$1.50

Washington Apple Pi

Volume 3 January 1981 Number 1

Highlights

A Questionnaire Subroutine

Notes on Hi-Res Graphics Routines in Applesoft

Fast Walsh-Hadamard Transform

In This Issue

OFFICERS & STAFF

President	-Bernard Urban	(301) 229-3458
Vice President	-Rich Wasserstrom	(703) 893-9147
Treasurer	-Bob Peck	(301) 468-2305
Secretary	-Dana Schwartz	(301) 621-2894
Members-at-Large	-Mark Crosby	(202) 488-1980
	-Sandy Greenfarb	(301) 674-5982
	-Hersch Pilloff	(301) 292-3100
Editor	-Bernard Urban	(above)
Associate Editor	-Rich Wasserstrom	(above)
	-Genevie Urban	(301) 229-3458
Librarian	-David Morganstein	(301) 972-4263
Group Purchases	-Howard Lefkowitz	(301) 649-3373
SIG Chairmen:		
ASMSIG	-Jim Rose	(301) 730-2230
EDSIG	-Chuck Philipp	(301) 924-2354
NEWSIG	-Al Weiner	(301) 468-0663
Pascal SIG	-Tom Woteki	(202) 547-0984
SIGAMES	-Al Gass	(703) 371-3560

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

ABBS (301) 983-9317

Membership dues for Washington Apple Pi are $18.00 per calendar year. 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.

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 January meeting will be on January 24 and the February meeting will be on February 28.

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, or call the President at 229-3458.

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.

CLASSIFIEDS

FOR SALE: Integer Card, with all documentation. $125, call Bill Bowie, (301)924-3455.

FOR SALE: Disk and 16-sector controller with DOS 3.3 - $510; drive only - $415. If you want either of these, please call or write and I will bring them to the next meeting. David P. Kemp, 1307 Beltram Ct., Odenton, MD 21113, (301) 796-6546 (work), (301) 674-2690 (home).

WANTED TO RENT: For a trial term, APPLE II with disk drive and printer. Call or write Marguerite Kelly, 420 Constitution Ave, NE, Washington, DC 20002, (202)544-5698.

EDITORIAL

First, let me thank the Maryland Apple Corps and you for nominating me for re-election as the Director, East Coast, of the International Apple Core. IAC has grown in less than a year to an APPLE Users Group comprised of more than 200 clubs. While it has provided member clubs with APnotes, software and the Apple Orchards, it has only scratched the surface in providing you the services I envision such an organization is capable of providing. Hopefully, if re-elected, I shall be in a position to make some of these things happen soon.

You will find reproduced within this issue the first issue of the IAC News Bulletin, which I have volunteered to prepare monthly. Many of you have wondered out loud about the IAC. The Apple Orchard which is published quarterly cannot provide information about IAC developments in a sufficiently timely way. Thus my reason for the News Bulletin. The January issue which has already gone to press will tell you a little about the IAC SIGs and Committees. Let me also say that Mark Crosby will be featured in the Orchard as editor of new product information, and "Dr. Wo" is chairman of a SIG on APPLE languages, e.g. Pascal, Forth, Fortran, etc. He will be responsible for a workshop/panel at the annual IAC meeting next May in Chicago.

Second, let me thank all December attendees for the overwhelming show of support for the officers, as shown by your vote in favor of the 1981 budget. Wow!

Finally, let's all help in making our new SIG, the SIG/Disabled, a major success. It's needed. We need some volunteers if possible to act as interpreters for the deaf. Any takers?

I would like to remind you to pay your dues for 1981, if you have not already done so. (See Dues Reminder, page 4 .) We are beginning to be more than just a newsletter/software source. Your dues make these new services possible.

MINUTES

EXECUTIVE BOARD MEETING

The Washington Apple Pi Board meeting of December 10, 1980, was held at the home of the President and was called to order at 7:00 PM, with 12 members present. The primary topic of the meeting was a review of the proposed club budget for 1981. Rich Wasserstrom, Vice President, chaired this portion of the meeting. The proposed budget was reviewed, discussed and voted on, line by line. The results of this were reflected in the budget mailed to the members in December and voted on at the December 20 meeting.

contd.

Other business involved a discussion among NOVAPPLE Board members Theron Fuller, Tom Lucas and Nick Cirillo, and Washington Apple Pi Board members. This discussion centered around the relationship of the two clubs. The WAP Board voted to extend the present special arrangements with NOVAPPLE for six months. This gives NOVAPPLE members the right to buy our newsletters at dealer costs, to purchase library disks at member costs, and to participate in WAP group purchases. NOVAPPLE will in turn offer member privileges to WAP. Discussions will continue between the two clubs to work out future relationships.

It was voted to charge dealers 85 cents per copy for the newsletter beginning January 1981. The meeting was adjourned at 12:45 AM.

Genevie Urban

GENERAL MEMBERSHIP MEETING

The Washington Apple Pi meeting of December 20, 1980 was called to order at 9:35 AM by the President with approximately 160 persons in attendance.

After a short review of how the business of the day was to be conducted, the program was turned over to Eric Ganien of the Source for a presentation on some of the features of his company's services. A lively question and answer session followed, with both praises and criticisms of the Source being voiced.

The business portion of the meeting began with a report on group purchases status by Howard Lefkowitz. The proposed 1981 budget was then presented to the membership by the budget committee. A motion to accept the budget as proposed was passed by an overwhelming majority.

The meeting was adjourned at 11:00 AM for SIG meetings and Holiday refreshments provided by several generous members.

Dana J. Schwartz, Secretary

NOVAPPLE MINUTES

The NOVAPPLE meeting of December 10 at Computers Plus was called to order at 7:30 PM by the President Theron Fuller. The first order of business was the meeting schedule for 1981. We voted to make the Saturday meeting a permanant addition to our schedule. We will meet on the second Saturday of the month at the Kings Park Library on Burke Lake Road in Fairfax County. The emphasis of the Saturday meetings will be reviews, evaluations and demonstrations of hardware and software. NOVAPPLE will also continue to meet on the fourth Thursday of the month at Computerland of Tysons Corner. This will be the regular NOVAPPLE meeting with business meetings and programs. On the second Wednesday of each month we will be using Computers Plus in Franconia to hold classes and tutorials. The classes and tutorials will not be business meetings at all, and will be open to NOVAPPLE and Washington Apple Pi members only on a signup basis.

Tom Lucas then distributed some copies of the International Apple Core application notes. These notes may be borrowed by members, and returned at a later meeting. Any member who can make additional copies is requested to do so.

The disk librarian gave a report that NOVAPPLE now has six disks ready to start the library. They will be ready for distribution as soon as they are reviewed, and a distribution process can be set up. It was voted that all disks be distributed for a one dollar per side copying fee, and the user will supply his own disk. The disk library should be ready for distribution by January.

The remainder of the meeting was devoted to a discussion of modems and computer bulletin boards. Craig Vaughan gave instructions for using his bulletin board and discussed some of the conversations and incidents that have occured on his system.

The meeting of December 13 was called to order at 1:00 PM at Kings Park Library by President Theron Fuller. Shelley Kramer gave the status of the third purchase of Dysan diskettes for reduced price. This will probably be the last time to get the current price.

We discussed the current and future relationship between NOVAPPLE and Washington Apple Pi. Washington Apple Pi has offered to all NOVAPPLE members the purchase of the newsletter at dealer cost, the purchase of their disk library at member price, and the right to participate in group purchases. This relationship will continue for at least the next six months. In return NOVAPPLE will offer Washington Apple Pi members the right to attend all presentations and tutorials, the right to purchase NOVAPPLE disk library at member prices, and the right to participate in NOVAPPLE group purchases of hardware and diskettes. In addition the board of NOVAPPLE and the board of Washington Apple Pi will continue to meet to determine how the two clubs can work together for their mutual benefit.

PROGRAMMING TIP

If you are programming in Applesoft and you want to determine if a file of a particular name is already on a diskette then setup an ONERR condition and RENAME the file to itself. If the ONERR is not executed then the file was present. If the ONERR was not taken then there was no file by that name.

The remainder of the program was on computer generated music. There were demonstrations of the nine voice ALF II board and the three voice ALF board.

"Every start upon an untrodden path is a venture which only in unusual circumstances looks sensible and likely to be successful."

-Albert Schweitzer

SIG-NEWS

SIGAMES is the special interest group of computer hobbyists interested in having fun with their APPLEs through some aspect of games. The main meeting of this group is held at a location announced at (if Bernie gives me a chance) and following the Washington Apple Pi monthly meeting.

This month's meeting will continue arrangements for construction of joy sticks for the APPLE II's game I/O socket. There appears to be interest in joy sticks at two different price ranges. Come to this month's meeting to express your inputs in design and arrangements for their construction. Group purchase of the components needed to build joy sticks is planned. If you can't attend the SIGAMES meeting, call Bill Bowie at (301) 924-3455 to let us know of your interest. Group sessions will be held during the construction phase to insure that novices get the right things soldered together. Bill Bowie and Brian Dormer are co-chairing this project. Thomas Steck will start the meeting off with a product review of NORTH ATLANTIC CONVOY RAIDER by Avalon Hill.

Al Gass

The NEWSIG group, by request of the thirty members present at the December meeting, has changed its meeting time to immediately following the regular Washington Apple Pi meeting.

For brand new APPLE users who need real beginning information, our kindergarten level will take place during the regular WAP meeting. The grade one level will meet immediately after the regular meeting, and will cover a multitude of questions and have a hand-out that we can give to all new members.

HELP - We need volunteers to help answer the questions. We also need you new members to call in the questions before the meeting. In addition, we need to know of some good tutorial software that is available for beginners just setting up their APPLE.

The next meeting, on January 24, will feature Steve Sondag, who will talk to the kindergarten group during the regular meeting. He will tell the brand new members about Washington Apple Pi, how we operate, how to order disks, and other ground level information that new members need to know, and he will answer their questions. The regular, or grade one level, will meet after the WAP meeting. All of the programs on the DOS 3.3 Master Disk will be discussed, along with some of the programs on Vol. 1 and Vol. 2 of the utilities.

The following members have graciously volunteered to help and have allowed their phone numbers to be published to help those in need:

Bob Chesley	Paper Tiger, Supertext II.	560-0121
Sara Lavilla	Kindergarten.	926-6355
Boris Lavine	Kindergarten, Debugging a copied program.	229-5730
John H Smith	Lo-res graphics, LOAD, SAVE on cassette tape, simple Basic and Applesoft.	(301) 439-4388
Steve Sondag	Z80 Card, modem, printer.	281-5392

The Pascal SIG meets on the 3rd Thursday of each month at the Uniformed Services University of the Health Services, Bldg.A, Room A2054 (2nd floor), near the National Medical Center at 4301 Jones Bridge Road, Bethesda, MD. Look for "USUHS" sign, underground parking. For further information call Dr. Wo at 547-0984.

EDSIG will meet immediately following the regular WAP meeting on the 4th Saturday.

ASMSIG, the Assembly Language SIG, meets on the 3rd Monday at 7:30 PM at the Computer Science Center, University of Maryland.

SIG/Disabled
by Curt Robbins

The December meeting of Washington Apple Pi was my first - and certainly not my last. The thing that turned me on was the abundance and availability of software and hardware at discount prices, and a glorious collection of excellent brains to pick for stimulating new ideas, keeping abreast with new technologies, getting and/or giving assistance for debugging programs or problems, and so many more reasons for typical WAP socialization. I think I need this interaction and interface so much more than most of you can imagine - and I am not alone.

There was so much information flowing around that there is a definite need to provide certain accomodations for disabled APPLE owners who wish to attend WAP meetings. For example, I am hearing impaired. I would gain so much more from WAP than my eyes can bear if an interpreter was present. (An interpreter is a hearing person with sign language skills with an ability to interpret, or translate, verbal communications into signs.) I know about five or six hearing impaired APPLE owners. Offhand, one or two would be willing to sacrifice their time and efforts trying to communicate their APPLE needs to WAP members without the assistance of an interpreter. Most of the others would rather stay home and tackle the communication problems they encounter with the APPLE than come to a WAP meeting - trying to tackle verbal communication problems. Surely, with an interpreter they would gain a great deal at any meeting. Thus the purpose of this committee.

Also, there are other APPLE owners who are physically disabled and may not (or could not) attend WAP meetings unless certain accommodations are available to them. Needless to ask, if they are APPLE owners like you and me, don't you think they deserve a fair share of your enthusiasm for learning more about APPLE II?

If you know any disabled APPLE owners, please let me know. Also, I will need some assistance from the WAP membership. I need ears and eyes!! Write me at 8805 Barnsley Court, Laurel, MD 20811 or catch me at the January meeting if you have any information or wish to assist in this committee.

Notices

DUES REMINDER

Those members who have not brought their 1981 dues up to date should do so before the end of January in order to assure continuation of your newsletter. Please fill out the form in your December newsletter and send it in with your check. Dues for the calendar year 1981 (January 1 to December 31) are $18.00. Most members who have not renewed owe for the entire year. This will be indicated by a 8012 expiration date following your WAP number on your December label. Exceptions, with their tabulated amounts, were listed on the December renewal form. We are trying to get everyone on the same expiration schedule, and this is why the varying amounts.

MEMBERSHIP CARDS

We will be issuing WAP membership cards for 1981. The cards will be distributed as soon as we can get them typed up, hopefully no later than the February meeting or in your newsletter if you do not attend.

SALES AND PURCHASES

Sales and distribution of group purchases will take place between 8:15 and 9:30 AM before the general monthly meeting. Library disks will be distributed only between 8:30 and 9:30. Members are urged to order their library disks by mail in ample time to have them prepared for pickup before the meeting. A $1.00 surcharge per disk will be added to those ordered and picked up on the spot.

Also, members are urged to pay with checks at the meetings. Our Treasurer does not like to be responsible for large sums of cash. So remember to bring your checkbook if you plan to do money-business before the meeting.

ABBS

If you want to sign onto the Washington Apple Pi ABBS, please call the club phone, 468-2305, and leave your name and WAP number. This message will be forwarded to John Moon who will take care of signing you up.

BACK ISSUES OF NEWSLETTER

Those of you who have requested back issues of the newsletter, please bear with us. We are trying to get some copies reproduced and they should be ready soon. Feel free to "nag" if you have requested back issues and do not receive them in the near future (Genevie Urban, 229-3458). Also, we are in the midst of compiling the "Best of Washington Apple Pi", 1979 and 1980. This is a time consuming job, but we hope to have it ready for sale in the near future.

WAP NUMBER 500?

Our membership is now past 450. To our 500th member we are planning to award a special prize of 10 free library disks, of the recipient's choice.

PEN PAL, ANYONE?

We have a new member from England (Canterbury, Kent) who would like to exchange information with any member(s) who are interested (sort of a one-man NEWSIG). He reports that there are very few resources for APPLE support there. For anyone inclined and willing, it should prove an interesting and informative experience. Write Duncan Langford; 20, Lesley Avenue; Canterbury, Kent; ENGLAND.

A Page From the Stack - Librarian's Corner by Dave Morganstein

We have many new releases to add to the library. These include: five new EAMON adventures; a new "games" disk; three special interest disks; and the first DOS 3.3 format disk. Below I'll fill in a bit.

The three special interest disks are: a check-book based personal accounting package called CRAS; a SOLAR TUTOR disk; and a GARDEN MANAGEMENT package. The first came from the San Diego Apple Core. The SOLAR TUTOR came from Jim Reilly. Fred Sharp assembled the GARDEN MANAGEMENT programs.

The DOS 3.3 disk is a utility disk. It came via the IAC and contains most of the Lawrence Hall of Science utilities modified for a 16-sector format.

Commercial Software: Several games which have impressed me include: A STELLAR TREK from Rainbow Computing and GALACTIC TRILOGY from Broderbond. The former is a Hi-res version of the classic Star Trek, adding realistic ships and stars to the quadrant display. Many embellishments can be found among the expanded command list.

GALACTIC TRILOGY is a brand new idea which is very well done. There are three games in the series, each with a different theme. One is a trading adventure, buying and selling as you move about the universe. The other two involve conflict and combat. They are all Hi-res and use a "computer" to call up data and calculate navigation parameters. The jump into hyperspace is done quite well.

See you again ....

An Invitation to be a Lobbyist by

Charles C. Philipp

This is the time of the year when most public school systems are working on school budgets for the fiscal year starting July 1, 1981. It is a good time for interested citizens to start the process of becoming knowledgeable about how learning opportunities for children are developed and funded. There are many public documents available and most officials are responsive to citizen inquiries about school instructional programs. For these reasons, it is an appropriate time to express interest in and ask critical questions about what schools are doing to promote computer literacy for children.

Your efforts in this area may be personal and limited to talking about this subject with your friends. This is helpful because your interest adds to public awareness about the educational needs of children. If you are a member of a PTA, you might mention computer learning opportunities for students and ask for a presentation or information about school system plans. Another possibility is to obtain a copy of the proposed school budget and to then ask questions and make notes. Save the notes for future reference.

If you have a deep interest in children and schools, you may wish to make a presentation at a public hearing or write a letter. Most school systems have budget development procedures that provide for citizen comments. The focus of your concern may be one student, one classroom, a school or a whole school system.

How you structure what you say is important. One kind of message is to express support for an existing program. Sometimes program support efforts are overlooked because parents feel secure about already having computer learning opportunities for students. However, in a era of tight resources almost any instructional program may suffer funding cuts. This is particularly true for computer literacy programs because many people do not yet recognize that learning how to use computers is one of the most important basic skills.

When you support an existing program try to state a few specific things that are positive results of that program. What you talk about may be related to formal school work or may have to do with independent learning activities that developed as a by-product of a school program. Both kinds of information are useful to school budget makers and curriculum specialists.

If your school system does not provide a computer literacy program for students, or has a very limited one, you might want to send another kind of message to the people who make budget decisions. That message should focus on the rationale for developing and/or expanding a program for students. Use your own knowledge base to build that argument.

A particularly good approach is to use information that you have about trends in business and industry. Describe what you know about skill requirements in various careers, now and in the future.

Another way to build a case for K-12 program development is to point out that there is a growing educational equity issue. Some children are learning about computers and how to use them because their parents have purchased a computer for home use. This means that certain students are acquiring valuable skills that make them attractive candidates for post-secondary education and job opportunities. Other equally bright but less fortunate children will be graduated from high school lacking appropriate skills.

Much of what is currently done in K-12 classrooms is justified in the name of basic skill development for children. Too frequently the reality of "basic" education means long hours of drill and practice at a mechanical level. The fact is that the word "basic" means different things to different people. The National Council of Teachers of Mathematics (NCTM) has recognized this problem. In a recent publication, AN AGENDA FOR ACTION, the NCTM made eight recommendations for the 1980s. The first three recommendations are that --

1. problem solving be the focus of school mathematics in the 1980s;
2. basic skills in mathematics be defined to encompass more than computational facility;
3. mathematics programs take full advantage of the power of calculators and computers at all grade levels.

A copy of the above mentioned publication may be obtained by contacting NCTM, 1906 Association Drive, Reston, VA 22091. If you have the time and inclination to be a continuing advocate for the use of computers in the classroom, this publication is an excellent resource.

Your efforts to promote the use of computers in K-12 classrooms do not have to focus on the learning of mathematics. There are many other applications that range from language learning to art, music, social studies and industrial arts. In addition, elementary and secondary science programs are in desperate need of support.

It is no secret that computer science is here to stay and that educators ought to do something to develop programs. Public school systems have been slow to act because of a lack of money for equipment and the fact that relatively few teachers know much about computers.

Another problem is that public school curriculum developers are not sure about what a K-12 computer program should contain. There are two very different points of view about how computers should be used.

In the past, much money was spent developing computer-assisted instruction (CAI) programs. For the most part, these

contd.

programs were designed for relatively large and expensive computer systems that provided for many computer terminals, devices that have a video screen and a keyboard. Students would sit at the terminals and focus their attention on a video presentation, sometimes supported by books, and then respond to questions.

When students use CAI programs, the computer, or more properly the computer program, is in control of what students are doing. The educators who design the programs, sometimes called course-ware, make assumptions about what students should learn and how they ought to learn it. In short, the process and products of learning are pre-determined by "teachers" who do not know the students. CAI programs cost a great deal of money to develop and have a record of rather poor quality. When educators talk about the cost of course-ware or software to support computer education opportunities for children, they are generally thinking about CAI applications in the classroom.

There is another point of view about children, computers and education. It is based on the belief that children ought to have the opportunity to learn to control computers. When computers were large and expensive, not many children had the chance to demonstrate that they could do this. The microcomputer has changed that. There is now much evidence that children can learn to program and control computers. This point of view is developed in an elegant way by Margot Critchfield in an article contained in the October, 1979, issue of EDUCATIONAL TECHNOLOGY; its title is, Beyond CAI: Computers as Personal Intellectual Tools.

The July, 1980, issue of BYTE, the small systems journal, contains several articles about computers and education. Three of the articles are not technical and are excellent for providing background information about education. One article, Books as an Antidote to the CAI Blues, is by Tom Dwyer. The case for CAI is built by Lou Frenzel in The Personal Computer--Last Chance for CAI? The third article, Computer Illiteracy--A National Crisis and a Solution for It, is by Arthur Luehrmann of the Lawrence Hall of Science in California.

The above suggested readings represent one way to get started in the process of educating yourself about computers and education. You do not have to be any kind of an expert to do this. Another way is to find small children operating micro-computers; watch them and ask questions. Notice how they do not worry about making errors. Also notice how they correct the errors (this process is called debugging) by trying different things, by asking another child, or by comparing one thing with another. If you are fortunate, you will find children who have had access to computers long enough to be making their own programs. Computers are like pencils and children need practice time to get them to do what they want them to do. Observe that children tend to acquire knowledge of a programming language in a way very similar to the learning of a spoken language.

You may have difficulty finding children doing the above described things. The reason is that a special kind of learning environment is required for that to happen. Too often adults make judgments about what children can or cannot do based on what adults know about their own learning ability. Most parents and teachers tend to be more comfortable when viewing children as dependent learners and generally cause this to happen.

The point of the above two paragraphs is that serious advocates for the use of computers in the classroom should spend some time observing children. The focus of observation should be on the thinking skills that children use as they learn to control computers. Computers will not replace teachers or books, they will add something qualitatively different to education. What is added will greatly facilitate the traditional kind of school learning and will also provide for new levels of individual creativity. This is especially true of the microcomputer. In the words of Margot Critchfield, "...the microcomputer is a distinctly new phenomenon. It is a technology that resists being used as the pipeline for pre-digested ideas and, by contrast, lends iteself to independent creation by the individual."

Frequently, concerned citizens do not know enough about the operation of schools to ask the right questions at the right time. Thus, as you talk to friends, attend PTA meetings, read school budgets, and perhaps write letters, think about the following questions:

1. Does the school system have a plan for implementing computer education activities for all students? If yes, ask for a copy and/or a description of the plan.

2. Does the plan provide for the training of teachers and administrators? If administrators don't understand the plan and its goals, it will never be implemented.

3. What is the instructional focus of the plan? Is it all CAI? Will children learn how to program and control computers? Being educated by computers is not helpful in getting a job. What counts is knowing how to use and control computers.

4. Is the computer equipment also used for administrative purposes? If yes, what are the rules about who gets to use the equipment and when? Think about a typing class--do administrators ever take a machine away from a student for office use?

5. How much equipment is available? When you have a number, break it down to the number of minutes per day per student. Is it reasonable?

6. Is there computer equipment in the library or a learning lab? Students should have access to computers as a support for doing homework and independent learning.

7. Do business students have an opportunity to learn to use word processors?

contd.

This skill is also an advantage for college preparatory students.

8. Are microprocessor applications taught in science and other related technology courses? Microcomputers are now commonly used to monitor temperature, pressure, light intensity, etc. These kinds of applications are appropriate for high school students and ought to be included in the curriculum.

The list could go on, but this is a good start. Keep in mind that there is probably not a public school system of any size in the entire country that is now doing all of the things suggested above. However, the above questions are not unreasonable. In fact, when one considers what is now happening in business, industry, government, and the military, the questions are really rather conservative. Think about the students who are in the high school graduating class of 1992. What kinds of jobs will they find and enjoy in the next century?

The Wonderful World of B.I.T.S - a Review by Gerald S. Eskelund - NOVAPPLE

It is not often that you get new software and know you just have to write a review. This is one of those times. The software package known as B.I.T.S was written by Craig Vaughan, Microsoftware Systems, 7927 Jones Branch Drive Suite 400, McLean, Va. 22102. The programs are for transferring information via a modem, as well as for having your computer act as an intellegent terminal. The system works with an APPLE with as little as 32K and a disk drive. You need a modem and some sort of serial card to complete the package. The software supports a number of peripherals such as Micromodem, Double Vision Card,and Dan Paymar's lower case adapter. It also supports the serial AIO card and/or Apple Communication Card with an acoustic coupler. It allows one the freedom to contact new friends all over the world via phone. The software is easy to use and the documentation is well laid out and easy to read. Outside of a few typos the documentation is more than one normally gets with software. It has an index and is divided into three sections:

1) overview and startup
2) details of operation
3) support programs.

I have tried to exercise or use most of the features and have found them all to be excellent. For instance, the modem program allows you to set up your system once and then every time you boot the disk all of the program parameters are set for your system. If the system changes, then you can reformat the program by running the Bits Configure. The first thing that happens after you have set up your machine and booted the disk, in Applesoft, is that a list of modem numbers is loaded. Since I don't have a Micromodem, the program tells you to pick the number of the ABBS you desire from the list furnished. After you select the number, it tells you to dial and then hit any key to connect. The modem program can be formatted to the 40 character screen format. This is especially useful when you call a number of systems which are not APPLE systems. Using the modem, I have been able to talk to TRS-80 systems, large computers such as a CDC 6600 and several other homebrew CBBS's, with no problems. In addition to the modem program, there are several utility programs which make the transfer of files a piece of cake.

Before I discuss the utilities, there are several more features of the modem program which need to be mentioned. To enter the other options all you have to do is hit "ESC" and the other options are instantly available without losing the carrier. A menu is displayed with either 11 or 9 options depending on the type of modem. The Micromodem version is the 11 option mode. It allows you to exit and stay on line, exit and hang up, dial a system, turn copy on/off, save file, transfer file to remote system, turn off line formatting, set duplex mode, clear buffer, send break, and send escape. All of these commands offer complete control of the computer and the modem connection at the same time.

Files can be transferred easily from one terminal to another but they must be text files. The utilities are designed to expedite the making of text files. They are fast and simple to use. There are three programs to make text files, one for Integer, one for Applesoft and one for Assembly Language. When a program is received as a text file, then "File Print", a B.I.T.S utility, allows you to print the file to the screen or a printer, and/or edit the file. Another feature utility is called Spacecrunch. This allows one to compress a text file by removing the spaces so that a maximum of file can be sent in a minimum of time. After trying each of the parts of the software I have found all of them to be simple to use and they perform as documented.

The software costs $34.95 at your local dealer and if they don't have it have them contact Craig Vaughan at the address at the beginning of this article or on his ABBS which is 703-255-2192. Once you get B.I.T.S be sure to try the ABBS--it is the best I have seen to date.

DATA CAPTURE 4.0 BUG AND FIX

by R. J. Decker

Data Capture 4.0 has a bug which will cause the system to hang if an automatic overflow file generation is attempted while connected to a host system which does not echo the stop list character (Ctrl-S). Neither Source nor Micronet echoes the Ctrl-S. Dave Hughs of Southeastern Software has provided the following temporary fix to the program. Insert the following lines:

```
15035 POKE 2672,96
15045 POKE 2672,74
```

The "POKEs will defeat the echo-check in the send routine temporarily and will allow the overflow file to be written automatically as it should."

Using the Smarterm For PASCAL by Dr. Who?

>>>>>>>>>>"CRAWL AWAY"<<<<<<<<<<

Being a beginner in Pascal we will attempt to share some of our misfortunes and triumphs (not too many of these) with other novices.

We have had no major problems, thus far, in using the new SMARTERM board with either Basic. However, Pascal was another matter. Installing the board in slot #3 provides automatic operation (upper/lower case and glorious 80 columns) when you boot Pascal. The first thing we did was change the SYSTEM.MISCINFO file as suggested on page 8 of the SMARTERM instruction manual. This involves changing the SCREEN WIDTH to 80 instead of 79 characters. Lo and behold the FILER now provides complete prompt lines instead of just initials. Just follow the instructions in your friendly APPLE PASCAL instruction book. We had no problems in doing this.

Next we tried some of the Hi-Res graphics programs on the APPLE3:disk. It didn't work of course. All we got was the text. As pointed out on page 26 of the SMARTERM manual you must "Video source switch". Wow!! This means converting to 40 character APPLE video in order to use the Lo-Res or Hi-Res graphics. This is done using Ctrl-T A1. Direct commands in Pascal cannot be used for this. Also you must convert back to text mode using Ctrl-T B1. These must be done in the programs. Here's how:

(1) To see Hi-Res or Lo-Res graphics use:
```
WRITE (CHR(20), 'A1');
```
(2) To convert back to text use:
```
WRITE (CHR(20), 'B1');
```

Now the only problem is selecting the proper place to insert these lines. If the program uses text prompts (most do) you have to select locations carefully. Two examples follow.

For HILBERT on the APPLE3:disk you can put both lines in the main program as follows:

```
BEGIN (*MAIN PROGRAM*)
  WRITELN('WELCOME TO HILBERT');
  WRITELN('ENTER ORDER 0 TO QUIT.');
  REPEAT (* FOR EACH ORDER *)
    WRITE('ORDER:  '); READLN(ORDER);
    IF (ORDER < 1) OR (ORDER > 7) THEN
      BEGIN
        WRITELN('THAT''S ALL FOLKS...');
        EXIT (PROGRAM);
      END;
WRITE(CHR(20),'A1');
    INITTURTLE;
    DELTA:=2;
    (* CALC STARTING X,Y AND SIZE *)
    FOR N:=2 TO ORDER DO DELTA:=DELTA*2;
    DELTA:=DELTA-1;
    SIZE:=190 DIV DELTA;
    DELTA:=(DELTA*SIZE) DIV 2;
    MOVETO(140-DELTA,96-DELTA);
    PENCOLOR(WHITE);
    HIL(ORDER);
    (* WAIT FOR KEYSTROKE, THEN SWITCH TO
       TEXT *)
    READ(KEYBOARD,CH);
WRITE(CHR(20),'B1');TEXTMODE;
  UNTIL FALSE;
END.
```

For the GRAFDEMO program on APPLE3: you insert the turn-on graphics WRITE statement in the main program as follows:

```
BEGIN
  WRITELN('PRESS ANY KEY TO QUIT.');
  WRITELN('PLEASE WAIT WHILE CREATING
           BUTTERFLY');
  INITBUTTERFLY;
  INITTURTLE;
  FRAME(WHITE);
  RANDOMIZE;
  COLORS[0]:=WHITE;
  COLORS[1]:=REVERSE;
  COLORS[2]:=GREEN;
  COLORS[3]:=VIOLET;
  COLORS[4]:=ORANGE;
  COLORS[5]:=BLUE;
WRITE(CHR(20),'A1');
  REPEAT
    FAN;
    GRID;
```

The command to turn text back on should be inserted in the PROCEDURE GOODBYE as follows:

```
PROCEDURE GOODBYE;
BEGIN
  WRITE(CHR(20),'B1');
  TEXTMODE;
  READ(KEYBOARD,CH);
  WRITELN;
  WRITELN('THAT''S ALL FOLKS...');
  EXIT(PROGRAM);
END;
```

Remember you will have to change all your graphics programs in order to have them work. These examples should provide sufficient clues as to the proper procedure. If you figure out a better way be sure to let Dr. Who? know. Don't forget that after you add the new WRITE statements you have to compile the programs again and then replace the existing programs with the modified versions.

Now Dr. Who? really went bananas! How about all those nice cursor options? Could we have those in Pascal? Of course!! With daring, courage and a total lack of knowledge Dr. Who? attacked the problem. Progam written, compiled and executed. It works!!!! Now put it on APPLE1:, but just the code in order to save disk space. You can name this program SYSTEM.STARTUP and it will run each time you boot Pascal so you can select your favorite cursor mode. If you only like one cursor then throw away the CASE statement, BEGIN with the WRITE statement for the cursor you like and END. Try it, you'll like it. Now here's the program.

```
PROGRAM CONFIGSMART;
```

(This program permits you to select the type of cursor you desire when you are using a SMARTERM board in slot #3 with Pascal or Fortran. It also illustrates the proper procedure to use in initiating TERMINAL ESCAPES in a Pascal program.)

contd.

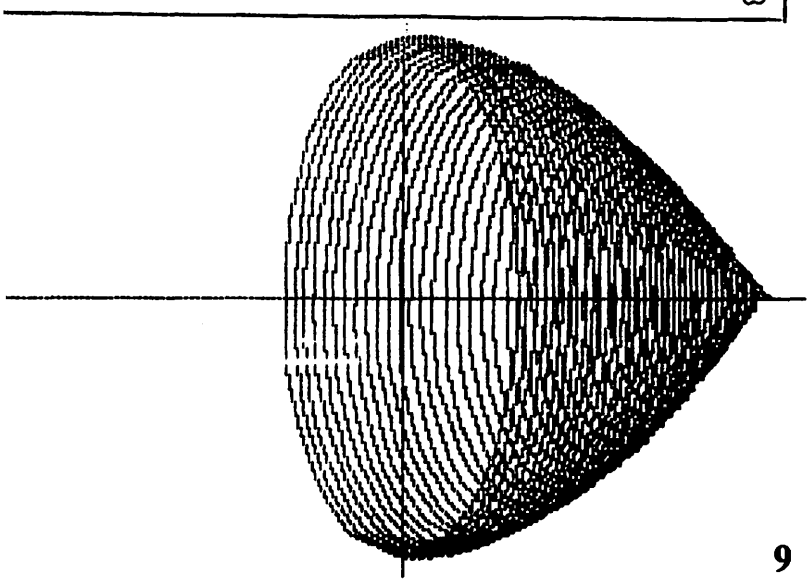
```
VAR cursor :integer;

BEGIN
  writeln('Select the CURSOR mode you
           desire and hit return');
  writeln('0 3.75hz blinking full block');
  writeln('1 3.75hz blinking half block');
  writeln('2 3.75hz blinking underline');
  writeln('3 1.875hz blinking full
           block');
  writeln('4 1.875hz blinking half
           block');
  writeln('5 1.875hz blinking quarter
           block');
  writeln('6 1.875hz blinking underline');
  writeln('7  blanked (black)');
  writeln;writeln;

  READ(cursor); {get input from the
                 keyboard}
    CASE cursor of {select the desired
                    mode}
      0:  write(chr(20),'c0'); {chr(20) is
                                Ctrl-T}
      1:  write(chr(20),'c1'); {see page
           25, SMARTERM manual for
           definitions}
      2:  write(chr(20),'c2');
      3:  write(chr(20),'c3');
      4:  write(chr(20),'c4');
      5:  write(chr(20),'c5');
      6:  write(chr(20),'c6');
      7:  write(chr(20),'c7');
    END {of case statement}
  END.
```

How much does the SMARTERM board draw from your APPLE power supply? When it is off: 256 milliamperes from the 5-volt supply and 18 milliamperes from the 12-volt supply. When it is on: 356 milliamperes from the 5-volt supply and 18 milliamperes from the 12-volt supply. By the way, M&R is coming out with a small fan for you board crazy types to relieve the heat prostration problem (about $50).

Does Dr. Who? like the SMARTERM board? Do girls like fur coats, Porsches???? I finally have an excellent Text Processing system which prints what the screen shows me. In fact the draft of this article was done on my new text system. Dr. Who? promises to return in the near future. REMEMBER.....

>>>>>>>>>> CRAWL AWAY <<<<<<<<<<

(Howard Lefkowitz)



Graph of a Trigonometric Function

by John Mahoney

The following is a listing and graph of a program which demonstrates the solid formed as the graph of f(x)=cos(x) is revolved about the x-axis on the interval [0,π].

This program is part of one of a dozen that I am writing to correlate with the high school advanced placement calculus course. I have received a grant from the Apple Education Foundation for this work. I plan to donate a disk containing these programs to the club library when I finish them this Spring. In this way math teachers (and fellow members) can easily obtain copies of them.

```
JLIST

10  REM   THIS PROGRAM SHOWS A SO
     LID FORMED AS THE GRAPH OF T
     HE FUNCTION F(X)=COS(X) IS R
     EVOLVED ABOUT THE X-AXIS.
20  REM  BY: JOHN MAHONEY, SIDWEL
     L FRIENDS SCHOOL, 3825 WISCO
     NSIN AVE., N.W., WASHINGTON,
      D.C., 20016
30  HGR : HCOLOR= 3
40  POKE  - 16302,0
50  HPLOT 0,95 TO 278,95
60  HPLOT 140,0 TO 140,190
70  HPLOT 140,5
80  FOR X = 0 TO 1.6 STEP .02
90 Y =  - 90 *  COS (X) + 95
100 P = 140 + 80 * X
110  HPLOT  TO P,Y
120  NEXT X
130  FOR X = 0 TO 1.56 STEP .04
140 Y =  COS (X)
150 P = 80 * X + 140:Q =  - 90 *
     Y + 95: HPLOT P,Q
160  FOR N = Y TO  - Y STEP  - .1

170 M =  - .5 *  SQR (Y ^ 2 - N ^
     2) + X
180  HPLOT  TO 80 * M + 140, - 90
      * N + 95
190  NEXT N
200  FOR N =  - Y TO Y STEP .1
210 M = .5 *  SQR (Y ^ 2 - N ^ 2)
     + X
220  HPLOT  TO 80 * M + 140, - 90
      * N + 95
230  NEXT N
240  HPLOT  TO P,Q
250  NEXT X
260  END
```

A Questionnaire Subroutine

by John L. Moon

While working on some enhancements for the WAP ABBS, I found I was programming questionnaires. That is, I was asking questions and collecting information. This tends to take up a lot of program space since copious explanations are usually required along with the questions. It occurred to me that there must be a better way (laziness being my mother of invention....), so in that vein I came up with a little routine to automate my information gathering.

I borrowed quite heavily some of the concepts of PILOT, a CAI coursewriting language, but simplified it considerably since I didn't need much flexibility (for example, loops are not allowed).

You might want to add some other features to the routine I have. In operation it is quite simple. It is called with the name of the file containing the questions and returns the answers in a string array. The file of questions is really a series of statements. The statements available are:

P:	Print a line
I:	Input an integer
A:n	Input a string up to n characters
L:n	A label
=:n	Assign current value to field n
!:	Load an immediate value
Y:L:n	Ask (Y/N), if Y skip to L:n
N:L:n	Ask (Y/N), if N skip to L:n
E:	Stop processing

Any lines not recognized are printed out as if they were preceded by "P:" The file is assumed to be ended by a null line (the line only has a carriage return). This is enough to do simple interrogations and data collection under computer control. For instance, on my ABBS this is used to control the questioning of users that do not have sign-ons. The controlling file is:

```
!:N
=:1
ARE YOU WILLING TO GIVE YOUR NAME, ETC?
N:L:1
ARE YOU ALREADY A WASHINGTON APPLE PI
 MEMBER?
N:L:1
DO YOU KNOW YOUR WAP NUMBER?
N:L:1
=:1
PLEASE ENTER YOUR NUMBER?
I:
=:2
L:1
PLEASE ENTER THE FOLLOWING INFO
LAST NAME?
A:20
=:3
FIRST NAME?
A:20
=:4
STREET ADDRESS?
A:30
=:5
CITY?
A:20
=:6
ZIP?
N:00000,99999
=:7
AREA CODE AND PHONE?
A:13
=:8
ANY GENERAL COMMENTS (UP TO 250
CHARACTERS
A:250
=:9
THANK YOU FOR YOUR INFORMATION
!:N
=:10
SHALL I LOG YOU ON THE SYSTEM NOW?
N:L:1
!:Y
=:10
L:1
E:
```

Using the equivalent Basic statements, I could duplicate the effect by typing a lot more characters (all the basic keywords). This also costs space within the program (a key factor in my ABBS program - it's beginning to stretch the limits of my machine). Roughly, the equivalent statements are:

```
P:ssss    PRINT "ssss"
I:        INPUT H$(0):H=VAL(H$):H$(0)
          =STR$(H)
L:n       Label (statement number)
=:n       H$(N) = H$(0)
!:ssss    H$(0) = "ssss"
Y:L:n     INPUT "Y/N ?";H$(0)
          IF LEFT$(H$(0),1)="Y" THEN
          GOTO (statement label)
N:L:n     (Similar to Y:L:n
E:        RETURN
A:n       INPUT H$(0);
          H$(0)=LEFT$(H$(0),n
```

Fancier versions could be written fairly easily. Some obvious improvements might be:

1. Allow comparisons of immediate data to current value so feedback can be provided on answers.
2. Add special print commands for blinking and inverse video.
3. Screen controls, i.e.
   clear the screen,
   position next print at some x,y location.
4. Allow default values for inputs.
5. Special graphics features:
   - load a graphics image file
   - allow input of paddle values
6. Move elaborate input routines with limit checking (such as those published by Paul Sand in a recent issue of WAP.)

Almost any of these could be selected piecemeal by using another letter to select it. What you would want would depend on your application. As it is, it can do most true/false, fill in the blank or multiple choice questions. For example, the following command file would do one of each type of question:

```
Quick Quiz:
 T/F Questions
 (Answer T for true, F for false)
 1. APPLE II's are fun to use!
    A:1
    =:1
 Fill in the blank, enter the phrase that
```


contd.

```
correctly replaces the missing part of
the sentence.
 2. The APPLE---- System is required in
    order to run Pascal and Fortran.
    A:10
    =:2
Multiple Choice
 3. An APPLE II+
     1. is missing some of the features
        of the APPLE II.
     2. has Applesoft built-in.
     3. costs a lot of money.
     4. all of the above.
    I:1,4
    =:3
```

The output of the subroutine depends on what you decide in the =: statements. A string array must be supplied which is filled in as you determine it. At the conclusion, the main program can process that data.

If you desire to do highly repetitive data entry controlled by such files, I would suggest reading the file into memory in a string array. Since this is inherently an interactive system, raw computer power doesn't mean much once you no longer have to access the disk over and over. In this case, loops could be added fairly trivially, using a sequential search in memory through the file for the label. Even a search of an 80-100 statement program wouldn't take an excessive time. Unless the main program is quite trivial, it should probably just batch the data, validate it and store it on disk for a later processing program. In this case, an initialization routine would be called to read the file into memory. As an example, the following interpreter implements loops in memory statements:

```
Initialize
 25000 PRINT D$;"OPEN";G$:H=1
 25001 PRINT D$:"READ";G$
 25002 INPUT H5$(H)
 25003 PRINT D$
 25004 IF H5$(H)="" THEN PRINT D$;"CLOSE";
       G$:RETURN
 25005 H=H+1:GOTO 25001
Do input
 26000 H1=0
 26010 H1=H1+1:IF H5$(H1)="E:" THEN RETURN
 26020 H2$=LEFT$(H5$(H1),2):H3$=
       MID$(H5$(H1),3)
 26030 IF H2$="L:"THEN 26010
 26040 IF H2$="Y:"THEN INPUT "Y/N?";H$(0):
       IF H$(0)="Y" THEN GOSUB 26100:
       GOTO 26010
 26050 PRINT H5$(H1):GOTO 26010
 26100 H2=0
 26101 H2=H2+1:IF H5$(H2)=H3$ THEN
       H2=H2-1:RETURN
 26102 GOTO 26101
```

In order to build an input file of addresses then:

```
 L:1                DIM H5$(100)
 Move data?         DIM H$(3)
 N:L:2
 L:3                G$="ADDRESSES"
 Name?              GOSUB 25000
 A:20               Open output
 =:1
 Address?           GOSUB 26000
 A:20               IF H$(1) <> "*****" THEN
 =:2                Write data out
 City, town?
 A:20
 =:3                Close output
 Move data?         End.
 Y:L:3
 L:2
 !:*****
 =:1
```

Inputs G$=filename, H=length of H array
Outputs H$=array of fields

```
25000 PRINT D$;"OPEN";G$
25010 FOR H1=1 TO H:H$(H1)="":NEXT H1:
      H2=0
25020 PRINT D$;"READ";G$
25030 INPUT H$(0)
25040 PRINT D$
25050 IF LEN(H$(0))=0 OR LEFT$(H$(0),2)
      ="E:" THEN PRINT D$;"CLOSE";G$:
      RETURN
25060 IF H2=1 AND H$(0)=H1$ THEN H2=0:
      GOTO 25020
25070 IF H2=1 THEN 25020
25080 H2$=LEFT$(H$(0),2):
      H3$=MID$(H$(0),3)
25085 IF H2$="L:" THEN GOTO 25020:
      REM IGNORE LABELS
25090 IF H2$="P:" THEN PRINT H3$:
      GOTO 25020
25100 IF H2$="!:" THEN H4$=H3$:GOTO 25020
25110 IF H2$="I:" THEN INPUT H4$:H4$=
      STR$(VAL(H4$)):GOTO 25020
25120 IF H2$="A:" THEN INPUT H4$:H4$=
      LEFT$(H4$,VAL(H3$)):GOTO 25020
25130 IF H2$="=:" THEN H$(VAL(H3$))=H4$:
      GOTO 25020
25140 IF H2$="Y:" THEN INPUT "Y/N?";
      H$(0):IF H$(0)="Y" THEN H1$=H3$:
      H2=1:GOTO 25020
25150 IF H2$="N:" THEN INPUT "Y/N?";
      H$(0):IF H$(0)="N" THEN H1$=H3$:
      H2=1:GOTO 25020
25160 PRINT H$(0):GOTO 25020
```

Questions, Questions, Questions
by Mark L. Crosby

Q. I have been fooling around with a program that attempts to put control characters in a text file. however, when the file is "EXEC'ed" I get a bunch of "SYNTAX ERRORs". Is this a no-no for a text file? (The control characters are inside a string).

A. When you "EXEC" a file it is just like typing at the keyboard. Any illegal usage will cause a syntax error from the language you are using. Use Applesoft if possible and use the CHR$() function, e.g., PRINT "PRINT CHR$ (7)" is the proper format to write a file that when EXEC'ed will ring the bell. It will read out as PRINT CHR$ (7) (CR)... which is legal. I don't think Integer BASIC will work properly since PRINT "" (CR) (imbedded CONTROL-G) is not legal in the immediate mode. The thing to keep in mind is that EXECing causes data to be INPUT from the disk terminated by carriage returns. While a CONTROL-G by itself is perfectly legitimate for a program to send out, it is not a legal line to INPUT to Integer BASIC.

Q. Is special software necessary to use the IDS 460G printer for text use or is it just necessary for the graphics dump from the APPLE II?

A. Text printing is done in much the same way as always using POKE 36,n instead of HTAB. Graphics dumps are much more difficult without some experience. You can buy the graphics dump program (which, by the way, works beautifully and is currently producing this newsletter's headings) from:

COMPUTER STATION, INC., 12 Crossroads Plaza, Granite City, IL 62040 (618) 452-1860 $44.95 either Applesoft or PASCAL.

Q. My Apple overheats. My primary use for it is as a terminal using the D.C. Hayes Micromodem. Particularly the "C" bank of RAM seem to get the hottest. After a long session I lose the display, and must RESET which causes serious problems. Any suggestions?

A. I've had the same trouble with overheating and it is caused by not enough air circulation within the Apple cabinet. Try purchasing an inexpensive ($8-$15) muffin fan from a local electronics or computer store and place inside the cabinet. I just laid mine (plastic frame) right on top of the RAM during the long hot Summer we had here and it worked fine. Alternatively, just take the top off for awhile. It you still have overheating and have other boards in your Apple - take them out if you are not using them. They still produce heat when not in use.

Q. I have had some problems using the Micromodem to control someone else's Apple (catalog, list, etc.). The results would only show on their screen but not on mine. How can that be fixed?

A. Follow this procedure: Have the other Apple set up with PR#n and IN#n (n=SLOT #). Then he should go into terminal mode. Then you do an IN#n on your Apple and call the other Apple to connect up. After connection you then send a CONTROL-R to access his BASIC. Because he initially did the PR#n you will get output on your screen. To send him a message, you do a PR#0 and CONTROL-T (terminal mode) then send the message. After the message send a CONTROL-R followed by a PR#n with HIS slot number. You will then be back in business.

Q. I have RAM Applesoft and can't seem to make the "&" work properly. I know about the POKE's that link the jump to any machine-language subroutine but it won't respond. Any help would be appreciated.

A. While using RAM Applesoft and DOS, the ampersand is disabled. To reconnect it POKE 2142,244: POKE 2143,3.

Q. Some games permit commands to be entered without ending them with a carriage return; i.e., they take action as soon as the character is hit. BASIC seems to require a carriage return before input data can be read. Is this correct? Is there a way around this so BASIC programs can read each input character as it is typed?

A. Yes there is. By using the keyboard data and strobe locations you can add some elegant features to that special game of yours. The procedure requires that you develop a short subroutine to which you will "GOSUB" rather frequently to determine if a key has been pressed. If no key is pressed then no action is taken and the program will continue. If a key is pressed, appropriate action can be taken. Here is a short illustraton:

```
10 TEXT :  CALL -936
30 GOSUB 100
40 IF KEY < 128 THEN 30
50 PRINT "ASCII VALUE OF KEY=";KEY;" "
60 GOTO 30
100 KEY = PEEK (-16384)
110 IF KEY > 127 THEN POKE -16368,0
120 FOR J = 1 TO 10 :  NEXT J
130 RETURN
```

(This will work in either BASIC).

NOTE: The delay in line 120 is necessary to prevent reading the keyboard too quickly and thereby getting unreliable results.

Q. I want to be able to copy data that is on the CRT from one VIS-I-CALC file on disk to another, and to store selected data displayed on the screen. Is it possible?

A. "PD" saves the displays, "SS" saves the sheets including formulae. You can save and load whole sheets but not partials. It is possible to load one sheet on top of another - it only clears if you do a "CY".

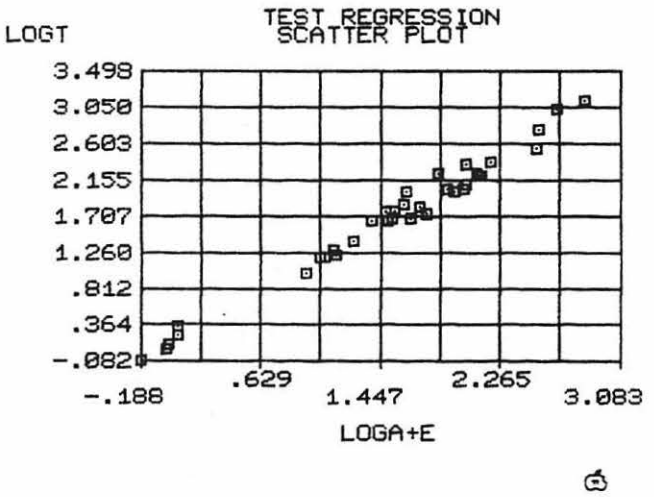
contd.

Q. I am trying to locate a HI-RES character or shape table on disk. Do you have any references?

A. See the January 1981 issue of Creative Computing for a shape table and printing algori thms for most ASCII characters. The Apple Software Bank Contributed Programs 3-5 includes a character generator and table which can be easily interfaced to BASIC programs. I use it all the time. See the example regression graph below.

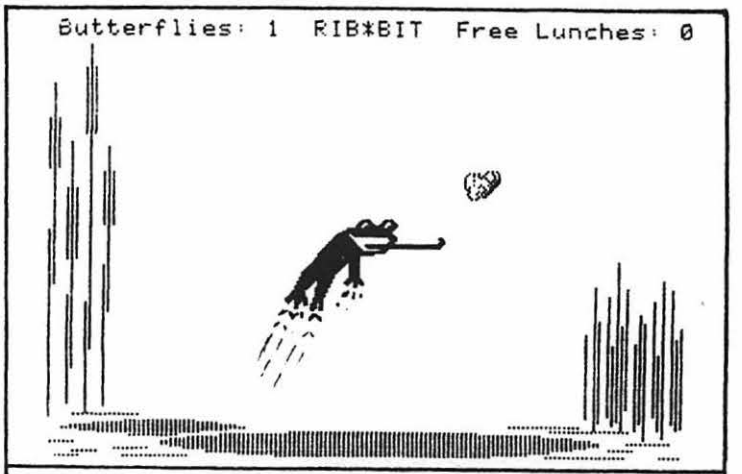
Q. Can anyone explain how to use the FRE (X) command? Every time I use it I get a syntax error.

A. Use it in the form: X = FRE (n) where n may be any dummy integer or variable. You may also PRINT FRE (n) to find out how much free space is left. This command forces Applesoft to "clean house" - to get rid of unused strings and free up memory space. This command should be used immediately after CHAINing to another program to avoid the annoying and technically erroneous "OUT OF MEMORY ERROR" (BEEP,BEEP).



FOR IDS 440/445/460 OWNERS

Newly released for the IDS 440/445 series of printers is APPLEWRITER GRAPHICS. This set of machine language and Applesoft programs (Integer too) can be used with DOS 3.3 and the new DOS TOOL KIT to produce printed copy of the Hi-Resolution characters sets available on the latter's disk. If you have Applewriter, it is compatible with that also. When in use the printed output matches the graphic character sets. Now you can do word processing using different typestyles as easily as normal printing is accomplished. Soon to be available for the IDS 460. Look for a review in the next issue. Available from Computer Station, #12 Crossroads Plaza, Granite City, Illinois 62040 (618) 452-1860. $34.95



Ladies and Gentlemen...
Apple Computer Inc. is proud to present
Maxwell, a small demonstration of Hi-Res
Programmable Character Set Animation !!!

TWO DEMOS FROM DOS 3.3 TOOLKIT

Hi-Res characters sets on Tool Kit disk.

##################################

ERRATA

Please note the following correction to "SAVE TAPE" Changes, page 20, December 1980 newsletter:

245 BUF=BUF+1:IF BUF < TOP THEN 260

##################################

Notes on Hi-Res Graphics Routines in Applesoft

by C.K. Mesztenyi

Checking out the entry points given by J. Crossley in the article "APPLESOFT INTERNAL ENTRYPOINTS" in the March/April 1980 Apple Orchard, I found the given entry points were 4 bytes off from the given ones in our APPLE II Plus. Furthermore, after checking out the routines in more detail, I thought to share my experiments with other APPLE II Plus owners interested in machine language programming. In the first section I describe the essential data storage area, in the second I give the entrypoints of the subroutines somewhat more detailed than in the above article, and in the last section I give some listings of instructions following the entrypoints so that one could identify it for different versions of Applesoft.

1. DATA STRUCTURE

There are four data in five memory locations which specify a point on the high resolution screen (whether the screen is displayed or not, is irrelevant). I call these data collectively as external cursor data. The five memory locations, and their contents are as follows:

- $E0: Low order bits of the horizontal screen coordinate
- $E1: High order bit of the horizontal screen coordinate
- $E2: Vertical screen coordinate
- $E4: Color masking word from the color table ($F6F6-$F6FD)
- $E6: Page indicator ($20 for Page 1, $40 for Page 2)

I have called the above set of data as external cursor data since the actual point plot is performed by the following five instructions:

```
    LDA   $1C
    EOR   ($26),Y
    AND   $30
    EOR   ($26),Y
    STA   ($26),Y
```

which uses data located at $1C, $26, $27, register Y and $30. The contents of register Y is always picked up from location $E5 prior to the above instructions, thus we may call the data in the following five locations as internal cursor data:

- $1C: The color masking byte shifted for odd address and none black or white, unchanged otherwise.
- $26, $27: (Low, high order) address of the byte corresponding to the page, vertical coordinate and leftmost seven points of the screen.
- E5 (register Y): The integer part of the horizontal screen coordinate divided by 7.
- $30: The bit position taken from Bit Position Table corresponding to the remainder of the horizontal coordinate divided by 7.

These two cursor data (external and internal) are equivalent in the sense that given one, the other can be derived from it. There would be no need to make any distinction if they would correspond to each other all the time but, unfortunately, this is not always the case, e.g. the following sequence of Basic instructions:

```
HCOLOR = 1
HPLOT 0,0 TO 10,10
HCOLOR = 2
HPLOT TO 10,50
```

plots two lines, (0,0) to (10,10) and (10,10) to (10,50), both with color 1, i.e. HCOLOR=2 has no effect. Actually it resets the color code in $E4 but it does not change $1C, and the statement HPLOT TO picks up whatever was left in $1C.

A machine language programmer can write his/her own graphics routines which takes time and uses sometimes much-needed memory space. Thus using the available programs in Applesoft ROM can be advantageous. If execution time is also important, as in the case of animation, then one should concentrate only on the internal cursor data, and modify the external cursor only when it is necessary. The entry points INTX and INTY, provide the basic routines for incremental plotting which are not available directly in Basic. Also modifying the external cursor coordinates allows the use of HLINE with off-set.

2. ENTRY POINTS IN APPLESOFT

Page and Color:

HGR2 ($F3D8): Displays page 2 with all graphics mode, sets $E6 to $40, clears page 2 (black) and sets $1C to zero (black I).

HGR ($F3E2): Displays page 1 in mixed mode, sets $E6 to $20,clears page 1 (black), and sets $1C to zero (black I).

BKGND ($F3F4): Clears the page defined by $E6 to the color defined by the contents of register A which should be one from the Color Masking Table. Also stores register A in $1C.

HCOLOR ($F6F0): Assumes register X contains the color index (0 to 7). The routine picks up the appropriate color code from the Color Masking Table and stores it in $E4.

Positioning Entries:

HPOSN ($F411): Assumes the input upon entry in the registers as:
register X = low order bits of the horizontal screen coordinate,
register Y = high order bit of the horizontal screen coordinate,
register A = vertical screen coordinate.
The routine stores the registers in $E0, $E1 and $E2. Then using $E6 sets $26, $27, $30 and $E5 together with register Y, and sets $1C to the contents of $E4. Thus this routine makes the internal cursor equivalent to the external one.

INTX ($F465): Modifies the internal cursor data in $1C, $E5, register Y and $30 so that it corresponds to incrementing/decrementing the horizontal screen coordinate X by one. Upon entry, if the N-flag is zero (positive) then it increments; if N is set (negative) then it decrements. The

modification has a wrap around feature, i.e., incrementing/ decrementing at the extreme sides of the screen defined by the internal cursor causes it to come back on the other side. The routine assumes that register Y corresponds to $E5 upon entry, and leaves the routine correctly modified if necessary.
Upon testing the N-flag the routine jumps to DECRX or INCRX.

DECRX ($F467): The routine modifies the internal cursor data by decrementing the horizontal screen coordinate by 1 (see INTX).

INCRX ($F48A): The routine modifies the internal cursor data by incrementing the horizontal screen coordinate by 1 (see INTX).

INTY ($F4D3): Modifies the internal cursor data in $26, $27 so that it corresponds to incrementing/ decrementing the vertical screen coordinate by one. Upon entry, the N-flag is checked, and if it is set (negative) then goes to INCRY to increment by one, if it is not set (positive) then goes to DECRY to decrement by one. Note that the sign convention is used opposite of INTX. These entries also have the wrap around features, i.e. if the incrementation/ decrementation causes the cursor to leave the screen on the bottom/top, then it comes back on the top/bottom.

DECRY ($F4D5): The routine modifies the internal cursor data by decrementing the vertical screen coordinate by 1 (see INTY).

INCRY ($F504): The routine modifies the internal cursor data by incrementing the vertical coordinate by 1 (see INTY).

IPOSN ($F5CB): Sets the external cursor data in $E0, $E1, $E2 equivalent to the internal cursor coordinate data.

Plotting Entries:

HPLOT ($F457): Assumes input data in the registers as HPOSN:
register X: low order bits of horizontal screen coordinate,
register Y: high order bit of horizontal screen coordinate,
register A: vertical screen coordinate.
The routine calls HPOSN with the above data, then goes to PLOT.

PLOT ($F45A): The routine executes the five instructions listed in the beginning of the article which plots a point using the internal cursor data. If this entry is used directly, then the user should make sure that register Y contains the data from $E5.

HLINE ($F53A): The routine assumes input in the registers:
register A: low order bits of horizontal screen coordinate,
register X: high order bit of horizontal screen coordinate,
register Y: vertical screen coordinate.
(Note that it is in different order than HPOSN.)
The routine then draws a line from the internal cursor position to the point defined by the input. Upon exit, it leaves the external cursor data corresponding to the input, the internal cursor data corresponding to the last plot point of the line. If the internal and external cursor data were not equivalent then an off-set occurs. This can be visualized as follows: Draw a linesegment from the external cursor coordinates to the input coordinates. Now move this linesegment parallel to itself so that the endpoint at the external cursor position gets into the interal cursor position. This is the actual linesegment which will be drawn. If it gets outside of the screen then a wrap around occurs, i.e. it comes back on the opposite side of the screen.

APPENDIX

The first few instructions are listed for each entry point so that one could identify them using the Monitor list feature.

Bit Position Table:

```
$F5B2:  $81 = 10000001
$F5B3:  $82 = 10000010
$F5B4:  $84 = 10000100
$F5B5:  $88 = 10001000
$F5B6:  $90 = 10010000
$F5B7:  $A0 = 10100000
$F5B8:  $C0 = 11000000
```

Color Masking Table:

```
$F6F6:  $00 = 00000000 (black I)
$F6F7:  $2A = 00101010
$F6F8:  $55 = 01010101
$F6F9:  $7F = 01111111 (white I)
$F6FA:  $80 = 10000000 (black II)
$F6FB:  $AA = 10101010
$F6FC:  $D5 = 11010101
$F6FD:  $FF = 11111111 (white II)
```

```
HGR2:   $F3D8: BIT $C055
               BIT $C052
               LDA #$40
               BNE $F3EA
HGR:    $F3E2: LDA #$20
               BIT $C054
               BIT $C053
               STA $E6
                ....
BKGND:  $F3F4: STA $1C
               LDA $E6
               STA $1B
               LDY $#00
                ....
HCOLOR: $F6F0: LDA $F6F6,X
               STA $E4
               RTS

HPOSN:  $F411: STA $E2
               STX $E0
               STY $E1
               PHA
               AND #$C0
               STA $26
                ....
IPOSN:  $F5CB: LDA $26
               ASL
               LDA $27
               AND #$03
               ROL
               ORA $26
                ....
INTX:   $F465: BPL $F48A
DECRX:  $F467: LDA $30
```


contd.

```
                LSR
                BCS $F471
                EOR #$C0
                 ....
INCRX:  $F48A:  LDA $30
                ASL
                EOR #$80
                BMI $F46E
                 ....
INTY:   $F4D3:  BMI $F505
DECRY:  $F4D5:  CLC
                LDA $27
                BIT $F5B9
                PNE $F4FF
                 ....
INCRY:  $F505:  CLC
                LDA $27
                ADC #$04
                BIT $F5B9
                 ....

HPLOT:  $F457:  JSR $F411
PLOT:           LDA $1C
                EOR ($26),Y
                AND $30
                EOR ($26),Y
                STA ($26),Y
                RTS
HLINE:  $F53A:  PHA
                SEC
                SPC $E0
                PHA
                TXA
                SPC $E1
                 ....
```

Fast Walsh-Hadamard Transform

by Jim Rose

One of the true advantages of belonging to a club such as Washington Apple Pi is finding kindred spirits with common interests both during meetings and in the pages of this newsletter. In particular is Bruce Field's article "FFT Subroutine for the 6502" which appeared in the September issue. That article explained and presented code for the Fast Fourier Transform, and in this follow-up article I want to discuss a faster 'Fast' transform: the Hadamard.

Being able to analyze a function in the frequency domain has led to important discoveries in the field of signal processing. Filtering and Fourier analysis of digital images is the subject of a good deal of current research in pattern recognition and image processing. In writing an image processing system for the APPLE (APPLEPIPS (C):APPLE-II Personal Image Processing System) the need to include such an analytical capability was apparent.

However, a two-dimensional Fourier transform, even one as fast as that presented in Field's article, would take more than ten minutes just to transform the image into the frequency domain. If one includes the application of a filter, and retransformation into the image domain, the user is waiting around for a half an hour or so for the analysis to be completed. This is unacceptable in an educational or training environment.

Our search for a faster transform was concluded with the (re)discovery of the Walsh-Hadamard Transform (2). The Hadamard transform of Walsh functions is a similar operation to the Fourier transform of a sine function. The essential difference is that Walsh functions are discontinuous (square-ish) waves which only take on the value of +1 or -1. This leads to the important fact that the Hadamard transform only involves addition and subtraction rather than multiplication. And this leads to an eight-fold speed improvement. Instead of 1.3 seconds for an FFT, it takes 0.16 seconds for an FWHT.

The simplicity of the FWHT has more ramifications than just speed improvements. Particularly, the code itself is shorter, and the memory requirements are more modest. Not only can we do without a table of sines, but also the transform into the 'sequency' (akin to 'frequency') domain is not a complex result. Thus, as the code below demonstrates, an inplace transform can be performed using only a single page for data. Such an inplace transform makes filtering and retransforming the image both space and time efficient.

The FWHT code clearly borrows heavily from existing sources. In a effort not to reinvent the wheel, blocks of code from Bruce Field's FHT algorithm are included (for which, much thanks), as is the FORTRAN structure of Beauchamp's inplace algorithm.

Following Bruce Field's lead, I also have included a higher level language demonstration program. This one, however, is in Integer Basic, speeding up the calculation of the input waveform from about 45 seconds to about 15. Applesoft is fine if floating point is really required, but Integer Basic is both fast and space efficient. But sines and cosines in Integer Basic? Yes, indeed! The old HIRES Graphics routines supplied by Apple contain a page of cosines for both their demos, and your own.

REFERENCES:

(1) 'Fast Fourier Transform Subroutine for the 6502', Bruce F. Field, WASHINGTON APPLE PI, September 1980

(2) WALSH FUNCTIONS AND THEIR APPLICATIONS, K. G. Beauc8amp, qcademic Press, London, 1975

(3) PATTERN CLASSIFICATION AND SCENE ANALYSIS, Richard O. Duda & Peter E. Hart, John Wiley & Son, 1973, p308-325

(4) ISSUES IN DIGITAL IMAGE PROCESSING, Robert M. Haralick & J. C. Simon, Sijthoff & Noordhoff, 1980, p142ff

contd.

```
:ASM
0000:                1 *------------------------------
0000:                2 *  FAST WALSH-HADAMARD TRANSFORM
0000:                3 *       BY JIM ROSE
0000:                4 *
0000:                5 *    FROM: WALSH FUNCTIONS AND THEIR
0000:                6 *          APPLICATIONS, K.G. BEAUCHAMP,
0000:                7 *          ACADEMIC PRESS, LONDON 1975
0000:                8 *    ALSO CODE BORROWED FROM:
0000:                9 *         FAST FOURIER TRANSFORM SUBROUTINE
0000:               10 *         FOR THE 6502, BRUCE F. FIELD
0000:               11 *         WASHINGTON APPLE PI, SEPT 1980
0000:               12 *
0000:               13 * ----------
0000:               14 *
0000:               15 *  THIS SUBROUTINE PERFORMS AN 'IN PLACE' FAST
0000:               16 *  WALSH-HADAMARD TRANSFORM ON THE DATA.
0000:               17 *  AS IN BRUCE FIELD'S ARTICLE THE INPUT DATA
0000:               18 *  (AND OUTPUT DATA) IS ASSUMED TO BE
0000:               19 *  TWO'S COMPLEMENT.
0000:               20 *
0000:               21 *  THE HADAMARD TRANSFORM IS REAL RATHER THAN
0000:               22 *  COMPLEX; AND IT IS ALSO REFLEXIVE OR SYMMETRIC.
0000:               23 *  THUS F(X)=H(H(F(X))): THE TRANSFORM OF THE
0000:               24 *  TRANSFORM OF A FUNCTION IS THAT FUNCTION
0000:               25 *  (EXCEPT FOR PROBLEMS OF LEAKAGE, ALIASING,
0000:               26 *  AND ROUNDING).  AN IN-PLACE TRANSFORM, SUCH
0000:               27 *  AS THE ONE BELOW, PROVIDES A UNIQUE LABORATORY
0000:               28 *  FOR TESTING THAT PROPERTY.
0000:               29 *
0000:               30 *
0000:               31 *------------------------------
0000:               32 ;
0000:               33 *NOTE: THIS IS THE INPLACE FWHT
0000:               34 *
6400:               35              ORG $6400
6400:               36              OBJ $6400
6400:               37 DATA         EQU $6300           ;DATA ARRAY
6400:               38 *
6400:               39 *
6400:               40 * INITIALIZE MAJOR LOOP
6400:               41 *   DO 4 L=1,M
6400: A9 08         42              LDA #8              ;M=ALOG2(FLOAT(N))
6402: 8D 2F 65      43              STA LCOUNT          ;N=256 M=8
6405: A9 01         44              LDA #1
6407: 8D 2E 65      45              STA L
640A:               46 ;
640A: AE 2E 65      47 NEXTL        LDX L               ;START MAJOR LOOP
640D: BD 13 65      48              LDA NZ,X            ;NZ=2**(L-1)
6410: 8D 13 65      49              STA NZ              ;(THE 0 OFFSET POSITION)
6413:               50 ;
6413: BD 1C 65      51              LDA NZN,X           ;NZN=N/(2*NZ)
6416: 8D 1C 65      52              STA NZN
6419:               53 ;
6419: BD 25 65      54              LDA NZ2,X           ;NZ2=MAX(1,NZ/2)
641C: 8D 25 65      55              STA NZ2
641F: 20 EA 64      56              JSR SCALE
6422:               57 *
6422:               58 * INITIALIZE INTERMEDIATE LOOP
6422:               59 *   DO 3 I=1,NZN
6422: AD 1C 65      60              LDA NZN
6425: 8D 31 65      61              STA ICOUNT                                  contd.
```

```
6428: A9 00      62              LDA #0
642A: 8D 30 65   63              STA I                ;ACTUALLY (I-1)
642D:            64 ;
642D:            65 *   JS=(I-1)*NZI
642D:            66 *     OR
642D:            67 *   JS=(I-1)*(2**L)
642D: AC 2E 65   68 NEXTI        LDY L
6430: 18         69              CLC
6431: AD 30 65   70              LDA I                ;REMEMBER=(I-1)
6434: 2A         71 JS2          ROL A                ;*2
6435: 88         72              DEY                  ;ENOUGH SHIFTS?
6436: D0 FC      73              BNE JS2              ;NO DO MORE
6438: 8D 33 65   74              STA JS               ;YES, STORE IT
643B: CE 33 65   75              DEC JS               ;LESS ONE (FOR 0 SUBSCRIPTS)
643E:            76 ;
643E:            77 ;
643E:            78 *INITIALIZE MINOR LOOP 1
643E:            79 *  DO 2 J=1,NZ2
643E: AD 25 65   80              LDA NZ2
6441: 8D 32 65   81              STA J
6444:            82 ;
6444: EE 33 65   83 NEXTJ1       INC JS               ;JS=JS+1
6447: AD 33 65   84              LDA JS
644A: AA         85              TAX                  ;FOR INDEXING LATER
644B: 18         86              CLC                  ;GET READY TO ADD
644C: 6D 13 65   87              ADC NZ               ;J2=JS+NZ
644F: A8         88              TAY                  ;FOR INDEXING
6450:            89 ;
6450:            90 *  HOLD=A(JS)+A(J2)
6450: 18         91              CLC
6451: BD 00 63   92              LDA DATA,X
6454: 79 00 63   93              ADC DATA,Y
6457: 8D 35 65   94              STA HOLD
645A:            95 ;
645A:            96 *  A(J2)=A(JS)-A(J2)
645A: 38         97              SEC
645B: BD 00 63   98              LDA DATA,X
645E: F9 00 63   99              SBC DATA,Y
6461: 99 00 63  100              STA DATA,Y
6464:           101 ;
6464:           102 *  A(JS)=HOLD
6464: AD 35 65  103              LDA HOLD
6467: 9D 00 63  104              STA DATA,X
646A:           105 ;
646A:           106 *  2 CONTINUE
646A: CE 32 65  107              DEC J                ;FINISHED WITH LOOP1?
646D: D0 D5     108              BNE NEXTJ1           ;NOT YET
646F:           109 ;
646F:           110 *  IF (L.EQ.1) GO TO 3
646F: AD 2E 65  111              LDA L
6472: C9 01     112              CMP #1
6474: F0 31     113              BEQ ENDJ
6476:           114 ;
6476:           115 * INITIALIZE MINOR LOOP 2
6476:           116 *  DO 1 J=1,NZ2
6476: AD 25 65  117              LDA NZ2
6479: 8D 32 65  118              STA J
647C: EE 33 65  119 NEXTJ2           INC          JS ;JS=JS+1
647F:           120 ;
647F: AD 33 65  121              LDA JS
6482: AA        122              TAX                  ;FOR INDEXING
6483: 18        123              CLC                                    contd.
```

```
6484: 6D 13 65    124             ADC NZ              ;J2=JS+NZ
6487: A8          125             TAY                 ;FOR INDEXING
6488:             126 ;
6488:             127 *  HOLD=A(JS)-A(J2)
6488: 38          128             SEC
6489: BD 00 63    129             LDA DATA,X
648C: F9 00 63    130             SBC DATA,Y
648F: 8D 35 65    131             STA HOLD
6492:             132 ;
6492:             133 *  A(J2)=A(JS)+A(J2)
6492: 18          134             CLC
6493: BD 00 63    135             LDA DATA,X
6496: 79 00 63    136             ADC DATA,Y
6499: 99 00 63    137             STA DATA,Y
649C:             138 ;
649C:             139 *  A(JS)=HOLD
649C: AD 35 65    140             LDA HOLD
649F: 9D 00 63    141             STA DATA,X
64A2:             142 *  1 CONTINUE
64A2: CE 32 65    143             DEC J               ;DONE YET?
64A5: D0 D5       144             BNE NEXTJ2          ;NO, DO SOME MORE
64A7:             145 ;
64A7:             146 *  3 CONTINUE
64A7: EE 30 65    147 ENDJ        INC I               ;NEXT I
64AA: CE 31 65    148             DEC ICOUNT          ;ARE WE DONE YET?
64AD: F0 03       149             BEQ TESTL           ;YES, DON'T GO BACK
64AF: 4C 2D 64    150             JMP NEXTI           ,NOPE, GO BACK
64B2:             151 ;
64B2:             152 *  4 CONTINUE
64B2: EE 2E 65    153 TESTL       INC L;              NEXT L
64B5: CE 2F 65    154             DEC LCOUNT          ;ARE WE DONE YET?
64B8: F0 03       155             BEQ REVBIT          ;YES, NOW BITREV
64BA: 4C 0A 64    156             JMP NEXTL           ;NOPE, DO SOME MORE
64BD:             157 ;
64BD:             158 ******************************
64BD:             159 ;ALL DONE WITH THE LOOPS
64BD:             160 ;NOW DO THE BIT REVERSAL
64BD:             161 ******************************
64BD: A2 00       162 REVBIT      LDX #0
64BF: A0 08       163 BITREV      LDY #8              ;INIT BIT REV COUNTER
64C1: 8E 35 65    164             STX HOLD            ;TEMP FOR X
64C4: 6E 35 65    165 BRV1        ROR HOLD            ;ROTATE INTO CARRY
64C7: 2A          166             ROL A               ;AND INTO ACCUM
64C8: 88          167             DEY                 ;DECREMENT COUNTER
64C9: D0 F9       168             BNE BRV1            ;SEE IF DONE
64CB: 8E 35 65    169             STX HOLD            ;RESTORE POINTER FOR COMPARE
64CE: CD 35 65    170             CMP HOLD            ;IF ORIG>BIT REV DON'T
64D1: 90 13       171             BCC BRV2            ;DO THE SWAP
64D3: A8          172             TAY                 ;BIT REV TO Y
64D4: BD 00 63    173             LDA DATA,X          ;GET VALUE
64D7: 8D 35 65    174             STA HOLD            ;HOLD IT A MINUTE
64DA: B9 00 63    175             LDA DATA,Y          ;GET SWITCHER
64DD: 9D 00 63    176             STA DATA,X          ;SWITCH
64E0: AD 35 65    177             LDA HOLD            ;GET OLD X
64E3: 99 00 63    178             STA DATA,Y          ;FINISH SWITCH
64E6: E8          179 BRV2        INX                 ;INCREMENT LOOP COUNTER
64E7: D0 D6       180             BNE BITREV          ;NOT DONE YET
64E9:             181 ;
64E9:             182 ** ALL DONE **
64E9: 60          183             RTS
64EA:             184 *
64EA:             185 ;
```


contd.

```
64EA:                  186 ;
64EA:                  187 *
64EA:                  188 * SCALING ROUTINE
64EA:                  189 *
64EA:                  190 *  CHECKS ALL DATA TO ANTICIPATE POSSIBLE
64EA:                  191 *  OVERFLOW IN THE NEXT PASS
64EA:                  192 *  IF NECESSARY DATA IS SCALED DOWN BY TWO
64EA:                  193 *
64EA: A2 00            194 SCALE       LDX #0;INIT          LOOP COUNTER
64EC: BD 00 63         195 SCL1        LDA DATA,X           ;GET VALUE
64EF: 10 06            196             BPL SCL11
64F1: C9 C0            197             CMP #$C0
64F3: 90 0A            198             BCC SCL2             ;OV IF <$C0
64F5: B0 04            199             BCS SCL12
64F7: C9 41            200 SCL11       CMP #$41
64F9: B0 04            201             BCS SCL2             ;OV IF >$40
64FB: E8               202 SCL12       INX                  ;INCREMENT LOOP COUNTER
64FC: D0 EE            203             BNE SCL1             ;SEE IF DONE
64FE: 60               204             RTS                  ;NO OVERFLOW
64FF:                  205 *
64FF:                  206 * SCALE EVERYTHING DOWN
64FF:                  207 *
64FF: EE 34 65         208 SCL2        INC SCLFCT           ;INCREMENT SCALEFACTOR
6502: A2 00            209             LDX #0               ;INIT LOOP
6504: 18               210 SCL3        CLC
6505: BD 00 63         211             LDA DATA,X           ;GET VALUE
6508: 10 01            212             BPL SCL4
650A: 38               213             SEC                  ;PUT SIGN BIT IN CARRY
650B: 6A               214 SCL4        ROR A                ;DIVIDE BY 2 WITH SIGN
650C: 9D 00 63         215             STA DATA,X
650F: E8               216             INX
6510: D0 F2            217             BNE SCL3             ;SEE IF DONE
6512: 60               218             RTS
6513:                  219 ;
6513:                  220 ** NOW FOR SOME DATA **
6513:                  221 ;
6513:                  222 *   NZ=2**(L-1)
6513: 00 01 02
6516: 04 08 10
6519: 20 40 80         223 NZ          DFB 0,1,2,4,8,16,32,64,128
651C:                  224 *
651C:                  225 *   NZN=N/(2*NZ)
651C: 00 80 40
651F: 20 10 08
6522: 04 02 01         226 NZN         DFB 0,128,64,32,16,8,4,2,1
6525:                  227 *
6525:                  228 *   NZ2=MAX(1,NZ/2)
6525: 00 01 01
6528: 02 04 08
652B: 10 20 40         229 NZ2         DFB 0,1,1,2,4,8,16,32,64
652E:                  230 *
652E:                  231 ****************************
652E:                  232 *
652E:                  233 *...AND SOME LOCAL DATA
652E:                  234 *
652E: 00               235 L           DFB 0
652F: 00               236 LCOUNT      DFB 0
6530: 00               237 I           DFB 0
6531: 00               238 ICOUNT      DFB 0
6532: 00               239 J           DFB 0
6533: 00               240 JS          DFB 0
```

contd.

```
6534: 00          241 SCLFCT      DFB 0
6535: 00          242 HOLD        DFB0
6536:             243 *
6536:             244 ****************************

*** SUCCESSFUL ASSEMBLY: NO ERRORS

>LIST
    9 REM -----HIRES ENTRIES---------
   10 INIT=3072:CLEAR=3086:POSN=3761:PLOT=3780:LINE=3786:SIN=3840
   11 REM ----------------------------
   12 REM -----LOCAL ENTRIES---------
   13 SINE=100:CHART=50:DRAW=400
   14 REM --------------------------
   15 IF PEEK (3072)#169 THEN PRINT "BLOAD HIRES"
   16 REM       (CHECK IF HIRES IS LOADED)
   19 REM ----- CHART/GRAPHING PARAMETERS
   20 DIM L(5):L(1)=255:L(2)=255:L(3)=0:L(4)=0:L(5)=255
   30 DIM U(5):U(1)=16:U(2)=144:U(3)=144:U(4)=80:U(5)=80
   31 REM --------------------------
   35 LOC=99*256:LASTLOC=100*256-1
   40 CALL INIT: POKE 801,0: POKE 812,255
   45 GOTO 1000
   49 REM --------   CHART   ---------
   50 POKE 800,0: POKE 802,16: CALL POSN
   60 FOR I=1 TO 5: POKE 800,L(I): POKE 802,U(I)
   70 CALL LINE: NEXT I
   80 POKE 800,0: POKE 802,80: CALL POSN
   90 RETURN
   91 REM --------------------------
  100 DC=0:A1=30:A2=60:A3=15
  110 F1=1:F2=2:F3=8
  200 B1=0:B2=0:B3=0
  210 MAX=(DC+(A1+A2+A3)*255)/255
  220 FOR I=0 TO 127
  230 B1=(B1+F1) MOD 256:B2=(B2+F2) MOD 256:B3=(B3+F3) MOD 256
  240 A=DC+A1* PEEK (SIN+B1)+A2* PEEK (SIN+B2)+A3* PEEK (SIN+B3)
  250 A=A/MAX: POKE 800,I: POKE 802,A/2+16
  260 CALL POSN: POKE 802,80: CALL LINE
  265 POKE 800,255-I: POKE 802,144-A/2: CALL POSN: POKE 802,80: CALL LINE
  270 A=A-128: IF A<0 THEN A=A+256
  280 POKE LOC+I,A: POKE LASTLOC-I,255-A
  290 NEXT I: RETURN
  399 REM -------   DRAW   --------
  400 FOR I=0 TO 255:A= PEEK (LOC+I)
  410 POKE 800,I: POKE 802,80: CALL POSN
  420 IF A>127 THEN A=A-256
  430 POKE 802,A+80: CALL LINE
  440 NEXT I: RETURN
  444 REM  ------------------------
 1000 GOSUB 50: REM  GRAPH
 1010 GOSUB 100: REM  SINE FUNCTION
 1020 CALL 25600: REM  FHT
 1030 CALL INIT: REM  CLEAR SCREEN
 1040 GOSUB 50: REM  GRAPH
 1050 GOSUB 400: REM  TRANSFORM PLOT
 1060 PRINT "HIT 'RETURN' FOR INVERSE"
 1065 INPUT "HIT 'CNTL-C' TO QUIT",A$
 1080 GOTO 1020: REM  FHT+GRAPH+PLOT
 9999 END
```

```
10  DIM A$(20,20): HOME : VTAB 4:A$ = "PERPETUAL CALENDAR": FOR I = 1 TO 9: PRINT  MID$ (A$,I,1): P
RINT : NEXT : VTAB 2: FOR I = 10 TO 20: HTAB 40: PRINT  MID$ (A$,I,1): NEXT
20  VTAB 1: HTAB 2: PRINT "P E R P E T U A L     C A L E N D A R"
30  REM

*********************************

   FROM 'MICRO' - APRIL, 1980
          PAGE 23:27

*********************************

40  REM

*********************************

 WRITTEN BY:
  MEL EVANS
  1027 REDEEMER
  ANN ARBOR, MI 48103
  FOR THE "AIM" COMPUTER

*********************************
50  REM

*********************************

REVISED FOR THE "APPLE" COMPUTER
AND THE "PAPER TIGER" PRINTER BY:
60  REM
        DONALD E. KAHLER
        3834 CARPENTER ST., S.E.
        WASHINGTON, D. C. 20020
          DECEMBER 18, 1980

*********************************

80  POKE 34,3: POKE 32,10: POKE 33,25
90  CLEAR : PRINT
100  DIM A(12),R$(12)
110 K = 0
120  FOR I = 1 TO 12: READ A(I): NEXT I
130  FOR I = 1 TO 12: READ R$(I): NEXT I
140  VTAB 4
150  INPUT "HOW MANY MONTHS ";N
160  IF N = 1 THEN  INPUT "MONTH # ";M
170  IF N > 1 THEN  INPUT "FIRST MONTH # ";M
180  INPUT "YEAR ";Y
200  PRINT "VIDEO OR PRINTER? (V/P) ": GET X$: IF X$ = "P" THEN  TEXT : PR# 2: PRINT  CHR$ (01): PR
INT  CHR$ (28): HTAB 17: PRINT Y: HTAB 17: PRINT "****": PRINT : PRINT : PRINT  CHR$ (29):SW = 1: G
OTO 220
210  IF X$ <  > "V" THEN 200
215 SW = 0
220  HOME : VTAB 4
230  REM  CONVERT TO ZELLER MONTH & YEAR
240 MZ = M - 2:YZ = Y
245  IF SW = 1 THEN 260
250 K = K + 1: IF K > 13 THEN K = 2
255  IF K = 3 OR K = 5 OR K = 7 OR K = 9 OR K = 11 OR K = 13 THEN  VTAB 21: PRINT "HIT ANY KEY TO C
```

contd.

Perpetual Calendar
A Revision

by Donald E. Kahler

The April, 1980 issue of MICRO contained an article and program by Mel Evans of Ann Arbor, Michigan for a PERPETUAL CALENDAR for the AIM computer. It required little revision to run on the APPLE, but needing some extra calendars for 1981, I decided to fancy it up a bit for the Paper Tiger printer.

As written by Mr. Evans, the program will make a calendar for any year, any month, or any combination of the two. (HEDGE) At least, I haven't found a combination that wont work.

As revised, the program will either display two months at a time, vertically, on the video screen; or simultaneously print out two months at a time, horizontally, with a full year on a page.

```
CONTINUE": GET X$: PRINT X$: HOME : VTAB 4
 260  IF M = 1 THEN MZ = 11:YZ = Y - 1
 270  IF M = 2 THEN MZ = 12:YZ = Y - 1
 280  REM  FIND STARTING DAY-OF-WEEK
 290 CZ =  INT (YZ / 100 + .005):YZ = YZ - 100 * CZ:DM = 1
 300 D1 =  INT (2.6 * MZ) + DM + YZ
 310 D1 = D1 +  INT (YZ / 4 + .1) +  INT (CZ / 4 + .1) - 2 * CZ
 320 DW = D1 - 7 *  INT (D1 / 7 + .01) + 1
 325  IF SW = 0 THEN 430
 330 M = M + 1: IF M > 12.5 THEN M = 1:Y = Y + 1
 340 N = N - 1
 350 MZ = M - 2:YZ = Y
 355  IF M = 1 THEN MZ = 11:YZ = Y - 1
 360  IF M = 2 THEN MZ = 12:YZ = Y - 1
 370 K = K + 1:CZ =  INT (YZ / 100 + .005):YZ = YZ - 100 * CZ:DM = 1
 380 D2 =  INT (2.6 * MZ) + DM + YZ
 390 D2 = D2 +  INT (YZ / 4 + .1) +  INT (CZ / 4 + .1) - 2 * CZ
 400 DX = D2 - 7 *  INT (D2 / 7 + .01) + 1
 410  REM   PRINT HEADER
 420  PRINT R$(M - 1);: PRINT " ";: PRINT Y;: PRINT " ";: PRINT "***";: HTAB 24
 430  PRINT R$(M);: PRINT " ";: PRINT Y;: PRINT " ";: PRINT "***"
 440  PRINT " S  M  T  W  T  F  S  ";: IF SW = 0 THEN  PRINT : GOTO 460
 450  HTAB 24: PRINT " S  M  T  W  T  F  S  "
 460  REM  BUILD FIRST DATE-LINE & PRINT
 470 L$ = "":D1 = DW - .5: IF SW = 0 THEN 490
 480 L2$ = "":D2 = DX - .5
 490  FOR I = 1 TO 7
 500 DT = I - DW + 1
 510  IF I < D1 THEN L$ = L$ + "  "
 520  IF I > D1 THEN L$ = L$ + " " +  CHR$ (48 + DT)
 530  IF I < 6.5 THEN L$ = L$ + " "
 532  NEXT I
 535  IF SW = 0 THEN 590
 538  FOR I = 1 TO 7
 540 DU = I - DX + 1
 550  IF I < D2 THEN L2$ = L2$ + "  "
 560  IF I > D2 THEN L2$ = L2$ + " " +  CHR$ (48 + DU)
 570  IF I < 6.5 THEN L2$ = L2$ + " "
 580  NEXT I
 590  PRINT L$;: IF SW = 0 THEN  PRINT : GOTO 600
 595  HTAB 24: PRINT L2$
 600  REM  CHECK FOR LEAP-YEAR
 610 C =  INT (Y / 100 + .005):YC = Y - 100 * C
 620 A(2) = 28
 630  IF YC = 4 *  INT (YC / 4 + .1) THEN A(2) = 29
 640  IF YC < .5 THEN A(2) = 28
 650  IF YC < .5 AND C = 4 *  INT (C / 4 + .1) THEN A(2) = 29
 660  REM  BUILD REMAINING DATE-LINES AND PRINT
 670 EN = 0
 675 EO = 0
 680 L$ = "":L2$ = ""
 690  FOR I = 1 TO 7
 700 DT = DT + 1: IF SW = 0 THEN 707
 705  IF DT > A(M - 1) + .5 THEN EN = 1: GOTO 758
 706  IF SW = 1 THEN 710
 707  IF DT > A(M) + .5 THEN EN = 1: GOTO 830
 710 D1 =  INT (DT / 10 + .05):D2 = DT - 10 * D1
 720  IF D1 < .5 THEN L$ = L$ + " "
 730  IF D1 > .5 THEN L$ = L$ +  CHR$ (48 + D1)
 740 L$ = L$ +  CHR$ (48 + D2)
```

contd. on page 25

Flavors - Little Tidbits

by Burton S. Chambers III

(Flavors chosen for each tidbit are not always an indication of content. My intent is to share with you information that I believe may be generally unavailable elsewhere and do so in a timely fashion. Unfortunately, this may give the appearance of my being overly concerned with problems. Let me assure you, I am extremely pleased with the hardware and software produced by Apple and many of their associated vendors.)

APPLE: A new release of Apple Pascal: Version 1.1

Finally, the new release of Apple Pascal is out. You can now introduce lower case letters from your standard unmodified Apple. Unfortunately, Apple only included translation of the letters, I would have liked to have seen the remainder of the set generated as well. I hope they had a good reason for only going half-way.

Nevertheless, the new release allows some neat features that greatly increase the Apple's utility. The capability to execute TEXT files has been added, where the TEXT file serves as the keyboard to all the system routines. This allows you to use the Filer without being its slave. For example, lets say you don't like your directory listing with .TEXT files mixed in with .CODE files. You could generate the following TEXT file to list the .TEXT and .CODE files, that are on your boot diskette, onto the console and a listing of all files onto the printer. The F gets you into the Filer and the Q quits the Filer.

```
FE*:=.TEXT
E*:=.CODE
E*,PRINTER:
Q
```

And there is much more. You can also chain programs, and pass information as well. This allows you tremendous flexibility. In addition, up to 15 segments (procedures, functions, or Regular UNITs) are now allowed plus the main program segment. These features and many more make it well worth it to change to the new release, especially since some of the more frustrating problems have been fixed, such as Error 407. Futhermore, Regular UNITs can now use Intrinsic UNITs. But Separate UNITs have been deleted from the system. They didn't work anyways, and with Chaining, I suspect we can live without them. Packed Array of BOOLEAN also now works as well as the way NOT is performed. Apple lists 31 corrected problems some of which were really troublesome.

It all costs $60, UNLESS you are still under warranty (extended or otherwise) in which case its FREE.

BANANA: BIOS for Apple Pascal version 1.1

The BIOS has been changed in the new release. Does anyone have the new BIOS specification? Incidentally, I recommend you keep backup copies of the old release. Programs that relied heavily on release dependencies may not run in the new release. If any program you 'own' is only available in p-code, you may need to keep backups of the old release indefinitely. I just talked to the good folks in Customer Service at D. C. Hayes Microcomputer, Inc. Unfortunately, they didn't know about the new release, and therefore, they haven't modified Datacomm yet to interface with the new BIOS. In fact, Customer Service doesn't think they even have the new BIOS description.

Just keep a backup of your old version. This is generally good practice anyways.

CHERRY: Watching out for Byte-eating Compilers.

An interesting thing happened to me while following Apple's 'advice', I lost 1,000 bytes! The compiler ate them! This may be a serious problem or it may be a quirk, but since I don't know, I wanted to warn you. Consider the following coding:

```
FUNCTION NEGATIVE: BOOLEAN;
  CONST MAXBYTE=255;
        MINBYTE=0;
        FLIP=-12524;
  BEGIN
    CASE BYTEVALUE(FLIP) OF
      MINBYTE: NEGATIVE:=TRUE;
      MAXBYTE: NEGATIVE:=FALSE
    END
  END;
```

contd.

This is taken from the Apple Silentype(tm) Operation and reference manual (top of page 55). Looks simple enough, now here's my version:

```
(*-------------------------------------------------------------------------*)
   FUNCTION negative: BOOLEAN;
   CONST minbyte = 0; maxbyte = 255; flip = -12524;

   BEGIN
     negative := ( bytevalue(flip) = minbyte )
   END;
(*-------------------------------------------------------------------------*)
```

Ignore the fact that neither is the best coding, since neither defines any error condition if BYTEVALUE returns a number outside the acceptable range. The compiler, incredibly, sets up coding requiring 554 bytes plus the expected 3 words for the declarations in the first example, whereas in the second example it only needs 24 bytes plus 3 words. Whatever the reason, the same result is obtained for both releases of the Operating System. I lost 1,000 bytes because two such routines exist in the Silentype manual.

The moral of this story is to get a compiler-generated listing of each of your programs from time to time, and then study them a bit. Finally, let us all remember that the people who generate this software are also fallible, and hence, it makes good sense to be somewhat skeptical of what our loving Apple's are telling us.

DATE: Sometime near the end of the Old Year.

Now is the traditional time of the year for New Year resolutions. I recommend the following resolution for the coming year: Spend some time with the family, away from the Apple.

FIG: Style

Please note the style in my example above. Not only does the row of dashes set off each procedure and function, but its use allows the compiler to tell you how big in bytes the previous routine was. Similarily, the blank line between the declarations (CONST, TYPE and VARs) and the BEGIN allows the compiler to tell you how many words are taken in the declaration part of your routine. This is handy information when you are trying to figure where all the bytes are going. (Remember the hungry compiler!)

---

contd. from page 23

```
750  IF I < 6.5 THEN L$ = L$ + " "
752  NEXT I
753  IF SW = 0 THEN 830
755 L2$ = ""
758  FOR I = 1 TO 7
760 DU = DU + 1: IF DU > A(M) + .5 THEN EO = 1: GOTO 830
770 D2 =  INT (DU / 10 + .05):D4 = DU - 10 * D2
780  IF D2 < .5 THEN L2$ = L2$ + " "
790  IF D2 > .5 THEN L2$ = L2$ +  CHR$ (48 + D2)
800 L2$ = L2$ +  CHR$ (48 + D4)
810  IF I < 6.5 THEN L2$ = L2$ + " "
820  NEXT I
830  PRINT L$;: IF SW = 0 THEN  PRINT : GOTO 840
835  HTAB 24: PRINT L2$
840  IF EN < .5 THEN 680
842  IF SW = 0 THEN 850
845  IF EO < .5 THEN 755
850  PRINT
860  REM  DO AGAIN FOR NEXT MONTH
870 M = M + 1: IF M > 12.5 THEN M = 1:Y = Y + 1
880 N = N - 1: IF N > .5 THEN 240
890  TEXT : VTAB 22: PRINT  CHR$ (02); CHR$ (30): PR# 0: END
900  REM  DATA:MONTH LENGTHS AND NAMES
910  DATA 31,28,31,30,31,30,31,31,30,31,30,31
920  DATA    *** JANUARY,** FEBRUARY,***** MARCH
930  DATA   ***** APRIL,***** MAY  ,****** JUNE
940  DATA  ****** JULY,**** AUGUST,* SEPTEMBER
950  DATA *** OCTOBER,** NOVEMBER,** DECEMBER
960  PR# 2: LIST
```

contd.

]

1981
****

*** JANUARY 1981 ***

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

** FEBRUARY 1981 ***

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

***** MARCH 1981 ***

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

***** APRIL 1981 ***

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

***** MAY 1981 ***

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

****** JUNE 1981 ***

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

****** JULY 1981 ***

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	
26	27	28	29	30	31	

**** AUGUST 1981 ***

S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

* SEPTEMBER 1981 ***

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			
27	28	29	30			

*** OCTOBER 1981 ***

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

** NOVEMBER 1981 ***

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

** DECEMBER 1981 ***

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

INTERNATIONAL APPLE CORE TM

NEWS BULLETIN

This is our first issue of the IAC News Bulletin, the purpose of which is to provide to you on a monthly basis everything you ever wanted to know about the IAC but were afraid to ask.

Our format will change in response to feedback from you.

First off, we would like your suggestions for a name - IAC News Bulletin doesn't hack it.

So we hereby announce a name contest to begin as of the date of this issue. The contest entries must be postmarked no later than January 15, 1981. Entries will be judged and the winner selected by the IAC Board. We'll announce the winner and the prize in the February issue.

We have a constitution and bylaws! After considerable time and effort we are officially an organization. Sorry about not involving more of you - too difficult to pull off. However, according to powers provided to you, you may call for changes through special meetings and procedures. Check your copy for further details. Joe Budge, our Secretary, will mail them to you soon.

We also elected Jerry Vitt of Dallas to be Chairman of the Board.

Coming up - a call for nominations for Board Directors. Joe has sent out information describing the procedures. Be sure to advise your entire club membership. By all means, participate. IAC is here to serve you.

The annual IAC general meeting will be held in Chicago May 2 and 3. More information to follow as to time and place.

Now that we are "organized" we can get on to the business of defining more precisely what there is in it for you to be a member of IAC. Under consideration are such goodies as workshops for small businessmen (doctors, lawyers, bowling alleys, etc.), IAC ABBS through low cost WATS type lines, review of serious works and major hardware, ombudsman services, sponsorship of major projects, professional certification of members, and more. More on that later...

Reminder - renewal memberships will be due March 31, 1981. Your club will be billed. Also we prefer individual membership subscriptions to Apple Orchard. Please accept our apologies for the delay in getting Vol. 1, No. 2 of the Orchard to our subscribers. Yes, we goofed! However, from now on, subscribers are to get first preference, meaning three weeks earlier than either through the club or at the computer stores.

contd.

P. O. BOX 976, DALY CITY, CALIFORNIA 94017 USA

A plea - Val Golding needs your input. It's your journal - remember, your magnum opus gets considerably greater visibility through publication in the Orchard. Try it!

Have you heard...?

...Craig Vaughan of Peripherals Unlimited has moved closer to the SOURCE. He now resides in Virginia. His new firm, Microsoftware Systems, has acquired the rights to all Peripherals Unlimited's software, and he can be reached at (703) 385-2944.
...Programma International has been bought out by Hayden Publications. Our Treasurer, Dave Gordon, is Vice President and General Manager for Programma.
...Videx has announced a new "keyboard enhancer" which provides upper and lower case display and entry with the standard 40-column APPLE keyboard.
...Novation has announced a new modem interface card for the APPLE.
...The "Smarterm" 80-column cards distributed by Apple are on their way to the dealers.
... Apple sent a note to all of their Level One Service Centers stating that old disk drives may need some modification to their analog cards to work properly with 16-sector DOS 3.3 and Pascal.
...The IAC Board has awarded a contract to DiLithium Press to print issues No. 4 through 7 of the Apple Orchard. Grawin Publications will do issue No. 3. We thank Grawin for the excellent job they have done on No. 2.

Your IAC disks and APnotes should be coming to you directly from the mail order firm responsible to Joe Budge. We're sure you will agree that the Apnotes are great and we plan to improve the quality and documentation of our disks. Very shortly you will be getting standards notes.

Check the Help Wanted section of your next issue of the Orchard...

About the DOS 3.3 problem. Apple tells us that it was an unfortunate oversight that caused the problem - how many of you have 32K machines? As soon as they heard, they broke records patching the system and had it completed in one week! They do try hard.

Let us know of news, rumors, whatever you would like to share with your counterparts. Comments and feedback should be sent to Bernie Urban, Editor, at the IAC address. Comments may also be passed along to Chairman of the Board Jerry Vitt, who may be reached on his daytime modem (214) 369-0427. We encourage you to pass on this information to your membership through your newsletter or otherwise. Feel free to reproduce anything herein.

December 1, 1980

Please note the new mailing address for subscriptions to the Apple Orchard.

APPLE ORCHARD SUBSCRIPTIONS
P.O. Box 1493
Beaverton, OR 97075, U.S.A.

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

*Vol. 181 required with these disks.

TOTAL ORDER = $ ------

Check here if you want these shipped---

NAME ----------

ADDRESS ----------

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

TELEPHONE ----------

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

Make checks payable to "Washington Apple Pi"

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

INTERNATIONAL APPLE CORE TM

APPLE ORCHARD SUBSCRIPTIONS

P.O. BOX 1493 BEAVERTON, OR 97075, USA

The International Apple Core will make individual subscriptions to "The Apple Orchard" available commencing with Volume 1, Number 3.

NAME _____

STREET _____

CITY _____ STATE _____ ZIP _____

COUNTRY _____

Annual Subscription Rate: $10.00 per year (Four issues - Published Quarterly)
First Class Postage: $5.00 per year additional (required for Canada, Mexico, APO, and FPO addresses)
Overseas and other foreign air mail postage (required): $10.00 per year additional

TOTAL REMITTANCE ENCLOSED: $(USA) _____

Make check or money order payable to "International Apple Core" and return with this form to:

APPLE ORCHARD SUBSCRIPTIONS
P.O. Box 1493
Beaverton, OR 97075, U.S.A.

1/81

(Reprinted with permission from the November 15, 1980 edition of URBAN FUTURES IDEA EXCHANGE.)

* Computer Home Banking Has Arrived...at least in Knoxville, TN. An Express Information bank-at-home service has been started up by a group of computer-oriented firms working along with the United American Bank. With this system, consumers are able to use the bank's services via a home computer, according to the group. The hardware consists of a TRS 80 computer and keyboard that plug into both a TV set and the telephone. This costs between $15 and $25 per month. The package of services purchased includes the in-home banking plus news reports, financial advisory information, bill paying capability, account status information, bookkeeping services, and loan services. Will it balance the checkbook? Other services such as electronic mail, tax services and more are potential add-ons. The group announced the Knoxville launch in 1980 and proclaimed plans to go national in 1981.

Urban Futures Idea Exchange ®

MESA, INC. ANNOUNCES THE NEWEST PAPER TIGER

IDS-445

*ADVANCED BALLISTIC HEAD *CARTRIDGE RIBBON

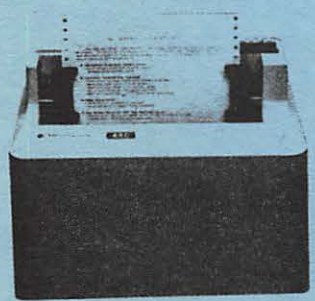
MESA OFFERS DISCOUNTS TO QUALIFIED APPLE PI MEMBERS ON THE PURCHASE OF:

INTEGRAL DATA SYSTEM'S "PAPER TIGER" PRINTERS

* MESA, INC. IS AN AUTHORIZED DISTRIBUTOR FOR IDS

* SERVICE - AUTHORIZED DEPOT SERVICE CENTER - SPARE PARTS & FACTORY TRAINED STAFF

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



MESA, INC.

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

Integral Data Systems, Inc.

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

First Class