



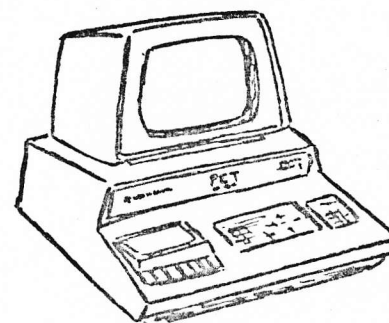
THE PAPER

P. O. BOX 43  
AUDUBON, PA. 19407

To:

AN ARESKO PUBLICATION

THE



PAPER

BOX 43 AUDUBON PA 19407

\$2.00

ISSUE # 4  
VOLUME 1

## EDITORIAL

This has been a most exciting month for everyone - the NCC in Anaheim, California, and the Midwest conference in Detroit. We'll have lots of new information for you as soon as we get it all together.

In this issue, we have not included the list of hardware and software and paperware we usually include. There is a new publication, the PET GAZETTE, published by Len

Lindsay, of Microcomputers Research Center. The Gazette contains info about every conceivable PET peripheral in existence, and there's no fee or subscription cost. We received the Gazette information too late to include it in this issue, but we'll be glad to send the address to you if you want it.

THERE WILL BE NO PAPER PUBLISHED IN JULY. Please remember this when you don't get your expected copy!

The PAPER is published ten times per year and mailed to subscribers on the last day of each month except July and December. Single copy price is \$2.00 per issue, subscription price is \$15.00 per year (all ten issues of one volume.) Dealer prices upon request. Outside of Continental U.S. and Canada, add \$10.00 per subscription for postage (\$1.00 for single copy).

Readers are encouraged to submit articles of general interest to Pet Owners. Material submitted will be considered free of copyright restrictions and should be submitted by the tenth day of the month in which publication is desired. Non-profit organizations (i.e., computer clubs) may reprint any part of the PAPER without express permission, provided appropriate credit is given with the reprint. Any other persons

or organization should contact the editor for permission to reprint PAPER material.

Advertising rates are as follows:

1/4 page -- \$ 50  
1/2 page -- \$ 90  
3/4 page -- \$130  
full page -- \$160

Less than 30% of the PAPER will be available for advertising. Please send camera ready copy in the exact page size of your ad on 8 1/2 x 11 white stock by the 10th day of the month in which you'd like the ad to appear. Photos should be glossy black & white in the exact size to be printed. Payment required with copy.

The PAPER is an Aresco Publication, edited by Terry L. Laudereau. For information contact Editor, The PAPER, P.O. Box 43, Audubon, PA 19407.

## PET PARADE

Terry- Recently, I became the owner of a Model 28 teletype. Perhaps you could tell me if there is a way to interface this unit with my PET. It certainly would be nice to be able to get hard copy of existing programs at a reasonable cost. - Albert E. Nichols

Albert - No, not that we know of. - Terry

Terry- Thank you for printing my name and address in The PET Prose, Issue No. 3.

I would like to address the following questions to The PET Parade:

I read with interest that Commodore has developed a printer for my PET, however, to date, I have received no information of its availability. Why hasn't Commodore announced this option to PET owners? Also, is it possible to interface my PET with an IBM Selectric typewriter?

I enjoy your paper very much, and appreciate your candor with your readers. - George Ettershank

George - You're quite welcome. We're as puzzled as you are by Commodore's non-communicative attitude. But send them your name and PET serial number to "get on the mailing list." Yes, it's possible - we're researching it now. - Terry

Terry - The compETator is recommended as a replacement for The PET Paper. - Enrico Conti

Enrico - Thank you. We're still looking - lots of new ideas have come in. - Terry

Terry - I just finished re-reading issue #2 and I think it's great...better than the User Notes in my opinion. Congratulations, and thanks for an interesting and informative publication. - John Quass

Terry - How do I keep my PET from dropping 0's when I want something like \$19.30 to be printed? Now I just get \$19.3 printed. - Peter L.A. Oakes

Peter - Use the NUMBER EDIT on our Useful Routines tape. As soon as we get our printer, we'll publish a listing of it.

Terry - I have figured out how to split and merge programs without having an Executable Return. You simply poke the new instruction (disinterpreted) in the proper place. Tricky, but it can be done. - George Sherritt

George - Haven't had a chance to try it yet. Maybe one of our readers can use it, though. - Terry

Terry - I am thinking very seriously of filling my RAM from FORETHOUGHT PRODUCTS, however, I hesitate to make such an investment with so little information. Do you plan to evaluate this or similar products in kit form?

For your information I talked to Commodore yesterday by phone and while they were willing to accept payment for the 2nd cassette and the printer, a shipping date for the cassette was vague and for the printer even vaguer. Also they are waiting for a new type cassette which is supposed to "look" better. They now quote three weeks on PET. Has anyone

figured out how to translate TI or HP calculator programs into "PET"? - James B. Trimble

James - We'll do an evaluation as soon as we can - but not for a few months yet. You might write to FORETHOUGHT and see if you can get more info. - Terry

Terry - After an inauspicious start, I think issue 3 is finally what I hoped to get. Good shot!

The stuff you've reprinted from SPHINX is some of what information I find I need. I wonder: could you send me an address to communicate with that group and get on their mailing list? Teacher's Shelter, Oakland Public Schools doesn't

sound awfully promising to me for a mailing address.

Here is how PET stores floating point numbers (a little contribution in exchange).

Bytes 1 and 2: the name and type of the variable

Byte 3: Exponent

Byte 4: First bit (high order) is sign; rest of byte is bits 2-8 of mantissa

Bytes 5-7: more mantissa.

Where did bit 1 of mantissa go?

This bit is implied. The scheme is this:

zero is represented by exponent and all mantissa bytes zero.

a non-zero number is given by

Example:  $2 \uparrow (\text{byte } 3-129) * (-1) \uparrow (\text{first bit of byte } 4) * (1 + \sum_{K=1}^{31} \frac{K^{\text{th}} \text{ next bit of mantissa}}{2^K})$

$$5/16 = 1/4 + 1/16 = 1/4 * (1 + 1/4) = 2 \uparrow (-2) * (-1) \uparrow 0 * (1 + 0/2 + 1/4 + 1/8 + \dots + 0/(2 \uparrow 31))$$

STORES AS

BYTE 3=127, BYTE 4=32=00100000<sub>2</sub>, BYTES 5,6,7=0

SIGN MANTISSA

-5/16 STORES AS

BYTE 3=127, BYTE 4=160=10100000<sub>2</sub>, BYTES 5,6,7=0

This has some implications for how the arithmetic subroutines must work (I haven't verified this yet, though). In multiplication, two numbers with mantissas  $M_1$  and  $M_2$  stored has a product with mantissa  $M_1 + M_2 + M_1 * M_2$ . In addition, when the exponents are equalized by

shifting the mantissa of the smaller number, the implied 1 has to be inserted (probably by a SEC before a bunch of ROR's) or maybe the whole thing is done with some clever logic tricks, so that the justification of the exponents and the addition is done all at once.

I hope this helps with the decoding/documentation problem with the BASIC. The annoyance of not having such documentation, together with the childish trick of defeating simple PEEKs into the Basic area, is what triggered off my own decoding project when it turned out that I was losing too many significant digits due to roundoff error; a 10 byte mantissa would probably be OK for me, but there's no SET DIGITS instruction.

(I'd appreciate it if you withheld my name and address from publication. From what I've read, Commodore is very paranoid about losing the secrets of their BASIC, and I don't want to get on their blacklist.) - A. Nony Mous

Dear A.N.M. - Thank you. SPHINX can be reached via Pete Rowe, Computer Science Dept., Lawrence Hall of Science, Berkeley, CA.

Terry - I've thought of something you just might like:

As you probably know, the RAM that holds the contents of the screen starts at 8000 hex. Well, have you ever tried POKEing into that memory? I have.

Below is a program that will fill the screen with random characters in random positions.

```
10 SC=16↑3*8
20 ?" © "
30 X=INT(RND(5)*40)
40 Y=INT(RND(3)*25)
50 C=INT(RND(9)*256)
60 POKE SC+X+Y*40, C
70 GOTO 30
```

Line 60 will take a character corresponding to the number C and put it in position X,Y, where X is from 0 to 39 and Y is from 9 to 24.

Note that the number C is not always the same as the value you get for that character using the ASC function. The easiest way to get the value for a character is to put that character into the first position of the screen and type "?PEEK(16 3\*8)".

That brings me to another advantage: you can tell what's on the screen in a certain position by using the PEEK command.

A word of warning -- the values for a character go from 0 to 255.

Using POKE and PEEK does wonders for all of my graphic games so I thought I'd tell you about it. - Russell Selph

Thanks Russell. - Terry.

Rick - I received your sample copy of the PET Paper and, really, didn't think much of it (maybe your format??) until I started to play with some of the short software exercises. I read further! Enclosed is an order for your PAPER. - Gene Planchak

Gene - Hope this format change does good things for your interest. - Rick

Terry - There is a glitch in my copy of Useful Routines. There is a Sort Demo, but I cannot

load Sort. Please send me a listing of Sort. I had some difficulty loading this tape, but succeeded after several tries. Everything works except Sort.

Here are my suggestions for names for the paper:

1. "Pet Paper" Use upper case and lower case letters. Commodore uses all upper case.
2. "Vet Paper" Sounds like PET and we are all veterans or veterinarians of the PET (or will be).
3. "PUP PAPER" For Pet Users Paper (excuse the slight

redundancy), and, a PUP is a PET.

- Hews McCann

Hews - SORT ROUTINE

```
10 FOR I=1 TO N
20 FOR J=1 TO N-1
30 IF A$(J)>A4(J+1) GOTO 50
40 S$=A$(J) : A$(J)=A$(J+1)
:A4(J+1)=S$
50 NEXT J
60 NEXT I
```

N is the number of items to be sorted. S\$ holds the smaller value until the larger value is moved up. - Terry

## PET PROSE

There are many people who want specialized application programs to use on their PETs, but who are not knowledgeable enough to design or to write them. If you can write significant software in some specialized field (i.e., education, engineering, business, etc.) and are willing to do so, we may be able to help you find the people who are willing to pay you to write programs for them. Send us \$25.00 and we'll print your name, address, phone number, and field of expertise in the next 10 issues of The PET Paper.

Wes Fasnacht (215) 436-2886  
West Chester State College  
West Chester, PA 19380  
\*Programming & Systems  
Development

Don Pagnataro  
1420 Centre Ave. #1907  
Pittsburgh, PA 15219  
\*Education, Engineering,  
Business, Games, Photographic  
Applications

We report this incident without comment:

David Smith has 3 brand new 019 ROMs - 2 of them don't work. (L. Tramiel) says it's impossible - the ROM absolutely, positively works every time without fail - it passed all Commodore's ROM test programs - but the 019 ROM still won't work.

Gary Wagner drove from LA to Palo Alto to hand deliver his PET for repair to the cassette: the cassette would not recognize a fileheader. Commodore said they couldn't fix it while he waited - not even within the next day or so. He drove home, waited 3 weeks for his PET - it arrived. First thing he noticed was a new hairline crack on the

display housing. He turned it on, typed 10 REM and pressed Return; the cursor disappeared. Commodore had replaced his original ROM with 019.

Tried to load a program - but no-go - PET still wouldn't recognize a file header. He opened it up and found a loose ground wire - the screw was inside the cassette.

Called Commodore - they said to send it back. He said he wanted the original ROM instead of the 019 - Commodore said it would cost \$30.00.

So - now keyboard locks out - he can't load or save programs and Commodore won't admit the 019 ROM could possibly be faulty.

Please publish my name as a prospective purchaser/author (circle one) of customized PET Programs. I enclose full payment as described in the PET Prose Column.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

MC/VISA/BAC# \_\_\_\_\_ Exp. Date \_\_\_\_\_

MC Interbank No. \_\_\_\_\_

Required Credit Card Signature \_\_\_\_\_

Area(s) of Interest \_\_\_\_\_

MAIL TO:

# INTRO TO BASIC

In our Intro to Basic, we have covered the following topics:

LINE NUMBERS	KEYWORDS	LIST
EXPRESSIONS	OPERANDS	RUN
CONDITIONS	GOTO	TAB
VARIABLES	IF...GOTO	READ
DATA..RESTORE	PRINT	CONT
FOR...NEXT...STEP		

ARITHMETIC OPERATORS

Now let's look at INPUT, GOSUB, RETURN, and IF...THEN. Once we've finished this section, we'll have enough information to actually write a real working program. Look back at issues #1, 2, and 3 for Tricks to Help Train Your PET - you'll find many useful little techniques you're ready to use effectively. Read the article in this issue entitled, "How to Write a BASIC Program." Then write one. Let us know the results of your efforts. Does your program do what you expected it to do?

**INPUT** The INPUT keyword instructs PET to wait for you to enter data from the keyboard. Instead of storing information in a DATA statement, data can be entered directly during program execution:

```
10 INPUT A$
```

You can enter more than one item at a time, using a comma to separate them:

```
10 INPUT A$,B,C$
```

When you are responding to INPUT during a program run, you must enter as many separate items of information as the program expects. And each of the items must be separated by a comma. For example:

```
10 INPUT A$,B$
```

requires two items to be typed in during the program. User response might be SMITH, JOHN. SMITH will be stored in A\$, while JOHN is stored in B\$.

The INPUT statement has a rather helpful feature called a "prompt," which permits you to tell the user what he/she is being asked to do when PET is waiting for input from the keyboard:

```
10 INPUT "ENTER LAST NAME";A$
20 INPUT "ENTER NAME (LAST, FIRST)";A$,B$
```

Notice the semi-colon following the prompt and the comma separating the data items.

Warning: if the user presses RETURN without typing any data, PET jumps out of the program and execution stops. Issue #1 has an article on the GET keyword which will help alleviate this problem. (Also see corrections in #2.)

**GOSUB** Sometimes a group of instructions may be repeated several times during a program. It is useful to be able to enter the code once for these instructions once, jump to them when they're needed and go back to the original program flow when finished. This can be accomplished by using the GOSUB and RETURN keywords, which transfer control to and from a section of code called a subroutine.

```
10 INPUT A
20 GOSUB 100
30 PRINT B
40 GOSUB 100
50 PRINT B
60 END
100 B=A*A This is the
110 A=B subroutine.
120 RETURN
```

Note the "END" keyword in line 60. This is necessary when using subroutines to keep your program from "falling" into the routine. Without the END, PET

doesn't know enough to stay out of the subroutine, and you'll see a ?RETURN WITHOUT GOSUB ERROR message on the screen.

Note that the line number in the GOSUB statement is the first line of the subroutine and when PET encounters the RETURN keyword, it returns control to the statement immediately following the GOSUB. In our example, the code is executed like this: Suppose we input 5 for A

```
1. LINE 10 INPUT A
2. LINE 20 GOSUB 100
3. LINE 100 B=A*A
4. LINE 110 A=B
5. LINE 120 RETURN
6. LINE 30 PRINT B
7. LINE 40 GOSUB 100
8. LINE 100 B=A*A
9. LINE 110 A=B
10. LINE 120 RETURN
11. LINE 50 PRINT B
12. LINE 60 END
```

Try this program to get a 'feel' for subroutines.

**IF...THEN** We've already seen how the IF...GOTO keyword works, permitting control to branch to another statement depending on the truth of a specified condition.

```
10 IF A=5 GOTO 100
```

IF...THEN works much the same way, but it permits you to use any keyword except GOTO, following THEN:

```
10 IF A=5 THEN B=10
20 IF A=10 THEN PRINT A
30 IF A=15 THEN GOSUB 200
40 IF A=20 THEN INPUT"ENTER A NUMBER";B
```

Sometimes you may want to branch to a different section of code for each value of a specific variable, and an INDEXED GOTO would be helpful. PET utilizes the keyword ON...GOTO to accomplish this.

```
10 PRINT "DO YOU WANT TO 1. ADD"
20 PRINT " 2. SUBTRACT"
30 PRINT " 3. MULTIPLY"
40 PRINT " 4. DIVIDE"
50 PRINT " 5. QUIT"
60 INPUT A
70 ON A GOTO 100, 200, 300, 400, 80
80 END
100 A$="+ "
110 GOSUB 500
120 C=A+B
130 GOSUB 600
140 GOTO 10
200 A$="- "
210 GOSUB 500
220 C=A-B
230 GOSUB 600
240 GOTO 10
300 A$="*"
310 GOSUB 500
320 C=A*B
330 GOSUB 600
340 GOTO 10
400 A$="/"
410 GOSUB 500
420 C=INT(A/B)
430 GOSUB 600
440 GOTO 10
500 A=INT(10*RND(1)+1)
510 B=INT(10*RND(1)+1)
520 IF A>B THEN RETURN
530 A1=A:A=B:B=A1:GOTO 520
600 PRINT A;A$B;"="
610 INPUT D
620 IF C=D THEN PRINT "CORRECT":RETURN
630 PRINT "SORRY...TRY AGAIN.":GOTO 600
640 END
```

Can you follow the code and see what happen? We've introduced a new keyword, RND, in lines 500 and 510. This keyword uses PET's random number generator to randomly select a number in this program, the number will be between 1 and 10. (See the article in this issue on RND.)

This program is a simple arithmetic drill. You can make the problems more complex by increasing the numbers in lines 500 and 510 to 100 (instead of 10)

500 A=INT(100\*RND(1)+1) for instance.

The INT function assures that the resulting number will be an integer. The fractional parts of the result are dropped, not rounded, so the INTeger will always be the next smallest whole number.

```
INT(3.142) becomes 3
INT(3.942) becomes 3
INT(-3.142) becomes -4
INT(-3.942) becomes -4
```

There's one more topic we need to discuss at this point: the DIM keyword.

DIM allows you to create lists or tables of data, all the items in which have the same name.

These lists are usually referred to as "arrays." PET presently has a "bug" in the array handling part of its software, so you are restricted to a maximum of 255 elements (data items) in any one array. The theory is that you can use 3 or 4 dimensions in an array, to form a table, but PET doesn't always perform properly to permit it.

This is a one-dimensional array. It has only one list of 25 data items.

```
DIM A(25)
```

This is a two dimensional array. It has two lists of 5 elements.

```
DIM B(2,5)
```

The FIRST parameter inside the parentheses represents the number of ROWS. The SECOND parameter, separated from the first by a comma, represents the number of COLUMNS. So array B would look like this:

```
B(1,1) B(1,2) B(1,3) B(1,4) B(1,5)
B(2,1) B(2,2) B(2,3) B(2,4) B(2,5)
```

Note that the FIRST number inside the parenthesis is the same as the number of rows and the second number in parens is the same as the number of columns. The values in the columns change more frequently than the values in the rows. A good example of this is our numbering system itself. We have digits 0 through 9 to work with - we can have as many rows as we like, but only 10 columns

```
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
```

In this example, the rows represent the "tens" place and the columns represent the "ones" or "units" place. And, as you can see, the "units" change more frequently than the "tens." Let's try it.

```
10 DIM A(10,10)      ALLOCATE STORAGE
20 FOR I=0 TO 9      SET UP ROWS
30 FOR J=0 TO 9      SET UP COLUMNS
40 A(I,J)=I*J
50 NEXT:PRINT
60 REM NOW PRINT
70 FOR I=0 TO 9      SET UP ROWS
80 FOR J=0 TO 9      SET UP COLUMNS
90 PRINT A(I,J)      PRINT DATA
100 NEXT:PRINT
```

Some warnings:

Pet permits up to 255 elements in a ONE dimension array. You'll have to experiment with your unit to determine the number of elements permitted in a two dimension array - and the number permitted won't be consistent. I can get A(8,28) and A(115,2) - which doesn't help me program "portable" software (able to run on another PET), since other units may not permit the same number of elements. A close friend can

get A(8,26) and A(118,2) - so check your unit before doing any extensive coding to handle arrays.

PET allows 11 elements in a single dimension without requiring a DIM statement. DON'T TAKE ADVANTAGE OF THIS FEATURE! It will make your program difficult to read and/or modify later.

It's a good idea to put your DIM statements at the very beginning of your program. You may NOT dimension the same variable twice. If you try it, you'll get a ?REDIM'D ARRAY ERROR MESSAGE.

Arrays can be dimensioned dynamically during program execution:

```
10 INPUT "HOW MANY PEOPLE ARE HERE";A
20 DIM B$(A)
30 FOR I=1 TO A
40 PRINT "NAME OF GUEST#";I;:INPUT B$(I)
50 NEXT I
```

Well, that about covers the INTRO TO BASIC subjects. Beginning with the next issue, we'll discuss some of the less frequently used keywords, keeping in mind that you may still consider yourself a "beginner." If there are specific topics you'd like to talk about, let us know.

YES! I'd like to subscribe to the PAPER and receive all ten issues of this year's volume! I enclose \$15.00 in full payment.

Name \_\_\_\_\_  
(Please print or type)

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Cash, Check, Money Order Enclosed \_\_\_\_\_

MC/VISA/BAC Number \_\_\_\_\_ Exp. Date \_\_\_\_\_

MC Interbank No. \_\_\_\_\_

Required Credit Card Signature \_\_\_\_\_

You may let other PET owners in my area know I have a PET, so they can contact me.

I'd like to see articles in the PAPER about:

I am interested in forming/joining (circle one) a PET Users group

MAIL TO: PAPER, P.O. Box 43, Audubon, PA 19407

## STANDARD SYMBOLS

Ⓜ	Cursor HOME	Ⓞ	Reverse Field OFF
Ⓒ	CLEAR Screen	Ⓢ	Carriage Return
Ⓡ	Cursor RIGHT	Ⓢ	Shift On
Ⓛ	Cursor LEFT	Ⓢ	Delete
Ⓤ	Cursor UP	Ⓢ	Shift Off
Ⓣ	Cursor DOWN	Ⓢ	Space
ⓕ	Reverse FIELD on	Ⓢ	Stop

## TRICKS TO TRAIN YOUR PET

Readers have offered suggestions on how to avoid "falling through" an INPUT statement when the RETURN key is accidentally pressed.

First one: Open the keyboard as a logical file: Use only strings or you'll get a "REDO FROM START" when RETURN is pressed.

```
10 OPEN 1,0,0
20 INPUT #1,A$
```

You can't get out of it except with a shifted RUN/STOP key.

Second, from George Sherritt:

```
10 POKE 3,1
20 INPUT A$
30 PRINT A$
40 POKE 3,0
```

This eliminates the INPUT question mark and will not accept a null string for an answer. And you can't escape this without a shifted RUN/STOP, either.

\*\*\*FLASH\*\*\*

The PET has command completion.

Type L shift i instead of LIST or type R shift u instead of RUN. BASIC recognizes these abbreviated commands and executes the proper token. Also works for GOTO and other commands in a program where space is needed. (or speed of typing. space would be same)

\*\*\*\*\*

John Groot noted that his PET returns a string ("1") at the end-of-file. If he checks for that string, his PET never fails to find the EOF.

The series of articles "Introduction to BASIC," which continues in this issue, is intended to get you familiar with the syntax or grammar of a specific programming language - BASIC.

Now that you know how to speak correct BASIC - or at least know many useful "phrases" - what are you going to say?

Each PET purchaser probably has a slightly different reason for buying his or her PET. For some it was pure pleasure - computers like PET are fun! For others it was as a learning tool, others may intend to use it in their work, or even to start a business built around the PET!

So no matter how many programs other people write and offer for sale, sooner or later you are going to want to write a program precisely aimed at your original reason for purchasing your PET.

Your first few programs will probably be short calculation programs or simple number-oriented games. These you can program "on the fly" - figuring out what you are going to do and typing in the few lines of BASIC you need as you go along. This is fine for simple programs, but as you write longer programs this method will get you in trouble. You will find yourself constantly LISTing your program to find out what you named a variable earlier in the program which you now need. Worse, when you come back later to add a new feature to the program, you will have forgotten how the program "works" and will end up adding "bugs" or errors, rather than features.

So what do you do when you have a reasonable grasp of BASIC, and have written some small programs and are ready to take on your first "Big" program? Well since computers are very methodical beasts, you can understand that they get along best with methodical people. I am not suggesting that this makes programming less fun; indeed programming is more fun when you are not spending most of your time trying to track down bugs in a poorly-planned program.

Here is the formula I propose for preparing a big project (big for the PET is a program of more than 40 to 60 lines):

1. Define the task the program is to perform.
2. Draw a flow chart of the program operation.
3. Define names for the key variables you will use in your programs.
4. Write and test any special-purpose routines you plan to use.
5. Write your program in modules and test each module as it is completed.
6. When the program is completed, thoroughly test it; first with data you expect to use, then with "unexpected" data.
7. Add final "internal documentation" remarks to the program.
8. File your flow chart and other paperwork where you can find them later.
9. Congratulate yourself on a well done project. Show it off to somebody (now, not back when you were still getting rid of the errors!)
10. Look at programs written by others, read the PET Paper and other publications. If you see

a "neat" way to do something, file it away for your next programming effort.

Now for a few comments on some of the steps.

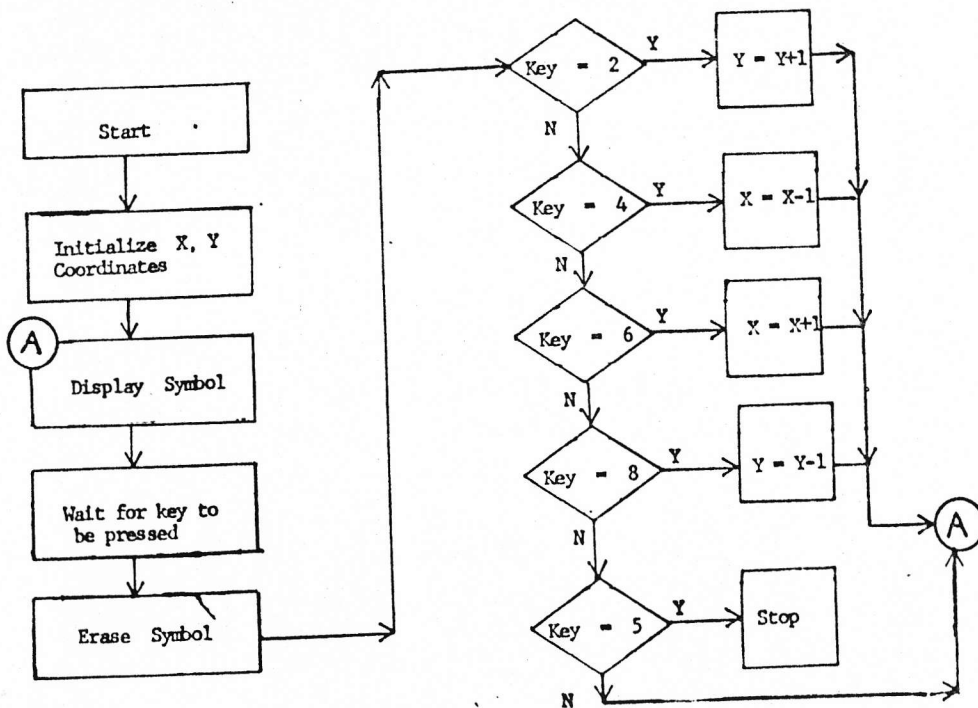
Defining the task - now is the time to think the program all the way through. What formulas do you need? What will the computer dialog with the user look like? What will the program do if the user types the wrong response? What graphics do you intend to use? Will the user understand what the program wants him to do? After this step you should know exactly what a program RUN will look like.

Flow charting - experienced programmers often refuse to flowchart their programs and make fun of those who do. This is called "job security", since

a program of any size without a flowchart is extremely difficult for anyone to modify (even the original programmer). These are the same people who scoff at putting comments or remarks into a program.

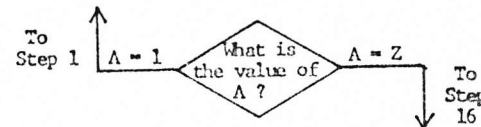
What is a flowchart? It's just a road map of the program you intend to write. People have dreamed up all sorts of special symbols to use in flowcharting different computer operations. I only use two symbols: a box to indicate some operation and a diamond to represent a decision by the computer.

Here is a flowchart for a program to put a symbol on the PET screen, then move it up if the user presses the "8" key, down if 2 is pressed, left for 4 and right for 6. All other keys are ignored:



Look closely at the flowchart. Will the symbol "blink" even if I depress the "0" key? What feature did I add that I didn't tell you about? Can you see where the GOTO statements are going to have to go to?

Notice that each diamond contains only two exit points: "Yes" and "No", since the PET can only make a yes or no decision. This also avoids the problem of a decision such as "IF A=Z GO TO STEP 1 AND IF A=1 GO TO STEP 16" where we haven't told the computer what to do if A isn't equal to 1 or Z.



**DON'T DO THIS!**

Define names for key variables - PET BASIC allows you to use longer names for variables than do most other versions of BASIC. This allows you to use meaningful names for your variables. So if you are writing a program to calculate loan interest, don't call the annual percentage rate variable "B," call it "RATE" or "APR." Write it down so you don't call it "RATE" in one part of your program and "APR" somewhere else!

Test special features first - it is easiest to find a "bug" if you know where to look. There is nothing more frustrating than entering an entire program which doesn't work - and you don't know where to start looking for the problem. Routines which do tape input and output, special graphic displays, and string manipulation routines are all

good examples of program segments which should be written and tested before they are used in a larger program. You usually add a few statements before these partial programs to feed them test data. If you have flowcharted your program you can assign blocks of line numbers for each step in the flowchart and use these pre-assigned line numbers when you write your special routines. Go back to your flowchart and add the line number for the start of the program segment performing the operation defined in each box or diamond.

Test the finished program with "unexpected" data - if other people are going to use your program, let them test it out using whatever instructions you are going to provide with the tape. You are going to provide instructions, aren't you?

If the user types in an incorrect response which causes your program to misbehave, what are the consequences? The user playing a game may not mind, but if your program incorrectly computes a payroll check as a result of user misunderstanding, you may be in trouble.

Some readers may object at this point, "Wait a minute - remarks take up memory and so do longer variable names and program statements to protect the user from incorrect entries. What happens if my program ends up being too large for the memory in my PET?"

My response is that your problem is not that your program is too large and must be cut down. Your problem is that your PET is too small and needs more memory to do the job you have set for it. Memory expansion costs money, but in many applications a bad program will cost far more.



# From The Commodore Himself

A problem with opening files to write on either built-in cassette #1 or external cassette #2 has been discovered. When a file is opened, garbage will be written out instead of a proper data tape file header. Without this header, it is impossible to open the data file for input.

You may not have encountered this problem previously, because it is disguised by having loaded a program on the cassette prior to writing a data file. In this event, the start address of the buffer with the header information is initialized properly, but cassette data file operation could still be random.

Fortunately, there is a software patch you can implement in your BASIC program to force the open for write on tape to work every time. Before opening to write on cassette #1:

```
POKE 243,122
POKE 244,2
```

And, on cassette #2:

```
POKE 243,58
POKE 244,3
```

Locations 243 and 244 contain the lo and hi order bytes, respectively, of the address of the currently active cassette buffer. The start address of buffer #1 is \$27A (\$2=2, \$7A=122). Similarly, cassette #2 is \$33A (\$3=3, \$3A=58).

These POKE statements may remain in your BASIC program even after

you start using the new ROM software which corrects the problem.

RND(X) generates a random number between 0 and 1. The argument X controls the generation of random numbers as follows:

X<0 generates a new sequence of random numbers using X as a seed. Calling RND with the same X, when X<0, will generate the same random number for each X if X does not change.

Example: RND(-1) gives  
2.9919647E-08  
for as many times as X=-1.

RND(-2) gives  
2.99205567E-08  
for as many times as X=-2.

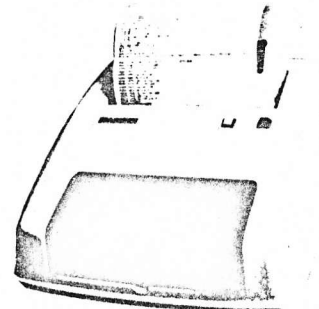
This is useful for debugging when you want the same random number generated. You can get a different but constant random number with any negative number:

X=0 generates .564705882 each time you call.

X>0 will generate the next randomly sequenced random number, if X doesn't change. If X changes, the new X is used as a seed to a new sequence of random numbers.

If you want to verify what the RND function actually does, enter the program:

```
10 INPUT R
20 X=RND(R)
30 PRINT X
40 GOTO 10
```



## Introducing the PET 2020 Printer for use with the PET 2001 Computer

The new PET 2020 Printer is a fast, intelligent peripheral designed to attach to the PET 2001 personal computer, or to any IEEE 488 interface. It will print 80 columns wide on any standard 8 1/2 inch width roll or fanfold paper, with a ribbon, or without a ribbon, on impact sensitive paper. It can handle all upper and lower case characters and all the PET 2001 graphics on a 7 by 8 dot matrix.

A programmable character capability allows the user to design a special character or symbol (a company logo, for instance). Through its character enhancement feature, any character may be doubled in width. A reverse field feature allows the effect of white on black printing.

Programmable Character	Enhanced Character	Reverse Field
Q	Commodore	Commodore
Q	Commodore	Commodore
Q	Commodore	Commodore
Q	Commodore	Commodore

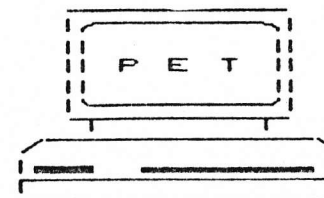
The 2020 prints through three copies at 120 cps. The net speed is 50 characters per second. And that includes the time between lines.

The PET printer can, of course, list any PET programs. It also has full formatting capability allowing

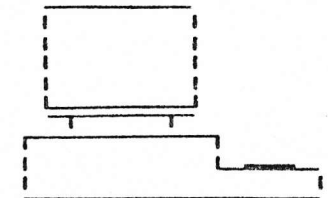
Suppressed leading zeros	9.45
Forced leading zeros	0009.45
Floating dollar sign	\$595.00
Fixed dollar sign or Alphanumeric	\$ 595.00

An example format might be  
AAAAAAA 999 \$\$\$\$.99

which would format data like this  
APPLES 25 \$4.45



PORTRAIT  
OF 2001  
BY 2020



The PET 2020 begins listening to data on the IEEE bus when it recognizes its primary address which is factory assigned as device #4.

(The primary address can be modified by jumper for 4-11 so multiple printers may be connected on the same interface.)

Software standard in the PET 2001 allows transmission of data to a particular primary device number, as well as to secondary addresses, which allow extended printer commands.

There are six secondary addresses which may be used on the PET 2020:

- #0 Print data exactly as received
- #1 Print data to format
- #2 Use data as format
- #3 Set number of lines per page
- #4 Enable verbose diagnostic messages
- #5 Use byte data for programmable character.

The printer has an MOS Technology 6504 micro-processor, 32 kilobytes of ROM and 256 bytes of RAM. Through its own diagnostics, executed every time the printer is powered-on, it will check out the RAMs and the ROMs, and turn on the ready light if all is in order.

The PET 2020 printer operates on 120 volts, 60 Hz, 120 watts. A 220 volt connector is available as a power option. Line frequency can be modified at the factory to 50 hz for operation in those counties requiring such frequency.

Dimensions:

7 1/2" high, 15" wide, 17" deep - 20 lbs.

 **commodore**

901 California Avenue • Palo Alto, California • (415) 326-4000 Telex 345 569 Cable Address COM BUS MAC PLA

# FLEA MARKET

## SOFTWARE EXCHANGE

THE FOLLOWING PROGRAMS ARE FOR SALE FOR \$5.00 EACH IF PURCHASED WITHOUT AN EXCHANGE PROGRAM. IF YOU HAVE A PROGRAM OF YOUR OWN TO ADD TO OUR FLEA MARKET, SEND IT TO US AND WE'LL SEND YOU THE PROGRAM YOU WANT. (PLEASE INCLUDE \$2.00 FOR POSTAGE AND HANDLING.) IF YOU DON'T SEE SOMETHING YOU WANT ON THIS LIST, WE'LL SEND YOU A COUPON WHICH YOU CAN RETURN (ALONG WITH THE \$2.00) AT A LATER TIME.

SNAKE - Game for small children. Good graphics - almost no interaction.  
 YAHTZEE - EXCELLENT GRAPHICS - Very well executed - one of the best we've seen.  
 SPADES - Great Graphics. Well done version of the familiar card game.  
 RND CHAR - POKES random characters on the screen. Useful for learning coding tricks.  
 SPIRAL - Pokes CHARACTERS on the screen in a spiral pattern. Useful for learning coding tricks.  
 TYPEWRITER -5- Mini word processor.  
 PRIME NUMBERS - Generates and saves prime numbers.

DIGITAL CLOCK - Hours and minutes. Good demo.  
 MORSE CODE TUTOR - Wire your PET to 'bark' (see issue #1) and use this program to learn Morse Code. Or learn it by sight only.  
 VACATION BUDGET - Budget in advance or fill out an expense report. Non-graphic.

See Issue #3 for descriptions.

ADDITION GAME  
 OTHELLO FOR ONE  
 CODES  
 ROAD RALLY  
 SLOT MACHINE  
 STATES & CAPITOLS  
 EMPEROR  
 GRADES  
 HEX-DEC  
 MATH QUIZ  
 FLEA-PAK I  
 USEFUL ROUTINES  
 PUBDOMDIS  
 NOTE  
 KLIDISKOP  
 KYBD UTILITY  
 RENUMBER & LIST  
 HIMONDIS & HIMON INTRO  
 LOMONDIS  
 AIR WAR  
 DOGFIGHT

### SOFTWARE GUARANTEE

If the tape you receive from us does not load on your PET, send it back and we'll record it on the older model PET. We do not guarantee that FLEA MARKET programs will do anything more than load, since they are submitted by readers and we do not check them out except to be sure they will load and are free from obvious errors.

# SOFTWARE SHELF

## SOFTWARE SALES

CHASE-\$10.00  
 BLOCKADE-\$10.00  
 SPACE TALK/SPACE FIGHT-\$10.00  
 BLACKJACK-\$10.00

The following seven programs constitute a set of routines which will carry out most of the sorts of statistical tests most people require. All the programs are conversational and interactive. The package will be mass duplicated and sells for \$20.00 complete. Order STAT II.

1. DESCSTATS -- takes a set of data, either as single values or as grouped data, and calculates descriptive statistics -- mean, variance, standard deviation, maximum and minimum values.

2. PROBF -- this program accepts test statistics produced from statistical tests -- the variance ratio, (F), Student's T, standardized normal deviate of chi square -- and calculates the probability of such a value. It replaces looking up the values in tables, and is also used as a subroutine in all the programs below.

3. 1WAYANOVA -- this program performs analysis of variance on two or more sets of data, each set being the responses of a group of cases to a treatment, for example, two drugs and a control (three treatments). The number of cases for each treatment may be the same or

DEFLECTION-\$10.00  
 HOME DATA RETRIEVAL-\$10.00  
 STAT-\$20.00  
 ADDRESS FILE \$10.00

different. In the special case of two treatments, it is identical with the well-known T-test.

4. 2WAYANOVA -- similar to program 3, but each case is subjected to two different treatments, A and B, simultaneously, e.g., treatment A might be different drugs (or different dosages of the same drug) and treatment B different diets; alternatively, the treatment B can be blocks of a block design.

5. ANOVA2WAYR -- as in program 4, but there are two or more cases (replicates) for each treatment combination. All the analyses of variance programs included tests of significance.

6. SLR -- this program calculates linear regression, correlation coefficients and coefficient of determination, and tests the significance of all statistics. This program is designed for grouped data, i.e., several values of Y (the dependent variable) for each value of X (independent variable).

7. XYSLR -- similar to SLR, but the X and Y values are in pairs.

# User Group Notes

\*\* North Orange County  
Computer Club News \*\*

The Pet User's Group held its first meeting last week at Computer Components, formerly Computer Playground, on Westminster Blvd. and Golden West. We had about 25 people show up for a lively give and take session with two representatives from Commodore, who were here for the N.C.C. and agreed to meet with us. Lennie Cooper represented service Dept. and she is one of several girls you're likely to contact if you call Customer Service (415) 327-4030. Leonard Tramiel, son of the president of Commodore represented Software Q.C. and generally filled us in on the state of things at Commodore. Generally, Leonard gave us the same excuses as we've been hearing all along: not enough people, and not enough time to get the work done. It seems the small technical staff that really know the PET are the same people who visit all the computer shows, answer technical phone calls, write the technical manuals yet to be released, and design the next peripheral. They won't hire any more engineers because they say that would drive the costs up, so everything is delayed. For example, Chuck Peddle himself is writing the more technical parts of the user's manual and you can imagine how much time he has for that! Anyway, here's what we learned.

The PET manual and monitor tape have "about one week of work left", but since everyone working on them will be booked up with trade shows, they won't have time to finish it until the

Reprinted with permission from  
David E. Smith

end of June. Then it will take a month and a half at the printers, and two weeks to mail. Translation: look for the manual and monitor tape by the end of August.

The printer that was supposed to be ready this month for delivery had this slight problem with its plexiglas cover for the paper roll: it didn't fit. So the printer will not be delivered until September if you send in your money by the end of June. The printer uses the 6504 processor, and has an operating system in ROM and a RAM buffer. It will accept one of six commands from the PET via the IEE 488 bus including formatting of output, define programmable character, and # of lines per page eject among others. It prints the entire graphics set, plus any character you wish to define yourself, has reverse field printing and continuous graphics in the horizontal direction only. Vertical has two dot spacing since it's 7X8 dot matrix. See printer copy example below. It's a terrific machine at \$595 but it's back ordered already.

The new basic ROM set is still being tinkered with on Leonard's desk, looking for more bugs. Because of lack of time again, the new ROMs (which will include the monitor in them) won't be ready until November. There is a single ROM fix for the lock out problem available for \$10.00. The ROM designated the 019 ROM and fixes the problem of Microsoft having written this "very small bit of non-interruptable code" for a

system that lives on interrupts from everything. Hence there is a small window where the PET goes out to lunch and you have to turn it off to recover.

On servicing, Computer Components now has the necessary diagnostic tools and tape to trouble shoot the PET. A "serviceman's package" will be available for \$30.00 that includes a schematic of the video board only and some scope tracings for that board, the connectors needed to implement the ROM diagnostics, and a PET TEST program tape. However, we now have all that stuff in the club as you'll see below.

The extra memory expansion box is "simple to whip together" but they had mechanical problems and not enough engineering time so it won't be ready until the end of the year.

The floppy disk is completed hardware wise, but the software isn't. By the way, Li Chen Wang, who wrote tiny BASIC for Dr. Dobb's Journal last year (which was stolen by Radio Shack for their level 1 BASIC), is now working for Commodore, and is writing the disk operating system. The floppy will have two 6504 microprocessors; one for formatting, and the other for communications with the PET. The controller will be a dual drive controller so you can add a second floppy later. The floppy will be a mini with 150K bytes per side. Since it's on the IEE 488 bus, there is almost no limit to the number of floppies you could add to your PET. The floppy should be available in October or November and cost "\$400 to \$1,000" so says Lennie.

The NCC featured several new products of interest to PET owners. HUH Electronics showed their S-100 interface for the

PET. It provides all the bus signals and will run with the Dazzler and dynamic memories very well. It uses a phase locked loop to synch up with the PET's clock. PSYNC and MWRITE is provided. It looks like a great product.

The real block buster at the show was a product that really wasn't pushed very well. Houston Instrument, the company that makes industrial plotters of all kinds and abilities, has just announced a digital flat bed plotter for personal computing for \$1,000. It plots on 8 1/2 by 11 paper and looks just like a \$4,500 H.P. plotter, except it doesn't change pens for you, but it does everything else you need. It looks like the first of real quality commercial products from the traditional computing industry tailored to this market. They already have a huge backorder as a result of the show. I'm getting one today! By the way, it has a serial and parallel interface so it will hook right up to the PET. See me for more info on it.

## HOW TO RUN THE ROM DIAGNOSTIC IN YOUR PET

Attach a cinch connector (available at computer components) to the 8 bit parallel port. The connector should be wired as shown with pin N to pin 5, pin B to 2, pin C to 3, pin D to 4, pin 8 to pin 7, and pin 7 to pin 6, pin K to pin 9, and pin L to pin 10. The keys should be between pins B and 2, and between pins M and 11. This assures proper connection.

In addition, you will need to jumper the keyboard pins on the main circuit board using small daisy hooks. We still

haven't found that 20 pin molex keyboard connector anyplace yet. On the Keyboard male connector, jumper pin 1 to 17 and to 9; pin 2 to 18 and to 10; pin 3 to 11; pin 4 to 12; pin 5 to 13; pin 6 to 14; pin 7 to 15; pin 8 to 16. Remove the keyboard connector before you do this.

Now with those two connectors in place, turn the PET off and then on again, and the ROM diagnostic test will be enabled. The PET checks to see if the connectors are in place on power up and if so, jumps to the diagnostic test. A cursor will move across the screen until it runs off the lower right corner. Then if the board checks out, the red led on the main circuit board will light up. This checks all the logic circuitry on the main board. A pattern will be put up on the screen to visually check the video circuitry. Other tests are available on tape that include a ROM test that checks the checksum for the ROMS and a memory test for RAM.

#### BASIC BUG FOR OPENING CASSETTE FILES

There is a bug that writes garbage on the tape for the file header when a file is opened. But it only occurs if you haven't loaded a program for tape previously. Because of this, it's impossible to read back data files. The fix is to POKE 243,122 and POKE 244,2 before doing OPEN statements to write to cassette number 1. For cassette number 2 POKE 243,58 and POKE 244,3. This restores the lo and hi order bytes of the address of the currently active cassette buffer. The addresses are Cassette #1 buffer: 27 AH and #2: 33AH.

#### HOW TO HOOK UP A JOYSTICK TO THE PET

(See Gary Wagner (714) 870-6709 for more on joysticks.)

Buy a joystick with four 100K pots on it. Hook each wiper of each pot to one of four data lines on the 8 bit user port. Connect the sides to ground. POKE 59459,255. This sets all the data bits to output mode. Then POKE 59471,255 to set all the bits high in the data register (ORA). Now POKE 59459,0 to set the bits to act as inputs, and PEEK(59471). When you move the joystick, you will ground the pins 0 to 4 depending on where the joystick is. When you input those bits you can tell which were grounded, and hence where the joystick is.

#### HOW TO PLAY MUSIC ON THE PET

(See Kim Clark (714) 774-1105 for more info on music and the PET)

Wire a space cinch connector as follows: connect pin M (CB2) to a 15 ohm followed by a 47K ohm resistor. Conenct the other end of the 47K to ground. Pick up the audio on each side of the 15 ohm resistor and plug into a small amplifier-speaker such as the Radio Shack \$3 job. Put the above three poke codes in a small program and vary C with a random number and you'll have instant music. Be sure to re-poke 59467,0 or else your tape drive won't work.

ACGNJ (Amateur Computer Group of New Jersey) PET Subgroup meets again on the 4th Friday at UCTI (7:00 to 10:30 PM).

SPHINX (Society of PET Handlers and INformation eXchange) meets the 2nd Thursday at 1025 2nd St. in Oakland, at 7:00. Contact Pete Rowe, Lawrence Hall of Science, Berkely, California.

Wisconsin PET Users meet every 1st Thursday. Call (608) 249-2666 for information.

PUG (PET Users Group, South San Francisco Bay area) meets the 1st Wednesday in the Mercury Room, 310 Showers Drive, Mountain View, CA, at 7:00 PM.

North Orange County Computer Club, PET sub group, has been formed by David Smith, 3030 Topaz, Apt. A, Fullerton, CA 92631.

Contact John Fung (612 - 376 - 5465) if you're in the Twin Cities, MN area.

PET Owners in Dallas, TX should contact Carl Martin, 2001 Bryan Tower Suite 3800 Dallas, TX 75201 for User Group info.

## SOFTWARE

Please ship the following taped programs. I enclose full payment as described in Flea Market or Software Shelf columns.

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

MC/VISA/BAC# \_\_\_\_\_ Exp. Date \_\_\_\_\_

MC InterbankNo. \_\_\_\_\_

Required Credit Card Signature \_\_\_\_\_

MAIL TO: