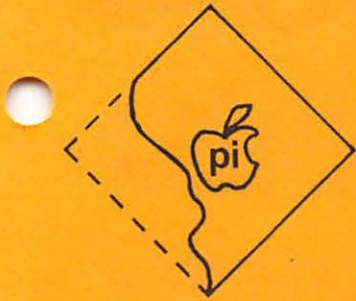


\$1.00



Washington Apple Pi

PO Box 34511 Washington, DC 20034

volume 1

August 1979

Number 7

Officers

President	John Moon	(202) 332-9102	
Vice President	Bernard Urban	(301) 229-3458	
Treasurer	Robert Peck	(301) 770-1954	
Secretary	Genevie Urban	(301) 229-3458	
Newsletter Staff:			
Editor	Bernard Urban	(301) 229-3458	
Associate Editor	Mark Crosby	(202) 488-1979	
Program Librarian	David Morganstein	(301) 474-5768	972-4263
Members-at-Large	Sue Eickmeyer	(301) 490-7627	953-7355
	Sandy Greenfarb	(301) 674-5982	<
	Mark Crosby	(202) 488-1979	
	Phil Eastman	() 971-2282	

Contents

	page#
President's Message.....	1
Minutes of our last meeting.....	1
NIBBLES: tidbits of interesting information.....	1
MODEMania: members who have modems.....	2
6502 Information Resources.....	2
Making Money From Your Programs by Andrew Rose.....	3
Changing "Beneath Apple Manor" by Nicholas B. Cirillo, M.D..	3
APPLE Programming With Style by John L. Moon.....	4
Minutes of NOVAPPLE meeting.....	5

NEXT MEETING: Saturday, August 25, 9:30 am at G.W. University -
 Tompkins Hall, School of Engineering, Room 206.
 23rd and H Streets N.W. Near Foggy Bottom/GWU
 Metro Station.

PRESIDENT'S MESSAGE

Dear APPLE Enthusiast:

I've just recently attached a modem to my APPLE and would like to spend some time talking about various opportunities and possibilities that I see in using telecommunications with a personal computer.

First of all, I was amazed to find out just how easy it was to use the thing. The first night that I took it home, I spent the night calling up computer bulletin boards and playing telepong instead of writing articles for this newsletter as I had promised Bernie (sorry Bernie, I already gave you my apologies on that score).

For the uninitiated, a little explanation is probably worthwhile. The modem (along with the controller card that plugs into the APPLE) allows my computer to send or receive data over a telephone line to other computers with compatible modems. In my case, the modem also can dial the telephone number or answer the phone if it rings. Firmware (software in ROM like the APPLE monitor) is on the controller card to allow simple use of the modem to act as a terminal or as a remote computer. More complex capabilities can be achieved through more complex user programming.

My first use of the modem was to call the AMRAD CBBS (AMateur RADio Computer Bulletin Board Service). Their number is 281-2125 and their computer answers automatically. A CBBS such as this reminds me a lot of a CB channel. There are lots of ways of doing your talking that you only learn after doing it for awhile. Also, you will find messages on all sorts of different topics. The basic capabilities of most CBBS's include ways to extract a summary of the current messages, enter new messages, read old messages, and delete obsolete messages. The commands to do these things are different from CBBS to CBBS. Typically, when you call up and hit the carriage return a few times, they will identify themselves and if they don't give you instructions, try entering "H" or "HELP". They are written to be "user-friendly" so they usually aren't too hard to figure out.

I currently know of three CBBS's in the immediate area (and many more nationwide). AMRAD has what seems to be the most established CBBS. It is operating 24 hours a day, 7 days a week unless it's broken. From what I can gather, it runs on a SWTP 6800 of some sort. A guy named Bruce Thompson has a CBBS running part time in Silver Spring on an 8080 based North Star. Look in the AMRAD CBBS for the phone number and hours of operation (leave a message asking for it if necessary!). He has some unique features such as games that can be played on-line.

Lee Hausman has recently brought up a system running on his APPLE. He wrote it himself; he uses APPLESOFT BASIC, the DOS, an IDS printer and a D.C. Hayes Micro Modem as the basic hardware. I spent an interesting evening talking to him on the phone and hope to invite him to one of our meetings to tell about the experience of writing the CBBS and the anomalies he ran into (such as callers with terminals that didn't have certain keys).

Other rumors and raw data that I've heard or seen lately include a "hard" disk for the APPLE. Advertised in BYTE magazine this month for about \$5,300 for the first drive with controller and \$2,900 for each additional drive is a "winchester" technology system by Corvus Systems Inc. From what I hear, this drive is interfaced through a patched DOS that uses the VOLume number parameter in the commands to make it

look like there are several hundred floppies out there where the hard disk is. I think this may be the system that some of you who went to the NCC may have seen.

On Wednesday, September 12 at 7:30 pm, The Source (that telecommunications/timesharing service that we discussed at the last meeting) is going to demonstrate their stuff on an APPLE at Tysons Corner Computerland. Anybody that likes to see before they buy might want to take a look.

A short paragraph in Electronics News alluded to a commitment by APPLE to buy some 10,000 of the H-plot digitizer pads. That sounds like some kind of graphics emphasis at some time in the future. I've seen some graphics papers by some researchers where they had written software for the APPLE to allow it to "look" like a Tektronix graphics terminal (a very widely used graphics terminal) to a computer.

That's all I have right now so I'll see you at the next meeting.

John L. Moon

MINUTES

A short business meeting was followed by a hands-on session at the many APPLE's available to us at the GWU Computer Center. Most of the business centered on various hardware and other general information:

- * A demonstration of an IDS 440 printer is planned for a future meeting.
- * The new Auto ROM is available in the metropolitan area. At the meeting, Computerland at Tysons was known to have it in stock. Other stores should have it by now too.
- * The California Drivers Test on a recent user submitted software disk from APPLE requires HIMEM be set to 32000 to operate correctly on larger machines.
- * We now have a PO Box. Jump to cover to check it out.
- * A fellow user from Europe joined us and we compared availability and costs of APPLE's.

NIBBLES

The Washington Amateur Computer Society (WACS) is an organization here in the nation's capital dedicated to personal computing. They are organized to provide a forum for the computer hobbyist and student of computing sciences. They meet the last Friday of each month at Catholic University. Call Chuck Patten 599-6962 6-9 pm or write: WACS, c/o 4201 Mass. Ave., N.W. # 168, Washington, DC 20016. WACS has their journal on-line at the University. Call 635-5710 for 110 baud or 635-5730 for 300 baud. Type a carriage return to get response then type "HELP WACS" to get the journal. They might not have their system up in the Summer but you can check it out. They also have a message box at (202) 281-2125. Type a carriage return for response and "H" for help information. The message box operates Full Duplex 8 bit no or even parity.

3-D graphics for the APPLE in an assembly language version is soon to be available. This will allow projection rates of 150 lines per second unclipped and 100 lps clipped. A 20 line drawing will be presented at about 5 frames per second unclipped. Routines allow orbiting scene, zoom, 42 ms. screen erase, page control gives erase, display and draw for "Ping-Ponging" between screens allowing smooth animation. Selective erase permits partial screen erase. Look for a review on it soon. Package includes a technical manual and cassette: \$45. Add \$10 for disk option. Order # A2-3D1. 1st Class add \$1.50, UPS add \$.75, COD add \$2.00. Order from SubLogic Box V, Savoy, IL 61874 (217) 359-8482.

Available at the next meeting will be corrected code for AMPER-SORT presented in Micro July 1979 by Alan Hill. This is a fast sorting code that interfaces to Applesoft allowing you to sort your arrays quickly. Also will be reprints of an article in another newsletter listing the Variables used by Basic and a Sub-routines list and including the Apple Basic Interpreter Instruction Set

MODEMania

We have members with modems. Here is a list of names and numbers given to us at our last meeting. Also, see ABACUS, July 1979 Volume 1 issue 7 for a list of Software and Services - many of which operate modems. If you have a modem but are not listed, please drop us a line and we will publish your name and number too...

NAME/RATE/ORIGINATE-ANSWER/TELEPHONE

Susan Eickmeyer/300/0/(301) 953-7355
 John Moon/110/300/0-A/(202) 332-9102
 Jim Manley/110/300/0/(301) 426-9248
 Hersch Pilloff/300/0/(301) 292-3100
 Hayden Porter/300/0/(301) 946-7786
 Howard Richoux/110/300/0/(703) 525-9889
 David Morganstein/300/0-A/(301) 585-4375

6502 Information Resources

by William R. Dial, 438 Roslyn Ave., Akron, OH 44320

Reprinted from MICRO 13 June 1979 with permission and updated by Mark Crosby.

Appleseed
 PO Box 68
 Milford, NH 03055

Applesauce \$10/yr.
 12804 Magnolia
 Chino, CA 91710

6502 User Notes
 \$13/6 issues
 Eric Rehnke
 PO Box 33093
 Royalton, OH 44133

Pet Gazette
 Free bi-monthly
 Contributions accepted
 Microcomputer Resource Ctr
 1929 Northport Dr. Room 6
 Madison, WI 53704

The Seed
 9711 Josephine
 St. Thornton, CO 80229
 \$1/copy

MICRO \$15/yr.
 PO Box 3
 S. Chelmsford, MA 01824

Ohio Scientific-Small
 Systems Journal
 \$6/yr./6issues
 Ohio Scientific
 1333 S. Chillicothe Rd
 Aurora, OH 44202

Robert Purser's Refer-
 ence List of Computer
 Cassettes. Nov 1978 \$2,
 Feb 1979 \$4
 Robert Purser
 PO Box 466
 El Dorado, CA 95623

The Paper(Pet)
 \$15/yr/10 issues
 The PAPER
 PO Box 43
 Audubon, PA 19407

Stems from APPLE
 Ken Hoggatt
 A.P.P.L.E.
 9195 SW El Rose Court
 Tigard, OR 97223

Kilobaud/Microcomputing
 \$18/yr
 Kilobaud Magazine
 Peterborough, NH 03458

Dr. Dobb's Journal
 \$15/y/10 issues
 Peoples Computer Co.
 Box E
 1263 El Camino Real
 Menlo Park, CA 94025

Recreational Computing
 (Formerly Peoples Computers)
 \$10/y/6issues
 Peoples Computer Co.
 1263 El Camino Real
 Box E
 Menlo Park, CA 94025

Popular Electronics
 \$12/yr
 Popular Electronics
 One Park Avenue
 New York, NY 10016

73 Magazine
 \$15/yr
 73, Inc.
 Peterborough, NH 03458

SSSC Interface
 Southern Calif. Comp. Soc.
 1702 Ashland
 Santa Monica, CA 90405

Radio Electronics
 \$8.75/yr
 Gernsback Pub., Inc.
 200 Park Avenue South
 New York, NY 10003

IEEE Computer
 IEEE
 345 E. 47th St.
 New York, NY 10017

Rainbow(Apple)
 \$15/yr
 Rick Simpson, Ed.
 PO Box 43
 Audubon, PA 19407

CONTACT -User Group Newsltr
 Gratis to APPLE owners
 10260 Bandlely Drive
 Cupertino, CA 95014
 (408) 996-1010

Computer Music Journal
 \$14/yr/6 issues
 People's Computer Co.
 Box E
 1010 Doyle St.
 Menlo Park, CA 94025

The Cider Press (APPLE)
 Scot Kamins
 Box 4816
 San Francisco, CA 94101

Apple Seed
 Bill Hyde
 The Computer Shop
 6812 San Pedro
 San Antonio, TX 78216

BYTE
 \$18/yr
 BYTE Publications, Inc.
 70 Main Street
 Peterborough, NH 03458

On-Line
 \$3.75/yr/18 issues
 D. H. Beetle
 24695 Santa Cruz Hwy
 Los Gatos, CA 95030

Interface Age
 \$18/y
 McPheters, Wolfe & Jones
 16704 Marquardt Ave.
 Cerritos, CA 90701

Personal Computing
 \$14/yr
 Benwill Publishing Corp.
 1050 Commonwealth Avenue
 Boston, MA 02215

Creative Computing
 \$15/yr
 Creative Computing
 PO Box 789-M
 Morristown, NJ 07960

Electronic Design News
 \$25/yr
 Cahners Publishing Co.
 270 St. Paul St.
 Denver, CO 80206

QST
 \$12/yr
 American Radio Relay League
 225 Main St.
 Newington, CT 06111

Polyphony
 \$4/yr
 PAIA Electronics, Inc.
 1020 W. Wilshire Blvd.
 Oklahoma City, OK 73116

Pet User Notes
 \$5/y/6+issues
 PET User Group
 PO Box 371
 Montgomeryville, PA 18936

Southeastern Software Newsltr
 \$10/10issues
 George McClelland
 Southeastern Software
 7270 Culpepper Drive
 New Orleans, LA 70126

Popular Computing
 \$18/yr
 Popular Computing
 Box 272
 Calabasas, CA 91302

Mini-Micro Systems
\$18/yr.
Modern Data Service
5 Kane Industrial Dr.
Hudson, MA 01749

Electronic Design
26 issues per year
Hayden Pub. Co., Inc.
50 Essex St.
Rocheille Park, NJ 07662

Digital Design
\$20/yr.
Benwill Pub. Corp.
1050 Commonwealth Ave.
Boston, MA 02215

Call-A.P.P.L.E.
\$10/yr includes dues
Apple Puget Sound
Program Library
6708 39th Ave., S.W.
Seattle, WA 98136

Software Central
1301 W. Estes Ave.
Chicago, IL 60626

Softape
10756 Vanowen
N. Hollywood, CA 91605

AppleSeed
PO Box 68
Milford, NH 03055

ARTICLES

MAKING MONEY FROM YOUR PROGRAMS by Andrew Rose

Most of you have probably written some good programs and are wondering what to do with them. I have done some research and compiled a list of publishing houses that will publish good programs (for the APPLE) and throw in some money too.

APPLE SOFTWARE BANK: If you send your software here and it is accepted (as it more than likely will be), you will receive a one time \$25 gift certificate for the APPLE. Or you will receive (your choice) 4 free programs from APPLE's small selection of pay-for programs.

MUSE: I sent a letter to MUSE asking for a software submission form and received none so there isn't very much to say. MUSE is looking for fairly high-class software (Tank-War, Mazegame, etc.). They pay royalties as long as the product sells.

INSTANT SOFTWARE: is run by Kilobaud-Microcomputing magazine. The company mostly backs TRS-80 and PET. They have only 3 Apple programs. If you are thinking of sending your software there, they do give royalties.

NewWare: Most of you have probably never heard of this company. They deal mostly in business software (DBMS, LEDGER, INVENTORY, Etc.). If you're interested in that area drop them a card.

SOFTWARE CENTRAL: Another unheard of company. They deal in things like word processors and index file systems. Again, if that's what you like, send them a letter.

SOFTAPE: This company publishes most of Bob Bishop's programs. They are looking for high quality games and demos. If your software is accepted, you receive \$25. Whether or not royalties are involved, I don't know.

APPLESEED: I received a letter from this company stating that they will publish programs for the APPLE at 20% royalties. This sounds like a good deal, but I'm not sure what kind of programs they want. I presume they want mostly games.

FOR INFORMATION, WRITE TO:

APPLE Software Bank
APPLE Computer Inc.
10260 Bandle Drive
Cupertino, CA 95014

The MUSE Company
PO Box 13365
Baltimore, MD 21203

Instant Software Inc.
Dept MA
Peterborough, NH 03458

NewWare
08525 Park Drive
Omaha, NE 68127



CHANGING "BENEATH APPLE MANOR" by Nicholas B. Cirillo, M.D. (NOVAPPLE)

Since my son and the neighborhood boys have a fanatical interest in warfare simulation board games it was natural for them to enjoy various computer games. Unfortunately, they quickly tired of the arcade-type games and after awhile even STARTREK couldn't hold their interest. I was discussing other possibilities with Rod Clarke at a NOVAPPLE meeting when he suggested I buy BENEATH APPLE MANOR by Don Worth (Software Factory).

Beneath Apple Manor is a "Dungeons and Dragons" styled simulation. It is constructed in three parts that work best on a 32K APPLE with DISK II. It can run in 16K from cassette but in this configuration it will require reloading a maze drawing machine language program.

Although I am not a game player, I did enjoy the few rounds I played after loading the program onto the disk. After the boys got hold of it I couldn't get near the computer for days. Parents of other boys were having to come over and almost physically remove their sons from the keyboard. These interruptions of their games led them to look further into the rather complete documentation and find a technique for noting down and saving their acquired attributes so that they could continue the game at a future time. This was awkward and time-consuming and required care so as not to crash the entire program and lose the data. Therefore I listed the code and developed the routines noted below to facilitate this process.

I suggest the converted programs be saved on another disk so that the original program can be preserved. Do not change the program names unless you are willing to list the entire program and change all the CHAIN commands.

Procedure: Load BENEATH APPLE MANOR (don't run it) then add the following lines:

```
16 DIM A$(15)
129 DIM NAME$(15)
130 PRINT "IS THIS GAME A CONTINUATION (Y/N)": INPUT A$:
    IF A$="N" THEN 140
132 PRINT "WHAT IS YOUR NAME ": INPUT NAME$
133 D$="": REM CTRL-D IN QUOTES
134 PRINT D$;"OPEN BAMUTILITY";NAME$: PRINT D$;
    "READ BAMUTILITY";NAME$
141 IF A$="N" THEN 160
142 INPUT PS,PSM,PI,PIM,PD,PDM,PB,PBM,EXP,GOLD,BANK,
    BSS,BSI,BSD,BSB,BSE
143 FOR V=1 TO 7 : INPUT MI(V): NEXT V
144 PRINT D$;"CLOSE"
146 CALL-936: FOR I=1 TO 8: PRINT: NEXT I
150 IF A$ # "N" THEN PRINT "REMEMBER TO GO ONE LEVEL
    DEEPER"
152 FOR I = 1 TO 1000: NEXT I
155 GOTO 200
```

Next, SAVE BENEATH APPLE MANOR (remember on a different disk). Then LOAD BAM1 (DO NOT RUN IT). Then type:


```

20530 TEXT: CALL-936: D$="": REM CTRL-D IN QUOTES
20545 PRINT"DO YOU WANT TO CONTINUE THIS GAME ":
PRINT"ANOTHER TIME (Y/N)"
20547 INPUT A$
20548 IF A$="N" THEN 20580
20549 PRINT"WHAT IS YOUR NAME ": INPUT NAME$
20550 PRINT D$;"OPEN BAMUTILITY";NAME$
20555 PRINT D$;"WRITE BAMUTILITY";NAME$
20560 PRINT PS:PRINT PSM:PRINT PI:PRINT PIM:PRINT
PD:PRINT PDM:PRINT PB:PRINT PBM
20562 PRINT EXP:PRINT GOLD:PRINT BANK:PRINT BSS:
PRINT BSI:PRINT BSD:PRINT BSB:PRINT BSE
20564 FOR V=1 TO 7: PRINT MI(V): NEXT V
20570 PRINT D$;"CLOSE"
20580 PRINT" SO LONG": END

```

Now SAVE BAM1 then transfer the machine language program to the new disk.

Now when you RUN the program it will ask whether this is a continuation. The code "NAME" will separate various players' data. Be sure to go the next level down otherwise you will have an unfair advantage over the monsters when you go into the program.

I hope you enjoy this game and its modification as much as we have.



APPLE Programming With Style
by John L. Moon

In this article, I will cover a number of situations that are often faced when writing programs. Without being dogmatic about it (I hope). I will try to give some guidelines as to how to accomplish various things so that a more or less standard way may be selected. In a very real sense, this is about the "style" of programming rather than the "subject".

The first topic in programming style is program structure. A program can be very much like a well-organized recipe - where you know from experience that the ingredients and the directions are separated. A program should have some kind of explicit structure too. It makes it easier to find your way around the code when you have a bug (it happens to all of us) and furthermore makes it much easier for the next user of your program to add those features that he thinks are necessary.

To end up with a structured program (not in the if-then-else, while-do sense for the computer scientists out there) you need to have that structure in mind from the beginning when you write the program. After you get the program running, it's not worth the effort to move all the lines around because of all the retyping that would be necessary. Besides, it is during the writing of the programs that the structuring of the program is most useful.

At the highest level of structure for an Integer BASIC program, I would suggest some structure modelled on the following:

```

10 GOTO 30000
100 - 19999 Subroutines
20000 - 29999 Main Program Code
30000 - 30999 Data Initializations
31000 GOTO 20000
31100 - 32764 Comments that are the program
documentation

```

The actual line numbers that I have used are not as important as the order. The reasoning is as

follows:

- 1) Data initializations are only necessary once, therefore they should be located beyond the main program and the subroutines.
- 2) For speed of reference, the subroutines should be placed in front of the main program. A typical ordering should place the ones used most often towards the beginning. Any speed-critical loops (for instance in graphics applications) should be in a subroutine near the front.
- 3) Main programs typically will have some form of menu or commands that the program is interpreting. They are highly dependent on the user of the program providing input, etc., therefore there need be no significant speed penalty to put this after the subroutines.
- 4) At the very end of the program should be comments that describe various sections of the program. A relatively easy way to go about generating these comments will be described below. By documenting the program inside the program it makes it easier to distribute the comments when you give someone a copy of the program.

Obviously, if your purpose is to make your program so confusing that no one can figure it out, don't do any of this!

A common occurrence is the need for a menu or command interpreter. At some point in the program, the user is offered a choice among a number of options. Typically, the whole program can be structured around such a menu. A menu can exist in several different forms. The simplest is to print out a list of options, giving each option a number, and then test the number for each of the options. For example, the following could be the menu for a railroad controller:

```

0 RETURN TO BASIC
1 START THE TRAIN
2 STOP THE TRAIN
3 SWITCH THE CROSSING

```

INPUT OPTION?

It is a good practice to reserve option "0" for all numeric menu selections as a return to BASIC. Also, the number that is input should be checked to verify that it is a valid option number. Always do something safe when the user mistypes an input. A typical response could be to print a message saying the input was out-of-range and re-run the menu.

Although BASIC will allow a number to be used directly in a computed GOTO (GOTO 1000+ INPUT VALUE) this should be avoided. It makes it very difficult for someone to renumber the program and get it to work properly afterwards. Although it would seem to be terribly inefficient, a string of:

```

IF I = 1 THEN GOSUB nnn
IF I = 2 THEN GOSUB mmm

```

is not as bad as you might think since it only happens during an operator input action, which to the computer is terribly slow. It is also a lot clearer as to what is happening for each option. For safety's sake, it is usually best to leave the input variable alone and to not change it in any of the subroutines until it is time to get the next option from the user.

In using menus, there are usually only a few options and, generally, a single digit could be used to enter the option. In these cases, it's a hassle to hit the option number and then hit the RETURN key. If the program does a direct read of the keyboard for a single character then the return would be unnecessary.

There are two problems to be overcome: first, if you are in Integer BASIC, it doesn't have a GET statement as in Applesoft. Secondly, when you PEEK the keyboard, the value you get must be converted from ASCII to a numeric value.

If you use a direct read of the keyboard (PEEK -16384) your program may become incompatible with some other systems. This is true if the program will eventually be used from a communications card, or if someone uses a separate terminal from the APPLE keyboard. The best solution is to use a machine language interface routine such as I have described in one of my earlier articles to use the Monitor "Get Character Routine". The next best solution is to write the direct read of the keyboard as a subroutine. Point this out in the documentation too.

To convert the ASCII numeric value into a binary numeric value is trivial. Assuming the variable "I" has the original keyboard value, then all that is needed is: `I=I-ASC("0")`. Remember to verify that it is in the correct range though!

A refreshing change from numeric options can be had by reading an input string instead of a number. In this case, the string of IF statements can be changed to something like:

```
INPUT "INPUT OPTION? ",I$
IL=LEN(I$): IF IL=0 THEN GOTO (Menu)
IF I$=ZEND$(1,IL) THEN GOTO (end of prog.)
IF I$=ZHELP$(1,IL) THEN GOSUB (help session)
etc.
```

ZEND\$ and ZHELP\$ have, of course, been initialized to the proper values. Several caveats are necessary: the above is set up to recognize any command that has the same initial letters as the length of the keyboard string. If ZHELP\$ = "HELP" then "H", "HE", "HEL" and "HELP" would all be recognized as the same command. To make it fool-proof, all of the Zxxx variables should be extended with 8 blanks or so. The check on the length would then be `IF IL=0 OR IL>8` then circle back and get the string again to prevent errors.

It is very much worth your while to document your program immediately while the functions are still fresh in your mind. It is surprising how much one can forget - even overnight. The easiest way to do this is to list your program in short segments (3-6 lines) and put a series of remarks at the extreme end of the program describing the functions of those lines. There is a real art to this. It is not worth much to simply repeat or describe what the code says, but if the function is described at an understandable, high level, it makes it very easy to locate a line within the program while debugging or modifying the program. For example, a typical line could be:

```
31100 REM 110-130 SUB TO INPUT AND CONVERT TO
      HEXADECIMAL
```

At the very least, each individual subroutine should be delimited along with the initialization sections and the main loop of the program.

Hopefully, this article has given you some ideas about how to set about writing that next program. I didn't really try to teach BASIC, but tried to show a methodology. With a sensitivity to things such as "style", programs begin to move out of the realm of extremely personal "ad hoc" figurings, into durable algorithms that can be shared and enhanced by anyone who has a need.

MINUTES OF NOVAPPLE August 8, 1979

The meeting was called to order by the President, Jim Nielson at 7:40 pm. He gave a brief description of coming events. The program for the next meeting will be on assembly language programming (Aug 23). On September 12 the program will consist of a demonstration and explanation of a time-sharing program known as "The Source". Beginning in October, the meetings will begin alternating between Computerland of Vienna, Virginia on the Second Wednesday and Computers Plus, Franconia, Virginia on the fourth Thursday.

In October, new elections will take place. All persons present were asked to provide their correct address and telephone number, as well as pay dues of \$6 for a half year so we can pay for the newsletters.

The program was presented by Bill Kennedy on Structured Programming. The trend to structured programming began in the 60's and the 70's and essentially was a method of programming using no goto's and a minimum of programming techniques. It stemmed from the fact that 75-80% of the computer scientist's time is spent maintaining old programs and the rest generating new lines. The cost of software has been going up and hardware down. It was necessary to minimize the time to maintain programs and modify them. Old programming methods require inordinate time to debug and modify because of the non-structured methods used. Programmers suggested that all programs be limited to 3 types of routines. They are 1) A DO list, 2) Branch, and 3) a Loop. It has been proven mathematically that all programs could be reduced to these 3 functions. They would be easy to run, have less mistakes, be easier to maintain, and, finally, they could be produced at a faster rate. Three additional structures were added by some programmers...to make it easier to write programs. They were 1) IF-THEN, 2) CASE, 3) DO-UNTIL. It is not easy for persons with bad habits to program using structured methods, but after being thoroughly frustrated by problems, one can be convinced of the need and utility of such methods. New programmers trained in these techniques are just now coming out of schools trained from the start in the new methods. If you are interested in better programming techniques, look through back issues of BYTE and other magazines or get yourself a book or two on structured programming.

The meeting adjourned at 9:15 pm.

Respectfully submitted:

Gerald R. Eskelund

EDITORS' CAGE

WE NEED ARTICLES, CARTOONS, NEWS ITEMS, IDEAS, ETC., ETC., ETC., ETC., ETC.,

Membership: \$6 for 6 months and 6 issues of newsletter. Outside U.S. add \$2 postage. Send Check or M.O. payable to Robert Peck (Treasurer) to PO Box on cover

p.s. We apologize for the late publication schedule we have been following lately. We soon will publish during the 1st week of each month.

Mark