



mini'app'les

apple computer user group newsletter

MAY 1980

VOL III No 5

OFFICERS:

President	Dan Buchler	890-5051 13516 Grand Burnsville Mn 55337	Vice President	Steve K Johnson	869-3447
Treasurer	Marilyn Thomas	824-9892 2735 Irving Minneapolis Mn 55408	Secretary	E Michael Young	884-2841 9319 Elliot Bloomington Mn 55420

MEMBERS-OF-THE-BOARD:

Newsletter Editor	Chuck Thiesfeld	831-0009 830-5020 8416 Xerxes Bloomington Mn 55431	Librarians:		
			-Acquisition	Keith Madonna	474-3876
			-Organization	Terry Pinotti	786-7118
Bibliographer	Chuck Boody	1-873-2227	Consultants:		
Prog.Editor	Ken Slingsby		-Communications	Nelson Capes	
			-Technical	Jim Henke	869-0361

IN THIS ISSUE

APNOTES from IAC
 Communications Corner by Nelson Capes
 Dan on Printers
 Expanded Disk by Dave Laden
 IAC Software and Related Matters by D.Buchler
 Minutes of APR 16th Meeting
 PASCAL Special Interest Group by K.Madonna
 Program Line Editor (PLE) by Chuck Boody
 Program Reviews by S.K.Johnson
 Scanning the Apple Newsletters by D.Buchler
 TABSETTER by D.Buchler
 TRAC & COPY-TRAC Files by D.Buchler
 Turning the Pages with Dave Laden
 What Flavor Diskette Do You Want? by Dave Laden
 Wild Colors by Dave Laden
 Word of Caution by Dave Laden

NEXT MEETING

Wednesday, May 21st, 7:30pm at the
 Minnesota Federal Savings and Loan
 9th Avenue South, Hopkins, Minn.

Program:

Large Disks.

We hope to have a demonstration of 8 inch floppies on an Apple and perhaps a hard disk. We also hope to show one or more of the 80 character display boards. Persons will be asked to comment on the pros and cons of the 3 available models.

JUNE MEETING:

Wednesday, June 18th, 1980.
 Representatives from 3M will explain the rules of a competition to create a demo program on the use and care of diskettes. They will answer questions about diskettes.

PROGRAM LINE EDITOR (PLE) of CALL APPLE
A Review by Chuck Boody

I promised to try to keep club members up-to-date with those new things I have discovered--in magazines and/or software. This past week I received my copy of the PROGRAM LINE EDITOR disk from Call A.P.P.L.E., and I want to talk about that here.

PLE is a well documented set of enhancements to the Apple II editor. When the program is run it hooks itself on at the top of memory and relocates the DOS buffers below it. It occupies about 1600 bytes of space. With it one can: Edit a program line without having to copy all of it. Delete portions of a program line. Insert materials into a program line without recopying any of it. Add visible (in inverse print) control characters within a program line. Insert upper and lower case letters (unfortunately not visible on the screen) into lines to be printed on a printer. Find the exact point in a line you wish to change quickly and easily without copying across the screen. Recall a line just edited to make other changes. AND, in short, cut the key strokes and time involved in editing down a great deal. I estimate that it cuts the time needed to make most changes by about 70%!!!!

And if that is not enough, PLE provides a whole set of escape functions that can be redefined by the user to suit his needs. The program comes with such commands as: 1. Catalog drive 1 disk. 2. give the length and address of the most recently loaded binary file. 3. Emulate all of the Autostart ROM editing features. 4. Enhance the right and left arrow so that they move the cursor 8 positions at a time if you wish. 5. Jump you into the monitor 6. Print common commands with a single keystroke when programming.

The package comes with the PLE in Integer and Applesoft versions (both work with either language, but the two versions provide any Apple owner with a disk access to the editor), the escape create program, (also in both languages) and a program to remove PLE from memory without damage to your program if you need the space. All this for \$20.00 if you're a Call A.P.P.L.E. member. The program (by Neil Konzen) has also been advertised on the retail market for \$35.00.--Either way is a real buy.

Does it have any drawbacks? Well, yes, a few. One feature does not seem to work under Applesoft (at least on machines with the language card), though fortunately that is a minor feature. I suspect a call to Call A.P.P.L.E. will fix this. The program hooks through the & hook (decimal 1013) and can interfere with other programs (like Ampersort) which use it. And finally, use of the PR# or IN# commands unhooks PLE, so that you have to re-hook it with a Call 1013 or & command (from Applesoft).

I can think of no piece of software that I own that will get more use or that is more

worth the very small investment. What a wonderful aid!!! (No I don't get a cut of the profits -- I wish I did).

MINUTES OF APR 16 MEETING

Meeting called to order by D. Buchler at 7:40. About 60 members in attendance. Minutes approved as printed.

OLD BUSINESS

1) A new disk has been added to the user bank.

2) Keith Madonna has volunteered to develop a PASCAL user bank disk which he will circulate at the meetings (sorry out of town members--maybe at a later date....).

3) Steve Johnson will undertake to develop a set of tape programs to supplement the earlier set Dan B. provided. Tape users can contact Steve about it.

NEW BUSINESS

1) Max Coe had volunteered to start an assembly language class if users are interested. A sign-up sheet was circulated, and others can contact Max through the club officers.

2) Dan described the new publication NIBBLE and described the club policy on circulating programs from such a journal. The programs will be circulated only if a person can provide proof of purchase or subscription to the journal. It is the feeling of the club that programs from journals whose reason for publishing is to provide programs should not be made generally available. To do so would be to deny those publications the income on which they depend and so would be a moral if not legal violation of copyright.

3) The June meeting will have representatives from 3M here to talk about their disks and to suggest a programming contest--They haven't really described the contest, but it sounds like fun...

4) John Olson moved, it was seconded and passed, to ammend the by-laws to include the elective office of Vice-President.

5) The list of candidates nominated for elected office was presented to the club. It was moved and seconded to accept the list of candidates and elect them by acclamation. Passed.

6) Newly re-elected President Dan Buchler asked that each of the newly elected Aofficers and appointed board members introduce themselves and talk about their position. (See last month's newsletter for names and positions).

7) It was suggested that the club contact the International Apple Corps and volunteer to copy the program disk locally, providing the money to the IAC. It was hoped that such an approach would provide IAC with income and remove some of the burden from them. Dan agreed to make the necessary contacts.

Meeting adjourned at c.8:45.

WHAT FLAVOR DISKETTE DO YOU WANT?

By: David Laden.

This article is a clarification of terms more than anything else. If you are confused about these terms: SYSTEM MASTER, 3.2 DOS, 3.2.1 DOS, STANDARD MASTER (DOS), PLUS MASTER (DOS), MASTER, SLAVE, or "LOST" DOS, continue reading. Your questions will probably be answered.

The SYSTEM MASTER DISKETTE is the diskette supplied by Apple Computer Inc. when you purchase a disk drive for your Apple. This diskette contains the Disk Operating System (DOS), various disk utilities, and demonstration programs.

The most current version of DOS is 3.2.1. Only minor changes were made to DOS 3.2 to get DOS 3.2.1. This revision consisted of changes made to the timing when seeking a drive and changes to the COPY program to facilitate more reliable disk-to-disk copying. If your diskettes contain DOS 3.2 you do not need to worry about updating them. The only time you have to have DOS 3.2.1 in your system is when you intend to make disk-to-disk copies using the new COPY program from the SYSTEM MASTER DISKETTE. Now, if you only have DOS 3.1, it is suggested that you update to 3.2.1. Major changes took place between 3.1 and 3.2. Programs (which use DOS) written using 3.2 will not work correctly with 3.1 and vice versa.

There are two versions of the SYSTEM MASTER DISKETTE: STANDARD and PLUS. The STANDARD SYSTEM MASTER contains utilities and programs written in Integer BASIC (for the Standard Apple system containing Integer BASIC in ROM). The PLUS SYSTEM MASTER contains utilities and programs written in Applesoft BASIC (for the Apple Plus system containing Applesoft BASIC in ROM). In addition, the PLUS diskette contains Applesoft Renumber and Append routines and Applesoft Chain. Both of these diskettes contain Apple's disk-to-disk copy routine and UPDATE 3.2.1. UPDATE 3.2.1 is used to change the DOS on your diskette from a previous version (like 3.1 or 3.2) to 3.2.1.

The term MASTER is not only used to mean SYSTEM MASTER or original. It is also used to describe a diskette which will boot any size system. This means that when DOS is loaded into your system, it will locate itself into the highest portion of RAM memory contained in that system. In addition to changing DOS to the current version, UPDATE 3.2.1 also makes your diskette a MASTER diskette. An updated diskette will be able to boot any size Apple.

Well then, if your diskette is not a MASTER it must be a SLAVE. A SLAVE is

created when you initialize a diskette via the DOS INIT command. The DOS that is placed on this diskette is the same DOS that is in the memory of your Apple at the time of initialization. Therefore, a SLAVE diskette is created with DOS that is system size dependent. This is why a SLAVE diskette will only boot a system of at least the same size as the system upon which the diskette was initialized. This also explains "LOST" DOS--when you try to boot a 32K system and nothing seems to be happening (as if DOS were lost). What is probably happening is that the system is trying to load a 48K copy of DOS into a 32K system and there is no place to put it. Another point about the SLAVE diskette that should be made is if you boot a 48K system with a smaller (32K) SLAVE diskette, you reduce your 48K system to a smaller (32K) system.

Now, after you have had a chance to read this article, I hope some of your questions have been answered and problems solved!

Pascal course offered at the University of Minnesota. CSci 3104 INTRODUCTION TO PROGRAMMING AND PROBLEM SOLVING. Writeup by David Laden.

During the Winter 1980 quarter, I was enrolled in CSci 3104 offered through Extension at the University of Minnesota. The class met one night a week for 2 1/2 hours and lasted eleven weeks (including the final exam).

Here is the course outline we followed and the approximate time spent in each area (times given in number of class periods):

Time spent	
1	Problem solving and algorithms
1	Pascal language Programming with Integers
1 1/2	Decision making/iteration
1/2	Programming with real numbers
1	Functions (subprograms): Standard, Library, and Programmer defined
1/2	Midterm exam
1/2	Data types: Characters, Subtypes/subranges, and Programmer defined types.
1	Arrays Procedures (subprograms)
1 1/2	Larger programs/projects

- 1/2 Sets
- 1 Datatypes: Records, Pointer variables, and Files
- 1 Final exam

Three books were recommended depending upon the students programming experience. The first book, "An Introduction to Programming and Problem Solving with Pascal" by G. M. Schneider, S. Weingart, and D. Perlman, was recommended for the beginning student. For the more advanced student, "Programming in Pascal" by Peter Grogono was recommended. "Pascal User Manual and Report" by K. Jensen and N. Wirth was cited for general reference.

I was somewhat disappointed with the course because we seemed to move rather slow at times and some of the features of Pascal were not covered in as much depth as I thought they should have been. However, this course has no prerequisites and is meant to be a beginning course for those with little or no programming experience.

CSci 3104 is being offered through Extension during the summer. It is scheduled to meet on Monday and Wednesday evenings from 6:00 to 7:40 and run for five weeks beginning July 21st. The cost for this four credit course is \$82 plus a \$4 special fee for a total of \$86. This course is also offered during the day and in the evenings at various times throughout the year. For more information call the University of Minnesota.

INTERNATIONAL APPLE CORPS SOFTWARE & OTHER MATTERS-----

Noted during the last meeting and in the last newsletter was the subject of IAC software. A preliminary announcement from unknown sources indicated that IAC would sell diskettes of the software described in the Apple Orchard at \$12. Well that is not true! IAC will supply member organizations with diskettes of that software at no charge. Since that software is all in the public domain, it will be placed on the Mini'apples user bank. In fact, we hope to have the first two diskettes from IAC in our possession by the May meeting. The first diskette will contain the aforementioned Apple Orchard software. The 2nd diskette will contain Integer Basic in ROM. You Apple Plus Users, perk up; we repeat Integer Basic in ROM. If time permits we will allow copying of these two diskettes! Warning, the Integer Basic won't handle all program cases. Also included on that diskette will be a mini-assembler that will work on an Apple Plus!

IAC plan to distribute software periodically and it will be free of charge. We are not allowed to charge for the software although we can enact a media/preparation charge.

IAC will cost the club \$50 per year, the next such fee being due in January 1981. This is a small charge amounting to .25 to \$0.50 per person per year. It's even less significant considering the benefits incurred.

The IAC publishes the Apple Orchard. 80 of you forked out \$1 apiece in advance and received your copy at one of the last two meetings. In addition, Apple Inc. purchased 25,000 copies which they distributed to persons on their CONTACT mailing list. Those of you who purchased Apples through approximately mid 1979 'may' have been on that list. With the mailing of Apple Orchard, Apple announced that no further distributions of Apple Orchard or Contact would be made except through IAC. Meanwhile IAC have announced that Apple Orchard will be distributed quarterly, the next issue being published Sept 1st. As a member of Mini'apples you may receive a copy through bulk orders by the club at \$2.00 each. Copies ordered in this way will only be distributed at meetings. If you are independent, then you may order an individual subscription at \$10 for 4 issues (\$2.50 each). Or you may purchase them at your friendly neighborhood computer store for \$3.50 per copy! Details of subscriptions will be forthcoming.

We do apologize to any of you who received two copies, one from Apple and one through the club. If you did, bring it to the next meeting. I am pretty sure we can find some takers from members who neglected to order copies in advance or who were not members at the time.

Finally, IAC has and will be distributing a set of documentation called APNOTES. Copies of the first set are part of this newsletter. We will continue to distribute them as part of the newsletter unless they get too voluminous.

Post Script on IAC--

We are part of the Northern region and are represented by two directors one in Detroit and one in Chicago. If you have any comments we will supply names and addresses.

-----Your President.

COMMUNICATIONS
CORNER

by Nelson R. Capes

Starting with this issue, I would like to begin a regular column devoted to the subject of communications and how it affects the Apple user. This installment will examine the subject and why it is so complex.

What makes communications complex and difficult to understand is the nature of the channel involved. Most communications takes place over standard telephone lines which tend to be quite "noisy". This means that for every message there is a significant chance that one or more bits will be corrupted during transmission so that what the sender sends is not what the receiver receives.

Consider the case of two armies, Red and Blue, set up on hills facing each other. In the valley is a Green army that is the enemy. Green is just strong enough to defeat either Red or Blue, but not both together. Red and Blue want to schedule a dawn attack on Green. Note that if they foul up and don't attack simultaneously, one of them will be wiped out. Red and Blue coordinate their attack over a noisy channel. They may use homing pigeons, drums, smoke signals, or the telephone. All of these channels are subject to noise: pigeons get eaten by hawks, drumbeats get distorted by echoes, smoke drifts with the wind, and telephone lines are subject to "crosstalk". So a message may get corrupted or lost, and both Red and Blue know this. Comes the dawn! Red sends an 'attack' message to Blue. But Red knows that the message may get corrupted, and so can't attack until Blue acknowledges the message. Likewise, Blue sends an 'attack' message to Red, but must wait for Red's acknowledgement. But things are much worse than this! When Blue sends an acknowledgement to Red's 'attack' message, Blue has no way of knowing that the acknowledgement will get through. If it didn't, and Blue attacks, Red won't attack and Blue will be wiped out. Similarly for Red. So Blue has to wait for an acknowledgement to his own acknowledgement, and so on. Without any way around this protocol, the two armies will never attack, but will merely acknowledge messages forever!

This little parable points out the essential characteristic of communications channels that makes communications complex: noise. It's fair to say that all of the schemes ever devised in communications are primarily oriented around insuring the integrity of messages. This is what makes communications different from any other specialty in data processing.

In the next few months, I will attempt to present the essentials of data communications as it relates to the Apple user: hardware, software, and services. One of the things I will try to do is to keep an eye on new products and services that have to do with Apple communications.

How can the Red and Blue armies get out of this mess? It's safe to assume that they can NOT improve the communications channel to the point where it is noise-free; not without an infinite amount of money! One way out would be to accept a certain amount of risk; that is, assume that after a certain number of acknowledgements, both sides know pretty certainly that the other side will attack. But in the Two Armies problem, the consequences of an error are so disastrous that this is probably not a good solution.

There is no easy solution to the Two Armies problem. As I'll show in the coming months, various strategies have been designed to ease the problem; but it will always be there!

For a look at what can be done to check for noise on a communications channel, see the May issue of Kilobaud/Microcomputing, p. 52. This issue of Kilobaud also has a good discussion of half-duplex and duplex communication in the Dialup Directory on p. 60.

P.S.:

I will be on vacation most of May, so if anyone has questions on communications, save 'em up and ask me after Memorial Day!

* WORD OF CAUTION * by: D. Laden

for those using Applesoft and high resolution graphics.

Before you use a "CALL 62450" in your program, be sure you have initialized the high resolution pointers by executing HGR or HGR2.

Depending upon the contents of the uninitialized pointer (\$E6), a "CALL 62450" may erase part of the program in memory. This is especially true after a disk-to-disk copy using Apple's copy program. Apparently, the copy program uses some of the same zero page locations as Applesoft and does not return them to their original state.

NOTE: If you do not wish to initialize with HGR or HGR2 before you do a "CALL 62450" you can POKE 32 (\$20) into location 230 (\$E6) for hi-res page 1, or for hi-res page 2 you can POKE 64 (\$40) into location 230.

TRAC & COPY TRAC FILES

by D. Buchler

In last month's newsletter, we published a short article about the new magazine called Nibble. Unfortunately, we lost 3 lines of the article somewhere in the publication process. Within those three lines was mention of a program published in the Jan Nibble called TRAC (Trend Reporting Analysis and Control). This writer has spent quite a bit of time with TRAC. As published in Nibble, there are several flaws and bugs, none of which are serious. One does wonder if they ever ran the program on an Apple? However the program is essentially a great tool. Chuck Boody and I, between us, have fixed most of the bugs and made minor improvements (We think).

For those of you who have not seen the issue of NIBBLE in question, TRAC does the following:-

-You enter all of your living expenses, either checks, credit cards or cash. Each time you enter an expense, you classify it into one of 24 categories. Examples of categories are-

- Food
- Heating
- Vacation
- Clothing

You define the 24 categories by modifying a data statement in the program.

Once the data has been entered, you may print a credit card or check summary report. This report which may be sorted by date, check number, etc, essentially provides a hard copy permanent record of the detail. You then 'delete' the records comprising your check and or credit cards. The deletion process creates a balance file which maintains month-to-date and year-to-date totals for each of the 24 categories. At any time you may print either of the two major reports which are

YEAR-TO-DATE SPENDING PROFILE and
TREND ANALYSIS FOR MONTH.

The YEAR-TO-DATE PROFILE prints a column for each month of the year and a row for each category showing the totals for each month and each category. (Dates are entered with each item during record entry process. You can enter a Jan expense in the middle of Dec if you wish).

The trend analysis shows you average expenses by month for each category on a year-to-date basis and indicates deviations from that average.

The program is easy to use and serves well as a budget control program. We also plan to use it as a method of tracking club expenses. Anyone wanting a copy, must show that they are a bonafide subscriber to Nibble. We will then give a copy or they may key it in for themselves.

Because I am a person of nervous disposition, I do not like to use financial accounting programs without some sort of redundancy. TRAC always reads and writes the same 3 files namely

CARD #
CHECK #
BALANCES

If the balance file were to get clobbered you would be in deep weeds! The solution is a simple copy program that copies the above 3 text files to 3 backup records identified by a suffix which you supply when the program is invoked. This program is called COPY TRAC.

When the program asks-
ENTER DATE SUFFIX

you might type '5/21'

This would cause 3 records called

CARD # 5/21
CHECK # 5/21
BALANCES 5/21

to be created. If you accidentally bomb any of the primary files, just rename the backup files. The program also pauses after each read and write so that you can switch diskettes if you want.

Listing of COPY TRAC follows:-

```

LIST
1  REM 4/26/80
10 REM COPY TRAC FILES
15 REM BY D.B.BUCHLER
16 GOSUB 12000
20 HOME : TEXT : PRINT
25 D$ = CHR$(4)
26 DIM A$(2000)
30 PRINT "ENTER DATE SUFFIX ";
40 INPUT DAYS
50 F$ = "BALANCES"
1000 ONERR GOTO 10000
1010 PRINT "READING ";F$
1030 PRINT D$;"OPEN ";F$
1040 PRINT D$;"READ ";F$
2000 FOR I = 1 TO 300
2010 INPUT A$(I)
2020 NEXT
2030 PRINT D$;"CLOSE";F$
2040 PRINT "DONE READING BALANCE
S"
3000 PRINT "READY TO ";; FLASH :
PRINT "WRITE";: NORMAL : PRINT
" FILE ";F$;" ";DAY$
3010 GOSUB 9000
3020 PRINT D$;"OPEN";F$;" ";DAY$
3030 PRINT D$;"WRITE";F$;" ";DAY
$
3040 FOR I = 1 TO 300
3050 PRINT A$(I): NEXT
3060 PRINT D$;"CLOSE";F$;" ";DAY
$
3070 PRINT "DONE WRITING ";F$;"
";DAY$
4100 F$ = "CHECK #": GOSUB 5000
4200 F$ = "CARD #": GOSUB 5000
4300 PRINT "ALL DONE": END

```

DAN ON PRINTERS

I figured most of you who were interested in printers were at our March meeting and saw plenty of interesting things. You may have noticed that Centronics was conspicuous by its absence. Centronics has been in recent months overpriced compared to the plan an across the board price reduction of maybe 30%. That should bring them back into line. You may have all seen the Centronics 730 at your local computer store. There's a companion printer to the 730 which sells for about 25% more but adds 2 interesting features: A 9 wire head instead of 7 proportional character spacing capability. We expect to see alot of new hardware in the printer areas in the next few months.

PROGRAM REVIEWS by S. K Johnson MAVP

I found a game program on the new user bank called HUSTLE written by C. Kellner of Salem Or. It uses low Res color graphics (although color is not absolutely necessary) to create a challenging game whose object is to acquire more points than the previous high total. Points are gained by controlling a computer generated line with the U,D,L, and R keys to intercept randomly appearing color rectangles (each color being worth a different number of points). Sometimes just as you get near a rectangle it will disappear. A catch in the game is that the computer generated line can not intersect itself or the border of the playing field. Doing this terminates I RUN. This means you must plan ahead and know what keys you are hitting. You get 5 runs to break the previous high total. A display on the screen identifies colors and their point values as well as your current total and score to beat. I found this program to be well written, and documented, and it runs. It's fun and challenging to try to get a high score.

An interesting use of Apple's ability for sound and low resolution graphics is found in TRISLOT by Richard C. Burr of Brisbane, Ca. It uses sound and animation to simulate a one-armed-bandit from insertion of one dollar and pulling of the arm to coins dropping out the chute when (if) you win. Speaking of winning, it appears the odds aren't bad. I started with \$100. one night and never lost more than \$50. before lady luck paid off. "APPLE JACK" in only 2 HRS. I quit with \$260 (too bad it's only play money). Richard says he's a novice but you couldn't prove it by this program. TRI(slot) it you might like it.

```

5000 PRINT "READY TO READ ";FS
5004 GOSUB 9000
5005 PRINT D$;"OPEN";FS;"L40"
5010 PRINT D$;"READ";FS;"R0"
5020 INPUT R
5030 IF R = 0 THEN 11000
5040 FOR X = 1 TO R
5050 PRINT D$;"READ";FS;"R";X
5060 FOR Y = 1 TO 6: INPUT A$(X)
- 1) * 6 + Y) : NEXT Y: NEXT X
5070 PRINT D$;"CLOSE"
5080 PRINT "DONE READING FILE ";
FS
5500 FS = FS + " " + DAY$
5600 PRINT "READY TO ";: FLASH :
PRINT "WRITE";: NORMAL : PRINT
" FILE ";FS
5610 GOSUB 9000
6000 PRINT D$;"OPEN";FS;"L40"
6010 PRINT D$;"WRITE";FS;"R0"
6020 PRINT R
6040 FOR X = 1 TO R
6050 PRINT D$;"WRITE";FS;"R";X
6060 FOR Y = 1 TO 6: PRINT A$(X)
- 1) * 6 + Y) : NEXT Y: NEXT X
6070 PRINT D$;"CLOSE"
6200 PRINT "DONE WRITING FILE ";
FS
6500 RETURN
9000 PRINT "HIT ";: INVERSE : PRINT
"RETURN";: NORMAL : PRINT "
TO CONTINUE";
9020 INPUT B$: RETURN
10000 PRINT PEEK (222); " ERROR"
: END
11000 PRINT "1ST RECORD=0 ERROR
IN FILE ";FS;: END
12000 HOME : TEXT : NORMAL : PRINT
12010 PRINT "PROGRAM TO COPY TRA
C FILES"
12020 PRINT
12030 PRINT "THE 3 TRAC FILES - "
12040 PRINT "BALANCES"
12050 PRINT "CHECK #"
12060 PRINT "CARD #"
12070 PRINT "WILL BE COPIED TO 3
NEW FILES NAMED-"
12080 PRINT "BALANCES XXX"
12090 PRINT "CHECK # XXX"
12100 PRINT "CARD # XXX"
12110 PRINT "PRINT " WHERE
XXX IS A DATE SUFFIX ENTERED
"
12120 PRINT " AS DIRECTED BE
LOW"
12130 PRINT : PRINT "THE PROGRAM
WILL AWAIT HITTING OF RETUR
N"
12140 PRINT "BEFORE EACH FILE WR
ITE SO FILES MAY BE"
12150 PRINT "COPIED TO A DIFFERE
NT DISK"
12160 PRINT : GOSUB 9000: RETURN

```

TURNING THE PAGES with Dave Laden.

INTERFACE AGE -- MAY 1980

Advertising with the Apple Computer by Howard H. Rothman. Pages 66-68. This article discusses three rules for advertising and how advertising via computers, such as the Apple, can comply with these rules. A short Integer BASIC program called "Scrolling Wonder" is listed at the end of the article.

Comparison Charts: A First-Buyer's Guide to Hardware. Pages 94-102. Comparison charts for floppy disk drives, terminals, and printers are included in the series. Also included is a list of manufacturers (with addresses) for each chart.

Personal Software's VISICALC software product review by Tom Fox, systems editor. Page 144.

Interface Age Magazine is published monthly. The subscription rate is \$18.00 per year. Orders should be sent to: Interface Age Magazine, P.O. Box 1234, Cerritus, CA. 90701

KILOBAUD MICROCOMPUTING -- MAY 1980

Dial-up Directory by Frank J. Derfler, Jr. Page 60-61. This is a description of computerized bulletin board systems and related data communication terms.

Hashing Revisited by Raymond T. Vizzone. Pages 78-82. An explanation of hashing used in data storage and retrieval is given. An Applesoft program using the methods described is listed at the end of the article.

Programming Optimization Techniques by W. A. Harrison. Pages 92-93.

Physician: Automate Thyself by Robert Charles Bauman. Pages 101-102. The ideas presented in this article are not necessarily limited to doctors.

Program Property Rights by Joseph G. Wackerman, JD. Pages 197-198. One of the problems with computer software is proving ownership. This article shows some sneaky and not so sneaky ways to prove program ownership.

You Can't Hurt It by Pressing the Keys by Wallace Kendall. Pages 204-205. This article talks about letting kids play with your computer.

An Easy Way to Aple Hi-Res Graphics by Malcolm J. R. Clark. Pages 222-224. This explains how to generate easy hi-res graphics using Integer BASIC and the Hi-Res machine language routines supplied by Apple. A program listing and example graphics screens are also printed.

Kilobaud Microcomputing is published monthly. The subscription rate is \$18.00 per year. Send subscripstin orders to:

Kilobaud Microcomputing
Subscription Department
P.O. Box 997
Farmingdale, NY. 11737

PERSONAL COMPUTING -- MAY 1980

Improve Your Programming Effectiveness by Charles D. Sternberg. Pages 40-46. This article discusses the steps that should be taken before, during, and after the coding of your program. Even though the author is using BASIC, the steps and suggestions certainly apply to any language (Assembler, BASIC, or Pascal). Forms (which may be copied for personal use) are also provided to help you in achieving effective programming.

How to Buy a Printer by Ken Mazur. Pages 51-56. This article gives general information on selecting and purchasing a printer. Things to do and things to watch out for before committing large sums of money.

Printer Buyer's Guide by Ken Mazur. Pages 58-72. This article contains short notes about various printers, a printer comparison chart which lists over 70 printers, and a printer vendor guide.

Personal Computing regularly runs articles dealing with computer chess and computer bridge.

Personal Computing is published monthly. Subscriptions are \$14.00 per year. Send subscription orders to:

Personal Computing
Circulation Department
1050 Commonwealth Avenue
Boston, MA. 02215

Next month.....more articles and magazines.

WILD COLORS.....by D. Laden

For those of you with color t.v.'s or monitors and Applesoft in ROM, try out this program:

```
10 HGR
20 FOR X = 0 TO 255
30 POKE 28, X
40 CALL 62454
50 FOR D = 1 TO 500 : NEXT
60 NEXT
```

The delay loop in line 50 may be adjusted if it is too long or short.

Application Note

MINI APPLET

MAY 1980 NEWSLETTER

Page 9

Address	Symbol	Description
1000	TABSETTER	
1010	4/17/80	
1020	*	
1030	D.BUCHLER	
1040	*	
1050	PSEUDO-DRIVER TO ISSUE SPACES	
1060	ACCORDING TO TABS DERIVED FROM	
1070	THE SETTING OF LOCATION 36 (\$24)	
1080	*	
1090	SOFTWARE INVOKE BY A CALL 770	
1100	*	
1200	OR \$300	
1210	CH .EQ 36 (\$24) HORIZONTAL CURSOR	
1215	VE .EQ \$36	
1220	DOSETT .EQ 1002	
1230	DRIV .EQ \$C305 HAYES ENTRY	
1270	LDA #START STORE START OF THIS	
1280	* DRIVER IN OUTPUT VECTOR	
1290	STA VE	
1300	LDA /START	
1306	85 37	
0308	20 EA	
030B	60	
1320	RTS	
1330	*	
1400	START PHA SAVE A-REG (CHAR)	
1410	AND #7F STRIP HI BIT	
1420	CMP #50D C/R ?	
1430	BEQ RESET	
1440	LOOP LDA 'AB	
1450	CMP CH	
1460	BCC BLANK	
1465	LDA CH	
1470	STA TAB	
1490	INC CH	
1500	EXIT PLA	
1510	JSR DRIV	
1520	RTS	
1530	*	
1600	RESET LDA #0	
1610	STA TAB	
1620	STA CH	
1630	BEQ EXIT	
1640	*	
1700	BLANK LDA #520	
1710	JSR DRIV	
1720	INC TAB	
1730	JMP LOOP	
1760	*	
1770	TAB .HS 00	
2000	.EN	
033D	- 00	
033A	- AC 13 03	
0337	- EE 3D 03	
0334	- 20 05 C3	
0332	- A9 20	
0330	- F0 F2	
032E	- 85 74	
032B	- 8D 3D 03	
0329	- A9 00	
0328	- 60	
0325	- 20 05 C3	
0324	- 68	
0322	- E6 24	
031C	- 8D 3D 03	
031A	- A5 24	
0318	- 90 18	
0316	- C5 24	
0313	- AD 3D 03	
0311	- F0 16	
030F	- C9 0D	
030D	- 29 7F	
030C	- 48	
0300	- A9 0C	

TABSETTER by Dan Buchler

Some of you have heard of my endeavours in trying to make such programs as the SC-ASSEMBLER and FILE CABINET run with the HAYES modem. The idea being that one can make use of remote quality printing terminals to which many of us have access. Suddenly, last month I realized I had been incredibly short sighted! Rather than customizing each program, why not provide a simple machine code routine which solved the major problem, that is, the setting of tabs.

Most sophisticated programs control tab positions by POKING the monitor horizontal cursor 'CH' the address of which is \$24 (36). Applesoft TAB function also updates CH. The standard Apple serial and parallel interface printer driver firmware makes provision for CH. Unfortunately the Hayes firmware does not. It simply ignores TABS and value of CH. By interposing a subdriver between the Monitor or Basic and the HAYES firmware the problem is solved. This subdriver, called TABSETTER, simply looks at CH and compares it to its own pseudo tab position called TAB. If CH gets ahead of TAB, TABSETTER issues SPACES until TAB catches up with CH. As with the Apple printer firmware, if the line width is greater than 40 the screen display must be turned off. This is because CH is used as an index into the buffer display area which is assumed to be mapped for lines of 40 lines or less in length.

The following is the recommended procedure for using TABSETTER

1. LOAD TABSETTER, A\$300
2. POKÉ 1912 + SLOT, 146
3. CALL 770

Step 1 loads TABSETTER

Step 2 sets the HAYES flag word with the following bits on:

Bit 7 (128) - Do not display

Bit 4 (16) - Send line feed

Step 3 turns on TABSETTER.

To turn it off simply do a PR#3 assuming you are in SLOT 3. The following listing of TABSETTER is set up for SLOT 3. For other slots change the value of locations \$327 and \$336 to \$C0 + SLOT.

SYMBOL TABLE

CH	0024	VE	0036	DOSETT	03EA
DRIV	C305	START	030C	LOOP	0313
EXIT	0324	RESET	0329	BLANK	0332
TAB	033D				



P.O. BOX 976 DALY CITY, CALIFORNIA 94017

Application Note

March 10, 1980

This is a list of known PASCAL problems. Updates and fixes will be announced

1. The integer value -32768 prints as "--2768" and causes a compile error in the expression `I:=-32768` where `I:Integer`.
2. A long integer compare causes the system to crash.
3. The compiler does not allow `R:Real; I,J:Integer; R:=I/J;` which should be legal according to Jensen & Wirth page 147.
4. The BREAK key (ctrl-shift-P) does not cause a break during the execution of some programs.
5. A variable of type TEXT can be passed as a VAR parameter. This is OK according to Jensen & Wirth, page 157.
6. The Editor sometimes ends a file with 00 instead of the required OD00. This results in trash on the screen.
7. TTLOUT in Applestuff does not work.
8. Transcendental functions are not included in the APPLE3:CALC program.
9. The compiler will allow more than 9 segment procedures but only 9 will function properly.
10. Separate units do not work.
11. Erroneous placement of control characters may cause the Editor to go out of control, requiring the user to delete text in order to recover.
12. When compiling using the `(*$L+*)` option, the compiler may damage the contents of the diskette.
13. Intrinsic units cannot use non-intrinsic units, and vice versa.
14. Intrinsic units cannot contain references to files.
15. Long integer constants are not implemented.
16. The IN function for set inclusion does not always work when the first argument is negative or greater than 511.
17. A run-time stack overflow crashes the system instead of re-initializing.

18. The MEMAVAIL function may return incorrect results in some cases.
19. .PUBLIC and .PRIVATE assembler variables may be relocated incorrectly at run-time.
20. There is no way to tell the compiler to allocate all available disk space for the code file, so the compiler may run out of room for the code file even if sufficient space is available.
21. The ORD function accepts REAL and pointer arguments even though this is incorrect.
22. Negation of BOOLEAN variables do not turn FALSE to TRUE and vice versa.
23. The compiler allows underscores in an identifier, but ignores them.
24. Functions cannot return STRING values, although this is implied in the Apple PASCAL Reference Manual.
25. Standard PASCAL syntax allows the field list and <variant> in a RECORD declaration to be null. The UCSD compiler does not allow these items to be empty.
26. Due to the normal inaccuracies in representing REALs, some equalities may not test true. For example, LOG(10) prints as 1.00000 but does not yield equality in the comparison LOG(10)=1.00000.
27. Documentation for the SCAN function is incomplete.
28. The example of MOVERIGHT is incorrect.
29. The Editor does not report assembler errors in the same way that it reports compiler errors.
30. The function KEYPRESS exists in Applestuff, but is not documented.
31. Overflowing the code file causes the system to crash in various ways.
32. The Editor informs the user when it is about to run out of space for the file buffer, but strange things may happen when the file is completely full.
33. The compiler INCLUDE directive does not always work as expected.

34. Use of the compiler swapping option causes global declarations to be ignored.
35. The manual does not clearly state that the string argument of the .TITLE pseudo-operation must be enclosed in double quotation marks.
36. Not all options of the Editor are listed on the prompt line, and the "?" option to get a list of the additional options is not supported.
37. The reference manual does not adequately describe how to set up a program to be executed automatically, using SYSTEM.STARTUP.
38. The Editor only accepts files with names of the form <file>.TEXT.
39. The Editor is insensitive to certain punctuation conventions during paragraph reformatting.
40. The Assembler should translate all alphabets to upper-case.
41. The documentation is inadequate in the description of assembly language and various options.
42. The Compiler may generate an error #407 (too many libraries).
43. The compiler generates code to always load segments 28-31 if they are present. This may cause undesired loading.
44. When the NOLOAD option is used for an intrinsic unit with a data segment, the compiler fails to generate code to unload the data segment at the end of execution. Thus, if the program is run again, the data segment will not be loaded since the interpreter segment table will show that it is already there.
45. The ATAN function gives an erroneous result for an argument less than -1.00.
46. BIOS does not turn off the high-order bit of characters handled by the remote I/O routines.
47. The P1 parallel printer PROM appears to be incompatible with BIOS. The P1-02 PROM is required.
48. The listing file produced by the Assembler does not follow the standard for text files as described in the Apple PASCAL Manual.

49. There is some confusion about the differences between the built-in variable KEYBOARD and the volume identifier SYSTEM:.
50. The fact that the Editor requires tedt files to have an even number of 512-byte blocks is not documented.
51. If a run-time error occurs while in graphics mode, the screen does not switch back to text mode to display the error message.
52. The Cross Reference program on APPLE3 does not close and lock the output file.
53. The Compiler does not check the declared length of STRING parameters passed by reference. Therefore, if the declared length of the actual parameter is less than that of the formal parameter, assignment of characters into the formal can clobber space beyond the end of the actual parameter without detecting any error condition.
54. Using an uninitialized long integer may cause unpredictable problems because there may be illegal bit patterns that are not representations of digits.
55. The Compiler QUIET option (Q+) does not turn off all output to the console.
56. The system will crash if the system disk containing the Editor is not on line when returning from a Copy File command when the system disk was replaced for a disk containing the file to be copied.
57. The interpreter code for floating point comparisons returns $0.0 > (-0.0)$.
58. DIV and MOD functions give incorrect results for certain combinations of signs for the arguments.
59. The code to clear SYSCOM is incorrect in the procedure BOOT.
60. The figures given for the maximum integer values for each declared length of long integer are incorrect in the Apple PASCAL Manual, page 198.
61. Source code for APPLE3:LINEFEED should be made available.

SCANNING THE APPLE NEWSLETTERS

Now that Dave Laden is providing an excellent abstract of Apple related articles from the national Microcomputer magazines (See TURNING THE PAGES elsewhere in this issue), I will restrict myself to club newsletter derived information.

From April 1980 Apple Pi of Colorado according to a letter from Dennis Baer of Hicksville, NY, a Canadian group is coming out with an Applesoft Compiler!

The 4 Chicago area Apple User groups have formed a consortium called the Chicago Apple Council. The member groups are:-

- Northwest Suburban Apple Users Group
- DuPage Apple Users Group
- Apple SIG-CACHE
- Apple PIE

They plan to publish a combined newsletter with co-editors from each of the four groups.

According to Harvest, the NSAUG newsletter (see above), FORTRAN will be available from Apple sometime in 3rd quarter of this year. Cost will be around \$200. Presumably it will require a language card!

PASCAL SPECIAL INTEREST GROUP by K.Madonna

A special interest group is going to be organized by Keith Madonna. About 20 minutes of each meeting is to be used to explore new ideas, methods, and exchange programs between users. The interest group meeting will start after the regular normal meeting. The pascal programs are only available at the meeting, and sometimes may be transferred by phone. At this meeting there will be a display of the Sup'R'Terminal board from M&R Enterprizes, with Pascal. The board does work well with U.C.S.D. Pascal by Apple. The board allows upper&lower case from the keyboard. The only modification was to MISCINFO by use of the SETUP program on the APPLE3: volume. This mod was to the screen width, to allow full prompt lines to be displayed. Horizontal scrolling and screen switching are no longer needed.

MINI'APP'LES
13516 Grand Avenue South
Burnsville
Minnesota
55337