

\$1.50

Washington Apple Pi



Volume 3

June 1981

Number 6

Highlights

TEXT FILE MANIPULATOR USING
APPLEWRITER

EPSON MX-80 PRINTER
INTERFACE PROGRAM

In This Issue

Page

MEMBERSHIP INFORMATION, EVENT QUEUE, EDITORIAL, MINUTES.	1
SIG-NEWS, NOTICES.	2
A NEW SIG FOR KIDS DAVID STERN	2
PRESIDENT'S CORNER DAVID MORGANSTEIN	4
THE \$10 JOYSTICK JAY THAL	6
QUESTIONS, QUESTIONS, QUESTIONS. MARK CROSBY	8
SNEAKY TEXT. ANDREW G O'BRIEN	9
APPLE II EUROPLUS OR THOUGHTS OF MEMBER 436. DUNCAN LANGFORD	11
EDUCATIONAL II: A REVIEW OF LIBRARY DISK 18 ROZ STERN	12
BENEATH APPLE DOS: A REVIEW WILLIAM SCHULTHEIS	14
DOS 3.3 AND DEMO'S FOR US CHEAPSKATES. BOB ANDERSON	15
TEXT FILE MANIPULATOR USING APPLEWRITER. BRUCE F FIELD	16
FORTUNE'S ANALYSIS OF THE PERSONAL COMPUTER INDUSTRY WALTON FRANCIS	18
FREE SECTORS BOOT PROGRAM. BRUCE F FIELD	19
EPSON MX-80 PRINTER INTERFACE PROGRAM. HERB KLONSER	20
IAC BULLETIN, MAY.	34

ComputerLand[®]

We know small computers.



 **apple[®] computer**
Sales and Service

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

OFFICERS & STAFF EDITORIAL

President -David Morganstein (301) 972-4263
 Vice President -Mark Crosby (202) 488-1980
 Treasurer -Dana Schwartz (301) 725-6281
 Secretary -Jesse Wagstaff (301) 937-4215
 Members-at-Large -John Moon ABBS Sysop
 -Nancy Philipp (301) 924-2354
 -Rich Wasserstrom (703) 893-9147
 Immed. Past Pres. -Bernard Urban (301) 229-3458
 Editor -Bernard Urban (above)
 Associate Editor -Rich Wasserstrom (above)
 -Genevieve Urban (301) 229-3458
 Librarians:
 Disk Mail Order -Sarah Lavilla (301) 926-6355
 Disk Pick-up -Bill Bowie (301) 924-3455
 Group Purchases -Dick Elkins (301) 384-9297
 Membership -Bob Peck (301) 621-2719
 Volunteer Coord. -Boris Levine (301) 229-5730
 SIG Chairmen:
 ASMSIG -Jim Rose (301) 730-2230
 EDSIG -Chuck Philipp (301) 924-2354
 NEWSIG -Paul Hoffman (301) 831-7433
 Pascal (PIG) -Tom Woteki (202) 547-0984
 SIGAMES -Al Gass (703) 371-3560
 SIG/DISABLED -Curt Robbins ABBS WAP428
 8805 Barnsley Court
 Laurel, MD 20811

Washington Apple Pi
 P. O. Box 34511
 Washington, D.C. 20034
 (301) 621-2719

ABBS (301) 983-9317

Apple user groups may reprint without prior permission any portion of the contents herein, provided proper author, title and publication credits are given.

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

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

EVENT QUEUE

Washington Apple Pi meets on the 4th Saturday of each month at 9:30 AM, at George Washington University, usually in Building C, on G Street at 23rd Street, NW. (To be sure of the exact location call the club phone or ABBS during the week of the meeting.) The June meeting is on the 27th and the July meeting on the 25th.

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

NOVAPPLE meets on the 2nd Saturday of the month at 1:00 PM at Kings Park Library on Burke Lake Road in Fairfax County; and on the 4th Thursday of the month at 7:30 PM at Computerland of Tysons Corner.

Yippee! It's vacation time! But... that means fewer articles for the next two issues. Or does it? All you procrastinators out there, drop your gin and tonics long enough to finish off that article you started last February, and send it in to our Editor-in-Residence Rich Wasserstrom. Rich, Steve Hadley, Dana Schwartz and others have generously offered to fill in for the Urbans while we are away. Help them out folks, please.

We've been getting quite diverse materials lately (and lots more ads). I like it that way. Do you? Let us know. *

CLASSIFIEDS

FOR SALE: Texas Instruments terminal, receive only, thermal paper. Model 742(RO), used with APPLE II. Cost new \$1085, selling for \$600. Grace Hayes, T.N. Dupuy and Assoc., 560-6427.

MINUTES

EXECUTIVE BOARD MEETING

There was no business meeting of the Executive Board during May.

GENERAL MONTHLY MEETING

The Washington Apple Pi meeting of May 30, 1981 was called to order at 9:35 AM by the President with approximately 350 persons in attendance.

The business portion of the meeting concerned the results of the election and the introduction of the new Officers.

The main presentation featured Steve Wozniak, co-inventor of the APPLE. Steve gave a short talk about the philosophy of Apple, Inc. He mentioned his belief that information should be freely exchanged between Apple and its users, and that this was an important factor in the success of the APPLE II. After the talk Steve answered questions from the floor covering a variety of subjects. The talk and discussion were very informative and enjoyed by all. Washington Apple Pi presented the Woz a certificate entitling him to lifetime membership in our club (or until IBM puts him out of business) and an apple pie.

SIG-NEWS

SIGAMES is the special interest group of computer hobbyists interested in using their APPLES for entertainment. The main meeting of this group is held at a location announced at and following the Washington Apple Pi monthly meeting.

June's program will consist entirely of young speakers and their impressions of APPLE games. Quite a few short presentations on the more popular games will be given. This should be a very enjoyable and interesting session. Steve and David Stern have been coordinating this program.

John Alden will be chairing July's meeting on hi-resolution adventure games.

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

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

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

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

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

⊗

A NEW SIG FOR KIDS

by David Stern

A new Special Interest Group for kids will meet during the regular Washington Apple Pi meeting. Those interested should meet outside the main meeting room at 9:30. This month's meeting will consist of organizing the group and choosing a group name. What we will be doing in future meetings will be determined during this month's meeting. The founders of this new group are:

- David Stern 881-2543
- Brian Van Flandern 363-3860
- Steve Green 460-8647

⊗

NOTICES

CONGRATULATIONS TO THE NEW OFFICERS *****

The May election resulted in the following new slate of Officers:

- President - David Morganstein
- Vice-President - Mark Crosby
- Secretary - Jesse Wagstaff
- Treasurer - Dana Schwartz
- Members-at-Large - John Moon
- Nancy Philipp
- Rich Wasserstrom

Paul Sand also won a Member-at-Large, but is moving away from the Washington area. (We'll miss you, Paul.)

We congratulate all our new officers, wish them well and pledge our support.

A THANKS TO OUR BALLOT COUNTER *****

We would like to thank Boris Levine (and his wife) for their long hours of work in tallying our election ballots. Boris wrote a program for the APPLE to tally the votes, and entered all the data from the approximate 250 respondents. So, if you get the job next year, your task will be easier, thanks to Boris' program.

APPLE FOR PRESCHOOLERS *****

Our member from England, Duncan Langford, writes to us requesting information on APPLE programs for four-year olds (see his article on another subject elsewhere in this issue). He would like to hear from anyone who has used their APPLE to aid a preschooler in learning to use a computer and for other educational applications. So, if you have any information to share with Duncan, please write him. I'm sure you will enjoy corresponding with him. Address: 20 Lesley Avenue, Canterbury, Kent, ENGLAND.

OUR GEOGRAPHIC DEMOGRAPHY *****

As you may have noticed with the May issue, we are now mailing our newsletters using a bulk mail permit. This entails sorting our mail in zip code order. As a result of this, we learned some facts about the geographic distribution of our members. About 90 of our members are scattered across the U.S. in states beyond the Maryland and Virginia area, with New York (26) and California (19) leading. Other states with more than five include Pennsylvania, North Carolina, Massachusetts, Florida, and New Jersey. Maryland (outside the DC Metropolitan area) has 57 and Virginia 13. Of the 57 Maryland members 31 are in the Annapolis, Columbia and Baltimore area. Within the Metropolitan area, the Maryland suburbs rank highest, Northern Virginia second and D.C third. We have five members with FPO or APO addresses, one member in England, two in Australia, and three in Canada. (We have had recent inquiries for applications from as far away as Thailand and Denmark.)

⊗

PRESIDENT'S CORNER by

David Morganstein

The first message I would like to convey, besides my thanks for your support, is an expression of the appreciation that I believe we all feel for Bernie & Gena Urban. These two people have eaten, breathed and slept Apple Pi since they helped found our group. They have supported us with their time and their energies and are among those responsible for its growth from a handful to almost 750 (by the time you read this we may have reached 800!). Thank you, Urbans...

I also want to add that I look forward to working with my esteemed opponent, the honorable Dr. Wo. Tom Woteki has and continues to contribute his time and talents to furthering the understanding and uses of the Apple. It is because of columns such as his that our "newsletter" has moved into the front ranks of information sources on the Apple and become a magazine with a valuable message. Tom's efforts at organizing meeting presentations have encouraged greater attendance and provided us with valuable knowledge supplied from users who have "been there already" and can save the rest of us a lot of heartache in the learning.

Now, on to the future. In my opinion, we have been facing a challenging problem, faced no doubt by other "volunteer" organizations which appear at the right time, at the right place. With each new member, the ability to provide the same service becomes more difficult. We are being told by G.W.U. that we will have to pay for our meeting space. The costs of producing, printing and delivering the newsletter grow. The number of library disks sold increases monthly. Each month some 40 or 50 new members join our ranks, many of whom are brand new to computing and electronic gizmos (and who don't know one end of a Peek from a Poke). Washington Apple Pi is dedicated to helping these new members become intimately acquainted with their computing machine and learn its many capabilities.

The question remains...How to maintain and increase the level of these services...sounds like a theme we read about daily. In our case though, I think the answer lies in you the members. The Pi needs volunteers. There are enough of us that it should be possible to spread the work around so as to not overburden anyone. A six-months stint at some responsibility is tolerable to most of us. Below I mention some of the immediate needs and ask for some members to step forward to help with them.

ARRANGEMENTS: We need to insure a Saturday morning meeting place. This may require finding another location other than G.W.U. Requires notifying the ABBS and the phone answering service in

advance of the meeting. (Contact me if you are interested.)

NEWSLETTER DELIVERY: Collect a team of deliverers for the MD, DC, and VA areas to deliver newsletters to the computer stores. We need one person to organize the team and see to it that they receive the required number of newsletters from Gena. We need team members to perform the deliveries. (Contact Gena if you want to be the organizer or a deliverer.)

ADVERTISING: One way to keep our treasury solvent is by advertising: in the newsletter, in the "Best of Apple Pi" and in the Membership Roster. We need someone to help develop an ad mail-out to both local stores and national manufacturers of Apple products. (Contact Mark Crosby or me if interested.)

ANSWERING QUESTIONS: Many of our new users would like to feel comfortable calling a more experienced owner to ask questions. Right now Mark is the only official "answerer". If we had a list of a dozen or more members, each willing to field questions on their specialties, the new members could get some mid-month help at "crisis" moments. If you feel competent and don't mind a few calls during the month contact Mark. We'll put your name and number in the newsletter. We can use "experts" in: Applesoft, Integer, Graphics, Hardware, Printers, Pascal, Pilot, etc., etc.

ORIENTATION PACKAGE: We have talked about a need for a new members orientation package. The package would contain hints and helps about the Apple, such as a glossary, and facts about the WAP and its services. We need volunteers to put such a package together. If you would like to help, call me.

LIBRARY: We always need people willing to spend time making disk copies. We also need help in selling disks at the meeting, by rotating this function no one will feel overburdened with the responsibility. (Contact Bill Bowie if you have time to contribute.)

SPEAKERS: Our Saturday morning meetings offer a learning opportunity. If you have some valuable knowledge and are willing to share it, please offer yourself as a speaker so that others may learn from your efforts. (Check with Tom Woteki if you have a topic you want to hear or a talk to give.)

ARTICLES: The newsletter is read by almost 1200 individuals. Writing up any of the following will provide enormous benefits to many: a program you have written; hardware & software reviews of products purchased; modifications to hardware & software you have made to improve them; tutorials on pet subjects;

or unusual applications you think may be of interest to others. (Speak with Bernie Urban about your subject.)

SOFTWARE CONTRIBUTIONS: The library thrives on donations. Please don't tell me you gave at the office... (Call Bill Bowie. Remember, we'll swap you a library disk for your donation.)

If you can contribute in any of the above ways, or have something else you can add, please contact any of the board members so that we can harness your energies!!!

COMPUTER ACE

P.O. BOX 728 / ADELPHI MD. 20783
PHONE: (301)587-7459

- APPLE w/16K.....\$1059
- APPLE w/32K.....\$1104
- APPLE w/48K.....\$1149
- DISK w/CONT.....\$ 535
- DISK only.....\$ 465
- PASCAL SYSTEM.....\$ 415
- INTEGER CARD.....\$ 150
- APPLESOFT CARD.....\$ 150
- SILENTYPE PRINTER..\$ 515

New!! A Finance Data Base System

... use double entry without being an accountant

- Balance Sheet
- Budgeting
- Save at Tax Time with Detailed Records
- Efficient Storage/Retrieval
- Human Engineered

Comprehensive manual in handsome 3-ring binder, (Requires DOS 3.3 and 48K)

The ACCOUNTANT FINANCE DATA BASE SYSTEM

Send check for \$89.95 or VISA/MC (#, EXP. date) to: DECISION SUPPORT SOFTWARE 1438 Ironwood Drive McLean, VA 22101 (703)241-8316

GROW WITH THE LEADER

Join our team! Be a part of the fastest growing distributor of microcomputer products on the East Coast. Our high volume operation has created personnel needs in several key areas. If you think you would be interested in growing with the leader, we want to hear from you. Send us your resume, and include a letter about your interests.

ATTN: R. BEALLE
MICRO DISTRIBUTORS, INC.
11794 Parklawn Drive
Rockville, MD 20852

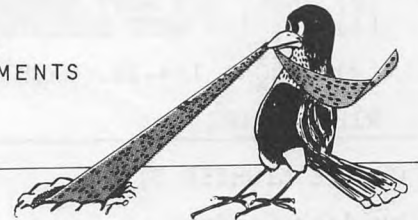
WE ARE NOT A RETAIL OUTLET.

PLEASE NO CALLS

ROBINS OFFERS A FULL LINE OF SUPPLIES FOR YOUR MICROCOMPUTER AT VERY REASONABLE PRICES

And convenient hours:
Tuesday-Friday: 9-5
Monday: 9-7
Saturday: 10-2

- ACOUSTICAL ENCLOSURES
- ANTI-STATIC MATS
- CASSETTES
- COMPUTER RIBBONS
- COPIER ROLLS
- DATA TERMINALS
- DISK CARTRIDGES
- DISK PACKS
- FACSIMILE PAPER
- FLOPPY DISKS
- FORMS TRACTORS
- MAGNETIC CARDS
- MEDIA STORAGE
- MODEMS
- NEC THIMBLES
- PERFORATOR TAPE
- PINFEED CARDS
- PINFEED PAPER
- PINFEED LABELS
- PRINT WHEELS
- RECYCLED RIBBONS
- SELECTRIC TYPE ELEMENTS
- SURGE SUPPRESSORS
- TELETYPE ROLLS
- THERMAL PAPER
- TYPEWRITER RIBBONS
- WORK STATIONS
- W-P RIBBONS



Roberts Information Services, Inc.
ROBINS 8306 Hilltop Rd. Fairfax, VA
22031 (in Merrifield) (703)560-5900

THE \$10 JOYSTICK

by Jay Thai

Game and graphics enthusiasts who know how to solder can save themselves \$40 or more with a few hours work.

Take a 7/64 bit and drill through both the joystick and the project case. Using the holes in the case to find the center, drill a one-inch hole. Mount the joystick with 5-40 screws through the holes.

A four conductor wire is sufficient to wire in the joystick; an added wire is necessary for each switch to be mounted in the case. The switch requires a 1/2-inch mounting hole.

Mount according to the schematic. Switch 0 is primary. Some programs requires Switch 1 also, and the rare ones (like missile defense) require three. However, if your future plans include a second joystick, be satisfied with one switch.

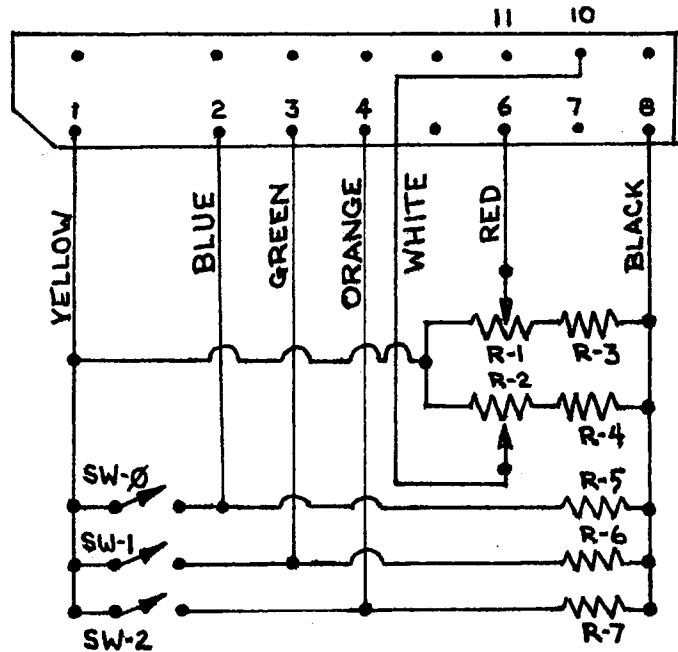
If two joysticks are used, a four conductor lead can be wired through to Case 1 (or five if a switch is included). The +5v and ground are wired to Pins 1 and 8, respectively; the center taps on Joystick #2 to Pins 7 and 11.

Vertical and horizontal tracking might be off by 90 degrees. This can be cured by moving the joystick 1/4 turn. Two annoyances: the stick is threaded 4 mm, and it may not reach the corners of a hi-res screen (file the block if necessary).

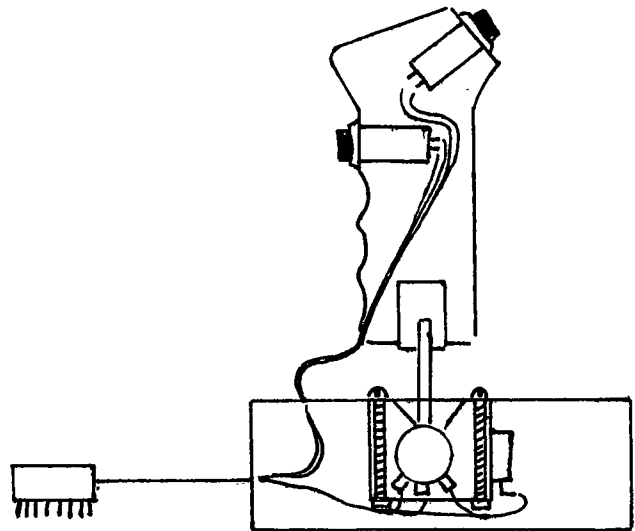
Design enhancements could include mounting the switches in a separate case to overcome right/left hand conflicts or, as I've done, build them into a pistol grip mounted on the joystick. The cases can be attached to a lapboard for use. The pistol grip I made was laminated balsa strips, rough cut to include wiring and switches, then carved to fit my hand. My index finger actuates Switch 0, the thumb Switch 1.

Parts List:

- JIM PAK/JAMECO JOYSTICK, 100K
- LINEAR TAPER POTS \$4.95
- RADIO SHACK DIP HEADER #276-1980 1.39
- RADIO SHACK PROJECT CASE #270-222 1.99
- RADIO SHACK MOMENTARY SWITCH #275-618 1.49
- (2) 360K, 1/4 WATT RESISTORS
- (1) 1K, 1/4 WATT RESISTOR (FOR EACH SWITCH)
- (4) 5-40 1 1/4-IN. MACHINE SCREWS AND NUTS
- RIBBON WIRE



- R1,2 - JOYSTICK, 100K LINEAR TAPER
- R3,4 - 360K, 1/4 WATT, 5%
- R5,6,7 - 1K, 1/4 WATT, 5%
- SW0,1,2 - MOMENTARY SPST



ON THE LIGHTER SIDE

by Jack Goodman

Recently my wife mentioned to her elderly stepmother that I had bought an APPLE. We just got a letter from her, which included the following:

"Tell Jack I am happy for him. I don't know what an APPLE computer does but they really clap loud when one is granted as a prize on TV."

There's only one place to buy apples.

AT FREDERICK COMPUTER PRODUCTS !!!

As an Authorized Apple Dealer we offer a full selection of both Apple Hardware and Software at low, low prices.

Also, as an Authorized Service Center we provide fast and efficient service on all Apple products.



FREDERICK COMPUTER PRODUCTS, INC.

Microcomputer Systems And Peripherals

5726 INDUSTRY LANE-FREDERICK, MD.- (301)694-8884

 **apple computer**
® Authorized Dealer

QUESTIONS, QUESTIONS, QUESTIONS

by Mark L. Crosby

Q. How do I turn on my printer while on the ABBS? I would also like to be able to turn it off and on without signing off and on the ABBS repeatedly.

A. The following will work with the Micromodem and the Communications Card:

First type Control-A, Control-X. This will disconnect the modem but should not disconnect the ABBS. Then type PR#s <CR> (where s is the slot # of the printer) to turn on the printer. Type PR#0 <CR> to turn off the printer. Then type IN#s <CR> where s is the slot # of your modem or comm card. Type Control-A, Control-F to reconnect the modem.

The way I do it is to hit the RESET button (on a Plus) which disconnects everything. The printer and/or the modem can then be reconnected as necessary. The ABBS will still be waiting for you to type something just where you left off.

Q. I am driving my TTY through a Communications Card. How can I set the card for 110 baud?

A. When the card is first initialized (IN#s where 's' = slot #) it comes up at 300 baud. To change to 110 baud, type Control-A, Control-1.

Q. I am writing a program (Applesoft) that must first delete several files on a diskette. The files may or may not be on the disk so ON ERR GOTO must be used to prevent a "FILE NOT FOUND" error. The subroutine is called with a GOSUB. It works beautifully until it hits the RETURN statement whereupon it gives a "RETURN WITHOUT GOSUB" error. Any help would be appreciated.

A. If an error is encountered during a FOR...NEXT loop or GOSUB you'll have to use a machine language patch that corrects problems with the "RETURN STACK". This "stack" is a list of addresses that tell Applesoft where to go to complete the "RETURN" from a GOSUB or a "NEXT" from a FOR. This is a "last-in, first-out" type of list. When an error is encountered and ON ERR is active, the last return address (if any) is removed from this stack and the address of the error is put on top of the stack. Before doing a RETURN or NEXT, the address of the error must be removed from the stack and the return address put back on the stack. The machine language subroutine below will do this. After each error perform a CALL 768. Then when you are ready, a RETURN or NEXT will function properly.

```
*0300- 68 A8 68 A6 DF 9A 48 98
0308- 48 60
```

Q. How can I input strings in Applesoft that contain commas, colons and quotes? I've tried using the GET statement which seems to work but the back-arrow key doesn't function properly or maybe I'm not doing it right. Is there a machine-language routine that would work?

A. Yes. As a matter of fact R.M. Mottola has written a short subroutine that works beautifully - even with disk INPUT commands (yes - now you can put commas, etc. on your disk files too!). This program was just published in Nibble Vol 2 No. 2 page 87. The machine language is located at \$300 (decimal 768) and is used with Applesoft. It is used in this manner:

```
10 PRINT "STRING: ";
20 CALL 768,A$
30 PRINT A$
```

Note in line 20 the CALL takes the place of INPUT A\$. The machine language follows:

```
0300- 20 BE DE 20 E3 DF A6 11
0308- 30 05 A2 A3 4C 12 D4 85
0310- 85 84 86 A5 B8 A4 B9 85
0318- 87 84 88 20 2C D5 AD 00
0320- 02 C9 03 D0 03 4C 63 D8
0328- A9 00 85 0D 85 0E A9 00
0330- A0 02 20 ED E3 20 3D E7
0338- 20 7B DA A5 87 A4 88 85
0340- B8 84 B9 20 B7 00 60 00
```

There is a slight change from the original article that corrects an error when the input string matches the first 2 letters of an Applesoft Reserved Word. Any string name may be used - including string arrays.

(Reprinted with permission of Nibble magazine, PO Box 325, Lincoln, MA 01773 - all rights reserved.)

Q. I am looking for a modification for the Epson printer to permit friction feed.

A. Check where you purchased the printer first since a friction feed is now available - or try Gary Cockrell, 1617 Sage Brush Ct., S.E., Albuquerque, NM 87123 (503) 298-2068. \$30 for the friction feed and \$10 for a roll paper holder.

Q. I want to use the shape table that is used in Apple-Plot. Can you tell me how to do it?

A. The shape table is located at decimal 17408. If you need to save the table just do a BSAVE TABLE,A17408,L1548. Then POKE 232,0 and POKE 233,68 (0 + 68 * 256 = 17408). To print a letter:

DRAW ASC (A\$) - 31 AT X,Y where A\$ is the letter and X and Y are the Hi-Res screen coordinates. Be careful that the ASC value minus 31 is between 1 and 66 inclusive or you will get an "ILLEGAL QUANTITY" error. The table should be relocatable if you need to move it.

- Q. I want to be able to save text from the ABBS session on disk and edit it later. Any suggestions?
- A. Although I don't have this software package, I have heard that ASCII Express works nicely storing text and putting it on the disk. Data Capture 4.0 works similarly. &

SNEAKY TEXT by Andrew G. O'Brien

One of the drawbacks in programming the APPLE in assembly language is the awkward manner in which you have to output text. The usual way to do it is to set up a loop and output one character at a time, as in the following example:

```

COUT EQU $FDED
*
LDY #$7
TOP LDA MESSAG-1,Y
JSR COUT
DEY
BNE TOP
BRK
MESSAG ASC "! OLLEH"

```

This is nice and simple as long as you know how long the message will be, and as long as you are not upset about entering the message backwards. In the above example, the last letter of the message will be displayed first, hence the necessity for storing the message backwards.

A more esthetically pleasing version of this routine would be:

```

COUT EQU $FDED
*
LDY #$00
TOP LDA MESSAG,Y
JSR COUT
INY
CPY #$7
BNE TOP
BRK
MESSAG ASC "HELLO !"

```

Now we can store the message as it will appear on the screen.

Another way is to use a special character as a delimiter, such as a \$, as in:

```

COUT EQU $FDED
*
LDY #$00
TOP LDA MESSAG,Y
CMP "$"
BEQ DONE
JSR COUT
INY
JMP TOP
DONE BRK
MESSAG ASC "HELLO !$"

```

This message affords us the luxury of not having to know how long the string is. The only thing we must do here is to ensure that the delimiter, "\$", is at the end of the string.

Well, I guess that just about covers all of the ways that you could possibly mention about outputting text in 6502 assembly language. Right? Wrong!

In my incessant disassembling of assorted machine code, I have come across a rather sneaky method of outputting text. While the theory involved is not my own, the code is. I am not condoning or condemning this method, but I am presenting it for your evaluation. However, if you want your code to look like jibberish to a disassembler, I highly recommend this method.

Consider the following example:

```

JSR SNEAK
ASC "HELLO !"
BRK
LDA #23

```

What subroutine SNEAK does is output the ASCII string until it sees a zero (BRK). It then returns to the line containing the LDA.

Listing 1 contains subroutine SNEAK. The routine itself is short and relatively simple. It does four tasks:

1. It gets the return address from the stack.
2. It adds 1 to this address to get the address of the first character to be displayed and displays it.
3. It increments the address and continues to display characters until the value at the address is zero.
4. When it finds a zero, it loads the stack with the current address and does a return.

Step 4 has the effect of returning to the address immediately following the address which contained the zero. In our example, the line contained an LDA.

One advantage of having the ASCII messages embedded in the source code is that you know immediately from looking at the listing what the string will be. The reader does not have to flip back and forth to see what message will be displayed where. One disadvantage to this method is that it takes up a lot of room. Unless you are writing a pretty big program, it really doesn't pay to use it.

LISTING 1:

```

*****
3      *                               SNEAK.ASM
4      *   Subroutine to print a string on the current
5      *                               output device
*****
7      *
8      *
9      TMP1      EQU   $E      Temporary variable 1
10     TMP2      EQU   $F      Temporary variable 2
11     OUTCHR    EQU   $FDED   COUT monitor routine
12     *
13     *                               ORG   $7000
14     *                               OBJ   $7000
15     *
*****
17     *   Get the return address from the stack
*****
19     *
7000: 68        20     GETADD      PLA
7001: 85 0E     21             STA   TMP1
7003: 68        22             PLA
7004: 85 0F     23             STA   TMP2
24     *
*****
26     *   Start outputting characters from the location
27     *   immediately following the call to this
28     *   subroutine until you hit a location with a
29     *   zero in it.
*****
31     *
7006: A0 00     32             LDY   #$0
7008: E6 0E     33     INCADD      INC   TMP1
700A: D0 02     34             BNE   START
700C: E6 0F     35             INC   TMP2
700E: B1 0E     36     START      LDA   (TMP1),Y
7010: C9 00     37             CMP   #$0
7012: F0 05     38             BEQ   PUTADD
7014: 20 ED FD  39             JSR   OUTCHR
7017: F0 EF     40             BEQ   INCADD
41     *
*****
43     *   Put the new return address on the stack and
44     *   return.
*****
46     *
7019: A5 0F     47     PUTADD      LDA   TMP2
701B: 48        48             PHA
701C: A5 0E     49             LDA   TMP1
701E: 48        50             PHA
51     *
701F: 60        52             RTS

```

--- END ASSEMBLY ---

TOTAL ERRORS: \$00

CODE STARTS AT \$7000
 ENDS AT \$701F
 RANGE = \$0020

APPLE II EUROPLUS OR THOUGHTS OF MEMBER #436 by Duncan Langford

I'm not sure just who holds the current record for the furthest member of Washington Apple Pi from Washington; at around four thousand miles, I'm certainly not able to drop into every meeting, much as I'd like to. From reading Pi, and other U.S. magazines, I've an idea of the U.S. APPLE scene. I thought it might be interesting to compare things with the English experience!

Personal computers took a long while to become established here. The very first time most people outside the computer business heard of the existence of such things - apart from the home-brewed kits - was in January, 1978, when national newspapers featured a tiny story about the 'amazing' new PET. I'd been interested in playing with computers since discovering 'Lunar Lander' on a PDP-10 at college. This looked too good to miss.

That PET took around three months to arrive. I had to pay the full purchase price before Commodore would take the order. For an original 8K MK1, this was about \$1600!

I rapidly discovered that there weren't many people around who knew what a microcomputer was, let alone who had one. This, combined with the notoriously bad documentation and rapidly obsolete ROMs of the PET, led to two years of experiments. A few months after I had my PET, the first APPLES began to arrive and I began to realize what I was missing!

Eventually, I was faced with the choice of either upgrading my PET for the third time, or starting over with a new system. I really wanted colour. I had tried 'high density' add-on graphics and wanted the real thing; and I was fed up with CBM, anyway.

When I started to look seriously at alternatives, the price difference started to really shock me...an ordinary 16K APPLE Europlus (basically, an APPLE II Plus set up for 240v) costs \$1400, plus 15% tax; a single disk and controller around \$700, plus tax; and so on. I wanted an APPLE, but...

Why not import a U.S. APPLE and run 110v from 240v via a transformer? Problems with warranty, conversion and dealer support, that's why. I was torn, though. It would sure save a lot of money. Eventually, I bargained for a Europlus 48K from a London dealer, my nearest. (It took almost six hours for the round trip.) I ordered my disk drive from San Diego, airfreight. It arrived promptly, works perfectly, saved \$250.

Software is even more crazily expensive. For a start, I thought the APPLE Software Bank would give me something to run, and to take apart to learn from. I understand

that it is - or was - free. Take in a disk to your dealer. I bought the five full disks from a New York store for \$25. Here, they're \$150.

That's the real difference. I can't drop into a local store, as there isn't one. I call up YOUR local store, instead and order through a charge card. It works fine.

APPLE users here seem to be mostly business and professional. I guess that's a reflection of the cost. My interest is mainly recreational, so I miss out on home-grown software. People take the APPLE so seriously here - I can't really be the only person who uses an APPLE for fun?

Home-grown computer magazines (worth buying a copy, to make you appreciate the U.S. models) make a big play of the TRS-80 and PET, and now the ZX-80; APPLE programs are rare. There is, just, a UK APPLE club. I'm a member, but it's too far to visit. I needed help - my APPLE is an order of magnitude above the old PET, and I was determined not to make the same mistakes as I had before, trying on my own. I telephoned Creative Computing, asking them to suggest a couple of the best APPLE clubs.

That's how I came to join Washington Apple Pi!

CONVERT BASIC FILES TO PASCAL

ANNOUNCING PUP I

A User Friendly Package
of Pascal 1.1 Utilities
Designed Specifically for the APPLE

FEATURES INCLUDE:

- Basic files (Applesoft, Integer, Binary) to Pascal conversion
- Automatic system date-set at boot
- Formatted printer listings of Pascal files
- 40/80 column board formatting
- Upper/lower case
- User modifiable Basic tokens
- Online user assistance
- Error checking with easily understood error messages
- Support of Pascal wildcards
- Single/multi drive systems
- Full user documentation
- Easy to use by Pascal Beginners

Available July, 1981. \$29.95

GRYPHON MICROPRODUCTS
P. O. BOX 6543
SILVER SPRING, MD 20906

EDUCATIONAL II: A REVIEW OF LIBRARY DISK 18 by Roz Stern

I have provided a brief review of each educational program on this disk. The programs are reviewed with regard to educational validity (whether it teaches anything or not), positive or negative reinforcement provisions, apparent maintenance of student interest, and grade level suitability. As with recent disk reviews, a general rating scale is as follows:

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

FLASH CARD - Provides visual drill by rapidly flashing sequence of related pairs. These pairs can be alphanumeric, such as math facts or word relationships. Program is educationally valid for rote drill and suitable for grades 4-6. Flash Card does not appear to provide positive reinforcement or maintain sustained interest. (If facts are flashed too rapidly, interrupt program execution and set SPEED = 100). Rating: ***

SPELLING BEE - Tests your spelling of a list of sixteen input words, either timed or untimed. This program has a nice lo-res display which provides an overall score and an oversized spelling of the word as it is keyed in. It requires two people, one reading the words to be spelled, the other keying in the words. You have two chances to spell the word correctly, otherwise you are in an assisted letter-by-letter mode. Spelling Bee rewards the student with graphics, prints "hurray", and plays a tune. This program is educationally valid, provides positive reinforcement, and maintains student interest. Rating: ****

PRESCHOOL LETTER RECOGNITION - Tests for matching of individual letters and words at 8 different levels of complexity. Letter or word is placed on the screen in text and child simply keys in same letter or word. Educationally valid for 3-4 year-olds if they can manipulate a keyboard (and you want them fooling around with your APPLE!). Rating: *

SHAPES - Is a visual memory test for shape and color patterns. The adult controls the machine, while the child draws reproductions of flashed designs on paper. This program is not educationally valid because lo-res graphics are not accurate enough for this type of program. Designed for grades K-3, it would tend to frustrate rather than test. Rating: *

GUESS MY WORD - Is a computerized form of the "Word Mastermind Game" in text form. It teaches deductive reasoning in a spelling game format. The student must deduce the correct letters and then sequence them correctly to advance to another challenge. This program is

educationally valid for grades 4-8 and appears to sustain interest. Rating: ***

SECRET MESSAGE - Consists of three menu-selected games: Spelling, Secret Message, and Math.

SPELLING - The APPLE flashes the last two letters of a three-letter word. Student is challenged to key in as many words as possible. Although educationally valid for teaching spelling, this program uses no graphics and does not appear to sustain student interest. Designed for grades 1-3. Rating: **

SECRET MESSAGE - Is a text oriented game which requires the student to solve simple math problems. As each problem is solved, APPLE prints another part of a word. Provides drill in simple math, but does not sustain interest. Designed for grades 3-4. Rating: *

MATH - The student inputs his grade level and is provided with arithmetic problems of appropriate complexity. If enough problems are successfully answered, the student is promoted to the next grade (level of complexity) with very positive lo-res graphic display and music. Provides good arithmetic practice and appears to sustain very high interest in grades 3-8. Rating: ****

MUSICAL MATH TEACHER - This is a text math quiz with positive and negative musical sounds plus a scoring summary after 20 problems. Does not correct for errors, rather keeps repeating same problem as often as ten times and keeps printing statements like "You should know better". Requires a third grade reading capability. This educationally valid program appears to sustain interest for grades 3-4. Rating: **

APPLE EDUC. PACK - Provides three menu-selected programs: Hammurabi, Hurkle and Multiply.

HAMMURABI - Is a simple version of "King". The student must rule a country without starving the population. It requires arithmetic estimation and/or computation and teaches a limited concept of economic geography. It provides both positive and negative reinforcement, sustains interest, and is suitable for grades 4-6. Rating: ***

HURKLE - Find the Hurkle is a grid game which uses directional terms (north, south, east, west). It provides practice in beginning map reading skills. This program has a major limitation in that it does not display current position. It may sustain

student interest and may be useful for grades 3-4. Rating: **

MULTIPLY - Is a math quiz with negative reinforcement mechanisms such as a bad buzzer sound and "Wrong answer you dummy, try again" statements. While the math problems are OK, the program is destructive, not constructive. Rating: *

PHOTOGRAPHY CAI - Is a tutorial designed to provide a short course on the effects of color filters in photography and light in general. It requires a color TV or monitor since it uses lo-res graphics as a teaching mechanism. This program provides definitions and sequential information on color filtration and follows with a scored quiz. The quiz does not provide correct answers; students must return to the tutorial and try again. Educationally valid and suitable for junior high and up. Rating: ****

ING ENDINGS - This program is designed to teach linguistic rules for adding "ing" to a root word. It has a tutorial format in text with lo-res graphics for positive reinforcement. It repeats previous words

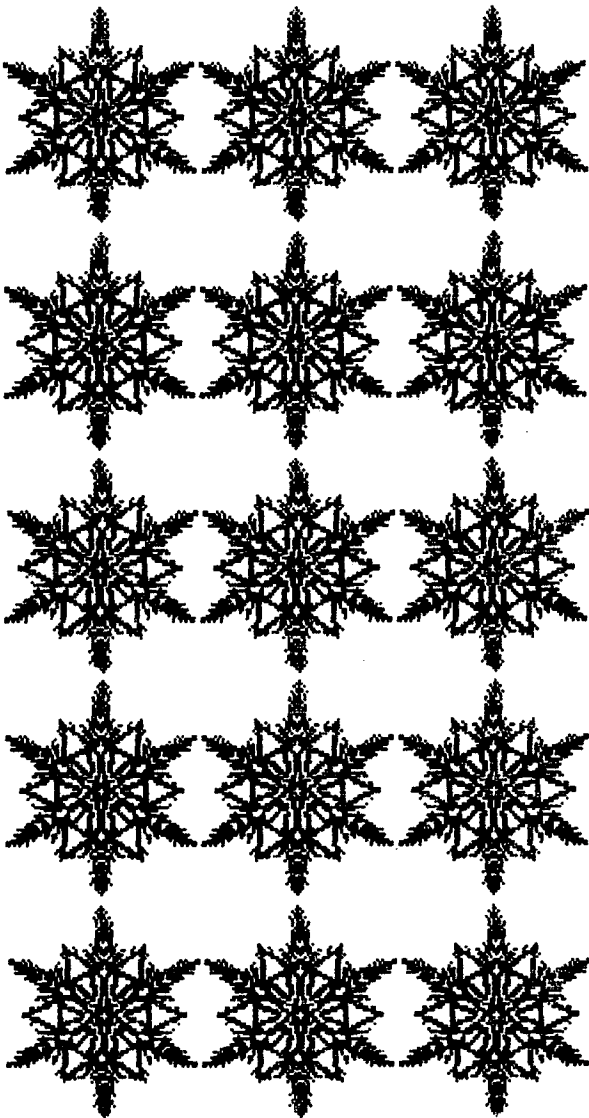
and gradually introduces new words. It is educationally valid, but does not sustain student interest. It is designed for grades 2-3. Rating: **

LONG DIVISION - Demonstrates long division of any input problem in text. Provides a fast and slow option; however slow option is far too rapid for student trying to learn procedure. You can slow down the process further by revising line 160 to read:

160 IF A\$(1,1) = S THEN NDLY = 3000. This program is educationally valid, sustains student interest for necessary time period, but has no game qualities. It is appropriate for grades 4-5. Rating: ***

MORSE CODE INSTRUCTOR - Provides sounds for keyed in letters. Gives code equivalent of letter keyed. This program is a good resource for teaching Morse Code to grade levels 4-6. Rating: **

(Editor's Note: Roz is an active member of EDSIG and an elementary learning specialist at The McLean School in Potomac, Maryland.)



MS.SPELLER

Apple II
UCSD Pascal

Compatible with TV or 80 column monitors
only 60.00
plus 2.50 shipping
Va. residents add 4% tax

spelling correction

dictionary with 2000 hard to spell words
user-created dictionary with options to add, correct, or skip a word while processing a text


utilities: list

- o entire dictionary
- o user added words
- o words added in current session

interactive query updating

includes **text formatter**
free with purchase of **ms.speller**

provides complete word processing capability with UCSD editor
Pascal program listings
documents
letters



Intelligent Computer Systems Corporation

722 South 24th St.
Arlington, Va. 22202
703 684-7389

BENEATH APPLE DOS: a review by William Schulteis

Have you ever wondered how DOS works? Have you ever wanted to modify DOS, recover from an I/O error, call DOS from assembly language, read DOS formatted disks from Pascal, or figure out the format of an uncopyable disk? If the answer to any of these questions is "yes", then you need "Beneath Apple DOS", a new book by Don Wirth and Pieter Lechner, published by Quality Software. The book provides 170 pages of incredibly detailed information about DOS. Most of it is new, not just a rehash of the Apple DOS manual. This is not a book for a novice. Some knowledge of assembly language would be required to use it most effectively.

This book contains eight chapters and three appendices, an index and a removable reference card. It is the same handy size as the Apple manuals, but the plastic binder is cheaper and the pages don't turn as easily. That is just about my only criticism of the entire production. It is full of useful tables, diagrams and clever illustrations. (My favorite includes the legend, "PLEASE DON'T FEED THE ASSEMBLY LANGUAGE PROGRAMMERS".)

Chapter 1 is an introduction which explains how the book was written and how it can be used. It is noted that the book was prepared without help from Apple, mostly by disassembling DOS and consulting with expert programmers. It concentrates on DOS 3.3 while providing a lot of information on earlier versions.

Chapter 2 describes the evolution of DOS. It describes the differences between versions 3.1, 3.2, 3.2.1 and 3.3. This might be useful to someone who has had problems switching between versions.

Chapter 3 describes the disk format. None of this information is in the standard DOS manual. It describes track and sector formats in detail starting with the bits and nybbles. It describes prologues, address fields, data fields, checksums, and epilogues. It covers both the 13- and 16-sector formats. This information is essential if you want to understand techniques used to make disks uncopyable. The chapter covers the sector interleaving methods used on 16-sector disks, comparing methods used by DOS 3.3, Pascal and CP/M. These are defined as software and the chapter includes a table which shows how to translate sector numbers when copying data between operating systems.

Chapter 4 describes diskette organization. It covers space allocation, the VTOC, the catalog, track/sector lists and file formats. Most of this information is not new but it is easier to follow and better illustrated than in the Apple DOS manual. The chapter ends with a discussion of how to recover information from disks with I/O error problems.

Chapter 5 describes the structure of DOS. It includes a memory map showing the major subdivisions of DOS. A particularly useful feature is a complete list of entries

in the page 3 DOS vector table, including the functions not described in the Apple manual. It also contains a detailed description of the boot process, including differences between 3.3 and earlier versions.

Chapter 6 covers using DOS from assembly language. If you are an assembly nut, this chapter is more than worth the cost of the book. It confirms what we always suspected: the main elements of DOS can be called directly without building and writing ASCII command strings. The chapter describes the functions of the DOS file manager and how to call them directly. It describes the associated parameter lists, buffers and buffer formats. It includes samples of assembly code to locate a free buffer, find out the version of DOS and find out which Basic is executing. The chapter also describes the I/O page addresses which control disk hardware and gives some simple examples of assembler code accessing disk controls.

Chapter 7 covers customizing DOS. Some of this material has appeared in Apple Orchard and other publications. Two potentially useful modifications are preventing the reload of the language card on every boot and setting up DOS to BRUN on EXECing a HELLO program.

Chapter 8 is another gold mine, containing a complete description of DOS 3.3 internals. DOS 3.2 would be quite similar except for RWTS. It seems to cover every internal routine or subroutine, buffer, table, work area and data area in DOS including both main memory and the boot routines in ROM. The internal subroutines are described in detail including input and output parameters. The book suggests using Chapter 8 along with a disassembly listing to best understand what DOS is doing.

Appendix A contains five example assembly language programs which use calls into DOS. DUMP reads a track in nybbles. ZAP is a program to read and write sectors. INIT reformats a single bad track without erasing the rest of the disk. FTS finds and lists track/sector lists to help you rebuild a damaged catalog. COPY is a sample of how to use the file manager calls to copy a B-type file to a T-type file. All examples are written for the Apple Tool Kit Assembler.

Appendix B describes disk protection schemes. The more advanced techniques include using non-standard synchronization patterns, unusual track formats, and even unconventional track spacing. The message here is that complex disk coding schemes will defeat even the controversial nybble-level copy programs, or at least make current ones ineffective.

Appendix C is a glossary of terms.

"Beneath Apple DOS" costs \$19.95 and it can be ordered from Quality Software. I expect you will be seeing it in the local stores soon.

DOS 3.3 & DEMO'S FOR US CHEAPSKATES!! by Bob Anderson

I recently scraped together sufficient funds to add a Disk II to my APPLE II Integer computer. In going through the DOS manual and in trying their demonstration programs, I ran into some problems. I felt it would be nice if I shared my problems and fixes with my fellow (male and female) APPLE novices.

The Hello program, the Basic's, and some of the demo programs on the disk assume you are rich enough to afford both Basic's in ROM or a language card. The Hello program assumes you have an APPLE II Plus or an Applesoft ROM card as the system master. It pokes a language card finder routine and then goes looking for it. If it finds the language card, it loads Integer Basic and the Programmer's Aid routines, and then proceeds to write protect the card. If it can not find one, it gives you a sad message and ends. With only Integer Basic, DOS loads and turns control over to the Integer system. Thus, whenever you type in FP to go to floating point, the disk whirrs and leaves you in integer. I copied their system master to another disk and proceeded to modify it. I changed their Hello program to a simple greeting followed by a catalog in Integer Basic. I deleted APPLESOFT, FP BASIC, and INT BASIC from the disk. I copied a disk version of APPLESOFT (available from various members) to the disk so that the FP command now works as it should.

Now, on to the demo's. Because disk APPLESOFT resides in page one of the high resolution graphics, BRIAN'S THEME dies and it takes a reset to get the machine's attention again. This happens because the HGR command in line 120 clears page one of the Hi-res graphics and most of APPLESOFT in the process. Changing that to HGR2 sends the graphic output of the program to Hi-res graphic page two and allows you to observe the patterns generated by the program. The comments and notes put out by the program still go to the text portion of page one. I have not gotten around to finding a fix for that. Disk APPLESOFT does not allow commands of the form: nnn H PLOT X1,Y1 TO X2,Y2 TO X3,Y3. Therefore, lines 300 and 420 have to be broken up into two statements of the form: nnn H PLOT X1,Y1 TO X2,Y2 : H PLOT X2,Y2 TO X3,Y3.

The EXEC DEMO program has similar problems due to the use of disk APPLESOFT. I made the following changes to allow the program to demonstrate its capabilities. I added a line 392 PRINT "NEW" in order to clear the EXEC DEMO program from memory. Otherwise, it gets incorporated into the demo itself and can cause some confusion as to what is happening. Starting on line 450, it goes into Integer Basic and asks to load COLOR DEMO, to list it, to get back into FP and finally to pause to examine the end of the listing. If you

have followed the instructions in the DOS manual concerning copying the program to another disk to run it, your immediate problem is that COLOR DEMO is not on the disk you are running. Assuming you caught that problem, you will now find that the call to FP clears the screen and prints the Applesoft legend at which time it pauses for you to look at it instead of the listing intended. I therefore made the following changes. I deleted lines 450, 460, 490, and then changed 470 to PRINT "LOAD EXEC DEMO". It now loads that program, lists it, and pauses for you to examine the last lines before moving on. A similar thing happens when it tries to show you the monitor memory lists. By changing 530 to PRINT "3DOG" you can re-enter APPLESOFT without reloading it from disk and destroying the screen contents. At 551 I added PRINT "FOR X = 1 TO 8000 : NEXT X" to give you time to examine the catalog. I added a 555 PRINT "NORMAL" to get you back into normal video. Their return to normal video is done outside of the statements which will be transferred to the DO'ER file, so you would be left in the inverse video mode at the end of the demo. I also added a 625 PRINT "DELETE EVEN MORE RECENT PROGRAM!!", which cleans the disk of this even more extraneous file. You thus exit the demo as cleanly as you entered. Note: line 570 has five bells (invisible Ctrl-G's) between the I and T of CHANGE IT; don't panic it is not a fire drill nor a system disaster, they just want to get your undivided attention.

These changes allow us cheapskates to run our new DOS disk complete with reasonably operating demo's. &

SPECIAL...SPECIAL...SPECIAL...SPECIAL

Almost Professional WORD PROCESSOR for your APPLE II or II+. 80 COLUMN Upper/Lower case. On Screen Formatting with SHIFT KEY operation. Works with any printer giving printers full capability. "Perfect" for the serious writer. Easy to learn and operate.

Smarterm 80 Column Board.....\$325
Letter Perfect Word Processor...\$130
Shift Key Modification.....FREE

HLA Inc.
11508 Colt Terrace
Silver Spring, Md.
649-4298

TEXT FILE MANIPULATOR USING APPLEWRITER by Bruce F. Field

Apple DOS text files are extremely useful but often underutilized because most users do not have a simple way of creating or manipulating them. This machine language program in conjunction with Apple's word processing program, Applewriter, allows the user to create or edit text files.

Editors are probably the most popular type of program for computers, so why should we use Applewriter? Each editor has its own commands and format structure, thus I find it difficult to remember if a control-D in editor "A" means do a disk command or delete my text. Obviously I spend a lot of time shuffling through manuals or reference cards. To try and minimize the number of editors needed, I use Applewriter whenever possible. Applewriter has an excellent easy to use editor, that also allows me to include lower case in the text.

Applewriter cannot be used directly with text files as the data is saved to disk as a binary file. Also, the characters are not represented in ASCII code, but are slightly scrambled. Paul Sand wrote two short Applesoft programs (see WAP newsletter, June 1980) to convert text files to Applewriter format and back. These work fine, but are somewhat cumbersome to use and are quite slow as a result of being written in Applesoft. Creating or editing a text file requires running one or both of these programs to generate an intermediate binary file that can be read by Applewriter. This is okay for occasional use but I wanted something faster as I edit files frequently.

The program presented in this article loads or saves a text file directly to or from the Applewriter buffer area in correct Applewriter format. It is written in machine code for rapid execution and is menu driven in a format identical to Applewriter to make it easy to use. The machine code for the program is listed at the end of the article, and the code should soon be available on a library disk. If you choose to type in the code I suggest you save it to disk as follows:

```
BSAVE TXFILER,A$803,L$500
```

As written, the program will only run on a 48K Apple (DOS 3.2 or 3.3), however it can be modified to run on a smaller machine.

To use the program simply "BRUN TXFILER", this will present you with a menu of 5 possible choices.

```
L - LOAD TEXT FILE FROM DISK
S - SAVE TEXT FILE TO DISK
R - RETURN TO THE EDITOR
P - PRINT FILE
Q - QUIT
```

These commands are for the most part self-explanatory and they operate almost

identically to the Applewriter commands. The appropriate letter should be entered with a return in response to the "SELECT:" prompt.

L - Load text file from disk. The user is prompted for the name of the text file. Disk parameters such as D1, D2, etc. may be appended to the name.

The user is asked if all upper case characters in the text should be converted to lower case. This conversion is desirable for upper case files as it is easier to manipulate lower case characters in Applewriter. The text file is then loaded into the Applewriter buffer area. The normal sequence is to then type R - return to the editor which will get TEDITOR the Applewriter editor from the disk.

S - Save text file to disk. The user is prompted for the file name to be used. If a file of this name already exists it will be deleted and replaced with the new file. Again the user has the option of converting all lower case characters back to upper case.

R - Return to the editor. As in Applewriter this gets TEDITOR from the disk and runs it. Unlike Applewriter, TEDITOR must be in drive 1. With a two drive system if you have TEDITOR and TXFILER in drive 1 and the text file in drive 2, the file may be edited and saved without swapping disks.

P - Print file. Identical to the Applewriter command except the PRINTER file must be in drive 1.

Q - Quit. Stops the program and returns to Basic. No warning is given to save your text.

Control - D (indicating a disk command) may be entered in response to the prompt for a menu character. This operates the same as Applewriter. Typing "Q" in response to the disk command prompt will return you to the menu.

To create a text file, run the Applewriter editor (BRUN TEDITOR) and type in the desired text. When finished, exit to the menu and type "Q" to quit the program. BRUN TXFILER, and when the menu appears type "S"; this will save the buffer contents as a text file.

It is somewhat inconvenient to type BRUN TXFILER to get from Applewriter to the text filer program; however it is easy to make a modification to TEDITOR to make text filer one of the menu options. I chose to replace the P - Print file option in TEDITOR with T - Text filer. Now simply typing "T" loads and runs TXFILER. Since I have included the print file option in the text filer menu, hard copy

listings are still available. To modify
TEDITOR:

BLOAD TEDITOR
CALL -151 (to get into the monitor)
12C3 : D4 D8 C6 C9 CC
1289 : D4
10D4 : D4
10DC : D4 C5 D8 D4 A0 C6 C9 CC C5 D2
3DOG (to return to Basic)
BSAVE TEDITOR,A\$803,L\$1040

The modified program must be saved under
the name TEDITOR; so to distinguish it
from the original I suggest you save it on
a separate diskette (with TXFILER and
PRINTER) and mark the diskette for text
file use.

*800.CEF

0800- 00 00 00 20 58 FC A9 03
0808- AD F3 03 A9 08 AD F4 03
0810- A9 FF 85 33 85 D8 A9 80
0818- 85 D9 85 76 A9 01 8D B6
0820- AA A9 4C 8D 5A 9D A9 09
0828- 8D 5B 9D A9 01 20 2C 09
0830- 20 6F FD AD 00 02 C9 CC
0838- D0 38 20 7A 09 A9 06 20
0840- A0 09 20 07 0A A9 08 20
0848- BB 09 A9 08 20 BB 09 BA
0850- 8E 7C 08 A9 00 85 14 A9
0858- 19 85 15 A9 BF D0 07 20
0860- 0C FD A8 B9 14 0D A0 00
0868- 91 14 E6 14 D0 F1 E6 15
0870- D0 ED C9 D3 D0 46 20 7A
0878- 09 A9 07 20 A0 09 20 CA
0880- 09 A9 08 20 BB 09 A9 09
0888- 20 BB 09 A9 08 20 BB 09
0890- A9 0A 20 BB 09 A9 01 85
0898- 14 A9 19 85 15 A0 00 B1
08A0- 14 C9 60 F0 0F A8 B9 14
08A8- 0D 20 ED FD E6 14 D0 ED
08B0- E6 15 D0 E9 A9 0C 20 BB
08B8- 09 4C 03 08 C9 D0 D0 05
08C0- A9 03 4C 2C 09 C9 D2 D0
08C8- 05 A9 04 4C 2C 09 C9 84
08D0- D0 41 A9 02 20 2C 09 A9
08D8- 00 8D 10 C0 20 6F FD AD
08E0- 00 02 C9 D1 D0 03 4C 03
08E8- 08 A0 00 B9 00 02 99 80
08F0- 02 C8 C9 8D D0 F5 A9 00
08F8- 99 80 02 A9 8D 20 ED FD
0900- A9 84 20 ED FD A9 80 85
0908- 05 A9 02 85 06 20 38 09
0910- 4C D2 08 C9 D1 D0 12 A9
0918- BF 8D F2 03 A9 9D 8D F3
0920- 03 A9 38 8D F4 03 4C D3
0928- 03 4C 03 08 0A A8 B9 43
0930- 0A 85 05 B9 44 0A 85 06
0938- A0 00 B1 05 F0 0D 09 80
0940- 20 ED FD E6 05 D0 F1 E6
0948- 06 D0 ED 60 E0 05 D0 10
0950- AE 7C 08 9A A0 00 A9 60
0958- 91 14 8D 9E 95 4C B4 08
0960- 86 05 20 8E FD A6 05 20

0968- 02 A7 20 DD FB A2 0D A9
0970- FF 20 A8 FC CA D0 FA 4C
0978- 03 08 A9 67 85 05 A9 0C
0980- 85 06 20 38 09 A9 00 8D
0988- 10 C0 20 6F FD A0 00 B9
0990- 00 02 99 EC 0C C8 C9 8D
0998- D0 F5 A9 00 99 EC 0C 60
09A0- 20 2C 09 A9 00 8D EB 0C
09A8- 8D 10 C0 20 6F FD AD 00
09B0- 02 C9 CE D0 05 A9 FF 8D
09B8- EB 0C 60 20 2C 09 A9 EC
09C0- 85 05 A9 0C 85 06 20 38
09C8- 09 60 A2 1F A0 DF 98 9D
09D0- 14 0D 88 CA 10 F8 A2 1F
09D8- A0 BF 98 9D 34 0D 88 CA
09E0- 10 F8 A2 1F A0 FF AD EB
09E8- 0C D0 02 A0 DF 98 9D D4
09F0- 0D 88 CA 10 F8 A2 1F A0
09F8- BF 98 9D F4 0D 88 CA 10
0A00- F8 A9 8D 8D A1 0D 60 A2
0A08- 1F A0 FF 98 9D 34 0D 88
0A10- CA 10 F8 A2 1F A0 1F AD
0A18- EB 0C D0 02 A0 DF 98 9D
0A20- 54 0D 88 CA 10 F8 A2 1F
0A28- A0 DF 98 9D 74 0D 88 CA
0A30- 10 F8 A9 8D 8D 21 0D A2
0A38- 7F 8D 14 0D 9D 94 0D CA
0A40- 10 F7 60 00 00 5D 0A 0F
0A48- 0C 43 0C 55 0C 67 0C 79
0A50- 0C 9E 0C C3 0C CA 0C D3
0A58- 0C DB 0C E2 0C 2A 2A 2A
0A60- 2A 2A 2A 2A 2A 2A 2A 2A
0A68- 2A 2A 2A 2A 2A 2A 2A 2A
0A70- 2A 2A 2A 2A 2A 2A 2A 2A
0A78- 2A 2A 2A 2A 2A 2A 2A 2A
0A80- 2A 2A 2A 2A 20 2A 2A 20
0A88- 20 20 20 20 20 20 20 20
0A90- 20 54 45 58 54 20 46 49
0A98- 4C 45 52 20 31 2E 30 20
0AA0- 20 20 20 20 20 20 20 20
0AA8- 20 20 2A 2A 20 2A 2A 20
0AB0- 20 20 20 20 20 20 20 20
0AB8- 20 20 20 20 20 20 20 20
0AC0- 20 20 20 20 20 20 20 20
0AC8- 20 20 20 20 20 20 20 20
0AD0- 20 20 2A 2A 20 2A 2A 20
0AD8- 20 57 52 49 54 54 45 4E
0AE0- 20 42 59 20 42 2E 46 2E
0AE8- 46 49 45 4C 44 2C 20 20
0AF0- 41 50 52 49 4C 20 38 31
0AF8- 20 20 2A 2A 20 2A 2A 20
0B00- 20 20 20 20 20 20 20 20
0B08- 20 20 20 20 20 20 20 20
0B10- 20 20 20 20 20 20 20 20
0B18- 20 20 20 20 20 20 20 20
0B20- 20 20 2A 2A 20 2A 2A 20
0B28- 20 20 20 20 20 20 20 20
0B30- 20 20 20 46 49 4C 45 52
0B38- 20 4D 45 4E 55 20 20 20
0B40- 20 20 20 20 20 20 20 20
0B48- 20 20 2A 2A 20 2A 2A 2A
0B50- 2A 2A 2A 2A 2A 2A 2A 2A

0B58- 2A 2A 2A 2A 2A 2A 2A 2A
 0B60- 2A 2A 2A 2A 2A 2A 2A 2A
 0B68- 2A 2A 2A 2A 2A 2A 2A 2A
 0B70- 2A 2A 2A 2A 20 8D 59 4F
 0B78- 55 20 40 41 59 20 43 48
 0B80- 4F 4F 53 45 20 46 52 4F
 0B88- 4D 20 54 48 45 20 46 4F
 0B90- 4C 4C 4F 57 49 4E 47 20
 0B98- 2D 20 8D 8D 3C 4C 3E 20
 0BA0- 4C 4F 41 44 20 54 45 58
 0BA8- 54 20 46 49 4C 45 20 46
 0BB0- 52 4F 4D 20 44 49 53 4B
 0BB8- 8D 3C 53 3E 20 53 41 56
 0BC0- 45 20 54 45 58 54 20 46
 0BC8- 49 4C 45 20 54 4F 20 44
 0BD0- 49 53 4B 8D 3C 52 3E 20
 0BD8- 52 45 54 55 52 4E 20 54
 0BE0- 4F 20 54 48 45 20 45 44
 0BE8- 49 54 4F 52 8D 3C 50 3E
 0BF0- 20 50 52 49 4E 54 20 46
 0BF8- 49 4C 45 8D 3C 51 3E 20
 0C00- 51 55 49 54 8D 8D 53 45
 0C08- 4C 45 43 54 3A 20 00 8D
 0C10- 8D 3D 3D 3D 3D 20 43 54
 0C18- 52 4C 2D 44 20 20 45 4E
 0C20- 54 45 52 20 44 49 53 4B
 0C28- 20 43 4F 4D 4D 41 4E 44
 0C30- 8D 20 20 20 20 20 28 51
 0C38- 29 20 3D 20 51 55 49 54
 0C40- 8D BA 00 84 42 52 55 4E
 0C48- 20 50 52 49 4E 54 45 52
 0C50- 2C 44 31 8D 00 84 42 52
 0C58- 55 4E 20 54 45 44 49 54
 0C60- 4F 52 2C 44 31 8D 00 8D
 0C68- 45 4E 54 45 52 20 46 49
 0C70- 4C 45 20 4E 41 4D 45 3A
 0C78- 00 8D 43 4F 4E 56 45 52
 0C80- 54 20 55 50 50 45 52 20
 0C88- 54 4F 20 4C 4F 57 45 52
 0C90- 20 43 41 53 45 3F 20 28
 0C98- 59 2F 4E 29 3A 00 8D 43
 0CA0- 4F 4E 56 45 52 54 20 4C
 0CA8- 4F 57 45 52 20 54 4F 20
 0CB0- 55 50 50 45 52 20 43 41
 0CB8- 53 45 3F 20 28 59 2F 4E
 0CC0- 29 3A 00 84 4F 50 45 4E
 0CC8- 20 00 84 44 45 4C 45 54
 0CD0- 45 20 00 84 57 52 49 54
 0CD8- 45 20 00 84 52 45 41 44
 0CE0- 20 00 8D 84 43 4C 4F 53
 0CE8- 45 20 00 00 00 00 00 00

Ⓔ



GRAPHICS ON THE EPSON

The following letter was received by the Editor from Kenny E. Burrow, one of our members from Millington, Tennessee.

"Recently I purchased an Epson MX-80 printer for my system. I was immediately impressed at the capabilities it possessed, especially for the price. I learned that the printer's block graphics could not be used with the APPLE and the APPLE-Epson printer interface card. This was not a serious problem since I did not need this capability. A short time later, I started to get curious about this and started to get greedy. I discussed this with Mark Kadrich, another APPLE owner. After examining the schematics of the interface board, we found out that data bit 7 was hardwired to ground. When we switched the jumper from ground to the data line, graphics were printed on the printer. I later bought a single pole, double throw switch and installed it on the board. The center lead was connected to the solder point going to the printer, one of the other connections wired to the ground point, and the remaining connection was wired to the data line. This gave me the ability to switch back and forth between graphics and regular print. To determine what character to use in my word processor, I find the graphic character's ASCII code, subtract 128 from it, and look up what APPLE character (page 138 in the Applesoft manual) this new number represents. With this modification, I have the use of all the graphic characters.

At the present time, we are working on a routine to mix the graphic characters with the normal print from within my Magic Window. I would be very interested to hear from anyone who has purchased the MX-80 printer to compare notes, especially if they are using the Magic Window word processor."

Kenny's address is: 8026B Corsair,
Millington, TN 38053. Ⓔ

FORTUNE'S ANALYSIS OF THE PERSONAL COMPUTER INDUSTRY

by Walton Francis

The latest (June 29) issue of FORTUNE magazine contains a wide ranging article on the development, current progress, and near term future of the personal computer industry. Titled "The Coming Struggle in Personal Computers", the article describes the various design and marketing strategies of the leading companies, why some have done so well and some not so well, and tries to crystal ball the future just a little. The article contains some fascinating tidbits (Commodore got into the game by pure accident), some words of wisdom on the advantages conferred by the sheer mass of software available for the leading machines, and brief stories on the people behind the leading software hits--VisiCalc, CP/M, WordStar, and Microsoft Basic. There is little here that isn't known to those who have read INFOWORLD and BYTE from cover to cover for the last year or two, but new APPLE owners especially will find it extremely informative. Incidentally, APPLE is now apparently a clear number one over Radio Shack, and Atari is losing big money on a surprisingly small market share. Ⓔ

FREE SECTORS BOOT PROGRAM by Bruce F. Field

This program is a modified version of my HELLO program that reads the track bit map on a disk and computes the number of free sectors. On a multiple drive system the program reads whatever drive was accessed last. The REM statements make the program self-explanatory.
JLIST

```
100 REM THIS PROGRAM READS THE TRACK BIT MAP AND COMPUTES
110 REM THE NUMBER OF FREE SECTORS ON A DISK
115 REM
120 REM A MACH. LANG. PROG. TO COUNT SECTORS AND
130 REM CALL RWTS IS POKED AT $3B8 TO $3CF.
140 REM
142 REM THE TRACK BIT MAP IS STORED IN THE LAST DOS
144 REM BUFFER AT $9600, CHANGE LINE 600 FOR APPLE
146 REM WITH LESS THAN 48K MEMORY.
148 REM
150 REM WORKS WITH EITHER DOS 3.2 OR 3.3, HOWEVER LINE 280
152 REM SHOULD BE CHANGED WITH CORRECT NUMBER OF SECTORS
154 REM (455) FOR DOS 3.2
156 REM
158 REM BRUCE F. FIELD MAY 1981
160 REM
190 GOSUB 500: REM POKE MACHINE LANGUAGE STUFF
200 REM GET TRACK BIT MAP
210 TRK = 17:SEC = 0: GOSUB 340: REM RWTS
220 REM COMPUTE FREE SECTORS
230 S = 0:BUF = HBUF * 256 + LBUF
240 FOR I = 68 + BUF TO 192 + BUF STEP 4
250 POKE 0, PEEK (I): CALL 952:S = S + PEEK (0)
260 POKE 0, PEEK (I + 1): CALL 952:S = S + PEEK (0)
270 NEXT I
280 P = INT (100 * (1.005 - S / 560)): REM 560=16 SECTOR
290 PRINT "VOLUME NUMBER "; PEEK (IOB + 14)
300 PRINT "DISK IS ";P;"% FULL, ";S;" FREE SECTORS"
330 END
340 REM SETUP SOME PARAMETERS AND CALL RWTS
350 VOL = 0: POKE IOB + 3,VOL: REM SET WILD CARD VOLUME
360 ICMD = 1: POKE IOB + 12,ICMD: REM READ
370 POKE IOB + 4,TRK: POKE IOB + 5,SEC: REM TRACK AND SECTOR
380 CALL 966: REM DO RWTS
390 IF PEEK (0) < 128 THEN 480
400 REM DISK ERROR, PRINT MESSAGE
410 PRINT CHR$ (7);"DISK I/O ERROR"
470 END
480 RETURN
490 REM POKE SECTOR COUNT ROUTINE
500 POKE 952,169: POKE 953,0: POKE 954,162: POKE 955,8: POKE 956,6
510 POKE 957,0: POKE 958,105: POKE 959,0: POKE 960,202: POKE 961,208
520 POKE 962,249: POKE 963,133: POKE 964,0: POKE 965,96
530 REM GET LOCATION OF IOB TABLE ADDRESS
540 IBADDR = PEEK (999) + 256 * PEEK (1000)
550 REM POKE RWTS CALL AND SAVE CARRY IN LOC $0
560 POKE 966,169: POKE 967, PEEK (IBADDR + 1): POKE 968,160: POKE 969, PEEK
(IBADDR)
570 POKE 970,32: POKE 971,217: POKE 972,3: POKE 973,102: POKE 974,0: POKE
975,96
580 REM LOOKUP IOB TABLE ADDRESS
590 IOB = PEEK (IBADDR) + 256 * PEEK (IBADDR + 1)
600 HBUF = 150:LBUF = 0: REM BUFFER=$9600
610 POKE IOB + 8,LBUF: POKE IOB + 9,HBUF
620 RETURN
630 END
```

EPSON MX-80 PRINTER INTERFACE PROGRAM by Herb Klonser

If you have an EPSON MX-80 printer, you have probably tried to print out text using many of the different print modes that the MX-80 can handle. If so, you found that there is not an easy way to do this. You can send out commands to set the printer up to certain modes (for example: 6 lines per inch, condensed print of 132 characters per line, and emphasized print) and then print a file, but there is no good way to change modes within a text file.

The following program, called PRINT, was written to ease the interface between a text file and the MX-80 printer. The program is written in PASCAL and it only works with PASCAL files. It allows a user to read in a text file and print it using whatever special MX-80 modes the user desires. Text can be printed at different lines per inch, different lines per page, and different modes (such as condensed, double strike, and double width), and these options can be changed within the text being printed.

The program allows two different methods of communicating special MX-80 commands to the printer:

I. GLOBAL MODE: When the program is executed (after responding to the reminder to turn your printer on), an option menu comes up with 6 options. Option 6 is to quit and leave the program. We will get to option 5 in a moment. Option 4 is to print a file (whose name the user has specified). Options 1 to 3 are the global options. The user can select option 1 to set the number of lines per page, option 2 to set the number of lines per inch, or option 3 to select print types. These selections will then be sent to the printer to set it up to those modes. Those modes will remain intact until an overriding command is sent out, or the printer is turned off and back on. Any text files that are printed (via use of option 4) will be printed with whatever modes have been selected. The option 3 print types are W (double width characters) or C (condensed characters), E (emphasized characters), or D (double print characters). E mode strikes a character twice to darken it. D mode does the same but shifts the print line 1/216 inch over between strikes.

The global mode is useful when a whole file is to be printed in the same format. For instance, you might want to print a whole report at 132 characters per line (condensed mode) and at 8 lines per inch. This would be set by selecting option 3 and "C", selecting option 2 and "8" (lines/inch), and then selecting option 4 to print the file. You supply the volume and file name to be printed:

II. EMBEDDED MODE: The second mode of sending commands to the printer is done by embedding the commands in the text to be printed. Any of the options that can be sent to the printer in the global mode can be turned on and off within the text file via commands that are put in with the text when the file is set up under the editor.

Embedded commands have the format {x} where x has to be one of the following commands:

0	= space two lines before printing
-	= space three lines before printing
+	= no space before printing (carriage return w/o line feed)
1	= eject page
G	= continuous print (don't eject at LPP count, only at "1")
C	= condensed print (132 characters per line)

W = double width characters (40 characters per line)
 D = double print (print, space 1/216" and print again)
 E = emphasized print (double strike)
 N = return to normal printing (cancels C, W, D, E; 80 cpl)
 B = ring buzzer on printer for 3 seconds
 6 = print 6 lines per inch
 8 = print 8 lines per inch
 T = print 10 lines per inch
 c,w,d,e = cancel commands for C, W, D, E embedded commands

The {x} format of entering commands was selected so that these commands can be embedded in PASCAL text files that are to be compiled. The compiler will take the commands to be comments and thus will ignore them. Yet when the text file is run through the PRINT program, the commands will be interpreted and the proper printer setting will be done. The only acceptable format for a printer command is where a left curly bracket is followed by one of the printer commands listed above, and followed by a right curly bracket. If there is more than one character between the curly brackets, the sequence of characters is not recognized as a printer command. Multiple commands can be put back to back such as {x}{x}, but not {xx}. The second case would be taken as a part of the text and not a command.

The program works by reading in the specified print file, scanning every character in the file for embedded commands, copying the file over to an output file while it inserts the proper MX-80 commands for the printer, and sending the file to the printer. The text files are read in and processed 10 blocks (512 bytes per block) at a time until the file is completely done. To process the 10 blocks, it may take up to 30 seconds. When the 10 blocks are copied over to the output file, they are printed. Then the next 10 blocks will be processed.

The embedded commands in the input text file are removed from the file so they are not visible on your printout. They are used only to select the proper printer commands. Any line with a command in it, will be printed 3 characters to the left of where it is on the screen.

A detailed explanation of each embedded command is given on the next two pages. The embedded commands are then grouped by type for convenient reference. Following that is a page of general rules to follow in entering the embedded commands.

A sample of a text file with embedded print commands is given followed by the resulting file as it is printed by the PRINT program.

EMBEDDED COMMAND EXPLANATION

- 0: Skip 2 spaces before printing. This inserts a Line-Feed (LF) into the output buffer. This can be put anywhere on a line to cause the spacing.
- : This is the same as "0" but 3 spaces are skipped instead of 2.
- +: No space. This can be used to do overstriking of a word or phrase or for underlining of text. This command should be put in columns 1-3 of the line. If it is put within a line, it will be ignored. When the command is processed, it is removed from the line (like all other commands are) thus the characters you put starting in column 4 will be left shifted 3 to start in column 1. This is done to allow an overstriking or underlining to be put on any character in the previous line (including any of

the first three characters). This mode only works when all the characters in the previous line were printed in the same mode. It will not work for mixed mode previous lines. For example, a "C" mode line with a "W"... "w" in the line will cause the underlining or overstriking on the line that follows to be in the wrong columns. Also for double width lines, because of the limitations of the printer, the characters printed on the + line will be a half space shifted to the right. That makes this mode useful only for C or N (132 or 80 character per line) modes.

- 1: Eject page. This will insert a Form-Feed (FF) into the output buffer. It can be put anywhere on a line.
- G: Go (Continuous print). Prevent page eject until a "1" command is hit. Normally, until a "1" command is hit in the text, a page eject is done after every lines-per-page (LLP) count is reached. LLP count is set by global option 1. If not set, the default is 58 LPP. As soon as a "1" is found, ejection based on LLP count is stopped. Ejection only occurs then when another "1" is hit. If the user wants to completely control page ejection (by "1" commands), from the beginning of the file, put a "G" at the start of the file. This will prevent the first page from ejecting before desired. For example, if printing at 8 lines/inch, 88 LPP can fit on a page, yet if "G" is not set, ejection will occur after 58 lines. "G" provides continuous print until a "1" is hit.
- W: Print double width characters. This takes 2 normal character positions to print one new character. The option may be turned on ("W") or off ("w") within a line or over many lines. The command may occur anywhere in a line. If the printout will not fit on a line anymore due to the extra spaces taken up by the double width fields, the end of the line will overflow and be printed on top of the same line (possibly in a different mode from the mode of the original line). Therefore, the user should make sure that his lines will fit on the current line length when in this mode. If the print type was in N (normal) mode when W was entered, only 40 cpl (characters per line) will be printed. Similarly, if in "C" mode, this gives 66 characters per line. This mode is turned off by a "w" or "N" command.
- C: Condensed print. This command should only be entered as one of the starting commands on a line. If it is found within a line, condensed mode is turned on for the whole line. This mode prints out 132 cpl maximum. The mode should be turned off on a new line after the last line to be printed in condensed mode by entering "c" or "N". If "E" or "D" is used while in "C" mode, the print goes to 80 cpl until "E" or "D" is turned off; then printing will revert to 132 cpl. "c" and "W" modes can be specified together to give 66 cpl. If "N" or "c" is specified on the same line as "C", the condensed mode will be cancelled for the whole line.
- E: Emphasized print. This double strikes each character (and slows the printing down from 80 characters per second to 40) to produce a dark professional looking printout. It can be turned on ("E") and off ("e") at any point in a line. When on, all characters are printed at 80 cpl, even if in "C" mode. If "C""W" or "N""W" modes are in effect, then 40 cpl will be printed. If "N" or "e" is specified on the same line as "E", emphasized mode will be cancelled for the whole line.
- D: Double print. This strikes each character, then shifts over 1/216" and restrikes each character. It produces a print similar to the "E" mode. It slows printing from 80 characters per second to 40. It can be turned on ("D") or off("d") at any point in a line. When on, all characters are printed at 80 cpl, even if in "C" or "C""W" or "N""W" modes are in effect. If "N" or "d" is specified on the same line as "D", double print will be cancelled for the whole line. This mode can be turned off by "d" or "N".

N: Normal print. This cancels any previously instated C,W,D, or E commands, thus restoring printing to 80 cpl and normal print (80 characters per second). It can be used, for example, at the beginning of a file to insure that normal print mode is set (as long as no global commands are desired). The C,W,D, or E commands can also be individually canceled by c,w,d, or e. If the "N" command is put on a line after any other commands on that same line, it will normally cancel all the other commands on that line. Therefore, it should be put on a line by itself for best usage or put as the first mode on a line so it will cancel any modes that were set up on previous lines before instating new modes.

c,w,d,e: Cancel C,W,D,E modes. These commands allow individual canceling of previously instated print commands. They can be put anywhere on a line. However, if put on the same line as the where the mode was originally set, they may cancel the mode for the whole line.

6,8,T: Line spacing. Normal print is at 6 lines/inch. This can be changed to 8 or 10 (actually 7/72") lines per inch. These commands should be put at the start of a line, but can be put at any point on a line. The line spacing will remain in effect until it is changed by another 6,8, or T command.

B: Ring the printer bell for three seconds.

EMBEDDED COMMAND GROUPS

The embedded print commands can be grouped in the following types. This is another way of grouping the commands for easier reference.

1. CHARACTER PER LINE MODE

N - normal 80 cpl (default).
C - condensed 132 cpl
NW - normal double width 40 cpl
CW - condensed double width 66 cpl

[Note: N,c,nw, or cw can be used to cancel the cpl mode.]

2. PRINT DENSITIES

E - emphasized (double strike)
D - double print (shift 1/216" between double strikes)
[Note: N or e or d can be used to cancel these modes.]

3. LINE SPACING

O - skip 2 lines before printing
- - skip 1 line before printing
+ - do not skip any lines & print (overstrike or underline)
1 - eject to top of next page
G - continous print (no eject til "1" command is found)
6 - space 6 lines per inch
8 - space 8 lines per inch
T - space 10 lines per inch

[The line spacing stays the same until changed by another line spacing command.]

4. SPECIAL

B - Ring printer bell for 3 seconds

contd.

RULES FOR EMBEDDED COMMANDS

The following rules should be followed in entering embedded commands:

1. Normally the command should be put in the first column of the line. This will avoid unusual results such as:
 - a. If "C" is put anywhere in a line, it turns condensed mode on for that line (132 cpl) and then stops. All the characters on the line where "C" was inserted up to 132 characters are printed condensed. Then the mode stops. If "C" is put on the first part of the line, "C" mode will remain in effect for all lines until the mode is set to something else.
 - b. If "W" is turned on within a line and then left on for the next few lines, remember that all the characters on the line before "W" was set on are in the previous mode that was set. The next lines will all be in "W" mode. Then, for example, if you wanted to print a double width title and underline it, if you put the "W" command followed by the title starting on character position 10, then the first 9 characters in that line are in the current mode (e.g. normal at 80 cpl). If you left "W" mode on and put '----' to underline your title on the next line, that would not line up properly with the title because the initial blanks there are at 40 cpl. If you had put your "W" command in column 1, however, all blanks are in that 40 cpl mode and it is easier to line the text up.

Exceptions to this rule are for using "W" commands to bring out a word or phrase embedded in the text. In that case the "W" command would be put before the word or phrase, and an "w" would be put following it.
2. The commands themselves are removed from the text so the rest of the line following the commands is shifted left to fill in. This gives the capability of embedding commands without having them break up the text that is printed.
3. The "no space" command is normally used to underline or double strike words. Remember that when you put the "+" command in the first column, it takes up 3 characters that will later be removed from the buffer. Therefore, remember to adjust your underlining or overstriking characters three to the left of where you want them to be. This allows you to use this feature to underline or overstrike any character in the previous line - including those in the first 3 columns.
4. Mixing character-per-line and print density commands on a line can cause the printer to get confused about what mode is to be selected. It is best to change modes on different lines. If you put a start and a cancel command on the same line, the mode is usually cancelled for the whole line. The exception to this is the "W" command.
5. Remember that the last commands sent after printing a file will remain in effect until the printer is turned off and back on. If one file is printed, then some editing is done and the file is printed again, the file will start printing according to the last commands output (of course any new embedded commands would override the old commands). It is a good practice to start the file with an "N" command to reset the print mode.

contd.

Sample text file before printing with "PRINT".

{W} EPSON MX-80 PRINTER SAMPLE

{W} -----

03/01/81

{-}{N}This is a sample of the types of printout that the MX-80 printer can produce. The normal print mode is 80 characters per line as typed here.

{+}

The other varieties of print are shown below.

{O}{W}This is normal mode double width characters: 40 char per line.

{+}

{N}{C}This is a sample of condensed print which is 132 char/line.

{+}

{W}This is condensed print with double width characters. This gives 66 characters per line.

{+}

{O}{N}Any of the above modes may be printed emphasized or double struck.

{E}This is emphasized print. It gives a darker print line.

{N}{D}This is double strike print. Each line is printed, then it is reprinted with the characters shifted over 1/216th of an inch. The mode takes twice as long to print since each line is printed twice.

{O}{N}All of the MX-80 printing can be done at 6 lines per inch (default) as is shown above or at 8 or 10 lines per inch. Some samples are shown below:

{O}{6}THIS IS 6 LINES PER INCH

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890!"#\$%&'()*+,-@+<;+/?^][_'{}~

{O}{8}THIS IS 8 LINES PER INCH

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890!"#\$%&'()*+,-@+<;+/?^][_'{}~

{O}{10}THIS IS 10 LINES PER INCH

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890!"#\$%&'()*+,-@+<;+/?^][_'{}~

{O}{8}{C}THIS IS 8 LINES PER INCH USING CONDENSED PRINT

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

1234567890!"#\$%&'()*+,-@+<;+/?^][_'{}~

{6}{N}

Double width characters can be mixed on a line to bring out that part of the printed text.

For example, you can {W}enlarge this{N} within a line.

{N}{C}

This is an example of condensed print with {W}double width{N}{C} characters embedded in it.{N}

{O}The {W}{C}MX-80 PRINTER {N}is a very versatile printer.{B}

Result of printing the sample text file using the "PRINT" program.

EPSON MX-80 PRINTER SAMPLE

03/01/81

This is a sample of the types of printout that the MX-80 printer can produce. The normal print mode is 80 characters per line as typed here. The other varieties of print are shown below.

This is normal mode double width characters: 40 char per line.

This is a sample of condensed print which is 132 char/line.

This is condensed print with double width characters.

This gives 66 characters per line.

Any of the above modes may be printed emphasized or double struck.

This is emphasized print. It gives a darker print line.

This is double strike print. Each line is printed, then it is reprinted with the characters shifted over 1/216th of an inch. The mode takes twice as long to print since each line is printed twice.

All of the MX-80 printing can be done at 6 lines per inch (default) as is shown above or at 8 or 10 lines per inch. Some samples are shown below:

THIS IS 6 LINES PER INCH
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890!"#\$%&'()*+=-@+><;+/?^][_'{}~

THIS IS 8 LINES PER INCH
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890!"#\$%&'()*+=-@+><;+/?^][_'{}~

THIS IS 10 LINES PER INCH
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890!"#\$%&'()*+=-@+><;+/?^][_'{}~

THIS IS 8 LINES PER INCH USING CONDENSED PRINT
 ABCDEFGHIJKLMNOPQRSTUVWXYZ
 abcdefghijklmnopqrstuvwxyz
 1234567890!"#\$%&'()*+=-@+><;+/?^][_'{}~

Double width characters can be mixed on a line to bring out that part of the printed text.

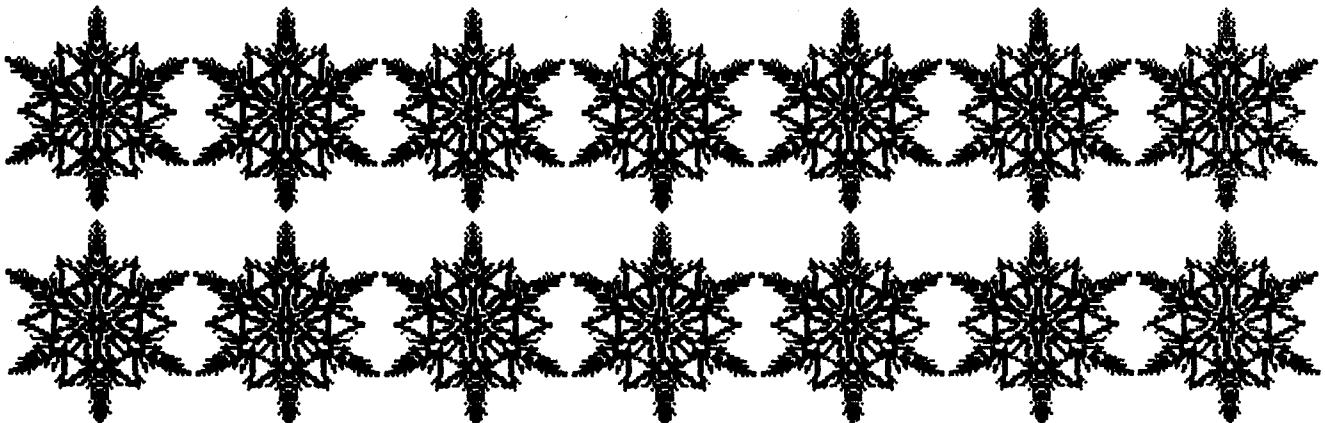
For example, you can **enlarge this** within a line.

This is an example of condensed print with double width characters embedded in it.

The **MX-80 PRINTER** is a very versatile printer.

contd.

"SNOWFLAKES"



```

PROGRAM PRINT;
(*$R-$) (*TURN OFF RANGE CK TO SPEED UP PROGRAM*)
(*=====*)
(* 03/14/81 CONTROL HOW A FILE IS PRINTED *)
(* REQUEST A FILE TO BE PRINTED ON THE MX-80 PRINTER. *)
(*=====*)

VAR OPTION, INCHAR :CHAR;
    PRTT :TEXT;
    PRT :FILE;
    LINES_PER_PAGE, LF72 :INTEGER;
    DBL_WIDTH, ONLY_6LPI_THIS_PAGE, C_ON :BOOLEAN;

(*-----*)
PROCEDURE INFO; (*----- INITIAL INFORMATION -----*)
BEGIN
    WRITELN('=====PRINTER FORMATTER PROGRAM=====02/17/81=====');
    WRITELN(' This program will allow you to set the MX-80 printer up to the');
    WRITELN('print that you select. After setting up the printer, you can');
    WRITELN('print an existing file via this program or you can quit the');
    WRITELN('program and use the Filer's T(ransfer command to write files');
    WRITELN('to the printer. The printer will stay in the same print mode til');
    WRITELN('the printer is turned off and back on. ');
    WRITELN(' Note that if the file is printed by the Filer, option 1(lines');
    WRITELN('per page) and option 3 W (double width chars) will not work. ');
    WRITELN('They only work when printed by option 4(print file). ');
    WRITELN; WRITELN
END;

(*-----*)
PROCEDURE OPTION_MENU; (*----- PRINT OUT THE OPTIONS MENU -----*)
BEGIN
    WRITELN('SELECT THE PRINT OPTIONS YOU WISH: ');
    WRITELN(' NO. OPTION DEFAULT ');
    WRITELN('-----');
    WRITELN(' 1 LINES/PAGE 58 ');
    WRITELN(' 2 LINES/INCH 6 ');
    WRITELN(' 3 PRINT TYPE NORMAL,8OCHAR/LINE ');
    WRITELN(' 4 PRINT A FILE ');
    WRITELN(' 5 INFO ON EMBEDDED COMMANDS ');
    WRITELN(' 6 QUIT ');
    WRITELN
END;

(*-----*)
PROCEDURE LINES_PAGE; (*-----SET PRINTER TO REQUESTED LINES/PAGE-----*)
BEGIN
    WRITELN('THIS OPTION WILL BE OVERRIDDEN IF ANY "EJECT PAGE" PRT COMMAND');
    WRITELN('IS FOUND IN THE TEXT. ');
    WRITELN('USE A LARGE # TO OVERRIDE END OF PAGE EJECT (e.g.3000). ');
    WRITE('ENTER # OF LINES TO PRINT PER PAGE (BEST 58): ');
    READLN(LINES_PER_PAGE);
    IF LINES_PER_PAGE < 1 THEN LINES_PER_PAGE:= 1;
    WRITELN(PRTT,CHR(27),CHR(67),CHR(LINES_PER_PAGE)) (*ESC C+n TO SET PRINTER*)
END;

(*-----*)
PROCEDURE LINES_INCH; (*-----SET PRINTER TO REQUESTED LINES/INCH-----*)
VAR LPI :INTEGER;
    INCHAR:CHAR;
BEGIN
    WRITELN('THE LPI SETTING WILL REMAIN IN EFFECT UNTIL AN EMBEDDED COMMAND ');
    WRITELN('IN THE TEXT CHANGES TO A DIFFERENT LPI. ');
    WRITE('ENTER # OF LINES TO PRINT PER INCH (6, 8, OR 10): ');
    READLN(LPI);
    INCHAR := '2'; (*DEFAULT TO 6 LINES PER INCH*)
    ONLY_6LPI_THIS_PAGE := (LPI = 2); (*TRUE IF 6LPI, ELSE FALSE*)
    LF72 := 12; (*DEFAULT TO 12/72" PER LINE= 6LPI*)
    IF LPI = 8 THEN BEGIN INCHAR := '0'; LF72 := 9; END; (* 8LPI*)
    IF LPI =10 THEN BEGIN INCHAR := '1'; LF72 := 7; END; (*10LPI*)
    WRITELN(PRTT,CHR(27),INCHAR); (*ESC 0 OR 1 OR 2 TO SET PRINTER*)
    END;

(*-----*)
PROCEDURE PRINT_TYPE; (*-----SET PRINTER TO REQUESTED TYPE OF PRINT-----*)
VAR PRTCHR : CHAR;
BEGIN
    WRITELN('TYPES OF PRINT: ');
    WRITELN(' W(idth dbl on or w(idth dbl off [each char takes 2*width] ');

```

```

WRITELN(' C(ondensed on or c(ondensed off [132 char/line]');
WRITELN(' E(mphasized on or e(mphasized off [double strike]');
WRITELN(' D(ouble prt on or d(ouble prt off [1/216" & reprint]');
WRITELN;
WRITELN('ENTER TYPE OF PRINT [upper case letter==> turn option ONJ]');
WRITE(' ',20,'[lower case letter==> turn option OFF] : ');
READLN(PRTCHR); (*W,C,E,D,w,c,e,d*)
CASE PRTCHR OF
'W' : DBL WIDTH := TRUE; (*SET FOR PRINT FILE*)
'C' : BEGIN WRITELN(PRTT,CHR(15)); C_ON:=TRUE; END; (*SI*)
'E' : WRITELN(PRTT,CHR(27),'E'); (*ESC E*)
'D' : WRITELN(PRTT,CHR(27),'G'); (*ESC F*)
'w' : DBL WIDTH := FALSE; (*SET FOR PRINT FILE*)
'c' : BEGIN WRITELN(PRTT,CHR(18)); C_ON:=FALSE; END;
'e' : WRITELN(PRTT,CHR(27),'F');
'd' : WRITELN(PRTT,CHR(27),'H');
END
END;

(*-----*)
PROCEDURE EMBED1_INFO; (*- PRINT INFORMATION ON EMBEDDED COMMANDS -*)
VAR INCHAR : CHAR; (*-----*)
BEGIN
WRITELN(' Printer commands can be set up globally via this program or by');
WRITELN(' inserting commands directly into the text (embedded commands).');
WRITELN(' {x} is a sample of an embedded printer command where x is the');
WRITELN(' command id. Acceptable command id's are:');
WRITELN(' 0 = space two lines before printing');
WRITELN(' - = space three lines before printing');
WRITELN(' + = no space before printing(carriage return w/o line feed)');
WRITELN(' 1 = eject page');
WRITELN(' 60 = continuous print (override lines/page eject)');
WRITELN(' C = condensed print (132 characters per line)');
WRITELN(' W = double width characters (40 characters per line)');
WRITELN(' D = double print (print, space 1/216" and print again)');
WRITELN(' E = emphasized print (double strike)');
WRITELN(' N = return to normal printing (cancels C, W, D, E; 80 cpl)');
END;

(*-----*)
PROCEDURE EMBED2_INFO; (*-FINISH PRINT : TOO LONG FOR ONE PROCEDURE -*)
BEGIN
WRITELN(' B = ring buzzer on printer for 3 seconds');
WRITELN(' 6 = print 6 lines per inch');
WRITELN(' 8 = print 8 lines per inch');
WRITELN(' T = print 10 lines per inch');
WRITELN(' c,w,d,e = cancel commands for C, W, D, E embedded commands');
WRITELN;
WRITE('HIT RETURN TO CONTINUE ==>');
READLN(INCHAR);
WRITELN('Commands can be entered back to back, such as {x}{x}. For example');
WRITELN('if the commands were C and W, then the text would be printed as');
WRITELN('condensed characters at double width (ie 66 characters per line)');
WRITELN('Any previously issued commands can be cancelled with a "N" which');
WRITELN('restores the printer to normal printing (80 characters per line).');
WRITELN('This capability provides the user with the option of printing');
WRITELN('different parts of one file out in different formats.');
```

```

(**I #5:PRINT_FILE.TEXT*) (*INCLUDE MEMBER TO PRINT THE FILE*)

```

```

(*----- MAIN PROGRAM -----*)

```

```

BEGIN
INFO; (*PRINT OUT INITIAL INFORMATION*)
WRITELN('TURN PRINTER ON AND MAKE SURE PAPER IS AT TOP OF FORM. ');
WRITE('ENTER Y WHEN READY: '); READLN(INCHAR);
IF INCHAR='Y' THEN
BEGIN
DBL WIDTH := FALSE; (*DEFAULT TO NOT DO ENLARGED CHARACTERS*)
LINES PER PAGE := 58; (*DEFAULT TO 58 LINES PER PAGE*)
REWRITE(PRTT,'PRINTER: '); (*OPEN PRINTER FOR TEXT OUTPUT*)
REWRITE(PRT, 'PRINTER: '); (*OPEN PRINTER FOR BLOCK OUTPUT*)
WRITELN(PRTT,CHR(27),'2'); (*DEFAULT TO 6 LINES/INCH: ESC 2 *)
C_ON := FALSE; (*DEFAULT TO NOT CONDENSED PRT*)
LF72 := 12; (*DEFAULT TO 12/72" PER LF=6LPI*)
ONLY 6LPI_THIS_PAGE:=TRUE; (*DEFAULT TO 6LPI*)
REPEAT
OPTION MENU; (*PRINT OUT THE OPTION MENU*)
WRITE('ENTER OPTION NUMBER: '); READLN(OPTION);
CASE OPTION OF
'1' : LINES_PAGE; (*LINES PER PAGE*)

```

```

'2' : LINES_INCH;          (*LINES PER INCH*)
'3' : PRINT_TYPE;         (*TYPE OF PRINT CHAR DESIRED*)
'4' : PRINT_FILE;        (*PRINT A FILE*)
'5' : BEGIN               (*INFO ON EMBEDDED CMNDS*)
      EMBED1_INFO;
      EMBED2_INFO;
      END;
      END (*CASE OPTION*)
UNTIL OPTION = '6'
END
END. (*MAIN PROGRAM*)

```

```

PROCEDURE PRINT_FILE;
(*=====*)
(* PRINT FILE 05/01/81 PRINT A FILE CKING FOR EMBEDDED COMMANDS *)
(* NOTE: THIS FILE IS INCLUDED BY "PRINT" *)
(*=====*)
CONST MAXBLKS = 10 ; (*MAX # OF 512 BLKS THAT WILL BE READ IN AT A TIME*)
      IP_STRT =512 ; (*START OF INPUT BUFFER IN BUFF ARRAY*)
      (* NOTE : IF MAXBLKS OR IP_STRT CHANGES, CHANGE VAR BUF BELOW TO *)
      (* O..MAXBLKS*512 +-IP_STRT + 2*)

```

```

VAR BUF : PACKED ARRAY[0..5634] OF 0..255;
      LINE_CNT, I, J, BLKNO, MAXLENGTH, PAPERLNTH : INTEGER;
      BLKS_XFER, NBLKS, NULLS, BUF_END : INTEGER;
      LPP_EJECT : BOOLEAN;
      INFILE : FILE;
      INNAME, LAST_NAME : STRING;
      TEMPSTR : STRING[1];
      CMND_ID : PACKED SET OF 0..255; (*EMBEDDED COMMANDS*)

```

```
PROCEDURE CK_EACH_CHAR;
```

```

(*-----*)
(*THIS VERSION OF EJECT PAGE IS USED BY CK EACH CHAR*)
PROCEDURE EJECT_PAGE; (*---EJECT TO TOP OF NXT PG BASED ON LINE SPACING---*)
VAR N, N85, LOOP, NREM, LPI_VALUE : INTEGER;
BEGIN (*EJECT PAGE*)
  (*THERE ARE 792 - 1/72" IN A PAGE(11"). SUBTRACT #/72" ALREADY DONE&*)
  (*SPACE THE REST TO DO A PAGE EJECT. #/72s HAVE BEEN ADDED UP AT CR's*)
  IF ONLY_6LPI_THIS_PAGE = TRUE THEN
    BUF[J] := I2 (*FF*)
  ELSE
    BEGIN
      N := 792 - PAPERLNTH; (*#/72s TO SPACE*)
      N85 := N div 85; (*# OF 85/72s *)
      NREM := N mod 85; (*REMAINDER OF LAST 85/72s BLOCK*)
      IF N > 0 THEN (*IF NOT AT TOP OF PAGE*)
        BEGIN (*ENTER PRT CMNDS TO SKIP BLKS OF 85/72"*)
          FOR LOOP := 1 TO N85+1 DO
            BEGIN
              BUF[J]:= 27; BUF[J+1]:= 65; (*ESC A 85 - SPACE 85/72's inch*)
              IF LOOP <> N85+1 THEN BUF[J+2]:=85 ELSE BUF[J+2]:=NREM;
              BUF[J+3]:=27; BUF[J+4]:=50; BUF[J+5]:=10; (*ESC 2 LF*)
              J := J+6;
            END;
            (*--- RESET PRINTER TO CURRENT LPI MODE---*)
            BUF[J]:=27; BUF[J+1]:=65; BUF[J+2]:=LF72; (*ESC A n:OLD LPI*)
            BUF[J+3]:=27; BUF[J+4]:=50; (*ESC 2 FOR 6LPI*)
            IF LF72 = 7 THEN BUF[J+4] := 49; (*ESC 1 FOR 10LPI*)
            IF LF72 = 9 THEN BUF[J+4] := 48; (*ESC 0 FOR 8LPI*)
            BUF[J+5]:=13; BUF[J+6]:=10; J:=J+6; (*CR LF TO DO SPACE *)
            PAPERLNTH := LF72*2; (*RESET CNT OF 1/72" ALREADY PRINTED*)
            IF LF72 = 12 THEN ONLY_6LPI_THIS_PAGE := TRUE
            ELSE ONLY_6LPI_THIS_PAGE := FALSE;
          END;
        END;
      END;
    END;
  END; (*EJECT_PAGE*)

```

```

(*-----*)
PROCEDURE PRT_COMMANDS;
VAR CI, LASTLN_CHR : INTEGER;
(*THIS VERSION OF EJECT PAGE IS USED BY PRT COMMANDS*)
PROCEDURE EJECT_PAGE; (*---EJECT TO TOP OF NXT PG BASED ON LINE SPACING---*)
VAR N, N85, LOOP, NREM, LPI_VALUE : INTEGER;
BEGIN (*EJECT PAGE*)
  (*THERE ARE 792 - 1/72" IN A PAGE(11"). SUBTRACT #/72" ALREADY DONE&*)
  (*SPACE THE REST TO DO A PAGE EJECT. #/72s HAVE BEEN ADDED UP AT CR's*)
  IF ONLY_6LPI_THIS_PAGE = TRUE THEN
    BUF[J] := I2 (*FF*)

```



```

ELSE
  BEGIN
    N := 792 - PAPERLNTH; (*#/72s TO SPACE*)
    N85 := N div 85; (*# OF 85/72s *)
    NREM := N mod 85; (*REMAINDER OF LAST 85/72s BLOCK*)
    IF N > 0 THEN (*IF NOT AT TOP OF PAGE*)
      BEGIN (*ENTER PRT CMNDS TO SKIP BLKS OF 85/72"*)
        FOR LOOP := 1 TO N85+1 DO
          BEGIN
            BUF[J]:= 27; BUF[J+1]:= 65; (*ESC A 85 - SPACE 85/72's inch*)
            IF LOOP <> N85+1 THEN BUF[J+2]:=85 ELSE BUF[J+2]:=NREM;
            BUF[J+3]:=27; BUF[J+4]:=50; BUF[J+5]:=10; (*ESC 2 LF*)
            J := J+6;
          END; (*----- RESET PRINTER TO CURRENT LPI MODE-----*)
          BUF[J]:=27; BUF[J+1]:=65; BUF[J+2]:=LF72; (*ESC A n:OLD LPI*)
          BUF[J+3]:=27; BUF[J+4]:=50; (*ESC 2 FOR 6LPI*)
          IF LF72 = 7 THEN BUF[J+4] := 49; (*ESC 1 FOR 10LPI*)
          IF LF72 = 9 THEN BUF[J+4] := 48; (*ESC 0 FOR 8LPI*)
          BUF[J+5]:=13; BUF[J+6]:=10; J:=J+6; (*CR LF TO DO SPACE *)
          PAPERLNTH := LF72*2; (*RESET CNT OF 1/72" ALREADY PRINTED*)
          IF LF72 = 12 THEN ONLY_6LPI_THIS_PAGE := TRUE
          ELSE ONLY_6LPI_THIS_PAGE := FALSE;
        END;
      END;
    END;
  END; (*EJECT_PAGE*)

```

```

(*)-----(*)
PROCEDURE NO_SPACE;
(*----- PROCESS A "+" EMBEDDED COMMAND -----*)
(* NO SPACE BETWEEN LINES. FIND & REMOVE CR, FILL LINE TO END *)
(*-----*)
VAR INDEX, M, N, CPL, NEWJ, CMND_CHR : INTEGER;
BEGIN (*NO SPACE*)
  IF (J > 0) THEN (*NOT 1ST CHAR*)
    BEGIN
      INDEX := 0; (*-----FIND CR BEFORE THE "+" CMND-----*)
      (*POSSIBLE CASES AFTER A CR: *)
      (* CR x x= any character*)
      (* CR W W= dbl wdth chr *)
      (* CR DLE # x *)
      (* CR DLE # W x *)
      (*-----*)
      NEWJ := J-1;
      IF BUF[NEWJ] = 14 THEN NEWJ := NEWJ-1; (*PASS OVER "W" IF IN BUFF*)
      IF (BUF[NEWJ]=13) THEN INDEX:= NEWJ; (*PREV CHAR WAS CR*)
      IF (BUF[NEWJ-2]=13) AND (BUF[NEWJ-1]=16) THEN (*CR,DLE,#*)
        BEGIN
          INDEX := NEWJ-2;
          N := BUF[NEWJ] (*SAVE BLANK COUNT*)
        END;
      (*PAD PREVIOUS LINE WITH BLANKS FOLLOWED BY NO CR SO THE *)
      (*NEXT LINE WILL OVERLAY THE PREVIOUS ONE. *)
      IF INDEX <> 0 THEN
        BEGIN
          IF C_ON = TRUE THEN (*---FIND CHAR/LINE BASED ON PRT MODE---*)
            IF DBL_WIDTH = TRUE THEN
              CPL := 66 (*CONDENSED PRT, DOUBLE WIDTH CHAR *)
            ELSE
              CPL := 131 (*CONDENSED PRT, NORMAL WIDTH CHAR*)
            ELSE
              IF DBL_WIDTH = TRUE THEN
                CPL := 40 (*NORMAL PRT, DOUBLE WIDTH CHAR *)
              ELSE
                CPL := 79; (* NORMAL PRT, NORMAL WIDTH CHAR *)
                (*-----FIND PREVIOUS CR-----*)
            NULLS := INDEX;
            CMND_CHR := 0; (*CNT OF EMBEDDED CMNDS INLINE*)
            REPEAT
              IF BUF[NULLS] IN [7,10,12,14,15,18,20] THEN
                CMND_CHR := CMND_CHR +1;
              IF BUF[NULLS] = 27 THEN CMND_CHR := CMND_CHR +2; (*ESC x *)
              NULLS := NULLS -1;
            UNTIL (BUF[NULLS]=13) OR (NULLS=0); (*EXIT WHEN PREV CR IS HIT*)
            (*-----CNT BLNKS TO ADD-----*)
            LASTLN_CHR:= INDEX - NULLS - 1 -CMND_CHR; (* # CHAR IN PREV LINE*)
            (*ADD IN INITIAL DLE CHAR COUNT*)
            IF BUF[NULLS+1] = 16 THEN LASTLN_CHR:= LASTLN_CHR+BUF[NULLS+2]-34;
            NULLS := CPL - LASTLN_CHR; (*CHAR BLANKS TO PAD *)
            IF NULLS > 0 THEN (*-----ADD BLANKS TO PREV LINE-----*)

```

```

BEGIN
  NULLS := NULLS + INDEX;
  FOR M := INDEX TO NULLS DO BUF[M] := 32; (*BLANK*)
  IF DBL WIDTH = TRUE THEN
    BEGIN BUF[NULLS+1]:=14;
          NULLS:=NULLS+1;
    END; (*"W" ON NXT LN*)
  IF INDEX = NEWJ-2 THEN
    BEGIN
      BUF[NULLS+1] := 16; (*DLE*)
      BUF[NULLS+2] := N;  (*BLANK COUNT*)
      NULLS := NULLS + 2;
    END;
    J := NULLS
  END;
END
END; (*NO_SPACE*)

```

```

-----*)
PROCEDURE ESC_CMND(ESC_VALUE: INTEGER); (*PUT AN "ESC x" CMND INTO BUF*)
BEGIN
  BUF[J]:= 27;
  J:=J+1;
  BUF[J]:= ESC_VALUE;
END; (*-----*)

```

```

PROCEDURE LF_CMND; (*INSERT LF AND "W", IF ACTIVE *)
BEGIN
  BUF[J]:= 10;
  IF DBL WIDTH = TRUE THEN
    BEGIN
      J := J+1;
      BUF[J] := 14; (*W COMMAND*)
    END
  END;
END; (*-----*)

```

```

BEGIN (*PRT_CMNDS*)
(* CHECK HERE FOR ANY EMBEDDED COMMANDS AND INSERT ASCII COMMANDS INTO *)
(* THE OUTPUT BUFFER WHEN A COMMAND IS FOUND IN THE INPUT BUF. *)

```

```

IF BUF[I+1] IN CMND_ID THEN (*IF A VALID EMBEDDED CMND*)

```

```

  BEGIN
    WRITELN('Embedded command "',CHR(BUF[I+1]),'" found in input file');
    CASE BUF[I+1] OF
      49:BEGIN (*----- 1 = EJECT PAGE -----*)
          LPP EJECT := FALSE; (*STOP EJECT AFTER 58 LINES IF *)
          EJECT_PAGE;
        END;
      48:LF_CMND; (*LF*) (*----- 0 = SPACE 2 LINES-----*)
      45:BEGIN (*----- - = SPACE 3 LINES-----*)
          BUF[J]:=10; J := J+1; LF_CMND (*LF*)
        END;
      43:NO SPACE; (*----- + = NO LINE SPACE-----*)
      78:BEGIN (*----- N = NORMAL PRINT -----*)
          BUF[J] := 20; (*DC 4 CODE FOR CANCELLING W *)
          BUF[J+1] := 18; (*DC 2 CODE FOR CANCELLING C *)
          BUF[J+2] := 27; (*ESC F CODE FOR CANCELLING E *)
          BUF[J+3] := 70;
          BUF[J+4] := 27; (*ESC H FOR CANCELLING D *)
          BUF[J+5] := 72;
          J := J + 5; (*INCREMENT OUTPUT BUFFER POINTER*)
          C ON := FALSE; (*TURN OFF FOR CK IN "+" CMND *)
          DBL_WIDTH := FALSE; (*TURN OFF DBL WPTH PRT FOR "+" *)
        END;
      66:BUF[J] := 7; (*----- B = RING BELL, 3SEC ----*)
      67:BEGIN (*----- C = CONDENSED PRT-----*)
          C ON := TRUE; (*SET ON FOR CK IN "+" CMND*)
          BOF[J]:=15; (*SI*)
        END;
      71:BEGIN (*----- G = OVERRIDE LPP EJECT---*)
          LPP EJECT := FALSE;
          J := J-1; (*RESET J- NOTHING IS PUT OUT *)
        END;
      87:BEGIN (*----- W = DBL WPTH PRT -----*)
          DBL WIDTH := TRUE;
          BUF[J]:=14; (*SO*)
        END;
      68:ESC_CMND(71); (*ESC G*) (*----- D = DBL STRIK PRT-----*)
      69:ESC_CMND(69); (*ESC E*) (*----- E = EMPHASIZD PRT-----*)
    END;
  END;

```



```

ELSE      (*INCR PAST ALL NULLS AT END OF 2 BLKS*)
  WHILE (BUF[I+1] = 0) AND (I+1 <> BUF_END) DO I := I+1;
END
END
END
END;

(*-----*)
BEGIN (* PRINT FILE *)
  PAPERLNGTH := 0; (*# OF 1/72s OF PAPER PRINTED*)
  CMND_ID := [43,45,48,49,54,56,66,67,68,69,71,78,84,87,99,100,101,119]; (*DEFINE THE SET OF EMBEDDED PRINT COMMANDS*)
  (*-----GET FILE NAME-----*)
  WRITELN('ENTER VOL:NAME.TEXT TO BE PRINTED OR');
  WRITELN('W" FOR WORK FILE OR');
  WRITE('S" FOR SAME FILE AS LAST TIME      : ');
  READLN(INNAME);
  IF LENGTH(INNAME) = 1 THEN
    BEGIN
      IF INNAME = 'S' THEN INNAME:=LAST NAME; (*USE SAME FILE AS LAST TIME*)
      IF INNAME = 'W' THEN INNAME:='*:SYSTEM.WRK.TEXT'
    END;
    (*I-*) (*TURN OFF AUTO ERROR CHECKING.*)
    RESET(INFILE, INNAME); (*OPEN THE FILE, IF IT EXISTS. *)
    IF IORESULT <> 0 THEN (*IF FILE DOESN'T EXIST *)
      BEGIN
        WRITELN('FILE NOT FOUND. ');
        (*I+*) (*TURN ERROR CHECKING BACK ON*)
      END
    ELSE
      BEGIN
        (*-----PROCESS INPUT FILES-----*)
        (*I-*) (*TURN ERROR CHECKING BACK ON*)
        LPP_EJECT := TRUE; (*ALLOW PAGE EJECT AFTER 58 LINES.*)
        BLK_NO := 2; (*SKIP 2 HEADER BLKS WRITTEN BY PASCAL*)
        MAXLENGTH:= MAXBLKS*512 -1; (*MAX LENGTH OF INPUT BUFFER*)
        LINE CNT := -1; (*COUNT OF LINES*)
        REPEAT (*-----READ MAX BLKS/TIME TIL DONE----*)
          BLKS_XFER := BLOCKREAD(INFILE, BUF[IP_STRT], MAXBLKS, BLK_NO);
          IF BLKS_XFER <> 0 THEN
            BEGIN
              BUF_END := BLKS_XFER*512 +IP_STRT-1; (*END OF DATA READ IN*)
              BLK_NO := BLK_NO + BLKS_XFER; (*BLK TO READ NXT PASS*)
              J:= 0; (*START OF OUTPUT BUF: IP & OP BUFS OVERLAY EACH OTHER*)
              (* WITH IP STARTING 512 AFTER IP *)
              IF DBL WIDTH = TRUE THEN (*GO TO DBL WIDTH CHAR IF RQSTD AT STRT*)
                BEGIN
                  BUF[0] := 14; (*PRT CMND FOR DBL WPTH*)
                  J := 1
                END;
              WRITELN(BLKS_XFER, ' Blocks of input file being processed. ');
              WRITELN('It may take 30 sec to process 10 blocks. ');
              CK_EACH_CHAR; (*MOVE EACH CHAR IP ==>OP BUF, CK FOR PRT CMNDS*)
              IF J > I THEN
                BEGIN
                  WRITELN('***ERROR*** OUTPUT FILE OVERLAPPED INPUT FILE');
                  WRITELN('CHANGE IP_STRT TO A LARGER #, RECOMPILE, & RUN AGAIN');
                END;
                (*--FILL LAST PARTIAL OP BLK WITH NULLS--*)
                NBLKS := (J DIV 512) +1; (*#BLKS TO OP-FULL BLKS + 1 PARTIAL*)
                FOR NULLS:= J TO (NBLKS*512 -1) DO BUF[NULLS]:=0; (*FILL IN PARTIAL*)
                NULLS:= BLOCKWRITE(PRT, BUF, NBLKS) (*-----WRITE TO PRINTER--*)
            END
          UNTIL BLKS_XFER <> MAXBLKS; (*EXIT WHEN NO MORE BLKS OF IP FILE*)
          CLOSE(INFILE); (*CLOSE THE INPUT FILE*)
          LAST_NAME := INNAME (*SAVE CURRENT NAME FOR USE NEXT TIME IF RQSTD*)
        END
      END;
END;
END;

```





You should have seen it. The annual meeting was great! Good turnout and fine talks by all. The door prizes were most appreciated, and included among the many fine donated items which were worth approximately \$5000 were a graphics tablet and the brand new Apple CAT modem. Let's see more of you at the next meeting.

New Faces...

First, the Directors. Region 1, the Far West, now has Fred Wilkinson (President of the San Francisco Apple Core) as the incumbent, and Jim Simmons (of the original Apple Core) replacing Joe Alinsky who has moved to Chicago. Region 2, the North, has Harlan Felt (currently with Apple, Inc.) as the incumbent and Jim Hassler of Wyoming replacing Jon Lawrence. We'll miss ya, Jon. Region 3, the South (including Texas) has Scott Knaster of Denver as the incumbent and a new two-year term for Jerry Vitt of Dallas. Jerry was also re-elected as Chairman of the IAC Board. In Region 4, the East, Tony Cerreta is the incumbent and I am starting a new two-year hitch. Roger Keating serves as Director of Australia, Wolfgang Dederick as Director of Europe, and Auby Mandell as Director of Canada.

Now, the Officers...

For President we have...you guessed it...Ken Silverman of San Francisco. Jovial Dave Gordon of LA is in charge of Vice. Our Secretaryship goes to hard-working Joe Budge of North Carolina. Our new Treasurer is Dave Alpert of Chicago. I think it's a great lineup, ready to do great things!

The Orchard gets a new Editor...

Peter Weiglin of the San Francisco Apple Core has been appointed Editor of the Apple Orchard, replacing Val Golding of Call-A.P.P.L.E. fame. Val has assumed a host of new tasks for the Puget Sound folks, and both he and Ken agreed to the change. Don't go too far, Val. We expect guest articles by you and often! You can sample some of Peter's excellent style by reading his lead article in the latest Orchard. We're with you, Pete!

More on nybble and bit copiers...

Now I know that at least some of you read the Bulletin. I also know that my recent questionnaire wasn't too well designed. I thought I had set it up so that you could answer just one of the five questions, which would place you somewhere between complete endorsement of their sale and use, and total rejection and condemnation of such products, i.e. a spectrum of possibilities. Didn't work that way. You answered each question or came up with different responses. The responses were all over the spectrum but most felt that IAC should remain neutral on the subject.

I do stand corrected - Steve Lawson of the G.R.A.P.E.VINE of Port Orchard, Washington, pointed out that the word "condone" was out of place in the first question, "Should the IAC condone its sale and use?" I thought I could weasel out of it by finding an obscure or obsolescent definition. Nope. Should have used "sanction" or "encourage". However, there was no intention of condemning the programs before taking the survey. Just a mischoice of words on my part. By the way, Steve, the Apple Orchard is not a pseudo-official organ of Apple Computer, Inc. due to the incorporation of Contact within its pages. Apple graciously offered to help us get started by publishing in the Orchard materials which would otherwise have gone into Contact. The funds which would have been used for Contact were (and are) used instead for partial defrayment of the cost of production of the Orchard. Apple purchases copies of the Orchard to donate one free issue to each new Apple owner of record. Sure, this helps Apple but it helps the IAC as well. IAC, however, retains total control over the shape, direction and policy of the Orchard. Thanks, Steve and the rest of you, for providing thoughtful responses.

The United States Yacht Racing Union is soliciting any programs readers might have developed relating to sailing, race scoring, handicapping, measurement rules, and the like. They maintain a library of such contributions at their headquarters: USYRU, P. O. Box 209, Newport, RI, 02840, USA.

The International Apple Core would like to thank Tom Jacobsen and the Cedar Falls, Iowa Green Apples for their help in re-typing the illegible Anotes. Many of the Anotes you've received were re-typed by Tom and his club. They get special credit for re-doing the entire Pascal 1.0 Bios!

★ ★ ★ **STOCK MARKET UTILITIES** ★ ★ ★
4 STOCK MARKET PROGRAMS ON DISK

Four programs provide a complete programming system for entry and storage of stock data, data correction, autoscaling Hi-Res graphical display of performance, and building historical data files electronically (program to download data not included). **Now with HARD COPY GRAPHICS.**

STK. 1 (39 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, input historical data to a single data file, display contents of individual stock files from disk, option to reduce files to last 260 entries for high-res graphics. All data files are fully accessible.

DATA CORRECTOR (31 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 (transaction dollars constant) to provide continuous momentum and price curves, also correct for incomplete updating due, for example, to a power outage.

EVAL (22 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.

MIRQ (12 Sectors) is used to build historical data files electronically by converting downloaded stock price data obtained from CompuServe's Micro-Quote financial data base to data files compatible with these programs.

Programs written by H. S. PILLOFF. **DOS 3.3**
Requires Apple II,™ ROM Applesoft,™ 48K and Disk (DOS 3.2)

★ ★ ★ **ELECTRONIC STOCK PACKAGE** ★ ★ ★

A complete system including password and programs for accessing the Dow Jones Stock Quote Reporter (contains more than 6000 daily stock prices). Current rates permit nightly updating for about 1.5¢ per stock.

Downloading programs provide for auto dialing, logging on, retrieving daily data (prev. close, open, high, low, close, volume) for up to 200 stocks stored in easily edited file, disconnecting from system, and then writing data to a single file on the user's disk. Data can then be displayed or printed.

Conversion programs read this disk file, formats data (M/D/Y/VOL/FNL), and automatically updates each individual stock file. Format is fully compatible with STOCK MARKET UTILITY PROGRAMS.

Requires Apple II/II+,™ Applesoft,™ 48K, Disk (DOS 3.2 or 3.3), and D.C. Hayes Micromodem II™.

Electronic Stock Package (includes Dow Jones password) \$80.00

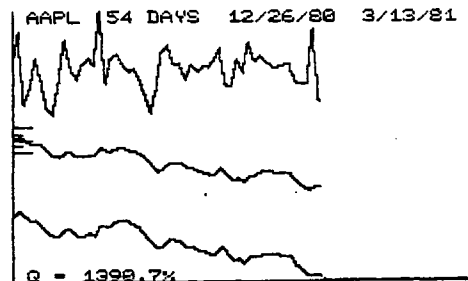
Stock Market Utility Programs \$59.95

COMING SOON!! Black-Scholes Stock Option Evaluation Programs
Specializing in Investment Software

Foreign orders add \$10 for shipping. **H&H SCIENTIFIC** VISA/Mastercard Accepted
13507 Pendleton Street, Oxon Hill, MD 20022 Tel (301) 292-3100

Apple II/II+, and Applesoft are trademarks of Apple Computer, Inc. Micromodem II is a trade mark of D.C. Hayes Assoc., Inc.

Bernie Urban, Editor
May 13, 1981



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. (If you order five or more the additional charge will be \$10.00 total.) They will come in a protective foam diskette mailer.

PROGRAM DISKETTES

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

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

- | | |
|----------------------------------|--------------------------------------|
| Volume 1 Utilities I () | Volume 33 Accounting () |
| Volume 2 Utilities II () | Volume 34 Solar Tutor () |
| Volume 3 Games I () | Volume 35 Garden Management () |
| Volume 4 Games II () | Volume 36 Games XII () |
| Volume 5 Games III () | Volume 37 Utilities IX () |
| Volume 6 Games IV () | Volume 38 Games XIII () |
| Volume 7 Games V () | |
| Volume 8 Utilities III () | |
| Volume 9 Educational I () | Volume 100 Dos 3.3 Utilities A () |
| Volume 10 Math/Science () | Volume 180 Dungeon Designer () |
| Volume 11 Graphics I () | Volume 181 Beginner's Cave () |
| Volume 12 Games VI () | *Volume 182 Lair of Minotaur () |
| Volume 13 Games () | *Volume 183 Cave of the Mind () |
| Volume 14 IAC Utilities IV () | *Volume 184 Zyphur Riverventure () |
| Volume 15 Games VII () | *Volume 185 Castle of Doom () |
| Volume 16 Utilities V () | *Volume 186 Death Star () |
| Volume 17 Graphics II () | *Volume 187 Devil's Tomb () |
| Volume 18 Educational II () | *Volume 188 Caves of Treas. Isl. () |
| Volume 19 Communications () | *Volume 189 Furioso () |
| Volume 20 Music () | *Volume 190 The Magic Kingdom () |
| Volume 21 Apple Orchard () | *Volume 191 The Tomb of Molinar () |
| Volume 22 Utilities VI () | *Volume 192 Lost Island of Apple () |
| Volume 23 Games VIII () | Pascal: |
| Volume 24 Games IX () | Volume 300 ATTACH BIOS () |
| Volume 25 Utilities VII () | Volume 301 FIG1: () |
| Volume 26 Stocks/Investments () | Volume 302 FIG2: () |
| Volume 27 Math () | Volume 303 FIG3: () |
| Volume 28 Planetfinder () | Volume 304 FIG4: () |
| Volume 29 Utilities VIII () | |
| Volume 30 Games X () | |
| Volume 31 Plot Utilities () | |
| Volume 32 Games XI () | |

*Vol. 181 required with these disks.

 TOTAL ORDER = \$ -----

Check here if you want these shipped---

NOTE: PLEASE ALLOW 6 - 8 WEEKS FOR MAILING.

NAME -----

ADDRESS -----

CITY, STATE, ZIP -----

TELEPHONE -----

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

Make checks payable to "Washington Apple Pi"

Send order to: Washington Apple Pi- ATTN: Librarian
 PO Box 34511
 Washington, DC 20034

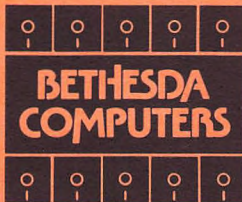
Eve
introduced
the apple to
Adam



Now
we've brought it
to Bethesda

Can we tempt you?

As your authorized Apple dealer, Bethesda Computers specializes in the sale of micro-computers, software, peripheral equipment & accessories. Drop by for a "hands-on" demonstration and professional analysis of your computer needs.



Bethesda Computers
8020 Norfolk Avenue
Bethesda, Maryland 20014
(301) 657-1992