. They also can destroy a report if you press the wrong button — making your precious words vanish without a trace.

There is something suspicious and grossly un-literary about the very name "word processor." How does it process words? Does it chew them? Does it digest them? Does it launder them? Does it dismember or scramble them? What does it do with Webster's finest inside that box?

As if to counteract a semantically cannibalistic image, manufacturers have given futuristic brand names like "Lextron" and "Vydec" to their word processing machines. But you still have the uneasy feeling that you're working with a Cuisinart of the King's English, and that what you get back is somehow tainted. I can't help thinking that — like a sinister preview of an upcoming Robert Heinlein novel — the word processor will somehow corrupt my latest magnum opus just for the heck of it. I can almost hear it chortling metallically. (*She wants to say "Studies showing the world will self-destruct in 1990 are still inconclusive?" Click. Click. I'll change it to "Studies show the world will self-destruct in 1990 conclusively."*)

With its insidious talents, a word processor gone wild could: jeopardize relations between the United States and the People's Republic of China; screw up Social Security payments; start an auto workers' strike; short McDonald's on a shipment of hamburgers.

A word processor is, of course, a computer. As such, it takes its place in that shadowy realm called "artificial intelligence," domain of robots and real-time control centers. In his book "The Digital Villain," Robert M. Baer, Ph.D., a University of California mathematician, notes drily: "The computer is the most beautiful and useful of man's inventions. The manufacturers' brochures say so."

Beloved by scientists to whom software means more than lounge pillows, deplored by liberal arts graduates imbued with Wordsworth and Keats, the computer-word processor has entered a world heretofore dominated by literary greats. Can you see Lord Byron, chest heaving, pouring out his heart's darkest secrets to a silent box? Can you picture Shakespeare composing "Romeo and Juliet" on a television screen?

The trick, of course, is not to capitulate to the mechanistic commands of the little box. Let it know immediately who's boss. You can always have the last word: just unplug it. As caustic critic Lewis Mumford put it, "No automatic system can be intelligently run by automatons — or by people who dare not assert human intuition, human autonomy, human purpose."

If you really want to get into all this in a scientific and non-emotional way, you could pore through back issues of something called the "International Journal of Man-Machine Studies," or, for those with slicker tastes, "Computerworld." But I, for one, would rather stick with Humpty Dumpty, who said with egg-like simplicity: "When I use a word, it means just what I choose it to mean — neither more nor less."

Take that, word processor.

DOUBLE-SIZE GRAPHICS FOR THE SILENTYPE

by BRUCE F. FIELD

Let me start by saying that I think the Silentype printer is potentially one of the most versatile printers on the market. Most conventional printers available for the APPLE control the printer operation via microprocessors in the printer with the printer functions coded in ROM. These instructions that tell the printer how to operate are hidden from the user and in fact are unalterable by the user. You can only do as much as the manufacturer will let you do.

The Silentype on the other hand uses the microprocessor in the APPLE to control the printer functions with the instructions contained on a ROM on the printer interface card. What this means to the user is that the hardware functions of the printer (i.e. moving the printhead, printing a dot, and advancing the paper) are all controllable directly by the APPLE! We are no longer tied to the software provided by the manufacturer and are free to do anything we want within the mechanical constraints of the printer mechanism. WOOPIE!

Our only problem now is to find out how to control the functions of the printer. Well APPLE "thoughtfully" didn't provide any information on the ROM software or the printer hardware to allow this. Letters to APPLE didn't produce any results either. Enter Sandy Greenfarb. Sandy also was trying to get information on the printer and obtained through IAC (International APPLE Core) a 14 page writeup on how to move the head and print a dot, and also a little on how the printer character set is defined and used. Just what I wanted. The result is the machine language routine presented here for printing either Hi-res graphics screen at double size.

Using the routine is very simple, it uses the standard Silentype parameters to control the printing. You simply BLOAD the DBL GRAPHICS routine into memory (it resides from \$300 to \$3C7 and so won't

conflict with Applesoft, Integer Basic, or DOS). You can generate or load your hi-resolution picture either before or after loading the DBL GRAPHICS routine. Then initialize the Silentype as you would normally (PR#1 from Basic if your printer is in slot 1) and setup the printer parameters as desired. Instead of typing a control-Q to print the Hi-res screen, from Basic you CALL 768 to activate this routine. The CALL can be placed inside a program or typed directly from the keyboard. If you are typing the CALL from the keyboard and don't want those words to appear on the printer, first reset the APPLE output to the screen (PR#0); the printer does not have to be "turned on" to print the graphics.

As it is written the routine is slightly slot dependant and assumes your printer interface card is plugged into slot 1. It is very easy to change this if you use a different slot. The offending byte is at \$30F hex or 783 decimal. This should be changed to \$C0 + printer slot (hex) or 192 + printer slot (decimal). Once you do this you can save it back to disk and not worry about it.

For those of you who are old hands at the APPLE and know how the Hi-res graphics screen works, you can go on to the next article; otherwise I'm going to try to describe how the routine works. First let me air a pet peeve. Most double size graphics printing routines I have seen (as for the Paper Tiger) chop off the sides of the screen when printed full double size because the picture is a little too wide for the printer. This comes about because the characters on the APPLE screen are 5 dots wide with a 2 dot space between them. Forty characters on the screen means 280 dot positions in the horizontal direction which is exactly the resolution of the Hi-res screen. Most dot matrix printers however print characters 5 dots wide with a one dot space between them. For 80 column wide printers this means 480 dot positions, not quite enough to print the graphics double sized. The obvious solution (at least to me) is to print the graphics screen sideways on the printer! The vertical resolution of the Hi-res screen is 192, times 2 is 384 which easily fits on a printer with 480 dot positions. So, guess which way my routine prints the screen?

Now that I have that off my chest, let's dig a little deeper into how the routine works. A rather poor map of the Hi-res screen is shown on page 21 of the new APPLE Reference Manual. (It would be handy to get it out now for reference. If you don't have one our club Treasurer has them for sale.) This map shows an array of 40 boxes in the horizontal direction and 24 in the vertical direction. In the horizontal direction each box controls 7 dots on the graphics screen. As explained on page 19 of the manual bits 0 through 6 control the seven dots with the least significant bit (bit 0) controlling the left-most dot of the group. In the vertical direction things are a little messier because the boxes are not organized in an orderly fashion. Each box represents 8 bytes in memory with each byte corresponding to a different vertical position on the screen and the bits within the bytes controlling the 7 horizontal dots as explained above.

Now lets see if I can straighten all this out with an example. Suppose we want to know the address of the byte that controls the 87th dot from the left on the 23rd line from the top of the screen. Twenty-three divided by 8 is 2 with a remainder of 7, so we go down to the third vertical box from the top which has an address of 8448 (decimal) and we want the 7th line in that box which means we add 6144 to 8448. (We go to the third box because box 1 controls lines 1 - 8, box 2 lines 9 - 16 etc. and we use the diagram in the lower right hand corner to find the number 6144 to be added to the base address.) Then we move over to the right to add in the contribution of the horizontal position. Eighty-seven divided by 7 is 12 with a remainder of 3. So we go over to the 13th box (which is labelled 12 because the numbering starts with 0) which adds 12 to the address. Now we total up these numbers to get 14604 which is the address of the byte we want. That wasn't so hard was it? By the way to get addresses for Hi-res page 2 we simply add 8192 to the address calculated above.

Since we now know how the screen works, let's print it. We are going to print it sideways so we start at the lower left-hand corner, get a byte, print it, and work our way up; then move to the right one row and repeat. The way I'm

doubling the graphics size is to print a cluster of four dots to represent one dot on the screen. This doesn't improve the resolution; it just makes the picture bigger.

The Silentype can print a line of 7 dots at one time. This is okay for regular size printing but for double size we are going to have to make two passes to print what corresponds to one column of bytes on the screen. Each byte contains 7 bits of screen data which we are going to print as 14 dots wide. The first pass starts at the label NXROW1 in the assembly code and gets the byte from the screen and looks up in a table what should be printed based on the 4 least significant bits in the byte. This is printed twice and the program loops around to get the byte from the row above. When this is completed the paper is advanced and the printer head returned to the left margin. The same process is repeated except this time bits 4 - 6 of the byte are used. This completes the printing of the first seven dot columns of the screen; repeating this 40 times finishes the graphics dump.

There are just two other things to take care of and actually these come first in the program. APPLE warns against running the printer and the disk at the same time as this overloads the APPLE power supply. To prevent this there is a wait routine at the beginning of the graphics dump to allow the disk to timeout and stop running from any previous disk operation. Also, since the graphics dump doesn't respect the right margin it is possible to set the left margin so that the print head is forced against the right side of the printer (bad, bad). Thus the left margin is checked and reset to a maximum of 18. Actually a left margin of 10 is my favorite as it centers the graphics on the page.



ASM

```

1000 *****
1010 *
1020 *      DBL GRAPHICS
1030 *
1040 *      PRINTS EITHER HIRES PAGE DOUBLE SIZE ON
1050 *      THE SILENTYPE PRINTER
1060 *
1070 *      PRINTER MUST BE INITIALIZED
1080 *      WITH CORRECT HIRES PAGE, INVERSE MODE,
1090 *      LEFT MARGIN, AND INTENSITY
1100 *
1110 *      B. F. FIELD   AUGUST 1980
1120 *
1130 *****
1140 *
1150 *
C100- 1160 SLOT   .EQ $C100      ADDR OF PRINTER
1170 *
0G1A- 1180 HBASL  .EQ 26        FOR SCREEN ADDRESS
001B- 1190 HBASH  .EQ 27        COMPUTATION
002A- 1200 XO     .EQ $2A       SCRATCH COLUMN VALUE
002B- 1210 YO     .EQ $2B       SCRATCH ROW VALUE
CF11- 1220 LFMG   .EQ $CF11     LEFT PRINTER MARGIN
CF13- 1230 HPAG   .EQ $CF13     HIRES SCREEN
CF2B- 1240 DOTS   .EQ $CF2B     DOT IMAGE TO BE PRINTED
CFFF- 1250 ROMS   .EQ $CFFF     CO-RES ROM SWITCH
CF14- 1260 INVRS  .EQ $CF14     PRINTER INVERSE FLAG
1270 *
CD02- 1280 SFTLFT .EQ $CD02     MOVE HEAD TO LEFT MARGIN
CB0B- 1290 PRNT   .EQ $CB0B     PRINT DOT IMAGE ROUTINE
CCAB- 1300 FEED   .EQ $CCAB     ADVANCE PAPER
FCAS- 1310 WAIT   .EQ $FCAS     MONITOR WAIT ROUTINE
1320 *
1330      .OR $300
1335      .TA $800
1340 *
1350 *      WAIT FOR 2 SECONDS TO MAKE SURE
1360 *      DISK ISN'T RUNNING
1370 *
0300- A2 0D 1380 DOUBLE LDX #13   COUNTER FOR WAIT LOOP
0302- A9 FF 1390      LDA #255
0304- 20 A8 FC 1400 LOOP JSR WAIT  MONITOR WAIT ROUTINE
0307- CA     1410      DEX        DEC LOOP COUNTER
0308- D0 FA 1420      BNE LOOP
1430 *
030A- AD FF CF 1440      LDA ROMS   SWITCH OUT CO-RES ROMS
030D- AD 00 C1 1450      LDA SLOT   SWITCH ON PRINTER ROMS
1460 *
1470 *      IF LEFT MARGIN >18 THERE ISN'T ROOM FOR GRAPHICS
1480 *      RESET MARGIN TO MAX OF 18 IF NECESSARY
1490 *
0310- A9 12 1500      LDA #18

```

0312-	CD 11 CF	1510	CMP LFMG	
0315-	B0 03	1520	BCS OK	
0317-	8D 11 CF	1530	STA LFMG	CORRECT LEFT MARGIN
031A-	20 02 CD	1540 OK	JSR SFTLFT	MOVE TO LEFT MARGIN
		1550 *		
		1560 *	GET EACH COLUMN OF BYTES ON THE SCREEN	
		1570 *	AND PRINT THEM TWICE	
		1580 *		
031D-	A0 00	1590	LDY #0	
031F-	84 2A	1600 NXCOL	STY XO	XO IS COLUMN COUNTER
		1610 *		
0321-	A9 BF	1620	LDA #191	INIT ROW
0323-	85 2B	1630	STA YO	
		1640 *		
0325-	20 76 03	1650 NXROW1	JSR GETPT	GET SCREEN VALUE
0328-	29 0F	1660	AND #\$0F	MASK UPPER BITS
032A-	AA	1670	TAX	
032B-	BD A8 03	1680	LDA TBL1,X	LOOKUP DOT IMAGE IN TABLE
032E-	4D 14 CF	1690	EOR INVRS	EXCLUSIVE-OR WITH INVERSE
0331-	8D 2B CF	1700	STA DOTS	
0334-	20 0B CB	1710	JSR PRNT	PRINT DOT IMAGE
0337-	20 0B CB	1720	JSR PRNT	TWICE FOR DOUBLE SIZE
		1730 *		
033A-	A5 2B	1740	LDA YO	TEST IF ROW=0
033C-	F0 05	1750	BEQ NEXT	
033E-	C6 2B	1760	DEC YO	DECREMENT AND CONTINUE
0340-	4C 25 03	1770	JMP NXROW1	
0343-	20 9F 03	1780 NEXT	JSR CRLF	DO RETURN AND LINE FEED
		1790 *		
		1800 *	SEVEN BITS IN EACH SCREEN BYTE REQUIRES	
		1810 *	PRINTING 14 DOTS ON THE PRINTER	
		1820 *	THUS HEAD MAKES TWO PASSES	
		1830 *		
		1840 *	THIS IS SECOND PASS	
		1850 *		
0346-	A9 BF	1860	LDA #191	
0348-	85 2B	1870	STA YO	
034A-	20 76 03	1880 NXROW2	JSR GETPT	GET SCREEN BYTE
034D-	29 78	1890	AND #\$78	MASK LOWER BITS THIS TIME
034F-	4A	1900	LSR	SHIFT FOR TABLE OFFSET
0350-	4A	1910	LSR	
0351-	4A	1920	LSR	
0352-	AA	1930	TAX	
0353-	BD B8 03	1940	LDA TBL2,X	GET DOT IMAGE
0356-	4D 14 CF	1950	EOR INVRS	EXCLUSIVE-OR WITH INVERSE
0359-	8D 2B CF	1960	STA DOTS	
035C-	20 0B CB	1970	JSR PRNT	PRINT DOT IMAGE
035F-	20 0B CB	1980	JSR PRNT	TWICE FOR DOUBLE SIZE
		1990 *		
0362-	A5 2B	2000	LDA YO	TEST IF ROW=0
0364-	F0 05	2010	BEQ NEXT2	
0366-	C6 2B	2020	DEC YO	DECREMENT AND CONTINUE
0368-	4C 4A 03	2030	JMP NXROW2	
036B-	20 9F 03	2040 NEXT2	JSR CRLF	

	2050 *	
	2060 *	ONE SCREEN COLUMN COMPLETED
	2070 *	INCREM COLUMN AND REPEAT
	2080 *	
036E- A4 2A	2090	LDY XO
0370- C8	2100	INY
0371- C0 28	2110	CPY #40 SEE IF ALL 40 COLUMNS DONE
0373- 90 AA	2120	BCC NXCOL NO, CONTINUE
0375- 60	2130	RTS ALL DONE, EXIT HERE
	2140 *	
	2150 *	COMPUTE MEMORY ADDRESS FOR HIRES SCREEN POSITION
	2160 *	Y VALUE IN YO, X VALUE IN XO
	2170 *	LEAVE WITH SCREEN VALUE IN ACC
	2180 *	
0376- A5 2B	2190 GETPT	LDA YO
0378- A4 2A	2200	LDY XO
037A- 48	2210	PHA
037B- 29 C0	2220	AND #5C0
037D- 85 1A	2230	STA HBASL
037F- 4A	2240	LSR
0380- 4A	2250	LSR
0381- 05 1A	2260	ORA HBASL
0383- 85 1A	2270	STA HBASL
0385- 68	2280	PLA
0386- 85 1B	2290	STA HBASH
0388- 0A	2300	ASL
0389- 0A	2310	ASL
038A- 0A	2320	ASL
038B- 26 1B	2330	ROL HBASH
038D- 0A	2340	ASL
038E- 26 1B	2350	ROL HBASH
0390- 0A	2360	ASL
0391- 66 1A	2370	ROR HBASL
0393- A5 1B	2380	LDA HBASH
0395- 29 1F	2390	AND #51F
0397- 0D 13 CF	2400	ORA HPAG
039A- 85 1B	2410	STA HBASH
039C- B1 1A	2420	LDA (HBASL),Y GET SCREEN VALUE
039E- 60	2430	RTS
	2440 *	
	2450 *	DO LINE FEED OF 4 DOT POSITIONS FIRST
	2460 *	THEN DO CARRIAGE RETURN
	2470 *	THIS ORDER REDUCES LINE STAGGER
	2480 *	
039F- A9 04	2490 CRLF	LDA #4
03A1- 20 AB CC	2500	JSR FEED ADVANCE PAPER
03A4- 20 02 CD	2510	JSR SFTLFT MOVE TO LEFT MARGIN
03A7- 60	2520	RTS
	2530 *	
	2540 *	TABLES FOR CONVERSION OF SCREEN BYTE
	2550 *	TO DOT IMAGE FOR SILENTYPE
	2560 *	
03AB- 00 60 18		
03AB- 78 06 66		

03AE- 1E 7E	2570 TBL1	.HS 0060187806661E7E
03B0- 01 61 19		
03B3- 79 07 67		
03B6- 1F 7F	2580	.HS 0161197907671F7F
	2590 *	
03B8- 00 40 30		
03BB- 70 0C 4C		
03BE- 3C 7C	2600 TBL2	.HS 004030700C4C3C7C
03C0- 03 43 33		
03C3- 73 0F 4F		
03C6- 3F 7F	2610	.HS 034333730F4F3F7F
	2620 *	
	2630	.EN

SYMBOL TABLE

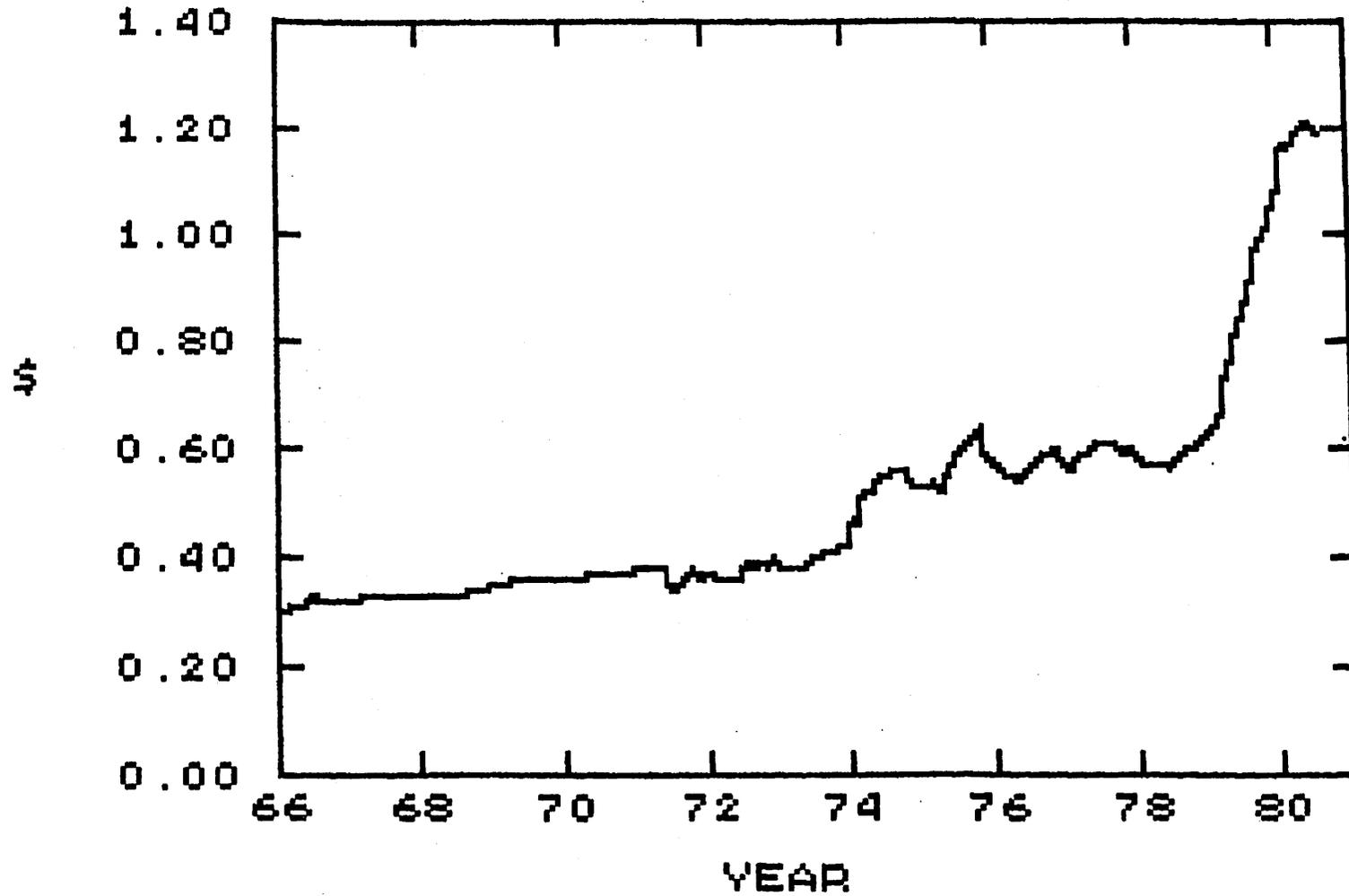
\$300.3C7

039F- CRLF	0300- A2 0D A9 FF 20 A8 FC CA
CF2B- DOTS	0308- D0 FA AD FF CF AD 00 C1
0300- DOUBLE	0310- A9 12 CD 11 CF B0 03 8D
CCAB- FEED	0318- 11 CF 20 02 CD A0 00 84
0376- GETPT	0320- 2A A9 BF 85 2B 20 76 03
001B- HBASH	0328- 29 0F AA BD A8 03 4D 14
001A- HBASL	0330- CF 8D 2B CF 20 0B CB 20
CF13- HPAG	0338- 0B CB A5 2B F0 05 C6 2B
CF14- INVR5	0340- 4C 25 03 20 9F 03 A9 BF
CF11- LFMG	0348- 85 2B 20 76 03 29 78 4A
0304- LOOP	0350- 4A 4A AA BD B8 03 4D 14
0343- NEXT	0358- CF 8D 2B CF 20 0B CB 20
036B- NEXT2	0360- 0B CB A5 2B F0 05 C6 2B
031F- NXCOL	0368- 4C 4A 03 20 9F 03 A4 2A
0325- NXROW1	0370- C8 C0 28 90 AA 60 A5 2B
034A- NXROW2	0378- A4 2A 48 29 C0 85 1A 4A
031A- OK	0380- 4A 05 1A 85 1A 68 85 1B
CB0B- PRNT	0388- 0A 0A 0A 26 1B 0A 26 1B
CFFF- ROMS	0390- 0A 66 1A A5 1B 29 1F 0D
CD02- SFTLFT	0398- 13 CF 85 1B B1 1A 60 A9
C100- SLOT	03A0- 04 20 AB CC 20 02 CD 60
03A8- TBL1	03A8- 00 60 18 78 06 66 1E 7E
03B8- TBL2	03B0- 01 61 19 79 07 67 1F 7F
FCA8- WAIT	03B8- 00 40 30 70 0C 4C 3C 7C
002A- XO	03C0- 03 43 33 73 0F 4F 3F 7F
002B- YO	:

:

contd.

RETAIL COST OF GASOLINE



GROUP PURCHASE PLAN

In order to more effectively plan for club group purchases we would like to have some idea of your needs. Please check the items of interest and either bring the form to the next meeting or mail it to:

Howard Lefkowitz
11508 Colt Terrace
Silver Spring, Maryland 20902

When we receive sufficient interest in an item, where the quantities qualify for group purchase, we will be in touch with you. A one-third deposit is needed for all items before we can place an order.

Hardware

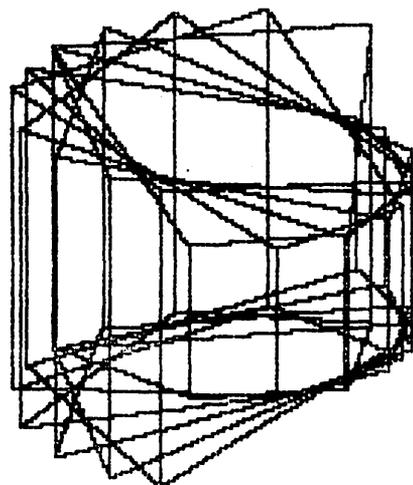
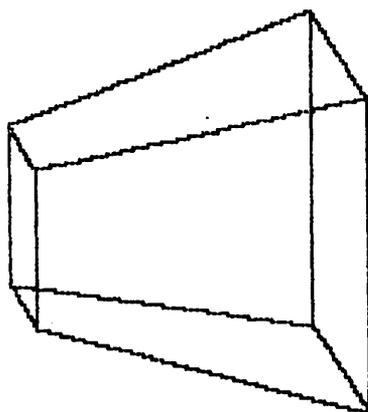
Floppy Disk: _____ ; Disk with Controller _____
Language System _____ ; DOS 3.3 _____
Applesoft Card _____ ; Integer Basic Card _____
Parallel Card _____ ; Centronics Card _____
Serial Card _____ ; Communications Card _____
Prototype Card _____ ; _____
New Apple 80 Column, Lower Case Card _____
Silentype Thermal Printer _____
IDS Paper Tiger Printer 440 _____, 460 _____
Hayes Micro Modem _____
Z-80 Microsoft System _____
ALF Music Synthesizer _____
Clock/Calendar Card _____
Graphics Tablet _____
CRT Monitors Sanyo 9" _____ Sanyo 12" _____
SOROC IQ 12 _____
Other _____

Software

Apple FORTRAN _____ ; PILOT _____
Apple Writer _____ ; Apple PLOT _____
Apple Post _____ ; DOS 3.3 Tool Kit _____
Visicalc _____ ; Desk Top Plan _____
CCA Data Mgmt. _____ ; Dakin-5 3.3 Utilities _____
Other _____

the **geometric projector**^c

A unique program. The two game paddles are used to control rotations of three dimensional geometric objects (a cube, pyramid, prism, etc, built into the program) through a full 360 degrees about body co-ordinate axes. It is thus possible to rotate the object to any orientation. The program can be toggled to modes in which successive images of the three dimensional object are superposed as the object undergoes arbitrary rotation. In this manner, an infinite variety of patterns of great symmetry, complexity, and beauty can be constructed. These images may be saved on disk and recalled or printed, as below. The program also contains sets of line-segments which may be rotated three dimensionally for the same purpose. In all, the program has 24 operating modes, 12 of which allow the construction of superposed images. Complete documentation included. Requires 48K RAM or ROM Applesoft*.



A classic lander for the Apple!* High resolution color graphics is used throughout. As the program begins, the LEM orbits high above the cratered surface of the moon. Using both game paddles, you must control craft orientation and rocket thrust in real time. When the LEM reaches the horizon, the scene automatically switches to a closeup of the mountainous terrain and landing pad to which the LEM must be brought. If the landing is successful, two astronauts emerge from the LEM and plant a ceremonial flag. After each landing, crash or safe, a status report is generated. There are four levels of difficulty, including a retrograde orbit. Complete instructions. 48K RAM or ROM Applesoft.*

a **decent moon lander**^c

* Apple and Applesoft are trademarks of Apple Computer, Inc.

© Copyright 1980 - R. Streitmatter**

THE GEOMETRIC PROJECTOR -- \$19.95
A DECENT MOON LANDER -- \$19.95

(on disk only)

To order: Send check or money order to

Robert Streitmatter
1703 GREENTREE CT.
CROFTON, MD 21114

**A Washington Apple Pi member.

Washington Apple Pi Membership Application

Dues for membership in Washington Apple Pi are \$12.00 per year. The dues year runs from January 1 to December 31. Members joining after February should pay at the rate of \$1.00 per month. Both new members and renewing members are asked to fill in the following application as completely as possible. Information gained here will help the club serve you better.

If you have any questions please call Genevieve Urban. (301 229-3458) or Bob Peck, Treasurer (301 468-2305).

Thank you.

NAME _____

ADDRESS _____

CITY, STATE, ZIP _____

TELEPHONE NUMBERS: HOME () _____ WORK () _____

PLEASE LIST HARDWARE YOU OWN: APPLE II _____ APPLE II PLUS _____ 48K _____ 32K _____ 24K _____ 16K _____
DISK II _____ SERIAL _____ /PARALLEL _____ I/O AI/O _____ RS-232 _____

OTHER:(e.g. MODEM, PRINTER) _____

OCCUPATION _____

I WOULD LIKE TO WRITE ARTICLES FOR THE NEWSLETTER (Y/N) _____

MY AREAS OF INTEREST ARE: _____

IF YOU WOULD LIKE TO ASSIST ON A COMMITTEE OR SPECIAL INTEREST GROUP, PLEASE SPECIFY _____

I AUTHORIZE THE RELEASE OF MY NAME _____, ADDRESS _____ AND TELEPHONE _____ TO OTHER MEMBERS.

(NOTE: Club policy prohibits releasing members' names and addresses unless you release that information by checking the appropriate areas above.)

PLEASE ENCLOSE PAYMENT WITH THIS APPLICATION.

MAKE CHECK PAYABLE TO "WASHINGTON APPLE PI"

MAIL TO: WASHINGTON APPLE PI Attn. Treasurer
PO BOX 34511
WASHINGTON, DC 20034

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.

Also available for purchase by members at a discount price is the new APPLE II REFERENCE MANUAL (replaces the Red Reference Manual). The price of this manual is \$17.00. You may pick it up at a meeting or have it mailed to you at no extra charge.

- | | Amount |
|---|--------|
| 1. New APPLE II REFERENCE MANUAL - \$17.00 each | ----- |
| 2. 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 28 Planetfinder | () |
| Volume 5 Games III | () | Volume 180 Dungeon Designer | () |
| Volume 6 Games IV | () | Volume 181 Beginner's Cave | () |
| Volume 7 Games V | () | *Volume 182 Lair of Minotaur | () |
| Volume 8 Utilities III | () | *Volume 183 Cave of the Mind | () |
| Volume 9 Educational I | () | *Volume 184 Zyphur Riverventure | () |
| Volume 10 Math/Science | () | *Volume 185 Castle of Doom | () |
| Volume 11 Graphics I | () | *Volume 186 Death Star | () |
| Volume 12 Games VI | () | *Volume 187 Devil's Tomb | () |
| Volume 13 Games | () | | |
| Volume 14 IAC Utilities IV | () | | |
| Volume 15 Games VII | () | *Vol. 181 required with these | |
| Volume 16 Utilities V | () | disks. | |
| Volume 17 Graphics II | () | | |
| Volume 18 Educational II | () | | |
| Volume 19 Communications | () | | |
| Volume 20 Music | () | | |
| Volume 21 Apple Orchard | () | | |

 TOTAL ORDER = \$ -----

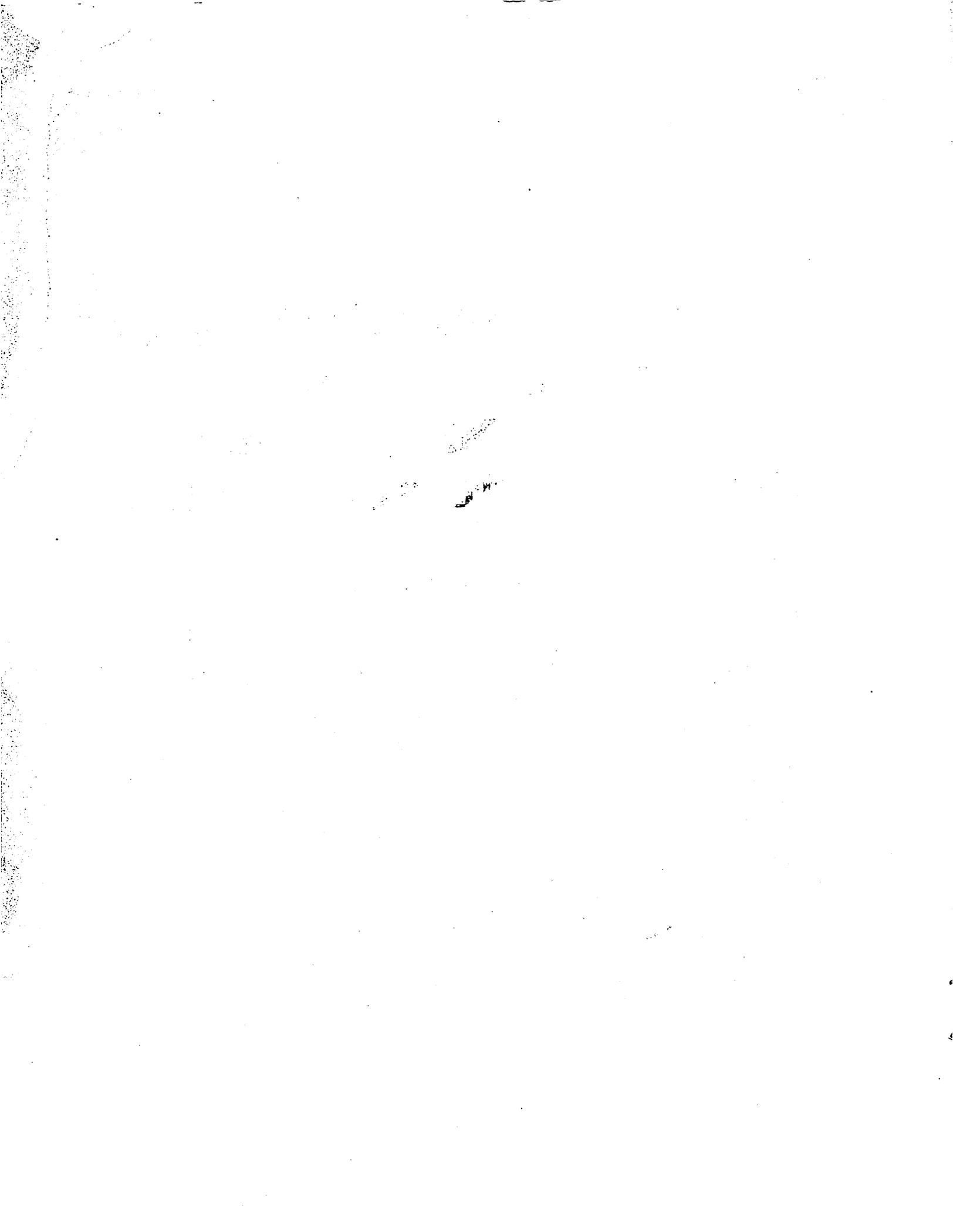
Check here if you want these shipped---

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, Knowlegdeable Atmosphere**
- Family Run Business**

Stop in and say hello to Gary, Ken or Betty

M-W	11:00 to 6:00 PM
TH-F	11:00 to 9:00 PM
Sat	11:00 to 6:00 PM

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



First Class

