(C) Copyright 1980

1

01/80

POCKET COMPUTER NEWSLETTER

POCKET COMPUTER MEMO PAD PROGRAM

"Practical applications," everyone cries, "Show us some practical applications."

STOP RIGHT THERE!

Here's a program for the Radio Shack TRS-80 Pocket Computer that opens up a whole slew of practical applications.

LIKE WHAT?

One way a PC (pocket computer) really helps me out on a daily basis is by serving as an electronic memo pad. I used to keep 'umpteen scraps of paper containing notes, reminders and messages to myself around the house and at my office. Besides being unwieldy, the system had one other major fault. Me. I was always misplacing the most important scraps of paper or discarding them at the wrong times.

Enough of that. I don't carry the scraps of paper around any more. I carry a pocket computer. I paid enough for it so that I am not likely to accidently throw it away. I use it to store my important notes electronically. Appointment dates and times, sales contacts, phone numbers, shopping lists, data obtained in the lab while on the job, all these various kinds of information can now be stored in my TRS-80 PC.

The information stored is retrieved at my command. If I want to recall someone's phone number, I enter that person's name and the PC plucks it out for me. If I need to pick up some groceries on the way home from work, my PC reminds me just what it was I was supposed to get. If I need to temporarily record some data from an instrument in the laboratory, it goes into the PC for later recall when it is time to write a report.

When information is no longer needed, it is removed from the pocket computer's memory.

Believe me, I now save a lot of paper.

HOW IS IT DONE?

A relatively straightforward program enables the TRS-80 Pocket Computer to perform these useful chores. The program enables the PC to perform three basic functions.

1.) It allows the entry of data into the computer in the format of a *key word* and a related *item*. For instance, I can assign a person's name as a key word. The person's phone number might then be the related item.

2.) The program enables you to edit information in its memory. You can add to, change or delete the data whenever desired.

3.) Finally, you may have the computer locate an item related to a key word. Type in a key word, such as a person's name, and the program will locate that person's phone number for you

ALL ABOUT THE PROGRAM

The program that performs the functions is shown in Listing 1.

Some discussion of this coding would probably be beneficial for those readers who are not familiar with some of the traits of the TRS-80 Pocket Computer. On a line-by-line basis, here is what goes on:

Line 600 provides a means for the user to clear out the contents of the entire memo pad if desired (by calling on the subroutine at line 750). Note that a check is not made for any response other than a 'Y' for yes. This is all that is required because the TRS-80 PC simply skips to the next *numbered line* in a program if a null input is received. Remember it goes to the next *line* on this condition, not merely the next statement in a line! The coding used in line 600 provides a "failsafe" format. You must enter a 'Y' in order clear out the contents of the memo pad. Any other input protects the current data.

Line 610 gives the user the option of either editing an entry or using the 'find' (search) mode to locate an item.

Line 620 along with 630 directs the program appropriately depending on whether the user selected the 'edit' or 'find' option. Since the TRS-80 PC does not have the 'ELSE' function in IF...THEN statements, the contents of these two lines cannot be combined.

Line 640 enables the user to specify which entry stored in memory is to be edited. The program as listed is designed to allow up to twenty key words and related items to be stored at a time. Thus, line 640 expects the number entered at this point to be within the range 1 to 20. If it is not in this range, line 650 causes the query to be repeated. When the number is in the allowed range, the balance of line 640 contains directives to set up pointers to the indicated storage locations (array elements) in memory.

Line 660 displays the number of an entry, the key word and the associated item. The TRS-80 PC has a restriction on the specification of the second and subsequent variables in a PRINT or PAUSE statement. The

variable symbols may not contain mathematical operators. That is why the variables named B and C are established in line 640. They can then be used here in line 660 to define the string variables A$(B) and A$(C).
**Note: If you have encountered problems when using the PRINT and PAUSE statements to display the values of variables augmented by operators, be sure to read the note at the top of page 57 in the TRS-80 Pocket Computer operating manual. (Radio Shack manual number 26-3501.)

The remainder of line 660 asks for a new key word. Since the TRS-80 PC will skip to the next numbered line in a program if a null input (pressing the ENTER key without having entered any other characters) is received, the current key word will remain unaltered unless the user types in a new key word. This is an appropriate response for this application.

Line 670 asks for a new data item. The current item is left unchanged if a new item is not entered.

Line 680 directs the program back to the edit or find option upon the conclusion of an editing operation.

Line 690 initiates the portion of the program used to search the contents of the memo pad for a specified key word. It asks the user to input the key word of interest.

Line 700 directs the program to search through all the positions in the memo pad storage area looking for the key word indicated by the user. The PAUSE statement is used to provide the user with a glimpse at each entry during the search process. This method was selected so that a user could review the memo pad's entire contents if desired. Of course, it is not necessary for a user to monitor the display. An IF statement in the latter part of the line tests for a match with the desired key word. If found, the program branches to notify the user.
**It should be noted that the PAUSE statement does considerably slow down the speed of the search operation. It adds approximately 0.85 seconds for each entry scanned. If you do not want to be able to visually scan the contents of memory, simply remove the PAUSE statement and associated display parameters in the line.

Line 710 displays the message: "NOT FOUND..." followed, at the end of the display line, by "C/R" which is used to remind the user to tap the ENTER key when ready to proceed.

Line 720 directs the program back to the option-selecting statement. A separate line is assigned to this GOTO statement. Remember, the TRS-80 Pocket Computer skips over all the remaining portions of a line if it encounters a null input. If the GOTO statement had been included in the previous line, the null entry, which is specifically asked for, would cause the program to skip right over the GOTO directive! Got it?

Line 730 causes the PC to give an audible beep if the specified key word is found. The entry number, key word and associated data item is then displayed. Again, the terminator C/R is used to remind the user to press the ENTER key in order to proceed.

Line 740 directs the program back to the option-selecting statement.

Line 750 consists of a subroutine used by line 600 to clear out everything in the memo pad if the user selects that option. The subroutine simply puts a single 'space' character in each array element reserved by the program.

USING THE PROGRAM

To get things started type RUN 600 or use the abbreviated form R.600. (Of course, you can also use the DEF mode to set up the PC so that you can invoke operation of the program using a single key.)

When the program is started from line 600 it will always ask:

```
CLEAR ALL (Y/N)?
```

Enter the letter 'Y' for yes only if you want all of the elements in the storage array (the memo pad) initialized to the blank condition.
**Note: The option to clear out all the memory elements should always be selected the first time the program is utilized. Failure to do so will result in an error condition when the program is used. This is because the TRS-80 Pocket Computer requires that an array element be assigned before it is accessed. The subroutine at line 750 that assigns a space to each element in the array accomplishes this objective.

Now, if you answer the 'clear all?' query with an 'N' or any input other than a 'Y,' the program will go on to ask:

```
EDIT OR FIND (E/F)?
```

Enter an 'E' and press the ENTER key if you want to edit the information currently stored in your electronic memo pad. Remember that editing in this context means adding, changing or deleting items in the data bank.

If you want to search for a particular item using a key word, then you enter the letter 'F' and press the ENTER key.

YOU CAN EDIT THE KEY AND THE ITEM

If you tell the program you want to edit an item, it first asks:

```
KEY #?
```

Tell it the number of the entry in the memo pad that you want to edit (in the range 1 to 20).

The program will then display the information stored under that number. The format of the display in this situation is: entry number, key word, data item, and then the prompt to press the ENTER key, 'C/R.' The procedure of displaying the information before any editing occurs allows you to verify that you have called up the correct entry.

When you have verified the entry, press the ENTER key. Now the unit will ask:

```
NEW KEY?
```

If you want to change the key word, simply type in the new one. If you want to leave the present word alone, just press the ENTER key. The program continues by asking:

NEW DATA?

Again, if you want to alter the data, then type in the new information. If not, just press the ENTER key.

That is all there is to editing information stored in the memo pad. Of course, you use this mode to enter original information since the memo pad is initialized so that all array elements contain a space character.

HOW TO PERFORM A SEARCH

To look for a particular data item under a key word, simply respond to the edit/find query with the letter 'F.' Now the TRS-80 Pocket Computer will display the question:

KEY WORD?

Enter the key word of interest. The program will respond by stepping through each entry and briefly displaying the contents as it goes along using the format: entry number, key word and data item. You don't have to watch the display during this time if you have other things to do. However, the ability to review the contents of memory is often beneficial.

If the key word is found, the pocket computer will issue an audible beep to let you know it has completed its mission. It will then display the information. When you have made note of the data you can press the ENTER key to direct the program back to the edit/find option point.

If the key word you specified is not located in the memo pad, then the message NOT FOUND is issued. Press the ENTER key gets you back to the edit/find option for further operations.

SOME APPLICATIONS

Applications! That's what it is all about, isn't it? Here are some of the practical uses I have been making of the memo pad program.

First of all, I use it as a personal appointment calendar. For instance, I can assign a day of a week as a key word. Then I place the time of day and the initials of the person I am to see as the data item for that entry. Thus, an appointment entry might appear as:

1. FRIDAY=11:00JB

or

2. SATRDAY=1P-WALT

Since you are limited to seven characters for the key word and seven for the data item, it is often necessary to abbreviate. But, it is easy to see that the system is quite adequate as a memory tickler.

Of course, the key word can be anything you want. If you have a lot of appointments in a day, you can change the key word so that it represents hours/minutes during the day. Entries in such a situation might appear as:

1. 10:00=BILL R.
2. 11:15=BRUCE
3. 12:30=LUN/SAL
4. 14:00=AIRPORT

and so forth. Can't remember what you had scheduled for 11:15? Just use the FIND mode, enter the key word as the time, 11:15, and let the TRS-80 Pocket Computer refresh your memory.

```
600 INPUT "CLEAR ALL (Y/N)? ";A$:IF A$="Y" GOSUB 750
610 A$=" ":INPUT "EDIT OR FIND (E/F)? ";A$:IF A$="E" THEN 640
620 IF A$="F" THEN 690
630 GOTO 610
640 INPUT "KEY #? ";A:IF (A>=1)*(A<=20) LET B=26+A:C=46+A:GOTO 660
650 GOTO 640
660 PRINT A;" ";A$(B);"=";A$(C);"  C/R":INPUT "NEW KEY? ";A$:A$(B)=A$
670 INPUT "NEW DATA? ";A$:A$(C)=A$
680 GOTO 610
690 INPUT "KEY WORD? ";A$
700 FOR D=1 TO 20:B=26+D:C=46+D:PAUSE D;" ";A$(B);"=";A$(C):IF A$
    (B)=A$ THEN 730
710 NEXT D:PRINT "NOT FOUND...        C/R"
720 GOTO 610
730 BEEP 1:PRINT D;" ";A$(B);"=";A$(C);"  C/R"
740 GOTO 610
750 FOR A=1 TO 20:A$(26+A)=" ":A$(46+A)=" ":NEXT A:RETURN
```

IT HAS YOUR NUMBER TOO

Or, say you are a salesman. In the morning you make up your mind to make telephone calls to 20 clients during the day. OK, you load up your memo pad by entering the names of your clients and their telephone numbers:

1. DECKERT=8771501
2. PECK=4551950

and so on. Now you work your way down the memo pad list. Each time you make your sales contact, you edit the call out of the machine. When your memo pad is back to all blanks, you have accomplished your sales objectives for the day. The memo pad becomes your personal little task master!

TAKE IT SHOPPING

Make up your shopping list using just the key word portion of each entry for the names of the goods you plan to buy. Then, when you are purchasing items you can enter the actual price of each product as the data portion of each entry. Later, at your leisure, analyze whether or not you got good buys. Keep the list for the next time you go shopping and see how prices change. Do comparison shopping. Make your computer earn its keep!

RECORD TECHNICAL DATA

Use the memo pad to store data you gather on your job. Group the data by key word category if you like. The entry number can be used to automatically order sequential readings.

IT IS YOUR SERVANT

Use the memo pad program to make your life easier and more enjoyable. Think about it. Where can *you* use the pocket computer memo pad to help *you?*

****

FROM THE POUCH

An informal poll of about 50 Radio Shack Computer Centers recently resulted in roughly 10 percent of the stores reporting they were out of stock on the TRS-80 Pocket Computer. Virtually every store indicated sales of the little machine were brisk, with most indicating they were selling them as fast as they get them in.

One story heard was that R.S. only got about 5000 units initially — to test the waters so to speak, and had to start pressing for more as soon as they saw the instantaneous warm reception. No doubt they have sold a lot more than 5000 already!

If you are thinking of picking up a TRS-80 Pocket Computer unit and your local store doesn't have one in stock, hurry up and try another one in your area. Not everyone is aware of the potential of these little beauties yet. A few of the regular Radio Shack stores we contacted (versus the R.S. Computer Centers) had yet to sell a single unit. Invariably these stores commented that they were in localities where the general public was "not much into computers yet." However, this was heard more than once in cities where the R.S. Computer Center had reported they were "sold out." So — an outlying store in your area might be stocked while a central shop was begging for more! You can bet R.S. will be working like crazy to keep the supplies up, but they may have a rough time meeting the demand for awhile. There were undoubtably some disappointments during the holiday season, by buyers and sellers alike.

On the matter of program packages for the TRS-80 Pocket Computer, not many stores were even able to demonstrate a package, let alone sell one. We did, however, locate a R.S. Computer Center that was demonstrating a math drill program and a business financial package. Both packages looked very good from what we saw. Word is that Radio Shack is currently releasing about 10 packages with perhaps twice that many currently under development. More information on these packaged programs, including select reviews, will be presented in later issues.

From the rumor mill comes word that R.S. Corporate has some people working on an interface that will let the TRS-80 PC communicate with a regular sized Radio Shack TRS-80. (Take notes in the field, then dump the data for massaging into the big beast back home or at the shop!) There is also scuttlebutt running around that a printer unit is in the works. Sounds exciting!

Meanwhile, Panasonic has announced a HHC — Hand Held Computer with a whole host of accessories. But from what we can tell so far, it is mostly talk. Haven't come across a sole yet with a real live unit.

More next issue....

(C) Copyright 1981

2

02/81

POCKET COMPUTER NEWSLETTER

P.O. Box 232, Seymour, CT 06483

PRODUCT REVIEW
BUSINESS FINANCE PROGRAM PACKAGE

Produced by: Radio Shack for the TRS-80 Pocket Computer.
List price: $19.95
Availability: Many Radio Shack Computer Centers and some Radio Shack stores.

Overview

You get seven programs spread over two cassettes in this package. The programs are repeated on each side of the tapes. This is probably a good policy, judging from our experiences. A number of times we have had difficulty loading a program from one side of a tape, but have then been able to access the listing using the alternate recording.

You might also want to check your package while you are still at the store. The very first package we looked at had two identically labeled tapes inside. It should have contained two different tapes. Maybe we ran into a one-in-a-million case of mispackaging?

It has been our experience that loading of the tapes can be a little touchy in regards to the setting of the volume control. This has been observed whether using the recorder (Minisette 9) that Radio Shack recommends or a Minisette VI or a CTR-34, though the CTR-34 seems to do a much better job than the Minisette VI we tried.

You might as well know that the tape loading process is relatively slow. This is particularly so if you are used to working with disk systems! Taking into consideration the time it takes to set the computer into the cassette interface, locating the desired program on a tape, performing the load operation, etc., you can count on it taking five minutes or so to change programs in your machine. Now that is really not bad for such a tiny system, but it is easy to become impatient when you are accustomed to a much speedier loading process.

Accompanying the cassettes in this package is a slim manual that briefly explains how to run each program in the package. An example is provided for essentially each option provided in a program. Listings of the programs are not provided in the manual, but naturally, you can view these easily enough with the computer in the PRO mode if you are curious.

A brief synopsis of each program in the package follows.

Dates

The first program provided in the package provides calendar functions using the Gregorian calendar. You can determine the day of the week and the day of the year, as well as the number of days between dates, for any period between October 15, 1582 and February 28, 4000 A.D. (You should live so long, eh?)

This program, as well as others in the package, is designed to run with the computer in the DEF mode. This enables the various functions provided by the program to be accessed with just two keystrokes: the shift key and a selected reservable key.

The program is initialized by putting in a reference date by month, day and year. About 11 seconds later the program comes back to tell you the day of the week that this day occurs on. You can then press the ENTER key to obtain the day of the year.

Once the program has been initialized, you can type SHFT/S to determine the day of the week, day of the year and number of days until (before) a second date. It takes about 11 seconds for the first answer to appear. (You do not have to watch the display in anticipation. The program issues a couple of audible beeps when it has calculated information ready for you.)

You can give the program the number of days before an event occurs and it will tell you the date, day of the week and day of the year for that event. It takes about 20 seconds for the computer to come up with an answer for this type of problem.

You have 89 program steps left in memory when this program is installed.

Loans

You can use this program to find the true interest rate on a loan in about 19 seconds — after giving the original loan amount and payment information.

It takes just 8 seconds to tell you what your periodic payments will be on a loan.

You can also determine the amount of loan you can afford at a given payment rate. Or, you can find out the number of payments required to pay off a specific loan amount.

The program also tallies the total payments and total interest on loan repayment schedules.

We have found this a very handy program to have around. We blew the minds of a few local bankers with this little machine telling the money men the terms of a loan before they could figure things out using their loan tables, etc.

One side effect might be noticed. The data this program provides might discourage you from borrowing money these days. The amount of interest you pay with interest rates up around 20 percent is truly frightening!

You have 195 program steps left when this program is in memory.

Future

It will take about 25 seconds, using this program, to find out the interest rate required to accumulate a given sum over a stated time period at specific payment amounts. The manual points out that these calculations are sometimes referred to as "sinking fund" calculations.

You can work these types of problems by having the computer determine the rate, payment amounts, number of payments, or the future value. This program provides a quick way for you to check up on how your IRA or other retirement plan is working out.

You will have 222 programming steps available with this program in memory.

Bonds

It takes about 20 seconds to find the price of a coupon bond once you have inputted the value at maturity, the number of years to maturity, the coupon value and the bond yield percentage.

Alternately, you can enter the current bond price, the maturity and the coupon value. It can take a number of minutes for the little PC to pump out the answer for this type of calculation. Thus, every 15 seconds or so during this operation, the computer flashes the message "*** COMPUTING ***" on the display so that you will know it is doing something useful! A sample problem to determine the yield of a bond having a 20 year maturity took approximately 7 minutes of computer time. Ah well, if you are one of the lucky few who has little else to do but sit around and clip bond coupons.... Of course, you can always be doing something else while the machine cranks away. Remember, these programs give two little beeps when the answer is ready. You may be surprised just how far away you can be from the machine and still hear those beeps — even in a noisy office environment.

You still have 424 programming steps left in the machine when this program is in memory.

Internal Rate of Return

It takes about 12 seconds to determine the present value of an investment after giving the PC the future value, number of years to maturity, the annual return and the rate of return.

Alternately, you can find the internal rate of return after inputting present and future values, the number of years over which the investment is being made and the annual rate of return. When this calculation is done over a number of years it takes some time for the program to crank out the answer. During the process the display flashes the message "*** COMPUTING ***" every 8 seconds or so. In a test problem it took approximately 4-1/2 minutes to solve an IRR problem projected over a five year period. It took roughly 5-1/2 minutes to perform the same calculation projected over a 10 year period.

You have 422 program steps left when this program is in memory.

Compound Interest

It takes about 8 seconds to find the earned rate after inputting the present and future values, number of periods per year and number of years over which a fund is deposited.

This program also enables one to determine the future or present value of a fund or the number of compounding periods required to reach a maturity sum.

You have 418 program steps available when this program is in memory.

Depreciation

It takes roughly 9 seconds to get the depreciation figure on an item using the straight line method.

The same calculation using the double declining schedule requires about 10 seconds.

Or, in about the same amount of time you can calculate depreciation using the sum of years-digits method.

The program leaves room for 423 program steps.

Summary

If your time is worth $5.00 or more per hour and you are a businessman or investor, this package is probably worthwhile.

While there is nothing fancy about the programs — indeed the programs are little more than straightforward implementations of common formulas — you would be hard pressed to locate the formulas, write them up in suitable program form and enter them into the computer in a half-hour or so. If it took you more time than that (per program) you might as well buy this package. What you are really buying is the convenience of not having to create and type in the programs.

The only real complaint we have about the package is that the programs are not really compact. This can not really be classified as a "fault" as it is simply a choice made by the package designers that we disagree with. The designers obviously made the decision to have the programs easy to use, with the user being cued at every step. This is certainly a legitimate philosophy and one that many beginning computer users applaud. However, the drawback is that you have to spend more time loading each program as you need it. It would be nice if three or four of these closely-related routines could be combined into one program. This might well be possible at some sacrifice in cueing the user, etc., *which could be compensated for by a more detailed instruction manual.* This is especially so since a number of the programs are similar in function and some leave a fair amount of memory unused. Some good subroutining might do the trick. In any event, it might be something to strive for in the future. From a practical viewpoint, a major attraction of a PC is its portability and convenience. These assets are seriously compromised when one has to carry a slew of paraphernalia (cassette interface, tape player, cassettes) around and spend five minutes or so diddling to put a different program in memory. The compactness of programs on such a machine is certainly a valid design criteria for sophisticated users.

Another solution to this aspect would be for Radio Shack — or anyone else for that matter — to come up with a system that would enable programs to be saved/loaded a lot faster. With the limited amount of memory in this machine it should be feasible to store all the contents on a small magnetic card or strip. Such a peripheral could greatly ease the task of changing the machine's function from moment-to-moment.

All-in-all we would give the Business Finance Program Package a rating of 6-1/2 on a scale of 10. Anything above a 3 on our scale is probably a worthwhile purchase for the average user.

****

ALARM CLOCK PROGRAM

There you are on a sales trip. You stop at your favorite low cost motel and get ready to catch forty winks. Suddenly you realize you forgot your trusy alarm clock. Your low cost motel doesn't offer a wake up service. What will you do? What will you do!?

But of course, you wouldn't travel without your trusty PC. So, grab it out of your traveling briefcase and stuff this quickie program into it. Presto, you will have time in your hands — or at least something close enough to it for you to catch those forty winks without fear of sleeping beyond your next sales meeting.

```
500 INPUT "ALARM TIME (HH.MM)? ",B:G=INT (B):L=(B-
    INT(B))*100
510 INPUT "CURRENT TIME (HH.MM)? ";A:H=INT(A):M=
    (A-INT(A))*100:S=0
520 PAUSE H;M;S:IF H<>G THEN 550
530 IF M<>L THEN 550
540 PAUSE H;M:BEEP 3:GOTO 540
550 S=S+2:IF S<60 THEN 520
560 S=0:M=M+1:IF M<60 THEN 520
570 M=0:H=H+1:IF H<24 THEN 520
580 H=0:GOTO 520
590 END
```

We will skip a tutorial analysis of this program. It should be pointed out, however, that the timing could vary slightly from unit to unit or even upon the condition of your batteries. We found the program accurate to within about two minute per hour. That is close enough provided you don't take excessively long naps! Also, take note of the fact that the clock loses track of time once the alarm goes off.

We would be interested in hearing from those of you that implement the program. How accurate was the timing on your unit? We will report on the census if we get enough responses to make it interesting.

Can someone come up with a more sophisticated version of the traveling alarm clock?

****

HOT AIR

Have you heard about the Panasonic/Quasar HHC (Hand Held Computer) units? Looks like someone sure wants you to hear about them. They have been receiving a lot of coverage in the popular press and trade magazines. There was even a big spread on the units in the January, 1981, issue of BYTE Magazine....

It seems the basic unit contains a real 6502 microprocessor running at a 1 MHz clock speed. It is reportedly equipped with a special "power-down" control circuit that minimizes current drain when the computer is not in actual use. Nickel-cadmium batters, which are rechargeable, enable the unit to run for a number of hours in portable use.

The "hand held" CPU portion has a QWERTY style keyboard and a 26 character liquid crystal display on its 9 inch by 3-3/4 inch top surface. The unit is reported to be about 1-1/4 inch thick. The display is said to handle upper and lower case characters. The keyboard has 2 key rollover protection. Weight is claimed to be 14 ounces.

Supposedely built into the HHC is an operating system including a 4-function calculator, editor, clock, and an electronic filing system. This package is apparently included on a 16K ROM that comes with the unit. There is also supposed to be some RAM included in the basic unit — at least enough to provide 500 or so characters of storage for the user in the basic unit. Indications are that RAM can be expanded up to 4K bytes in the HHC itself. Similarly, ROM is expandable to as much as 64K bytes in the basic unit when chips of projected capability become available in the future. It is also said that the machine will be capable of addressing up to 4 megabytes of RAM and/or ROM memory in external modules!?

The HHC will also be equipped with an audible beeper having the capability of emitting sounds over a 4 octave range.

A whole slew of plug-on extension modules are planned to support the HHC. These include RAM and ROM memory modules, a cassette interface, a video display module with color graphics capability, and a 16 character-per-line miniature printer. All of these devices are said to be designed so that various configurations will fit comfortably in an ordinary size briefcase.

No reliable pricing information on the Panasonic HHC has been obtained to date. However, industry sources have reported estimates ranging from $400.00 to $800.00 for the basic HHC unit.

Though reports, press releases, and comprehensive articles describing these units abound in the press, don't hold your breath waiting for delivery. After earlier rumors that the devices might be available in the latter part of 1980, it didn't happen. The latest word we were able to obtain, directly from Panasonic's marketing department, was that they would not be available until at least "mid-year" 1981!

Better keep working out on that little TRS-80 PC, friends. As they say, a bird in the hand is worth two in the bushes....

****

FROM THE WATCH POCKET

Adaptations of programs for the TRS-80 PC are popping out of the woodwork in short order. Page 182 of the January, 1981, issue of BYTE Magazine has a complete program that calculates discrete-Fourier-transforms. We loaded the program into memory and checked it out. It appears to work like a champ. There is sufficient memory left with the program installed for a good number of waveform points to be analyzed.

Page 184 of the same issue has another program that computes the time-domain response of systems using a given transfer program.

Both of these programs should be of considerable interest to engineers and technicians as well as scientists in a variety of disciplines.

It is nice to see this kind of quality "stuff" springing up so rapidly for these little PCs!

****

SHARE YOUR CREATIONS

What are you doing with your pocket computer? What kinds of application programs have you cooked up? How are you customizing your unit for your own use? Have you picked up any operating time-savers, programming tips, short cuts or technical information that might be of use to others with similar interests?

If you answer "yes" to any of the above questions, you are probably in a position to please a lot of people (our readers!) as well as earn some rewards for yourself.

What kind of rewards? How about hard cash. Plus the recognition and respect of your peers that comes from being published in a discerning medium.

If you are interested in these kinds of rewards, then hurry up and get busy! Write up your ideas. Send us listings of your programs. Tell us what they do, how they operate, how to use them, and explain their practical applications.

It doesn't hurt to assume a tutorial stance. Tell others how and why you developed a program. Explain the reasoning behind your programming techniques.

If your submission is good and we accept it for publication, you will hear from us within four weeks. If it is not up to our standards or we have already obtained similar material, we will endeavor to promptly return your material *provided you include a self addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials. The best safeguard is to make a copy for yourself before submitting material to anyone.

Get on our list of approved authors while we are still young and growing. The time for action is now!

Article submissions should be addressed to.

PC NEWSLETTER EDITOR
PO Box 232
Seymour, CT 06483

****

THE TAX MAN COMETH

It is that time of the year again. Uncle Sam is ready to take his bite out of everyone's pocket.

If you run a small business, you know that he takes more than just coins out of your pocket. He also takes a lot of time out of your life! You have to calculate witholding and FICA taxes for all your employees every week, keep summaries, make out W2 forms at the end of the year, file quarterly 941 forms, etc., adnauseum.

Well hang in there pard'ner. We have help coming for you in the next issue of PC NEWSLETTER. Yep, believe it or not, we will be publishing an incredible program that fits in the TRS-80 PC. It will calculate federal witholding tax, FICA, provide an option for other deductions, and present net pay for virtually any wage earner on your payroll. We think you are going to be amazed at how such a powerful program could be fit into such a small computer.

(C) Copyright 1981

3

03/81

POCKET COMPUTER NEWSLETTER

P.O. Box 232, Seymour, CT 06483

THE TAX MAN COMETH

And for most small businesspeople he comes every week! That means calculating the payroll deductions for each employee to come up with a net take home figure. This can become a downright depressing task, to say nothing about the amount of time it takes, what with having to use the FICA & FWT tables, calculator and so forth.

Well, you don't really have to go to all that trouble if you have a computer handy, right? You just load the FWT (federal witholding tax) tables into memory, make that deduction, crank out the FICA stuff using a formula, deduct that, and so forth.... But how much memory do you need in order to store that FWT table?

Friends, you don't need an FWT lookup table at all. Good old Uncle Sam came up with a piece-wise linear model of our sliding tax scale many years ago that enables FWT amounts to be directly calculated as a function of a person's exemptions, marital status and gross pay. This model fits easily in a TRS-80 Pocket Computer. And it is obvious from a lot of programs we have seen that a lot of people don't know about it. Witness the tape cassette program we recently reviewed that had a 55,000 byte FWT lookup table that it had to access each time an employee's witholding was determined.... and the table had to be stored on a tape cartridge. Why it took forever and a day to do a week's payroll.

The accompanying program assumes you know an employee's gross pay for a week. If you are starting with an hourly figure, you might want to remove line 10 and add a few lines to input hourly pay, number of hours worked, etc., ending up by assigning weekly gross pay to the variable G.

Note: this program calculates FWT specifically for **weekly** *pay periods.*

The program asks you to enter weekly gross pay, the number of exemptions claimed and marital status. It will then display the FWT deduction followed by the FICA deduction. Next the program allows you to enter a value for any other witholdings, such as, say, a deduction for insurance payments. Finally, you are given a net pay figure.

Pressing the ENTER key when the "net pay" figure is being displayed causes the program to loop back to handle the next employee.

It takes just a few seconds to perform the calculations on each employee. Quite a powerful little package to fit in your pocket!

Program Notes

Line 20 calculates FICA witholding. The individual amount in 1981 is 6.65 percent (0.0665). Remember, this percentage changes every year or so.

The subroutines starting at lines 60 and 80 calculate the FWT amounts for single and married persons respectively. The formulas used are based on the 1979 rates. These will have to be changed when the tax rate changes. *This may occur during 1981.* We will, of course, try to keep you advised accordingly.

A Warning

We are required to give the following warning to all users:

The PC NEWSLETTER *is not engaged in and does not represent itself to be engaged in the practice of giving tax or accounting counsel. The program and the data provided therein is for illustrative purposes only. We do not represent or warrant that the data produced by the program has any legal significance. You are solely responsible for your tax liabilities and should check with your own tax or legal counsel on matters concerning that liability.*

There, you get the message, right? Now pay those taxes!

```
10 G=0:INPUT "ENTER GROSS PAY? ";G
12 E=0:INPUT "# OF EXEMPTIONS? ";E
14 M=0:INPUT "SINGLE(0)/MARRIED(1)? ";M
16 IF M=0 GOSUB 60:GOTO 20
18 IF M=1 GOSUB 80:GOTO 20
19 GOTO 16
20 F=.0665*G
22 BEEP 2:X=INT(X*100+.5)/100:PRINT "FWT = ";X
24 F=INT(F*100+.5)/100:PRINT "FICA = ";F
26 Z=0:INPUT "OTHER WITHOLDINGS? ";Z
27 Z=INT(Z*100+.5)/100
28 G=G-X-F-Z:PRINT "NET PAY = ";G
30 GOTO 10
60 T=G-E*19.23
61 IF T<=0 GOTO 63
62 IF T>=27 GOTO 64
```

```
63 X=0:RETURN
64 IF T>=63 GOTO 66
65 X=.15*(T-37):RETURN
66 IF T>=131 GOTO 68
67 X=5.4+.18*(T-63):RETURN
68 IF T>=196 GOTO 70
69 X=17.64+.21*(T-131):RETURN
70 IF T>=273 GOTO 72
71 X=31.29+.23*(T-196):RETURN
72 IF T>=331 GOTO 74
73 X=51.31+.3*(T-273):RETURN
74 IF T>=433 GOTO 76
75 X=68.71+.34*(T-331):RETURN
76 X=103.39+.39*(T-433):RETURN
80 T=G-E*19.23
81 IF T<=0 GOTO 83
82 IF T>=46 GOTO 84
83 X=0:RETURN
84 IF T>=127 GOTO 86
85 X=.15*(T-46):RETURN
86 IF T>=210 GOTO 88
87 X=12.15+.18*(T-127):RETURN
88 IF T>=288 GOTO 90
89 X=27.09+.21*(T-210):RETURN
90 IF T>=369 GOTO 92
91 X=43.47+.24*(T-288):RETURN
92 IF T>=454 GOTO 94
93 X=62.91+.28*(T-369):RETURN
94 IF T>=556 GOTO 96
95 X=86.71+.32*(T-454):RETURN
96 X=119.35+.37*(T-556):RETURN
99 END
```

****

PRODUCT REVIEW
GAMES PACK 1

Produced by: Radio Shack for the TRS-80 Pocket Computer.
List Price: $14.95
Availability: Many Radio Shack Computer Centers and some Radio Shack stores.

Overview

You get eight programs spread over two cassettes in this package. The programs are repeated on each side of the tapes.

A 24 page manual that explains how to load and operate each program is provided. Example illustrations and samples of the programs' operation are included. Listings of the programs are not provided in the manual, but these can be obtained by placing the computer in the PRO mode.

A brief summary of each game program is provided here.

Missionaries and Cannibals

There are 352 program steps left in memory after this program has been loaded.

This is a game that tests your ability to think ahead. The object of the game is to get a group of three missionaries and three cannibals across a river using a boat that can hold no more than three people. Sounds simple enough, eh? The hitch is that the missionaries must not be left outnumbered by the cannibals at any time, on land or in the boat. There are one or two other obstacles to overcome also.

There was even the obstacle of an incorrectly working program in the tape that we purchased for this review. However, an errata was included with the manual to allow for the correction of this problem. Hopefully, later versions of this package have eliminated the programming error.

The game is simple and the programming execution left something to be desired in our opinion. For instance, when the boat was loaded with three missionaries and taken across the river, we found that the computer would still merrily load cannibals into the boat at our command even though they were now supposedly on the opposite bank of the river. Of course, the boat was then "stolen" by the illegally loaded cannibals. It might be more appropriate not to allow such illegal moves to occur.

On the positive side of things, the manner in which the program represents the characters being ferried on the boat and depicts the population on either side of the river is quite clever considering that the machine does not have any true "graphics" capability.

The game, in our opionion, is not very challenging This is particularly so for anyone accustomed to dealing with the logic of computers. We solved the problem on the first serious attempt at play and have no doubt that we could repeat the performance quite readily. Thus, the game offers no further challenge. It might, however, keep your children occupied for 15 or 20 minutes during a car trip. How much is that worth?

Nim

There are 763 program steps left in memory after this program has been loaded.

All this program really does is set up the playing board for the game of Nim. The board is represented as seven rows with seven items in each row. The game is intended for two players. The computer keeps track of the number of items removed from each row as the game proceeds. It will also identify the winner of the game. One observer commented that he thought the game was a lot more interesting when played with a handful of real pebbles.

Spaceship Lander

There are 840 programming steps left in memory after this program has been loaded.

Just about everyone is familiar with the old "lunar landing" game that has long been played by computer afficionados. This is a version of the popular game in miniature. Instead of starting at 120 miles out, you are put in command when the spaceship is a mere 2,678 feet above the surface of a planet. You control the thrust to slow the rate of descent of your craft. A good landing is considered to be anything that reaches the surface with less than 10 units of velocity.

Once again we felt that this version of the game was rather unexciting. It was very easy to win. A virtually perfect landing was made on the first play of the game. Gee whiz, what is left after that?

Treasure Hunt

You will have 447 program steps left in memory after this program has been loaded.

We found this game a little bit better than some of the others in this package for the simple reason that you get the dynamics of playing against the computer. The object of the game is to locate a treasure that has been hidden in a 10 by 10 matrix. You are given an imaginary "purse" of money at the beginning of a game with which you can "buy" clues regarding the location of the treasure. You lose the game if you run out of money before you have found the treasure. To play the game you need to either mentally or physically keep track of your moves on the grid. Radio Shack helps you in this regards by providing a full two pages (out of their 24 page manual) of blank grids.

To try and locate the hidden treasure you specify an X and a Y coordinate on the grid. Once this has been done the computer responds by telling you the number of grid units you are from the treasure, but not the direction. The computer then charges you a certain amount of money which is deducted from your purse. The amount of money is related to how far you have moved.

This game has a typographical error in the introductory message when it wishes you "goodluck" as one word.

Oh well, at least this game, from the description, appears as though it would last a little longer and give more variation than some of the other games in the package.

Unfortunately, we have not been able to actually play this game because even though both copies of the program provided on the tape would load without error, the program would not display the data as shown in the manual. Though we are not absolutely sure, it appears that our version of the program was not functioning properly. For instance, we found that the display would not show the dollar value remaining in our game purse or provide the hint value discussed in the manual. Maybe the real treasure hunt here is to see if the user can get the program to display the information in the same manner as shown in the instruction booklet.

Biorhythms

It looks as though this was Radio Shack's big project for this package. Why a whole 2-1/2 pages in the manual is devoted to explaining the theory behind biorhythms, complete with the famous three-line graph representing the proposed physical, emotional and intellectual conditions that the theory is claimed to be capable of predicting.

In any event the program provided certainly does give you information with which you could determine your biorhythm curves. In case you are the type that doesn't like to do a lot of grueling graphing, the computer will alert you when it detects a "mini-critical" or "critical" day in any of the characteristic curves.

There are only 319 program steps left in memory when this program is resident.

The program appeared to operate just as outlined in the manual. It indicated that the primary reviewer of this games package was at a physical high while simultaneously being on mini-critical low points on both the emotional and intellectual curves on the date that the review took place. Readers should thus be alerted that this might have some bearing on the overall rating given to the package.

Star Blaster

There are 227 programs steps left in memory when this program has been loaded.

This game has been around for some time and has popularly been termed "shooting stars." A similar game can be found on electronic toys such as the popular "Merlin" device. The purpose of the game is to convert the contents of a three-by-three matrix having the middle cell occupied and the surrounding cells empty, to the opposite configuration with the center cell empty and the surrounding cells occupied.

You re-arrange stars in the grid by "blasting" an existing star. This act causes stars in neighboring locations to be created or destroyed depending on the local pattern that exist at the time the star is blasted.

The game appears to function as described. However, it is somewhat difficult to visualize the true relationships with neighboring stars unless you actually draw the matrix with pencil and paper. This is because the contents of the matrix are represented on the single display line of the TRS-80 Pocket Computer.

We found this game somewhat more challenging than, for instance, the Missionaries and Cannibals game.

Craps

This program uses virtually all of available memory. There are only 13 program steps left after the program has been loaded.

This is a standard version of the dice game "craps." You are only allowed to play the pass line, but you do have the option of matching your bet to obtain "single odds" while shooting for your point.

Someone at Radio Shack wants to make "good luck" into a one word salutation.

This game seems to be fairly well executed. The dice action appears to be pretty mixed and "choppy" and the numbers come up fast enough to maintain interest in the game, though it is by no means a "fast" game. If you are a veteran gambler you will probably enjoy this somewhat limited version of the game. At least you can use it to whet your appetite for the real thing. You might even do pretty well since this version of the game doesn't permit any of the usual sucker bets.

Quick Watson

After this program has been loaded there are 65 program steps left in memory.

This is a detective game where, using deduction and information provided by the computer, you try to match suspects with crimes. In each game there are four types of crimes assumed to have been committed by any one of seven individuals. As each type of crime is presented you name a suspect. The computer provides clues to help you "peg" the guilty party. You are given eight "trials" in which to try and link up the perpetrators to the crimes. The game is fairly simple. It is slightly reminiscent of the board game "Clue." Kids might enjoy this game for a considerably longer period of time than the average adult.

Summary

Once again, there is nothing fancy about the programs supplied in this package. Most of the programs are straightforward implementations of commonly available computer games. Since the TRS-80 Pocket Computer does not have any kind of graphics capability, you have to use your imagination or pencil and paper in order to visualize some of the playing grids.

All-in-all we could not help but get the impression that there was not a great deal of thought put into the production of this package. It is as though someone decreed that Radio Shack must have a games package A.S.A.P. and so it was done. Virtually all of the programs provided could readily be implemented by the simple compacting of programs found in sundry computer hobbyist journals.

We are giving the games program package a rating of 4 on our scale. You may remember from the previous issue that we said anything above a 3 was probably a worthwhile purchase for the average person even if only because you saved the time of looking up and typing in the programs. That is about what you get here.

We know it is still a bit early in the game, but we hope that now that Radio Shack has fulfilled their "obligation" to have a games package available, that they will take the time and effort to come up with a little better quality stuff in Games Pack 2.

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others?

If so, write it up! If your submission is accepted for publication you will earn a cash reward. If we cannot use your material, we will endeavor to promptly return it *provided you include a self addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483

PRODUCT REVIEWERS WANTED

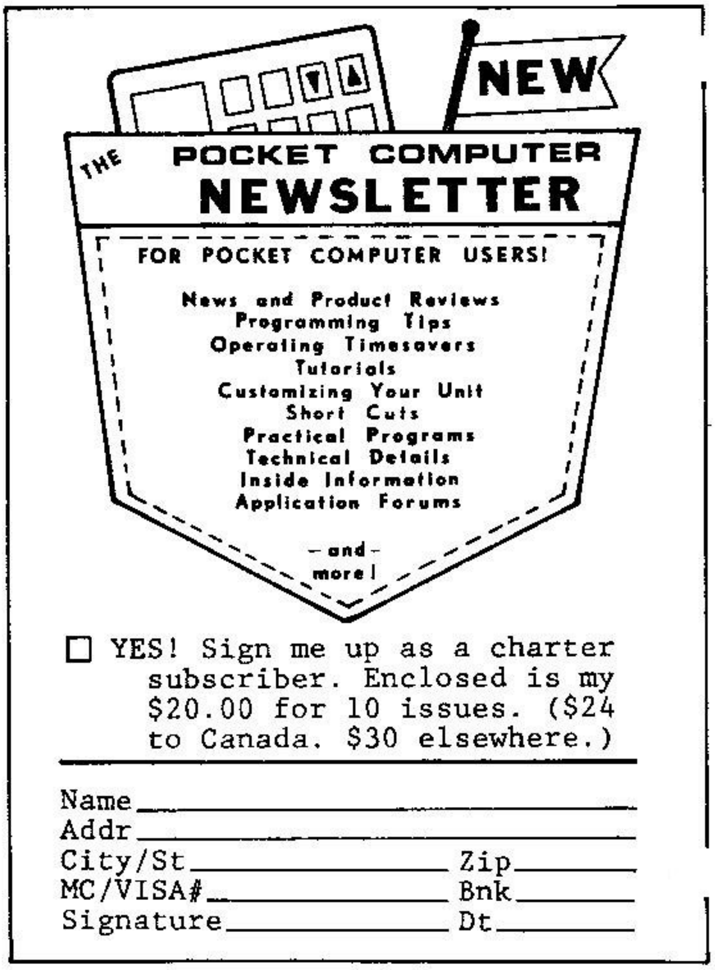
Do you have a few spare hours on your hands? Would you be available to review new pocket computer programs? We are looking for a few discerning individuals that feel they could write accurate, unbiased, reports on the operation of programs. Do they deliver what they advertise? Would you have bought the package? How could they do better?

If you would like to be on our list of volunteer reviewers, drop us a line, tell us what kind of pocket computer you have, let us know any particular areas of special interest or expertise.

What will you get for your time and effort? First, you get to keep the software package you review. Second, you will receive a cash bonus if your review is accepted for publication. Third, you will earn the respect and appreciation of fellow PC users who may be considering the software you report on.

You say you are not an expert writer? Just give us your findings as best you can. We will take care of polishing up the prose if that is needed. So, if you are interested, send in your name.... to the attention of the editor, of course.

****



1981 - Issue 04

(C) Copyright 1981

POCKET COMPUTER NEWSLETTER

P.O. Box 232, Seymour, CT 06483

SCELBI PUBLICATIONS IS OUR NEW PUBLISHER

On March 16, 1981, SCELBI Publications announced the purchase of the POCKET COMPUTER NEWSLETTER from its founding editor, Nat Wadsworth. SCELBI Publications, a producer of microcomputer products and microcomputer publications since 1973, will provide advertising, publication and fulfillment services. Nat Wadsworth will continue as editor.

RADIO SHACK TRS-80 PC INTERFACING SIGNALS

An investigation of the interfacing signals used on the TRS-80 PC has been underway for some time by a number of interested parties. The information provided here may be of value to fellow investigators. For reference purposes in the following discussion, we have assigned interface connector pin numbers as follows. When looking at the front of the tape cassette interface, the top pin is referred to as Pin 1 and the bottom as Pin 9, as shown in the accompanying diagram (Fig. 1).

Pins 1 through 5 are not associated with the use of the cassette interface. Indeed, they are not connected to anything on the cassette interface side.

Pins 6 through 9 are utilized by the cassette interface. Pin 9 appears to serve as a common connection. Using pin 9 as a reference point, the following information has been gleaned.

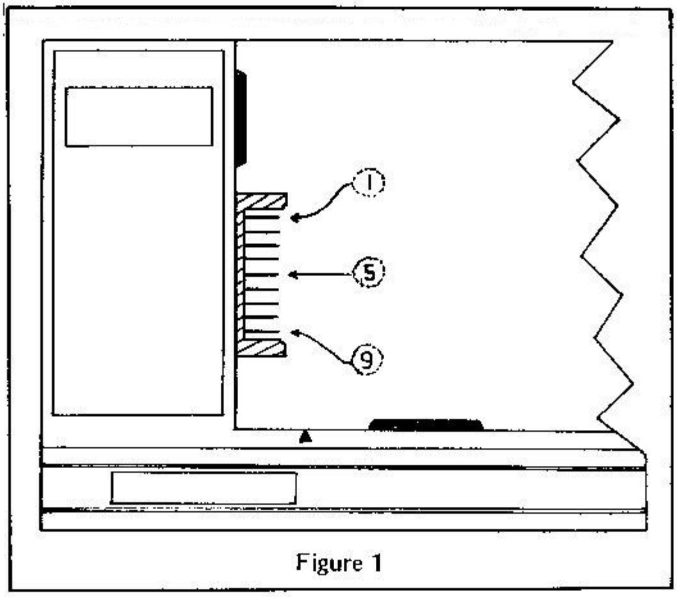


Figure 1

Pin 6 goes from a high state (4.5 volts) to a low state (0.4 volts) whenever the PC is in the tape read (CLOAD) or tape write (CSAVE) mode. It returns to the high state as soon as the operation is completed. This switching of states is synchronous with the audible tone that the PC emits whenever a tape read or tape write operation is in process. The signal on this pin activates a small relay that is part of the tape interface unit. The relay is activated when the signal on pin 6 goes low (0.4 volts). The contacts of the relay close under this condition to effectively "short" the plug that goes to the remote control jack on the tape unit.

Pin 7 routes data from the PC when the CSAVE command is used. During the time when tape "leader" is being generated (at the start of a tape save operation), a square wave of approximately 3800 Hertz is generated. When data is being transmitted, this frequency appears to alternate with a signal of half that rate (roughly 1900 Hertz).

Pin 8 passes data from the tape interface to the PC. Naturally, for data recorded using the PC, the received data is the same frequency as that used to store the information.

Does anybody know what pins 1 through 5 are used for on the Radio Shack TRS-80 Pocket Computer? We have lots of interested readers who would like to learn more about this area of the little computer's operation!

PROGRAMMING HINTS & TIPS

Norlin Rober of Marshalltown, Iowa, passes along these tips of interest to Radio Shack TRS-80 Pocket Computer users:

All closing parentheses may be omitted at the end of a line or at the end of a statement within a multiple-statement line or at the conclusion of the first part of an IF or FOR statement. Exception: The closing parenthesis used to complete a variable subscript may not be omitted.

The use of implied multiplication (e.g. 3AC rather than 3*A*C) when possible, can save bytes. Often rearranging operations within a statement permits greater utilization of implied multiplication. Also, expressions such as (5+A)B are legal, though not B(5+A).

Logical tests can often be replaced by logical calculations or by use of the SGN function; the result is often not only a shorter statement, but also the elimination of an extra program line, permitting more economization through the use of multiple statements. Some examples:

Use IF A rather than IF A > 0
Use IF –A rather than IF A < 0

Use D=D+(A < 5 rather than IF A < 5 LET D=D+1

Use D=D + ABS SGN A
rather than IF A <> 0 LET D=D+1

Use IF A IF B IF C as the logical
equivalent of IF A > 0 AND B > 0 AND C > 0

Closing quote marks may be deleted at the end of a line (though not at the end of a statement within a multiple-statement line).

Stored variables are rounded to 10 significant digits, whereas calculations performed within statements are performed using 12 digits. Thus maximum accuracy is retained by storing intermediate results as seldom as possible.

Norlin has also done some investigations into the accuracy of the built-in transcendental functions of the Radio Shack TRS-80 PC and reports that they are quite accurate, with the following exceptions:

Trigonometric functions are (unavoidably) inaccurate in RADIAN mode, for inputs near multiples of PI/2.

Example: tan 1.570796326 = 1258025227
TRS-80 PC gives 1273239545

Inverse cosine is weak for inputs near 1. (Accuracy is lost because the internal algorithm uses $1-X^2$, which will contain too few significant digits).

Example: ACS 0.999990007 = .2561450312 (Deg mode)
TRS-80 PC gives .2561456711

LOG is weak for inputs near 1 that are less than 1. The reason is that internally the computer calculates –LOG 1/X, giving an error in the 12th significant digit that becomes serious when X is near 1.

Example: LOG .99999684 = –.000001372372731
TRS-80 PC gives .000001372368394.

Exponentiation is weak in accuracy when results are very large numbers. It is worse still when the base used is close to, but less than 1, because of the poor accuracy in the logarithm in the first place.

Example: $.99999684^{71409500}$ = 9.989631172 E–99
TRS-80 gives 9.996756992 E–99

The EXP function is calculated internally by calculation of $10^{.434294481903X}$ and is accurate to 9 digits for large inputs and ten digits for smaller ones.

MEMORY SURPRISE!!

Here is another interesting tid-bit from *Norlin Rober.* Try the following sequence of operations on your Radio Shack TRS-80 Pocket Computer:

1) Clear the computer with the NEW command.
2) Enter this short program with the computer in the PRO mode: 1: 1OOOOOO (These are letter O's, not zeros.)
3) Stay in the PRO mode and clear out the above entry by keying: 1 ENTER.
4) Switch the computer to the RUN mode, key in 2Λ2 followed by the ENTER key.
5) Key A$(204) and press the ENTER key.
Hit the cursor "back" key, then the ENTER key.
Hit the cursor "back" key and the ENTER key again.
Press the CL key.
6) Key in MEM and hit the ENTER key.

Are you excited by what you see? Well, have fun experimenting. Norlin warns not to get too excited because he has discovered, for instance, that in this mode memory addresses 23 to 82 overlap 87 to 146. Also, A(83) to A(86) now appear to be the storage registers for FOR/NEXT loops in that they contain the address of the variable used as the index in a loop, the test value, step size and return address in hexadecimal format. If you do some experimenting, let us know what you come up with when using the computer in this mode.

TIMING COMPARISONS

Interested in programming for maximum speed on the TRS-80 PC? We have conducted a number of tests that indicate the order of variable useage should be as follows:

Fastest variables = W, X, Y and Z.
Followed by variables A through V.
Then A(23) through A(99).
Followed by A(100) through available memory.
Then the variables assigned as A(1)   A(9).
Slowest variables are those assigned A(10) – A(22).

The variables W – Z are about 18 percent faster than A through V. They are some 80 percent faster than using the designation A(10) – A(22)!

OOPS! THERE'S A HOLE IN OUR POCKET!

Two typographical errors snuck into the payroll tax program published in the POCKET COMPUTER NEWSLETTER, Issue 03. Correct the following lines:

```
65 X=.15*(T-27):RETURN
                    and
71 X=31.29+.26*(T-196):RETURN
```

SOLVING LINEAR EQUATIONS

This program calculates the solution of a system of up to 12 linear equations, using Gaussian elimination.

```
1 "A" INPUT "NO. OF EQ.?";A:A=A+1:B=0
2 "C" BEEP 1:FOR C=7 TO A+6:INPUT A(C+AB):NEXT C
  :IF B=0 GOTO 5
3 FOR D=1 TO B:E=A(A+AB-D+6):F=A(AD-D+6)
4 FOR C=7+AB TO A+C-1:A(C)=A(C)-A(C+AD-A-AB)E/F:
  NEXT C:NEXT D
5 IF A(C-B-1)=0 BEEP 1:PRINT "REJECTED":GOTO 2
6 B=B+1:IF A-B-1 GOTO 2
7 FOR B=1-A TO -1:F=6-AB:E=A(F)/A(F+B):A(F)=E:IF
  B+2 GOTO 9
8 FOR C=1 TO -B-1;F=AC+6:A(F)=A(F)-A(F+B)E:NEXT C:
  NEXT B
9 BEEP 1:FOR B=1 TO A-1:E=A(AA-AB+6):PRINT B;"    ";
  E:NEXT B:END
```

If an error in entry of coefficients is made, a correction may be made by keying SHIFT C, then re-entering the entire row of coefficients in which the error was made.

(continued on page 4)

```
5 P/R002 "ALLAN MCCROSKERY 1981"
10 "–" REM "P/R002"
20 P. "JOB ESTIMATE:SHIFT A"
30 P. "HOURS AND COST:SHIFT S"
40 P. "BILLING CHECK:SHIFT D"
50 GOTO 20
60 REM "ESTIMATE FOR JOBS"
70 "A":CL.:INPUT "BEVS DAYS? ";B:D=B*450
80 P. "BEV ";B;"DAY = $ ";D
90 INPUT "ALS DAYS? ";A:C=A*450
100 P. "AL ";A;"DAY = $ ";C
110 INPUT "TEDS DAYS? ";T:U=T*337.5
120 P. "TED ";T;"DAY = $ ";U
130 INPUT "GARY DAYS? ";G:H=G*243.75
140 P. "GARY ";G;"DAY = $ ";H
150 INPUT "MACES DAYS? ";M:N=M*243.75
160 P. "MACE ";M;"DAY = $ ";N
200 Q=B+A:P=D+C
210 P. "PART ";Q;"DS = $ ";P
220 P. "ASSOC ";T;"DS = $ ";U:R=G+M:S=H+N
230 P. "SR DSN ";R;"DS = $ ";S:E=P+U+S
240 GOTO 600
300 REM "HOURS AND COST CHECK"
310 "S":CL.:INPUT "BEVS HRS? ";B
320 INPUT "ALS HRS? ";A:Q=A+B:P=Q*60
330 P. "PART ";Q;"HRS = $ ";P
340 INPUT "TEDS HRS? ";T:U=T*45
350 P. "ASSOC ";T;"HRS = $ ";U
360 INPUT "GARYS HRS? ";G
370 INPUT "MACES HRS? ";M:R=G+M:S=R*32.5
380 P. "SR DSN ";R;"HRS = $ ";S:E=P+U+S
390 GOTO 520
440 REM "BILLING CHECK"
450 "D":CL.:INPUT "PART HRS? ";Q:P=Q*60
460 P. "PART ";Q;"HRS = $ ";P
470 INPUT "ASSOC HRS? ";T:U=T*45
480 P. "ASSOC ";T;"HRS = $ ";U
490 INPUT "SR DES HRS? ";R:S=R*32.5
500 P' "SR DES ";R;"HRS = $ ";S:E=P+U+S
520 P. "DSN COST = $ ";E
530 INPUT "CONN TAX Y/N? ";I:IF I="Y" GOSUB 700
535 IF I="N" GOTO 760
540 Z=E+1:P. "DES + TX = $ ";Z
550 INPUT "REIMBURSABLES? $ = ";F:X=X+F:GOTO 550
555 P. "REIMB TOTAL = $ ";X
560 W=E+X:P. "T/WOTX = $ ";W
570 Y=Z+X:P. "TOTAL = $ ";Y
580 END
600 P. "DSN COST = $ ";E
610 INPUT "CONN TAX Y/N? ";I:IF I="Y" GOSUB 700
620 INPUT "REIMBURSABLES? $ = ";F
630 W=E+F:INPUT "CONTINGENCY? % = ";J:L=INT (W*J+.5)/100
640 P. "DES + REIMB $ = ";W:O=W+L
650 P. "T/WOTX $ ";O:V=O+I
660 P. "T/WTX = $ ";V
670 END
700 I=INT(E*3.5+.5)/100
710 P. "CONN TAX = ";I
720 RETURN
760 INPUT "REIMBURSABLES? = $ ";F:X=X+F:GOTO 760
770 P. "REIMB TOTAL = $ ";X
780 W=E+X:P. "TOTAL = $ ";W
790 END
```

ASSIGNMENT OF VARIABLES

A = AL'S TIME
B = BEV'S TIME
C = AL'S COSTS
D = BEV'S COSTS
E = DESIGN COSTS
F = REIMBURSABLES
G = GARY'S TIME
H = GARY'S COSTS
I = CONN SALES TAX
J = CONTINGENCY %
K = ...not assigned.....
L = CONTINGENCY AMT
M = MACE'S TIME
N = MACE'S COSTS
O = DESIGN+REIMB+ CONTINGENCY
P = PARTNER'S COSTS
Q = PARTNER'S TIME
R = SD.DES TIME
S = SD.DES COSTS
T = TED'S TIME
U = TED'S COSTS
V = O + CONN SALES TAX
W = DESIGN + REIMBURSABLES
X = TOTAL REIMBURSABLES
Y = DESIGN+CONN TAX+REIMB
Z = DESIGN COSTS + CONN TAX

This is Allan McCroskery's *Estimating and Billing* program. He says it saves him a lot of adding machine tape! The program is divided into three sections. The first section is used to make estimates on a job. This section includes the costs of key persons working on the job, a lump sum figure for reimbursables and a "contingency factor" expressed as an added percentage. It is used to make quick estimates. The second section assists in the making up of bills. Actual hours spent on a job are inputted from time sheets. This section also adds a state service tax where applicable. The third section is used for verification purposes. Any questions? Al's address is: %Bevilacqua, McCroskery & Associates, Inc., 327 Main Avenue, Norwalk, CT 06851.

(continuation from page 2)

If the order in which the rows of coefficients are entered is not acceptable for the algorithm used by the program, "REJECTED" will be displayed (this happens infrequently) and the row just entered will need to be re-entered later. If this rejection occurs after every row has been entered, there is no unique solution (the system is inconsistent or dependent).

If desired, the display of the entire solution can be repeated by keying RUN 9, ENTER.

An example: To solve the system

$$\begin{array}{rcrcrcrcr} 3W & + & 2X & + & 6Y & + & 2Z & = & 2 \\ W & + & 4X & + & 3Y & + & 2Z & = & 6 \\ 2W & + & X & + & 9Y & + & 3Z & = & 2 \\ 4W & + & 3X & + & 6Y & + & Z & = & 8 \end{array}$$

which has the solution W=-2, X=3, Y=2 and Z=-5; proceed as follows.

SHIFT A (Computer must be in DEF mode.)

"NO. OF EQ.?" is displayed; input 4 and ENTER.

Beep prompts for input of equation; inputs to enter for this example are 3, 2, 6, 2, 2.

Beep prompts for input of next equation; input 1, 4, 3, 2, 6.

Next equation; input 2, 1, 9, 3, 2.

Next equation; input 4, 3, 6, 1, 8.

Beep indicates completion of execution, solution is displayed as follows:

```
1.  -2
2.   3
3.   2
4.  -5
```

Notes

The execution time required to solve a system varies according to the number of terms in the system. For a system of two equations, it is about 10 seconds; three equations, 21 seconds; four, 43 seconds; etc. A system of 12 takes a little over 14 minutes.

Practically all of the memory not actually used by the program statements is required for data. Any modification that lengthens the program will reduce the number of usable equations to less than 12.

This program was submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others?

If so, write it up! If your submission is accepted for publication you will earn a cash reward. If we cannot use your material, we will endeavor to promptly return it *provided you include a self addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: **The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483**

FROM THE WATCH POCKET

A program for the TRS-80 PC that helps restaurant owners forecast what sales revenues are needed to return a satisfactory profit is provided in the March/April issue of the 80-U.S. Journal. The article, written by John D. Farrell of Tacoma, Washington, starts on page 34 of the magazine.

Neil Maller of Fort Wayne, Indiana, reports that he has been successfully using a GE microcassette tape recorder, model 3-5340/SS 3 for program storage. Despite a tape speed of only 15/16 ips and the fact that the recorder is designed for dictation work, he reports results as "entirely satisfactory." He also indicates that the particular model mentioned has the handy feature of being operable in the fast forward and rewind modes without having to unplug the remote control plug. The price range of the GE microcassette is comparable to the tape recorder unit recommended by Radio Shack for use with their TRS-80 PC.

Did you know you could actually send a kid to summer camp to learn about computers? Yes, Dr. Michael Zabinski, a professor at Fairfield University in Connecticut, directs such a camp. Two sessions are offered during the month of July. For further information, contact Dr. Zabinski at (203) 795-9069 or write to: Computer Camp, Grand View Lodge, Box 22, Moodus, CT 06469.

Several people have reported difficulties with their TRS-80 PC "crashing" when in the PRO mode. The keyboard completely locks up. The only way to clear the situation is depress the reset switch on the back of the unit. Of course, that destroys whatever is in memory. Apparently Radio Shack doesn't have the answer to the problem. They have offered to exchange units exhibiting the problem, but one writer indicated that the new unit he received continued to behave similarly. Another writer reported that R. S. claimed the problem "might be caused by static electricity." Say, better be careful not to scruff across the carpets when you are working on a new program!

Radio Shack will not confirm rumors that new peripherals are being designed for the PC. Nor will they make any comments about sales to date. But, several industry observers peg the figure at around 50,000 units.

COMING ATTRACTIONS

In the next several editions of the POCKET COMPUTER NEWSLETTER we have all sorts of goodies scheduled. There will be more operating hints and tips. We have a whole series of programs ranging from some first class games to more serious applications such as finding prime factors and a speedy Shell/Walters sort routine. There will be more software reviews such as appeared in issues 02 and 03. We will be publishing a list of "tokens" used in the TRS-80 PC. Plus, news related to pocket computers from around the industry.

You will note that we have gone to a smaller type in this issue so as to cram in more information. If user contributions and interest expands, we will increase the number of pages per issue. We expect this to happen as soon as the Panasonic/Quasar units get into circulation.

1981 - Issue 05

POCKET COMPUTER NEWSLETTER

P.O. Box 232, Seymour, CT 06483

HEXED!

No, the TRS-80 PC doesn't have any real magical powers, at least none that I have discovered. "Hex" is short for *hexadecimal*, which is another way of denoting base 16. That is a numbering system near and dear to many a programmer's heart. Unfortunately, very few programmers are blessed with eight fingers per hand. Thus, unless they want to take off their shoes and socks, they have trouble counting in hexadecimal. Curses!

However, all is not lost. The pocket computer can change things to hexadecimal – once you show it how.

The accompanying routine converts decimal numbers to hexadecimal (label H) and hexadecimal numbers to decimal (label X). Since the PC can hardly handle groups of letters and since base 16 numbers use the letters A through F to represent the "digits" 10 through 15 respectively, you must key in hexadecimal numbers one digit at a time.

To convert 65535 to hexadecimal put the computer in the DEF mode (now it can't hear – sorry, I couldn't resist), type SHIFT-H and key 65535 ENTER. Following two beeps you see "65535 DEC" as verification of your input. Press ENTER and, a beep later, you will see "FFFF HEX".

To convert FFFF back to decimal, type SHIFT-X. The computer will ask for hexadecimal digits one at a time, going from left to right. Input "F ENTER" four times, then press the equal (=) key and ENTER for the result. "FFFF HEX" will be displayed after one beep to verify the input. Press the ENTER key one more time. Two beeps later you will see "65535 DEC" as the answer.

Please note the the somewhat unstructured arrangement of the program was purposely done to increase the speed of operation. The TRS-80 PC starts looking for labels and line numbers at the beginning of a program.

David Motto, 3639 Roosevelt, Jackson, MI 49203

```
1 "H" CLEAR:INPUT "DEC NUM?";M:M=INT ABS M:
  L=M:GOTO 36
2 "X" CLEAR:K=11:GOTO 38
3 A$(K)="0":RETURN
4 "0" L=16L:GOTO 38
5 A$(K)="1":RETURN
6 "1" L=16L+1:GOTO 38
7 A$(K)="2":RETURN
8 "2" L=16L+2:GOTO 38
9 A$(K)="3":RETURN
10 "3" L=16L+3:GOTO 38
11 A$(K)="4":RETURN
12 "4" L=16L+4:GOTO 38
13 A$(K)="5":RETURN
14 "5" L=16L+5:GOTO 38
15 A$(K)="6":RETURN
16 "6" L=16L+6:GOTO 38
17 A$(K)="7":RETURN
18 "7" L=16L+7:GOTO 38
19 A$(K)="8":RETURN
20 "8" L=16L+8:GOTO 38
21 A$(K)="9":RETURN
22 "9" L=16L+9:GOTO 38
23 A$(K)="A":RETURN
24 "A" L=16L+10:GOTO 38
25 A$(K)="B":RETURN
26 "B" L=16L+11:GOTO 38
27 A$(K)="C":RETURN
28 "C" L=16L+12:GOTO 38
29 A$(K)="D":RETURN
30 "D" L=16L+13:GOTO 38
31 A$(K)="E":RETURN
32 "E" L=16L+14:GOTO 38
33 A$(K)="F":RETURN
34 "F" L=16L+15:GOTO 38
35 "=" A$(K)="":GOSUB 39:GOSUB 40:GOTO "X"
36 K=K+1:N=M-16*INT(M/16):M=(M-N)/16:GOSUB 2N+3:
   IF M>0 THEN 36
37 GOSUB 40:GOSUB 39:GOTO "H"
38 K=K-1:PAUSE USING "###";11-K;"TH HEX CHAR?":
   INPUT A$(K):GOTO A$(K)
39 BEEP 2:PRINT USING "##########";L;" DEC":
   RETURN
40 BEEP 1:PRINT J$;I$;H$;G$;F$;E$;D$;C$;B$;A$;" HEX":
   RETURN
```

MORE OPERATING TIPS FOR THE TRS-80 PC

FOR loops are relatively slow on the Radio Shack TRS-80 PC. This is especially noticeable when using nested loops. Also, watch out for the 4 loop maximum!

The statement label capability of the TRS-80 PC is great, but GOSUBing to labels is much, much slower than using line numbers. In a particular application I was programming involving matrix manipulations, I saved 1/3 the program execution time by converting from labels to line numbers. If the application is short with few iterations of code, the difference may not amount to much. However, it can make a great deal of difference in a highly modularized program.

If you have several lines of similar code in a program, such as:

```
100 IF A=1 THEN 500
110 IF A=2 THEN 600
120 IF A=3 THEN 700
```

You can save a lot of keying by editing the line number and other differences after you have entered the original line. Since editing is done in a buffer, the original line is undisturbed. Just make the changes, hit ENTER, and presto!

Using the procedure discussed in the preceeding paragraph can give you the capability of a quasi-renumbering operation. Simply LIST the line, edit the line number and press ENTER. A few notes of caution are in order here. First, remember that you still have to manually change all line number references such as GOSUBs, etc. Second, don't forget that the original line still exists. To erase it, if desired, just key in the line number and hit ENTER.

When you desire to delete a series of lines in a program, start with the highest line number. This will save time. Why? Because making changes in the middle of program memory (whether editing or deleting) necessitates shifting the contents of memory (to the end of program storage) to accomodate the changes. Working at the end of the program reduces the amount of such manipulation.

The TRS-80 PC manual does not explicitly state that the PRINT USING statement does not perform rounding on the pocket computer as it does in some other dialects. On the TRS-80 PC it simply truncates or adds zeros. Remember to use a round-off algorithm, if necessary, when using the PRINT USING statement!

When editing numeric expressions or anything not enclosed by quotation marks, it is quicker to use the SPC key than the SHIFT DEL keys. When you hit the ENTER key the interpreter will remove any unnecessary spaces!

Submitted by: *C. C. Vans Evers II, 3607 Oklahoma, Tampa, FL 33611*

THE TAX MAN COMETH – AGAIN!

Issue 03 of *PCN* carried a neat program that calculated payroll deductions and net pay for employees paid on a weekly basis. One of our readers has written an even "neater" version of the program that we thought you might appreciate – so here it is. Thanks to: *David Motto, 3639 Roosevelt, Jackson, MI 49203*

```
10 USING "#####.##":H=E2
20 G=0:INPUT "ENTER GROSS PAY? ";G
30 E=0:INPUT "# OF EXEMPTIONS? ";E
40 M=0:INPUT "SINGLE(0)/MARRIED(1)? ";M:M=INT
   ABS M
50 IF (M<0)+(M>1) GOTO 40
60 X=0:T=G-19.23E:GOSUB 600+200M
70 F=.0665G
80 BEEP 2:X=INT (HX+.5)/H:PRINT "FWT= ";X
90 F=INT (HF+.5)/H:PRINT "FICA= ";F
100 Z=0:INPUT "OTHER WITHOLDINGS? ";Z
110 Z=INT (HZ+.5)/H
120 G=G-X-F-Z:PRINT "NET PAY= ";G
130 GOTO 20
600 IF T>=433 LET X=103.39+.39*(T-433):RETURN
610 IF T>=331 LET X=68.71+.34*(T-331):RETURN
620 IF T>=273 LET X=51.31+.3*(T-273):RETURN
630 IF T>=196 LET X=31.29+.26*(T-196):RETURN
640 IF T>=131 LET X=17.64+.21*(T-131):RETURN
650 IF T>=63 LET X=5.4+.18*(T-63):RETURN
660 IF T>=27 LET X=.15*(T-27)
670 RETURN
800 IF T>=556 LET X=119.35+.37*(T-556):RETURN
810 IF T>=454 LET X=86.71+.32*(T-454):RETURN
820 IF T>=369 LET X=62.91+.28*(T-369):RETURN
830 IF T>=288 LET X=43.47+.24*(T-288):RETURN
840 IF T>=210 LET X=27.09+.21*(T-210):RETURN
850 IF T>=127 LET X=12.15+.18*(T-127):RETURN
860 IF T>=46 LET X=.15*(T-46)
870 RETURN
```

JACKPOT SLOTS

Here is a very nicely written game for the Radio Shack TRS-80 Pocket Computer. Watch out – you can find yourself spending a lot of time with this game!

Jackpot Slots was submitted by: *Gary Heidbrink, Box 24, FPO New York, NY 09593*

```
5 "J":Z=3254:H=0:O=0:J=0
10 A$=" CHERRY"
15 B$=" APPLE"
20 C$=" PLUM"
25 D$=" BELL"
30 E$=" BAR"
35 F$=" 777"
40 G$=" ANY"
45 PRINT "    JACKPOT SLOTS":W=0
50 INPUT "NEED INSTRUCTIONS?(Y/N)";M$.IF M$=
   "N" GOTO 105
55 IF M$="J" PRINT "TAKE:$";J;" BAL:$";O:M$=
   "Y":GOTO 45
60 PRINT " EACH PLAY COSTS $ .25"
65 PRINT " PAY OFFS ARE:"
70 PRINT A$;G$;G$;"  $ .50"
75 PRINT A$;A$;G$;" $1.25"
80 PRINT B$;B$;B$;" $2.50"
85 PRINT C$;C$;C$;"  $3.50"
90 PRINT D$;D$;D$;"  $6.75"
95 PRINT E$;E$;E$;"  $50.00"
100 PRINT F$;F$;F$;" MIN. $50.00":PAUSE ""
105 INPUT " CHOOSE A MACHINE:";Z
110 INPUT " PLAY OR QUIT? (P/Q):";M$:IF M$="Q"
    GOTO 290
115 PAUSE " LEVER HAS BEEN PULLED!":J=J+.25:
    H=H+.05:W=W-.25:GOTO 125
```

```
120 Z=439147*Z+J:Z=23*Z-INT(Z*.23)*100:Y=INT
    (Z*.2)+1:RETURN
125 GOSUB 120
130 IF Y<8 LET I$=C$:GOTO 160
135 IF Y<14 LET I$=B$:GOTO 160
140 IF Y<17 LET I$=E$:GOTO 160
145 IF Y<19 LET I$=A$:GOTO 160
150 IF Y=19 LET I$=D$:GOTO 160
155 I$=F$
160 GOSUB 120
165 IF Y<7 LET K$=A$:GOTO 195
170 IF Y<12 LET K$=D$:GOTO 195
175 IF Y<16 LET K$=B$:GOTO 195
180 IF Y<19 LET K$=C$:GOTO 195
185 IF Y=19 LET K$=F$:GOTO 195
190 K$=E$
195 GOSUB 120
200 IF Y<10 LET L$=D$:GOTO 225
205 IF Y<15 LET L$=B$:GOTO 225
210 IF Y<19 LET L$=C$:GOTO 225
215 IF Y=19 LET L$=E$:GOTO 225
220 L$=F$
225 BEEP 1:PAUSE I$;" ";K$;" ";L$
230 IF I$=K$ GOTO 245
235 IF I$=A$ LET X=.5:GOTO 285
240 PAUSE " YOU LOSE!!":GOTO 110
245 IF I$=A$ LET X=1.25:GOTO 285
250 IF I$=L$ GOTO 260
255 GOTO 240
260 IF I$=B$ LET X=2.5
265 IF I$=C$ LET X=3.5
270 IF I$=D$ LET X=6.75
275 IF I$=E$ LET X=50
280 IF I$=F$ LET N=INT(H/.25)*.25:X=N+50:H=H-N
285 PRINT " YOU WIN!! $";X:W=W+X:GOTO 110
290 IF W<0 GOTO 300
295 PRINT " HOUSE OWES YOU $";W:O=O-W:GOTO 45
300 W=W*-1:PRINT " YOU OWE $";W:O=O+W:GOTO 45
```

TRS-80 POCKET COMPUTER TOKENS

Each byte of memory in the Radio Shack TRS-80 PC that is used for program storage of a user program contains a two-digit hexadecimal number. This number represents a particular character or instruction. The words making up the vocabulary of the computer, such as STEP, GOTO, INT, RETURN, and so forth, are thus *tokenized.* That is, each word is stored as a single byte that the interpreter is able to recognize as a symbol for the instruction or command. An instruction such as INPUT, though it may have been typed in by the user as a six letter word, is simply stored as a single byte after the program line is accepted. This process of *tokenizing* allows programs to be greatly compacted when stored in memory.

Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158, has spent a great deal of time and effort determining the tokens used in the TRS-80 PC. The hexadecimal tokens he has figured out are shown in the accompanying table.

It may be noted that the hexadecimal representation of the numeral digits 0 to 9 and the letters of the alphabet correspond with standard ASCII codes. However, punctuation marks and other symbols do not.

It is also interesting to note that a program line number begins with E0 to E9 as its first byte and 00 to 99 as its second byte. (Thus representing line numbers from 000 through 999.) A line in memory ends with the null byte, 00, which is produced by the ENTER key.

00	ENTER
11	SPC
12	"
13	?
14	!
15	#
16	%
17	¥
18	$
19	π
1A	√
1B	,
1C	;
1D	:
30	(
31	)
32	>
33	<
34	=
35	+
36	-
38	/
39	Δ
40	0
41	1
42	2
43	3
44	4
45	5
46	6
47	7
48	8
49	9
4A	.
4B	E
51	A
52	B
53	C
54	D
55	E
56	F
57	G
58	H
59	I
5A	J
5B	K
5C	L
5D	M
5E	N
5F	O
60	P
61	Q
62	R
63	S
64	T
65	U
66	V
66	V
67	W
68	X
69	Y
6A	Z
82	>=
83	<=
84	<>
90	TO
91	STEP
92	THEN
A0	SIN
A1	COS
A2	TAN
A3	ASN
A4	ACS
A5	ATN
A6	EXP
A7	LN
A8	LOG
A9	INT
AA	ABS
AB	SGN
AC	DEG
AD	DMS
B0	RUN
B1	NEW
B2	MEM
B3	LIST
B4	CONT
B5	DEBUG
B6	CSAVE
B7	CLOAD
C0	GRAD
C1	PRINT
C2	INPUT
C3	RADIAN
C4	DEGREE
C5	CLEAR
D0	IF
D1	FOR
D2	LET
D3	REM
D4	END
D5	NEXT
D6	STOP
D7	GOTO
D8	GOSUB
D9	CHAIN
DA	PAUSE
DB	BEEP
DC	AREAD
DD	USING
DE	RETURN
E0	Begin Line 0--
E1	Begin Line 1--
E2	Begin Line 2--
E3	Begin Line 3--
E4	Begin Line 4--
E5	Begin Line 5--
E6	Begin Line 6--
E7	Begin Line 7--
E8	Begin Line 8--
E9	Begin Line 9--

PROGRAM: ______________________ DATE: __________ PAGE ___ of ___

STEP		VARIABLE USAGE		
		A	1	
		B	2	
		C	3	
		D	4	
		E	5	
		F	6	
		G	7	
		H	8	
		I	9	
		J	10	
		K	11	
		L	12	
		M	13	
		N	14	
		O	15	
		P	16	
		Q	17	
		R	18	
		S	19	
		T	20	
		U	21	
		V	22	
		W	23	
		X	24	
		Y	25	
		Z	26	

NOTES:

TIC-TAC-TOE FOR THE RADIO SHACK TRS-80 PC

Here is a listing of a program that plays a non-losing game of Tic-Tac-Toe and keeps track of the score.

```
1 E=34:F=24:G=72:I=23:J=42:K=43:L=K:M=E:O=16:P=13
  :R=31:S=61:T=17:U=K:V=0:W=0
2 H=0:Q=0:GOSUB 11:IF X=5 GOTO 20
3 A=X:B=X-2*INT(X/2:Y=5:GOSUB 10:C=X:GOSUB 7
  :D=Y:N=A(D+12-9B):C=INT.1N:GOSUB 7:GOSUB 10
  :IF X+Y<>10 GOTO 13
4 C=N-10*INT.1N:GOSUB 7:GOSUB 10:IF X+Y<>10
  GOTO 13
5 IF B=0 IF D=8 LET C=6:GOSUB 7:GOTO 13
6 C=8+B*(D<9:GOSUB 7:GOTO 14
7 Y=INT((C+2)/3:Z=5-ABS(5-A:IF Z=3 LET C=6Y-C-2
8 IF Z=4 LET C=3C-8Y+6
9 Y=ABS(C-10*(A>5:RETURN
10 USING "##";PAUSE "MY MOVE,";Y·Q=10Q+Y
11 X=0:INPUT "YOUR MOVE: ";X:H=10H+X:RETURN
12 GOSUB 30:GOTO 11
13 W=W+V+1:V=0:Y=10-Y:Q=10Q+Y:BEEP 1:PAUSE
   "I WIN IN SQUARE";Y:GOTO 15
14 V=V+1:Q=10Q+Y:PAUSE "MY MOVE,";Y;"; TO
   KITTY."
15 PAUSE USING "###";"THE SCORE:":PRINT "YOU 0;
   ME";W;"; KITTY";V:GOTO 2
20 Y=1:GOSUB 10:A=X:Y=10-X+6*(X=9:GOSUB 10:B=X
   :IF ABS(A-3)=1 GOTO 25
21 Y=9-A+6*(A>6:IF X+Y<>10 GOTO 13
22 GOSUB 10:Y=A-4:IF B/2>INT(B/2 IF X+Y<>10
   GOTO 13
23 Y=6-X:IF X/2=INT(X/2 LET Y=10-X
24 GOTO 14
25 Y=10-B+(5-A)*(B=9:GOSUB 10:IF ABS(A-B+2)=1
   GOTO 27
26 Y=10-X+(X=9)*(X-8+ABS(B-3:GOTO 14
27 Y=1+(A+3)*(X=9:GOTO 13
30 PRINT USING "#####";"YOU:";H;"     ME:";Q:
   USING "##":RETURN
31 "D" GOSUB 30:GOTO 2
```

The spaces of the playing board are designated by number. The numbers correspond to their positions on the numeric keyboard of the pocket computer:

7	8	9
4	5	6
1	2	3

Execution of the program is begun by RUN, ENTER. Then, enter each of your moves when prompted by the display, entering the number of the space in which you want your mark. The computer will respond with its move. After each game, the running score is displayed. Key ENTER to begin the next game.

If you forget what marks have been made during the course of a game, key ENTER when it is your turn *without* having input your mark. The computer will display the marks made thus far by both you and it. Key ENTER again to resume play. (At the end of a game, key SHIFT D to display the marks made in the game just completed; keying ENTER will start the next game.)

This program was submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

A NUMERIC SORT

The accompanying Shell/Walters sort routine is a reasonably fast sort, even on the Radio Shack TRS-80 Pocket Computer. A 25-item random list was sorted in 110 seconds. That sorted list was resorted in reverse order in 101 seconds. This might be accomplished faster by hand. However, as a subroutine in a program, this amount of time might not be out of line considering the other types of operations that can be done by the PC at a much faster rate than manual computations!

```
1 REM NUMERIC SORT
10 CLEAR:PAUSE "NUMERIC SORT":INPUT "HOW
   MANY NUMBERS?";N:K=26
11 FOR I=1 TO N:PAUSE "# ";I:INPUT "> ";B:A(I+K)=B:
   NEXT I
12 PAUSE "SORT BEGINNING..."

988 REM SORT ROUTINE
989 BEEP 1:C=N
990 C=INT(C/3)+1:FOR I=1 TO N-C:IF A(I+K)<=A(I+C
    +K) THEN 996
991 T=A(I+C+K):B=I
992 A(B+C+K)=A(B+K):B=B-C
993 IF B>0 IF T<A(B+K) THEN 992
995 A(B+C+K)=T
996 NEXT I:IF C>1 THEN 990
998 BEEP 3:FOR I=1 TO N:S=A(I+K):PRINT "N(";I;")= "
    ;S:NEXT I
999 PAUSE "END OF RUN...":END
```

A(I) holds the array that is to be sorted starting at A(27). This puts the array up out of the way of other variables used by the sort itself as well as those used by any (user supplied) main program. N is the size of the array. K sets the beginning point of the array. N and K must be passed to the sort routine. K could be part of the sort routine if there were no reason to change it. I have found it easier to initialize K in the main program.

The maximum size (N) of the array is dictated by the amount of other resident code. Typing MEM will display how many flexible memories are available.

Note that this is strictly a *numeric* sort. The only string handling command for the TRS-80 PC is an equality (=) test. Less-than's and greater-than's are not supported. (If you want to use an IBM 370, buy one!)

This sort can be useful in any applications in which a list of numbers must be sorted (dates, invoice numbers, etc.). With a little programming effort, one could set up indexing for strings (7 characters only in the TRS-80 PC), but data memory capacity is so limited as to make such efforts essentially academic exercises.

Incidentally, this sort is faster than a Heap-sort that I converted. I think the primary reason is that the Heap-sort used two separate FOR loops rather than one. FOR loops chew up run time on the TRS-80 PC.

C. C. Vans Evers II, 3607 Oklahoma, Tampa, FL 33611

FINDING PRIME FACTORS OF A POSITIVE INTEGER

The program shown here for a Radio Shack TRS-80 PC lists the prime factors of any positive integer less than 1E10.

```
10 INPUT "INTEGER: ";I:J=I:M=0:T=2:GOSUB 50:T=3:
   GOSUB 50:T=5:GOSUB 50:T=1
20 T=T+6:GOSUB 50:T=T+4:GOSUB 50:T=T+2:GOSUB 50:
   T=T+4:GOSUB 50:T=T+2:GOSUB 50
30 T=T+4:GOSUB 50:T=T+6:GOSUB 50:T=T+2:GOSUB 50:
   IF TT<=J GOTO 20
40 GOTO 100
50 IF J/T-INT(J/T RETURN
60 IF T=I GOTO 130
70 M=M+1:J=J/T:IF J/T=INT(J/T GOTO 70
80 BEEP 1:PRINT USING "###########";T;" IS A
   FACTOR":PRINT USING "###";" OF MULTIPLICITY"
   ;M
90 M=0:IF TT<=J RETURN
100 IF I=J GOTO 130
110 IF J=1 PRINT " END OF LIST":END
120 T=J:J=1:M=1:GOTO 80
130 BEEP 1:PRINT USING "###########";I;" IS A
    PRIME":END
```

An Example

To factor 13510750, which equals $2x5^3x11x17^3$, key in RUN. When "INTEGER: " is displayed, input 13510750 and the ENTER key, which begins (and is used to continue) program operation. The results are shown thus:

```
 2 IS A FACTOR   OF MULTIPLICITY 1
 5 IS A FACTOR   OF MULTIPLICITY 3
11 IS A FACTOR   OF MULTIPLICITY 1
17 IS A FACTOR   OF MULTIPLICITY 3
   END OF LIST
```

Program Notes

The program is designed to hold down execution time. Multiples of 2, 3 and 5 (other than 2, 3 and 5 themselves) are omitted as trial divisors. The worst case, with 9999999967 as input, takes five hours; the number is a prime.

The program could be slightly shortened and speeded up by restricting usable inputs to smaller numbers, permitting tests of divisibility *not* requiring the use of the internally calculated 11th and 12th "guard" digits.

Variables Assignments

I = Integer to be factored
J = Given integer reduced by factors found to be prime
T = Test factor
M = Multiplicity count

This program was submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

THANKS, ROBERT!

The programming form included on page 4 of this issue is a slightly modified version of one submitted by *Robert J. Leahy of 15742 Williams Street, Tustin, CA 92680.* Note that there is no other material on that page so that you may reproduce the form in quantity for your own use!

FROM THE WATCH POCKET

No news yet on the release of the hand held computers being planned by Panasonic/Quasar. Some trade journal reports are pegging a "system" based on these units in the $2,000.00 price range.

Reports are now circulating that Sharp will introduce its PC-1211 Pocket Computer – virtually identical to the TRS-80 PC (after all, it is widely assumed that they built the R. S. unit) -- to the United States market in April or early May. Europeans are already familiar with the unit. Sharp also has the CE-122 combination printer and cassette interface ready to sell with the unit. The printer offers 16 characters per line. The unit is battery operated or can be used with an optional a.c. adapter.

The Sharp unit is reported to come with three manuals. One of these is a 300+ page "Applications Manual" which contains well over 100 programs. These range over the subject areas of mathematics, statistics, geometry, civil and electrical/electronic engineering, business and finance as well as a number of popular games. This manual would seem very attractive to current TRS-80 PC owners – keep an eye open for it.

There are now several publishers producing programs and booklets for PC users. One outfit, *Sebree's Computing, 456 Granite Avenue, Monrovia, CA 91016,* has a version of "Wumpus Adventure" available on cassette for $7.95 plus 95 cents shipping/handling. Another firm, *ARCsoft Publishers, PO Box 132, Woodsboro, MD 21798,* is working on a line of programs in booklet form and has pads of programming forms. Write to them for a catalog if interested.

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others?

If so, write it up! If your submission is accepted for publication you will earn a cash reward. If we cannot use your material, we will endeavor to promptly return it *provided you include a self addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483

POCKET COMPUTER NEWSLETTER

© Copyright 1981 -- Issue 06

THE SHARP PRINTER/CASSETTE INTERFACE

In early May, Sharp Electronics Corporation began marketing its new model CE-122 combination Printer/Cassette Interface for the PC-1211/Radio Shack TRS-80 Pocket Computer, in the United States (it had been available elsewhere earlier). It is a dandy little unit for its price of approximately $150.00.

The CE-122 combines a 16-column alphanumeric printer and an audio cassette interface in a single package measuring approximately 11 x 3-3/4 x 1-3/8 inches and weighing a little less than a pound, complete with built-in rechargeable ni-cad batteries.

The printer utilizes regular 1-3/4 inch wide rolls of paper and a tiny inked ribbon to produce characters in a dot-matrix fashion. The printer automatically indents and formats program listings stored in the pocket computer so that the up to 80-column lines are easily differentiated. Examples of actual print-outs will be provided by a number of program listings in future issues of *PCN*.

The cassette interface provides for recording programs and data from the pocket computer to an audio cassette or vice versa. A switch on the unit's console allows the operator to enable or disable the tape "remote" switch so that the remote control plug does not have to be unplugged in order to rewind a tape or place it in the fast-forward mode. This addition is a very nice convenience for those that frequently utilize a tape recorder with the pocket computer.

The Sharp PC-1211 or Radio Shack TRS-80 Pocket Computer mounts on and plugs into the printer/cassette interface in just the same manner as it does in the cassette interface that has been available for some time. A form of "handshaking" needs to be performed when using the printer so that the pocket computer knows how to output information.

To tell the PC that the printer is in use, the printer interface "power" switch and "printer on" switch are both placed in the "ON" position. Next, the power switch of the PC is depressed. When the PC has powered-up, its "ON" switch is depressed two more times. This process causes the computer to examine the status of the printer/cassette interface and determine that it is to route all PRINT statements to the printer unit. In this mode, only PAUSE statements will cause information to appear on the PC's display.

If the "printer on" switch is placed in the "OFF" position and the power switch of the PC is depressed twice again, the PC will switch back to outputting PRINT statements to the display of the PC unit.

The printer will also produce program listings — but only on an "all or nothing at all" basis. That is, you are not able to specify the printing of just a single program line or a group of lines. Only the full "LIST" command activates the printer when the PC is in the "program" mode, so a program must be listed in its entirety (unless you shut the unit off). From a practical viewpoint, this limitation isn't all that bad. Most users want a program listing in its entirety for documentation purposes. If you are just debugging, you can view single lines or groups of lines on the PC display.

The printer produces characters at the rate of about one line per second (16 characters per second). This is not lightning fast, but it is better than early teletype machines.

The unit is highly portable. It fits readily in even a thinline briefcase. The rechargeable batteries will produce about 8,000 lines when fully charged, so you can carry it around for awhile before juicing it up again.

The CE-122 Printer/Cassette Interface is standardly supplied with three rolls of paper, a ribbon cartridge, an AC adapter for recharging its batteries, cassette connection cables, an instruction pamphlet and a carrying case. The paper and ribbon cartridges are used in other Sharp calculator products so they can often be obtained locally at office supply stores that may not actually be carrying the Sharp PC or CE-122.

Sharp appears to be marketing the PC-1211 and accessories such as the CE-122 in the U.S. primarily through selected computer stores, including the Computerland chain, plus college and university stores and a limited number of mail order houses. It will be interesting to see how this approach fares against the might of the 7,000 outlets of the Radio Shack network that originally introduced the PC under their TRS-80 brand name.

RADIO SHACK ANNOUNCES PRINTER

As this issue of *PCN* went to press, Radio Shack announced the availability of a printer/cassette interface for their popular TRS-80 Pocket Computer. The unit appears essentially identicle to the Sharp CE-122 unit and is retailing in Radio Shack Computer Centers and selected Radio Shack Stores at a price of $149.95. Radio Shack has also announced that they will be stocking paper and ribbons for the unit which should be a boon for CE-122 owners as well!

THIS MAY JUST BOGGLE YOUR MIND!

The following article and accompanying program was submitted by: *David G. Motto, 3639 Roosevelt, Jackson, MI 49203.*

I am a "Boggle-maniac." Many times during the course of a day, in my off-work moments, I am drawn to the little letters in the squares as a moth is drawn to . . . . Well, you know the expression. My main problem is that I am afraid. Call me silly, but I do not want the world to know of my obsession.

Unfortunately, the makers of *Boggle* seem to be indifferent to my forlorn circumstances. They continue to manufacture their superb game with what I see as a perverse attempt to provide sound and motion to the game. Really, rattling those cubes around in that container sets my teeth on edge!

For those of you who have not played the game, here is a brief description of it. You have a translucent cube that can hold 16 dice. On each side of a die there is a letter of the alphabet (the side with a 'q' also contains a small 'u'). You rattle the cube in order to "randomize" the dice. When you stop shaking the cube, the dice fall into square depressions in the cube, leaving the top face of each die exposed. Next you turn over an egg timer, examine the array of exposed letters, and race to form words.

The way you form words is to start at any letter on a die and travel to an adjacent letter. You can move across, up or down, and diagonally. You write down the letters as you go. But, you cannot back-track to the same letter in a word. You may, however, re-use letters again when you form other words.

So, you hop around the cube writing down words. The longer a word, the more it counts. This process continues until the timer runs out.

(Seeing all this put down in print makes me wonder what all the fuss is about?)

So, let us go to the program that I have named *Word Squares*.

Line 1 is the introductory line. It is useful if you forget to place your PC in the DEF mode and press SHIFT and then the SPACE key.

Line 2 is the random number generator. It is concise and fairly reliable.

Lines 3 through 28 contain the letters of the alphabet. The order used will be explained shortly.

Line 29 is used to clear all variables and to input a fractional "seed" to the random number generator.

Line 30 sets up the constants for the letters. These constants are related to the frequency with which letters typically appear in the English language. I dug the figures out of a table in a book by Fletch Pratt entitled *Codes and Secret Writing*. The table showed that the letter E occurs 13.108 percent of the time, the leter T 10.471 percent. The letters A, O, N, R, I and S make up the next 42.525 percent, etc.

I used these figures to build up my table. Each number in registers T through Y contain the probabilities in the integer portion and the range in the fraction portion.

Line 31 accepts the size of the square, anything from 1 by 1 to 10 by 10. Use something around 6 to keep the run time short.

Program *Word Squares*

```
 1:" "GOTO 29
 2:Z=98212+.211
   327:Z=Z-INT
   Z:RETURN
 3:A$(P)="E":
   RETURN
 4:A$(P)="T":
   RETURN
 5:A$(P)="A":
   RETURN
 6:A$(P)="O":
   RETURN
 7:A$(P)="N":
   RETURN
 8:A$(P)="R":
   RETURN
 9:A$(P)="I":
   RETURN
10:A$(P)="S":
   RETURN
11:A$(P)="H":
   RETURN
12:A$(P)="D":
   RETURN
13:A$(P)="L":
   RETURN
14:A$(P)="F":
   RETURN
15:A$(P)="C":
   RETURN
16:A$(P)="M":
   RETURN
17:A$(P)="U":
   RETURN
18:A$(P)="G":
   RETURN
19:A$(P)="Y":
   RETURN
20:A$(P)="P":
   RETURN
21:A$(P)="W":
   RETURN
22:A$(P)="B":
   RETURN
23:A$(P)="V":
   RETURN
24:A$(P)="K":
   RETURN
25:A$(P)="X":
   RETURN
26:A$(P)="J":
   RETURN
27:A$(P)="Q":
   RETURN
28:A$(P)="Z":
   RETURN
29:CLEAR :INPUT
   "SEED? ";Z
30:T=13108:U=23
   579.01:V=661
   04.026:W=731
   66.08:X=8923
   3.096:Y=9816
   3.155
31:)NPUT "SIZE?
    ";S
32:FOR Q=1TO S
33:FOR P=1TO S
34:GOSUB 2
35:O=E5Z:L=20.6
36:FOR M=20TO 2
   5:IF O<A(M)
   LET L=E2*(A(
   M)-INT A(M))
   :M=25
37:NEXT M
38:GOSUB 2:L=
   INT L+E1Z*(L
   -INT L)
39:GOSUB 3+L
40:NEXT P
41:BEEP 1:PRINT
   A$;B$;C$;D$;
   E$;F$;G$;H$;
   I$;J$:NEXT Q
42:GOTO 31
```

Lines 32 and 33 set up a nested loop to print the square.

Line 34 obtains the first random number, which is less than 1.

Line 35 scales that number so that it can be compared to the table set up in line 30. It also sets up a default value for variable L.

Lines 36 and 37 make up a FOR—NEXT loop which finds a value for L based on the random number. This value is further modified in line 38 by another random number. The third digit of the fractional part tells which letter of a group of letters is chosen. Finally, line 39 selects the actual letter.

Line 40 is the end of the internal loop.

Line 41 tells the user to look at the display and write down the line of letters shown there.

This program shows, I believe, that the string handling capabilities of the Radio Shack TRS-80 PC are perhaps not as bad as you may have thought!

PRODUCT REVIEW
BUSINESS STATISTICS AND MARKETING PACKAGE

Produced by: *Radio Shack for the TRS-80 Pocket Computer.*
List price: *$19.95*
Availability: *Radio Shack Computer Centers and selected Radio Shack stores.*
Reviewer: *Allan McCroskery, 327 Main Avenue, Norwalk, CT 06851.*

The Radio Shack *Business Statistics and Marketing* package consists of programs that perform relatively sophisticated statistical analysis, with the exception of the "Management Decisions" routines which most retailers would consider standard formulas. Reviewer *Allan McCroskery* reports he had to purchase a college study guide to bone up on what the various programs were actually accomplishing!

The package contains the following programs:

Forecasting -- you can input up to 100 values taken over selected intervals of time. The program then makes projected data values, provides the coefficients of the trend line equation (based on the values entered) and an estimate of the standard error value.

Seasonal Variation — You can input up to two years of monthly data (or eight quarters) and this program will compute the seasonal indexes based on the ratio-to-centered-moving-average technique.

Moving Average — Up to 60 data values can be averaged.

Normal/t/F/Distributions — This program computes normal, t, and F functions using standard approximation formulas. If you provide a mean, standard deviation and X value, the program will output the Z score, f(X), cumulative percentile, and the area under the normal curve between X and the mean. You can enter new X values without entering new mean and standard deviation figures. The program can also determine one- and two-tail probabilities for the t distribution after it has been inputted a t value and the degrees of freedom have been specified. Finally, given an F-ratio and degrees of freedom in the numerator and denominator, it can provide the right-tail probability.

Descriptive Statistics — You can feed this program up to 80 data values, grouped or ungrouped. The program can then provide information on sample size, mean, median, mode, sum, sum of squares, standard deviation, and variance.

Multiple Regression — This program uses the least squares method for fitting a straight line to produce regression statistics.

Gompertz Curve Analysis -- Up to 90 data values are fit to to the Gompertz growth curve. The program also provides an unbiased estimate of the standard error of estimate and produces the constants log a, log b and log c for the curves equation: $\log y = \log a + (\log b)c^x$.

Management Decisions — This program is really a whole package of smaller programs that all reside in the PC at the same time. It may well be the most generally useful program in the package for many purchasers. It includes routines to perform the following computations:

Break-Even Analysis
Optimum Order Quantity
Make versus Buy
Profit Margin
Markup
Price Elasticity
Manufacturing Cost Curve
Random Sample

The routines use common business formulas to solve the problems in straightforward fashion. What is nice here is that many of the calculations and retailer or wholesaler frequently has to make during a business day are all available in the PC at one time.

Allan reports that all the programs in the package loaded and operated as described in the 30-page booklet that is provided with the cassettes. His only complaint had to do with the fact that some of the programs require you to be very careful when entering data or you are liable to lose your place. The reason this can happen is because the programs typically just flash the important information, such as "ENTER VALUE # 2" and then leave just the question mark "?" on the display. A program such as the *Gompertz Curve Analysis* can have you entering up to three "data sets" with four values in each set. If you don't concentrate and watch the display carefully, you can get fouled up and have to start all over.

Generally, however, Allan rates the package with approval, adding that even if you cannot use all the programs you may benefit from a study of the programming methods utilized.

MORE INTERFACING INFORMATION

In Issue 04 of *PCN* we asked if any readers knew the functions of pins 1 – 5 of the Radio Shack TRS-80 PC. While several readers wrote to say that they believed they were for the use of a printer or similar device, which we had certainly surmised, *Ken Slaughter, 2916 Bangor Avenue, Highland, CA 92346,* was the first to send in pin-specific data which he indicates he obtained from PC service manuals! Now, if he will just tell us where we can buy those manuals?! In any event, here is the scoop he provided. The pin order is as described in the earlier *PCN* article.

Pin #	In/Out	Description
1	Out	Data to printer
2	In	Printer Busy
3	In	Printer Identification
4	Out	Computer Busy
5	–	Ground (Common)

OPERATING TIP

When you first go into the PRO mode on a Radio Shack TRS-80 or Sharp PC-1211 PC you can get access to statements in memory without entering the command LIST. If you want to look at the highest line number, press the "scroll up" key. If you want to look at the lowest line number, press the "scroll down" key. Once the initial line has been displayed you may use the scroll keys in conventional fashion to reach the line(s) of interest.

PROGRAM MEMORY ADDRESSES

Norlin Rober, who has provided material for *PCN* previously, recently sent in a package containing an incredible wealth of information concerning the layout and use of memory in the Radio Shack TRS-80/Sharp PC-1211 Pocket Computers. Norlin is proving himself to be a "software sleuth" par-excellent. All of the information he is providing has been gleaned by patient analysis of the results from countless hours of creative operation of his pocket computer. This issue he provides the following information regarding program memory addresses.

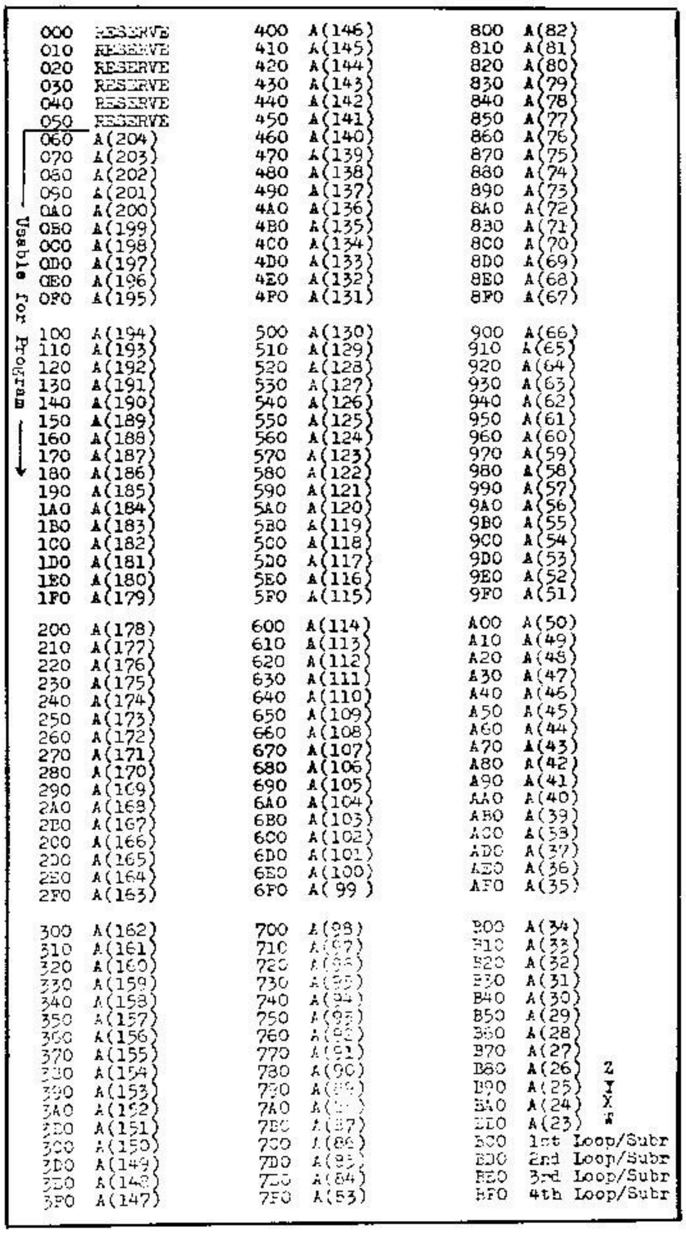
A three-digit hexadecimal address may be used to represent a count of the number of hexadecimal digits (which is exactly twice the number of bytes) that can be stored starting at the beginning of memory. Each memory address shown in the accompanying table is the location of the start of a block of 16 hexadecimal digits (stored in 8 bytes of memory). Each column of addresses in the table represents the storage space in one of three 512-byte RAM (Random Access Memory) chips in the pocket computer that are used for program storage.

Addresses 000 through 05F are used as REServe memory only. Addresses 060 to B7F may be used for program storage or for array variables in the positions tabulated. Addresses B80 through BBF hold the fixed variables A(23) through A(26), i.e., the variables named W, X, Y and Z. The registers used to store FOR/NEXT loop information and subroutine return addresses are designated BC0 through BFF.

A "pointer" register (located elsewhere in memory) holds the address of the first unused program byte. It is used to prevent storage or recall of array variables whose addresses are occupied by a program. When program lines are added or deleted, the contents of this pointer register are modified appropriately.

Thanks to: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158*

Table *Program Memory Addresses*



000	RESERVE	400	A(146)	800	A(82)	
010	RESERVE	410	A(145)	810	A(81)	
020	RESERVE	420	A(144)	820	A(80)	
030	RESERVE	430	A(143)	830	A(79)	
040	RESERVE	440	A(142)	840	A(78)	
050	RESERVE	450	A(141)	850	A(77)	
060	A(204)	460	A(140)	860	A(76)	
070	A(203)	470	A(139)	870	A(75)	
080	A(202)	480	A(138)	880	A(74)	
090	A(201)	490	A(137)	890	A(73)	
0A0	A(200)	4A0	A(136)	8A0	A(72)	
0B0	A(199)	4B0	A(135)	8B0	A(71)	
0C0	A(198)	4C0	A(134)	8C0	A(70)	
0D0	A(197)	4D0	A(133)	8D0	A(69)	
0E0	A(196)	4E0	A(132)	8E0	A(68)	
0F0	A(195)	4F0	A(131)	8F0	A(67)	
100	A(194)	500	A(130)	900	A(66)	
110	A(193)	510	A(129)	910	A(65)	
120	A(192)	520	A(128)	920	A(64)	
130	A(191)	530	A(127)	930	A(63)	
140	A(190)	540	A(126)	940	A(62)	
150	A(189)	550	A(125)	950	A(61)	
160	A(188)	560	A(124)	960	A(60)	
170	A(187)	570	A(123)	970	A(59)	
180	A(186)	580	A(122)	980	A(58)	
190	A(185)	590	A(121)	990	A(57)	
1A0	A(184)	5A0	A(120)	9A0	A(56)	
1B0	A(183)	5B0	A(119)	9B0	A(55)	
1C0	A(182)	5C0	A(118)	9C0	A(54)	
1D0	A(181)	5D0	A(117)	9D0	A(53)	
1E0	A(180)	5E0	A(116)	9E0	A(52)	
1F0	A(179)	5F0	A(115)	9F0	A(51)	
200	A(178)	600	A(114)	A00	A(50)	
210	A(177)	610	A(113)	A10	A(49)	
220	A(176)	620	A(112)	A20	A(48)	
230	A(175)	630	A(111)	A30	A(47)	
240	A(174)	640	A(110)	A40	A(46)	
250	A(173)	650	A(109)	A50	A(45)	
260	A(172)	660	A(108)	A60	A(44)	
270	A(171)	670	A(107)	A70	A(43)	
280	A(170)	680	A(106)	A80	A(42)	
290	A(169)	690	A(105)	A90	A(41)	
2A0	A(168)	6A0	A(104)	AA0	A(40)	
2B0	A(167)	6B0	A(103)	AB0	A(39)	
2C0	A(166)	6C0	A(102)	AC0	A(38)	
2D0	A(165)	6D0	A(101)	AD0	A(37)	
2E0	A(164)	6E0	A(100)	AE0	A(36)	
2F0	A(163)	6F0	A(99)	AF0	A(35)	
300	A(162)	700	A(98)	B00	A(34)	
310	A(161)	710	A(97)	B10	A(33)	
320	A(160)	720	A(96)	B20	A(32)	
330	A(159)	730	A(95)	B30	A(31)	
340	A(158)	740	A(94)	B40	A(30)	
350	A(157)	750	A(93)	B50	A(29)	
360	A(156)	760	A(92)	B60	A(28)	
370	A(155)	770	A(91)	B70	A(27)	
380	A(154)	780	A(90)	B80	A(26)	Z
390	A(153)	790	A(89)	B90	A(25)	Y
3A0	A(152)	7A0	[illegible]	BA0	A(24)	X
3B0	A(151)	7B0	A(87)	BB0	A(23)	W
3C0	A(150)	7C0	A(86)	BC0	1st Loop/Subr	
3D0	A(149)	7D0	A(85)	BD0	2nd Loop/Subr	
3E0	A(148)	7E0	A(84)	BE0	3rd Loop/Subr	
3F0	A(147)	7F0	A(53)	BF0	4th Loop/Subr	

Usable for Program

VEGAS CRAPS

This is the game of *Craps* designed for a single player. It allows most "right way" or so-called "bet to pass" options, with the exception of the "coming out" bet. A description of the various acceptable bets follows.

Pass Bet — This is the initial bet of any amount that lets the player in a game. The bet is won if the initial roll of the dice is 7 or 11. The bet is lost if the initial roll is a 2, 3 or 12 ("craps"). Any other number becomes known as the "point number" — this is the number the player will be shooting for on subsequent rolls of the dice. This number must be made before a 7 is rolled in order to win the bet.

Free Odds Bet — A bet of any amount up to the original amount of the initial "pass bet" may be made. Its name is derived from the fact that it is the only bet that gets even odds from a casino.

Field Bet — A bet of any amount that is won if a 2, 3, 4, 9, 10, 11 or 12 is rolled on the next toss of the dice. A 2 or 12 pays double money, the remaining "field" numbers pay even money. The bet is lost if a 5, 6, 7 or 8 is rolled.

Big 6 — A poor percentage bet that the number 6 will be rolled before a 7 out.

Big 8 — A similar bet for the number 8.

Bet the Hard Way — Any amount may be bet that any or all of the four possible "hard way" bets (4, 6, 8 or 10). The "hard way" bet is that the number will be rolled the hardest way — by rolling doubles — before the number is rolled some other way or the player rolls a 7. Pay offs for the hard way bets are listed in parentheses in the program.

Place to Win — A bet of any amount that the number selected (which may be a 4, 5, 6, 8, 9 or 10) will be rolled before a 7 comes up.

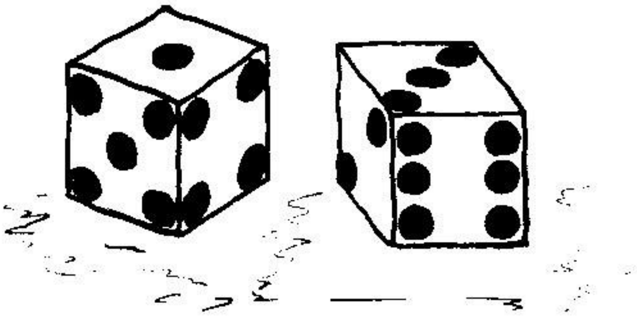
Have fun rolling your electronic dice!

This version of *Vegas Craps* was submitted by; *Gary Heidbrink, Box 24, FPO New York, NY 09593*

Program *Vegas Craps*

```
  5:"V":Y=3254:
    INPUT "RND#=
    ";Y
 10:FOR V=1TO 18
    :A(V)=0:NEXT
    V:PRINT "
       VEGAS CRA
    PS"
 15:IF P>0GOTO 3
    5
 20:INPUT "BET O
    R QUIT? (B/Q
    ):";Z$:IF Z$
    ="Q"GOTO 280
 25:INPUT "PASS
    BET:";A:Q=A:
    GOTO 120
 30:GOTO 20
 35:INPUT "CHANG
    ING BETS? (Y
    /N):";Z$:IF
    Z$="N"GOTO 1
    20
 40:INPUT "FREE
    ODDS BET:";B
    :IF B>ALET B
    =A
 45:INPUT "FIELD
     BET:";C
 50:INPUT "BIG 6
    :";D
 55:INPUT "BIG 8
    :";E
 60:INPUT "BET T
    HE HARDWAY?
    (Y/N):";Z$:
    IF Z$="N"
    GOTO 85
 65:INPUT "4 (7/
    1):";F
 70:INPUT "6 (9/
    1):";G
 75:INPUT "8 (9/
    1):";H
 80:INPUT "10 (7
    /1):";I
 85:INPUT "PLACE
     TO WIN? (Y/
    N):";Z$:IF Z
    $="N"GOTO 12
    0
 90:INPUT "4 (9/
    5):";J
 95:INPUT "5 (7/
    5):";K
100:INPUT "6 (7/
    6):";L
105:INPUT "8 (7/
    6):";M
110:INPUT "9 (7/
    5):";N
115:INPUT "10 (9
    /5):";O
```

```
120:PAUSE " DICE
      ARE BEING R
    OLLED"
125:GOSUB 290:T=
    W:GOSUB 290:
    U=W:S=T+U
130:PRINT " DICE
      ARE: ";T;"A
    ND ";U
135:IF P>0GOTO 1
    70
140:IF S=7LET Q=
    Q+A:FOR V=1
    TO 15:Q=Q-A(
    V):A(V)=0:
    NEXT V:GOTO
    240
145:IF S=11GOTO
    245
150:IF S=2GOTO 2
    50
155:IF S=3GOTO 2
    50
```



```
160:IF S=12GOTO
    250
165:P=13:PRINT "
    POINT # IS:
    ";S:Q=0
170:IF T=UGOSUB
    255
175:IF S=2LET Q=
    Q+C*2
180:IF S=3LET Q=
    Q+C
185:IF S=4LET Q=
    Q-F+C+INT (J
    *9/5):F=0:J=
    0:IF P=SLET
    Q=Q+B*2
190:IF S=5LET Q=
    Q-C+INT (K*7
    /5):K=0:IF P
    =SLET Q=Q+
    INT (B*3/2)
195:IF S=6LET Q=
    Q-C-G+D+INT
    (L*7/6):D=0:
    G=0:L=0:IF P
    =SLET Q=Q+
    INT (B*6/5)
```

```
200:IF S=7FOR V=
    1TO 15:Q=Q-A
    (V):A(V)=0:
    NEXT V:P=0
205:IF S=8LET Q=
    Q-C-H+E+INT
    (M*7/6):E=0:
    H=0:M=0:IF P
    =SLET Q=Q+
    INT (B*6/5)
210:IF S=9LET Q=
    Q+C+INT (N*7
    /5):N=0:IF P
    =SLET Q=Q+
    INT (B*3/2)
215:IF S=10LET Q
    =Q-I+C+INT (
    O*9/5):I=0:O
    =0:IF P=SLET
    Q=Q+B*2
220:IF S=11LET Q
    =Q+C
225:IF S=12LET Q
    =Q+C*2
230:C=0:IF S=P
    LET Q=Q+A:P=
    0
235:IF P=13LET P
    =S
240:IF Q<0LET Q=
    -Q:GOTO 250
245:R=R+Q:PRINT
    "YOU WIN $";
    Q:Q=0:GOTO 1
    5
250:R=R-Q:PRINT
    "YOU LOSE $"
    ;Q:Q=0:GOTO
    15
255:IF S=4LET Q=
    Q+F*7:F=0
260:IF S=6LET Q=
    Q+G*9:G=0
265:IF S=8LET Q=
    Q+H*9:H=0
270:IF S=10LET Q
    =Q+I*7:I=0
275:RETURN
280:IF R<0LET R=
    -R:PRINT "YO
    U LOST $";R:
    GOTO 10
285:PRINT "YOU W
    ON $";R:GOTO
    10
290:Y=439147*Y+R
    :Y=23*Y-INT
    (Y*.23)*100:
    W=INT (Y*.06
    )+1:RETURN
```

OOPS!

Line 10 of the payroll program in Issue 05, page 2, of *PCN* should end with the statement H=100. A typsetting error resulted in it reading H=E2, meaning the value 1 raised to the exponent 2. Sorry for the confusion that resulted when many readers took it to mean the variable E multiplied by 2.

OOPS, AGAIN!

The Tic-Tac-Toe game in Issue 05, page 5, of *PCN* contains a programming bug. Thanks go to *Robert Hatch, of Fort Sam Houston, Texas,* for being the first to report the bug to the program author. Author *Norlin Rober* quickly sent in the following patch to clean things up:

In line 22: Replace Y=A-4 by Y=14-A

In line 23: Replace Y=6-X by Y=6-B

Chances are many readers never detected the bug as it only showed up if the player made the moves 5349, 5672, 5729, 5834 or 5943. If you never played a game with those moves you would never know the program had a flaw.

INTERFACE YOUR PC TO A BIGGER COMPUTER!

A number of readers have inquired as to whether there is some type of device available that would allow a pocket computer to transfer data to a larger machine. The June, 1981 issue of *80 Microcomputing* (page 18) reports that an enterprising chap in West Germany has developed such an interface. It reportedly enables a PC to be hooked up to a regular disk-based Radio Shack TRS-80. The regular TRS-80 can then be used to transfer data or programs to the pocket computer. It is anticipated that programs that allow transfers in the reverse direction (PC to regular computer) will soon be available. For more information on this device, write to the developer: Reinhard Wiesemann, Winchenbachstr. 3a, D-5600 Wuppertal 2, West Germany.

OPERATING TIP

Ever been in the calculating mode , entered a long series of numbers in a complicated formula, hit the ENTER key, and get a result that doesn't make sense? You figure maybe you made an error in entry. So, you get ready to re-enter the whole darn formula again. *Wait!* On a TRS-80 or PC-1211 PC, all you have to do is press the forward or back cursor arrow key. Presto! The figures used in the calculation will be back up on the display. Use the same cursor control keys to review and edit the data as necessary. When you want to try for the answer again, press the ENTER key. You can do this as often as it takes to get the answer that satisfies *you.*

TRIP EXPENSES

The accompanying program is designed to keep track of various expenses during an automobile trip. It will tally such parameters as your average miles per gallon for an entire trip, the mpg for each tank, the highest and lowest mpg per tank, total trip mileage, gasoline and oil expenses, and expenses in other categories.

Here is a summary of the expense categories incorporated into the program.

GAS – Gasoline. A sub-account tallies oil usage.
MEAL – Food at regular meals.
SNACK – Records those between meals expenses.
ROOM – Lodging costs.
FUN – Another word for this is entertainment.
MISC – Expenses in all other areas.

The program should be executed in the DEF mode. There are four program labels used to enter various parts of the program. SHIFT N is used at the start of a new trip. This entry point clears all memories. SHIFT C is used to continue making entries during a trip. Use SHIFT = to call up current totals. SHIFT L may be used to recall the current high and low mpg figures.

Ready to take a trip? Fill up your gas tank, then press SHIFT N. Enter your starting mileage. If there are no expenses at this point, just turn the PC off. (Remember, you do not enter the cost of the initial tank of gas. However, to get an accurate record of a trip, you should always top your tank off at the conclusion of a trip and include that expense as part of the trip!)

Anytime you need to make an entry during a trip, just turn the PC back on and enter SHIFT C. When the message ENTER EXPENSE NAME appears, enter the appropriate category, such as GAS, MEAL, ROOM, etc. Answer the questions that are then provided by the program.

For example, if you entered GAS as an expense name, the program will ask you for the present mileage, the number of gallons purchased, the price, and whether or not you also bought oil. If you did, you will also enter its cost. When you are through making entries, just shut the PC off again.

When you want to examine totals to date, use the SHIFT = key. Incidently, you should run totals (by using the SHIFT = key) between gas purchases. Otherwise, you might miss a new low or high mpg per tank record. If you ask for totals and have not purchased gas yet, the trip mpg figure will not appear.

The program leaves only 13 steps of storage left. You are likely to run out of memory if you try to add much. I have tried to eliminate operator caused errors. If I have missed any, please let me know along with your thoughts on correcting the problem. The program could be compacted using the kinds of techniques that Norlin Rober mentioned in Issue 04 of *PCN.* Feel free to modify it as you see fit. This is the version I am using!

Submitted by: *Jack Forbes, 812 Bray Avenue, Rolla, MO 65401*

Program *Trip Expenses*

```
  5:"N"PAUSE "
       TRIP EXPE
    NSES":PAUSE
    "      BY JAC
    K FORBES"
 10:BEEP 1:PAUSE
    "  START WIT
    H FULL TANK"
    :CLEAR :
    INPUT "ENTER
     STARTING MI
    LEAGE ";K:G=
    K
 15:"C"INPUT " E
    NTER EXPENSE
     NAME ";A$:
    GOTO A$
 20:"GAS"INPUT "
    ENTER PRESEN
    T MILEAGE ";
    F,"HOW MANY
    GALLONS ";D:
    C=C+D:IF D=0
    THEN 60
 25:INPUT "WHAT
    WAS THE COST
     $";B:E=E+B:
    INPUT "DID Y
    OU BUY OIL Y
    /N? ";A$:IF
    A$="N"THEN 6
    0
 30:INPUT "    HO
    W MANY QUART
    S? ";B:L=B+L
    :INPUT "    C
    OST OF OIL $
    ";B:Q=B+Q:
    GOTO 60
 35:"MEAL"INPUT
    "ENTER COST
    OF MEAL $";B
    :O=B+O:GOTO
    60
 40:"FUN"INPUT "
     ENTER FUN C
    OST $";B:M=B
    +M:GOTO 60
 45:"MISC"INPUT
    " ENTER MISC
     COST $";B:N
    =B+N:GOTO 60
 50:"ROOM"INPUT
    " ENTER ROOM
     COST $";B:R
    =B+R:GOTO 60
 55:"SNACK"INPUT
    " ENTER SNAC
    K COST $";B:
    P=B+P
```

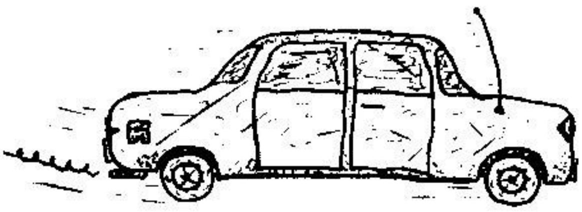
```
 60:INPUT "ARE T
    OTALS WANTED
     Y/N?";A$:IF
    A$="N"THEN 1
    5
 65:"="I=F-K:IF
    I<=0THEN 75
 70:PRINT " TOTA
    L TRIP MILES
     ";I
 75:IF (D=0)+(C=
    0)THEN 125
 80:B=I/C:PRINT
    " GRAND TOTA
    L MPG= ";B:H
    =F-G:B=H/D:
    IF B<=0THEN
    125
 85:G=F:PRINT "T
    RIP MILES PE
    R GAL ";B:IF
    S<>0GOTO 95
 90:S=B:T=B
 95:IF B<SGOTO 1
    10
100:IF B>TGOTO 1
    15
105:GOTO 120
110:BEEP 2:S=B:
    PRINT " NEW
    LOW MPG":
    GOTO 120
115:BEEP 2:T=B:
    PRINT " NEW
    HIGH MPG":
    GOTO 120
120:"L"U=INT (S*
    100)/100:
    PRINT "LO MP
    G= ";U;" HI=
     ";T
125:IF C=0THEN 1
    35
130:PRINT " GAS
    TOTAL IS ";C
    ;" GALS"
135:IF E=0THEN 1
    45
140:PRINT " TOTA
    L GAS COST $
    ";E
145:IF L=0THEN 1
    55
150:PRINT " TOTA
    L OIL USED "
    ;L;" QTS"
155:IF Q=0THEN 1
    65
```

```
160:PRINT " TOTA
    L OIL COST $
    ";Q
165:IF O=0THEN 1
    75
170:PRINT " TOTA
    L MEALS COST
     $";O
175:IF P=0THEN 1
    85
180:PRINT " TOTA
    L SNACK COST
     $";P
185:IF M=0THEN 1
    95
190:PRINT " TOTA
    L FUN COST $
    ";M
195:IF N=0THEN 2
    05
200:PRINT " TOTA
    L MISC COST
    $";N
```

```
205:IF R=0THEN 2
    15
210:PRINT " TOTA
    L ROOM COST
    $";R
215:B=E+Q+O+R+N+
    M+P:IF B=0
    THEN 225
220:PRINT " TOTA
    L TRIP COST
    $";B
225:IF I<=0THEN
    240
230:J=B/I:IF J=0
    THEN 240
235:J=INT (J*100
    0)/1000:
    PRINT " *COS
    T PER MILE*
    $";J
240:PRINT "     T
    OTALS COMPLE
    TED":GOTO 15
```

Variables Usage

A	String input
B	Variable input
C	Total gallons
D	Amount of gallons
E	Total gas costs
F	Present mileage
G	Starting mileage
H	Total trip miles
I	Trip mileage
J	Cost per mile
K	Starting mileage
L	Total quarts of oil
M	Total FUN costs
N	Total MISC costs
O	Total MEAL costs
P	Total SNACK costs
Q	Total OIL costs
R	Total ROOM costs
S	Low mpg figure
T	High mpg figure
U	Integer of S



FROM THE WATCH POCKET

Industry wags say Radio Shack has sold over 100,000 units of the TRS-80 Pocket Computer. R.S. dropped their price on the PC during the month of May from $249.00 to $199.00, maybe by pure coincidence, but this was the month in which Sharp Electronics chose to announce it was introducing its PC-1211 to the U.S. market at a list price of $249.00. After the May sale Radio Shack lifted the price up to $229.00.

The CE-122 combination printer/cassette interface appears to be receiving a warm welcome in the U.S., with comments generally favorable. The unit operates with either the Sharp PC-1211 or R.S. TRS-80 PC. Right now Sharp has the market for this peripheral all to themselves. However, Radio Shack is introducing an identical-performance printer to the U.S. market in early July at a competitive price ($149.95) and will also stock paper and ribbons for the unit.

Reliable sources indicate that the Panasonic/Quasar HHC is finally getting ready to hit the market, possibly as early as August/September. Production units are reportedly being placed in the hands of prospective distributors and large purchasers for their review at this time.

The Panasonic RL-H1000, whose design was licensed by *Friends Amis, Inc., 505 Beach Street, San Francisco, CA 94133,* has apparently been expanded in capability during recent months. The basic unit is now reported to have 8K-bytes of memory, to be capable of operating as both a stand-alone computer and a communications terminal, and to be supported with a variety of software packages. Add on units include a modem, mini-printer, ROM and RAM elements, and a video generator. Software packages include "spreadsheet" type programs, numerous business packages, operating systems that tailor the HHC to an application, and several languages.

The price of the basic HHC will apparently be about $500 with a typical small system for a particular application being in the $1000 -- $1500 range. All of this sounds exciting. Let's hope it all turns out to be true.

Another outfit that is specializing in PC programs is *Blanton Software Service, 4522 Briar Forrest, San Antonio, TX 78217.* They have programs such an an Appointment Calendar, Timer/Clock, Electronic Checkbook, Blackjack, a Personal/Business Expense ledger and others. I have taken a quick look at several of the programs and accompanying documentation and was favorably impressed. We expect to present reviews of several of the packages in coming issues of *PCN.* In the meantime, if you want to take a look at what they have, write to *them* for a catalog.

Meanwhile, an outfit mentioned in this column in the last issue, *Sebree's Computing,* did not receive a favorable review from our frequent games contributor *Gary Heidbrink.* He slapped a rating of about 2 (out of 10) on the package citing poor programming technique, such as an incomplete FOR—NEXT loop, as one of the reasons the program did not meet with his approval.

Electronic engineers and technicians are likely to appreciate a program for the Radio Shack and Sharp PCs that calculates LC resonance. Such a program is presented and described in the June 16, 1981, issue of *Electronics* on page 189. The program evaluates a resonant circuit having a lossless, single-layer airwound coil and a capacitor. Check it out if you are into that kind of design work.

Scott U. Johnson, in the June, 1981, issue of *Kilobaud Microcomputing,* responding to an earlier article of mine, mentioned an interesting comparison of the Sharp PC-1211 versus the Texas Instrument's TI-59 and Hewlett-Packard's HP-41C calculators. A bond yield formula was used as a benchmark. The HP-41C solved the test problem in 37 seconds, the TI-59 in 43 seconds and the PC-1211 in 51 seconds. But the problem only consumed 5.8% of the PC-1211's memory while it ate up 10.3% of the TI-59's and 15.6% of the HP-41C's storage. I'l take the convenience of the PC's BASIC over the almost machine language programming requirements of the other two units and live with the slightly lower execution time for the great majority of my applications. I really think the PC's showing is impressive when you consider it was up against two units one can assume were designed as streamlined number crunchers!

I still would like to see a faster method of loading programs into the PC. It is disconcerting, having the portability of the basic unit, to have to lug around a separate tape recorder and cassette tapes. Wouldn't it be nice if there were something, say about the size of the CE-122, that could read and write magnetic strips? Such strips could easily be carried in a jacket pocket or maybe even slipped inside the carrying case of the PC. The technology for storing the 1.4 kilobytes of memory used in the PC is certainly present today. Make the carrying around of supplementary programs more convenient, reduce the program/data loading time to a matter of seconds instead of minutes, and the PC will become a much more attractive unit to many prospective customers. — *Nat Wadsworth, Editor*

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others? Have you developed application programs that you would like to share with fellow pocket computer enthusiasts? Have you had experience with PC products or programs that others might find of value?

If so, write it up! If your submission is accepted for publication you will earn a cash reward as well as have the satisfaction of knowing you are helping other PC users.

We prefer that program listings be supplied on audio cassette tape so that they may be easily reviewed. We can also then make listings of accepted programs directly from the tape to avoid transcription errors. If we cannot use your material, we will endeavor to promptly return it *provided you include a self-addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483

POCKET COMPUTER NEWSLETTER

© Copyright 1981 — Issue 07

CASIO ANNOUNCES A HAND-HELD COMPUTER

A representative of Casio, Inc., indicated that the firm would begin U.S. deliveries of the Cas.o FX-702P hand-held computer sometime in September. The unit is said to closely rival the Radio Shack TRS-80 and Sharp PC-1211 pocket units in both size and capability. However, there are clear differences.

The Casio FX-702P has its keyboard arranged in linear alphabetical order instead of a typewriter-like QWERTY pattern. At 5/8 inch by 6-1/2 inches by 3-1/4 inches, the Casio unit is slightly shorter in length but slightly longer in width than the Sharp PC. The weight of 6.3 oz with batteries is almost identical to the Sharp/Radio Shack PC. The display of the Casio unit is dot matrix and capable of displaying 20 text characters in its window. The liquid-crystal display also simultaneously presents other information, such as operating mode and number of memory steps remaining available to the user.

The unit is programmable using a dialect of BASIC language. The programming capacity is reported as ranging from a maximum of 1680 steps with 26 variables to 80 steps with 226 variables. The number of possible variables decreases as program steps are utilized. Up to 10 user programs may be stored in memory at any one time. Subroutines may be nested to 10 levels and FOR-NEXT loops to 8 levels. Editing and debugging capability is provided. The Casio FX-702P hand-held computer (HHC) uses algebraic notation.

The HHC also has a variety of built-in functions such as trigonometric and inverse trigonometric, hyperbolic, logarithmic, exponential, powers, factorials, and so forth. Stastical functions include the sum, square sum, and two types of standard deviation operations. The unit also performs common types of linear regression analysis on command. Apparently many of these functions are implemented by calling upon ROM-stored subroutines. The FX-702P is also reported to support the LEN and MID character string functions. This would be a welcome addition for many HHC users.

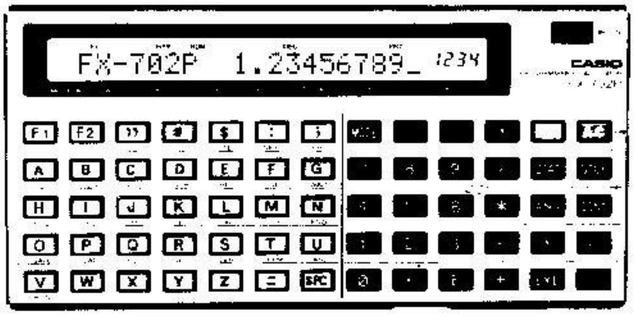
The capability of the FX-702P can further be augmented by installing a preprogrammed ROM cartridge into the back of the HHC. A number of different ROM cartridges are said to be under development to provide extended capabilities.

A distinct difference between the Casio HHC and the Sharp/Radio Shack PCs is that the FX-702P has interfaces built-in that will permit its direct connection to a cassette recorder for the storing and retrieval of programs and data, as well as its connection to small electro-discharge printer. The built in interfaces considerably lower the price of a complete system.

The initial price of the Casio FX-702P is reported to be set at $199.95 in the U.S., with the optional printer unit to carry a price of $79.95.

To obtain a brochure describing the unit, write to: *Casio, Inc., Consumer Products Division, 15 Gardner Road, Fairfield, NJ 07006.*

Figure *Casio FX-702P HHC*



FIRST PC PRICE WAR?

By a strange coincidence, as Casio prepares to launch its new FX-702P HHC at a U.S. price of $199.95, Sharp Electronics is reported to be significantly lowering the price of its popular PC-1211 pocket computer. Originally introduced to the U.S. at a list price of $249.95, the new list price is said to be reduced more than 25 percent to $179.95. Furthermore, the CE-122 combination printer/cassette interface is reported to have a new list price of $129.95, down from $149.95.

Even with this price reduction, the Casio FX-702P with its companion electro-printer will list some $30.00 lower than the PC-1211/CE-122 pair.

And where is Radio Shack in all this with its Sharp-equivalent TRS-80 PC? The betting here is that Tandy will soon join the fray. Perhaps soon coming in at a Sharp-matching $179.95 price or maybe even going for a $10.00 undercut.

And how long will Casio hold to its introductory price? And how long before others join the market? It won't be long. You can plan on seeing a lot of action in the PC/HHC arena during the next few months.

The $99.00 PC could be here soon, *real soon.*

Another *Personal Information*(TM) product from SCELBI Publications.

PRODUCT REVIEW
THE SHARP CE-122 PRINTER

Produced by: *Sharp Electronics, Inc.*
List price: *$150.00*
Availability: *Sharp distributors.*
Reviewer: *Ken Slaughter, 2916 Bangor Avenue, Highland, CA 92346.*

When the layman contemplates the term "computer" his first thoughts typically converge on the machine's mathematical prowess or its ability to store and retrieve volumes of information. These are powerful functions. However, what impresses me most about the computer is its ability to manipulate the environment.

A printer is the obvious extension of this idea. The addition of a printer to a computer greatly magnifies the utility of the machine. In fact, without the capability of providing hard copy of processed data, the practicality of the computer seems slight.

It is my feeling that the Radio Shack TRS-80 Pocket Computer should have had a printer accessory from the beginning. The reportedly light consumer acceptance of the unit, I believe, is due to its limited memory (without expansion capability at this time) and the absence of a printer.

What would an accounting firm do with a microcomputer that could only display spreadsheets on a television monitor? Consider the limitations imposed by an 80-character horizontal and 22-line vertical display format. Even sophisticated software, such as Visicalc, with its split-screen and adjustable "window" format, greatly handicaps the user. How do you provide information for your client, copy from the screen? If you want to view the data at another time, in or out of the office, you must stop and reload the program.

The pocket computer has singular advantages over competing calculators such as Texas Instruments' programmable MBA. But it was not until I connected my pocket computer to Sharp's CE-122 printer/cassette interface that I realized what an enormously powerful device the pocket computer really is!

The CE-122 is almost identical in appearance to the Radio Shack cassette interface except for the addition of about three inches to the left side which houses the printer hardware. The plastic case is an attractive, office decor brown with a subtle satiny finish. The seat for the pocket computer is angled about five degrees which greatly improves visibility of the PC's display during desktop use. Also, the CE-122 is heavy enough so that it doesn't scoot across the desk like a wet bar of soap when entering data with one hand. I had been searching for months for a calculator stand that would solve those problems.

The cassette interface part of the CE-122 is identical to the Radio Shack version with two welcome improvements. The cables leading from the interface to the recorder are detachable at the interface. Three sockets, like those on a cassette recorder, are provided at the rear of the unit. Better still, there is a "remote-defeat" switch on the printer's control panel that provides the user with manual control of the cassette recorder without having to unplug the cables.

There are no batteries to buy for this unit. It is powered by nickle-cadmium cells that recharge by way of an a.c. power cord supplied as a standard accessory. The power cord detaches from the unit near the cassette input jacks.

A low battery warning is provided by a LED (light emitting diode) on the printer control panel.

The printer uses paper 1.75 inches (4.5 cm) wide in tiny (I call them "hummingbird") rolls. Packages of five rolls are available directly from Sharp. The roll, with no spool, drops into its own compartment (nest) out of sight. Looks are deceiving as there is a satisfactory stretch of paper on a roll.

The microscopic ribbon comes as a replaceable cartridge. My first ribbon's ink was exhausted after three rolls of paper. To be fair, this represents a considerable amount of printing. The ribbon is small and ordinary printers are hard on ribbons. An owner of this unit would be wise to keep spares on hand. I am thankful that Sharp did not opt for a thermal printing method. I have found these to be notoriously unreliable. Incidently, ribbons are also available directly from Sharp. You can order ribbons and paper from the parts department of the Sharp service center listed on the back of the instruction manual that accompanies the printer until your local dealer gets the supplies in stock. *[Now that Radio Shack offers essentially the same unit, you can also get ribbons and paper from many Radio Shack stores! – N.W.]*

Sixteen extremely clear dot matrix characters are printed on each line. I initially thought 16 characters would be impractical. A 24-character format, matching the liquid crystal display, seemed more sensible to me, certainly for compatibility with the PC's display window. But, I've adjusted.

The printer unit is activated by depressing the PC's power ON key two times after the printer's ON switch has been activated. The printer then outputs under two conditions.

When the PC is in the PRO mode, typing LIST will cause the printer to output the entire contents of program memory, unless you press the BREAK key (or turn the printer unit off). You cannot specify the printing of just a single program line (such as LIST 120) nor can you list from a given line (such as LIST 120–). The printer seems to have some intelligent (?) discretion in formatting the program listing, though the rules are not entirely clear to me. It (the printer or the computer?) attempts to break a statement up at a logical point, such as at a colon or semicolon, but it will not break up a block of text that the user has put in quotes. My surmise is that this serves to preserve the integrity of text so that the user can see the exact spacing of quoted characters.

When the PC is in the RUN or DEF modes the printer outputs whenever a PRINT statement is encountered. Under this condition, nothing appears on the PC's display and *program execution is not halted.*

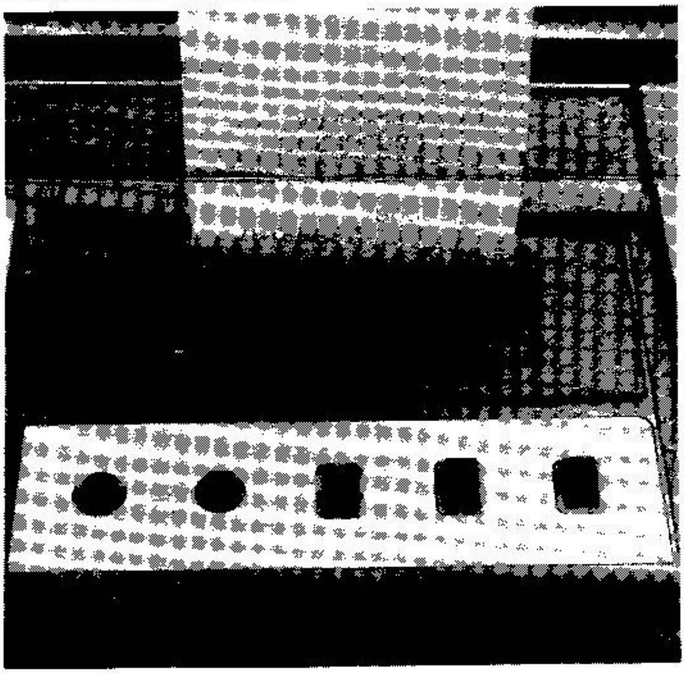
The Bottom Line

A tedious bookkeeping function required by the firm I work for is the preparation of a payment schedule for rental customers using the Rule of 78s. A 60-month amortization schedule required perhaps twenty-five tedious, boring minutes to prepare by hand using an ordinary calculator. Now the PC in combination with its printer does this nasty little job in a few minutes — unattended! No transposed numbers. No illegible characters. No eye fatigue. No sprained fingers.

Now, the PC can perform many new financial and marketing data processing functions that were simply impractical using a one-line display. Yes, it can even perform functions that would be impractical on a 22-line CRT display.

The CE-122 printer by Sharp comes complete with a sturdy vinyl carrying case. There is room in the case for several extra rolls of paper and the tape cassette interface cable. The PC can remain mounted on the CE-122 when it is installed in its case.

Photo *Sharp CE-122 Printer*



Is There More Coming?

On the back of the CE-122 interface there is a cryptic, sealed socket about the size of -- well, how should I put it? — a 9-pin connector! No, there is not a 9-pin connector behind the plate. But, it is my suspicion that this black hole is for memory expansion or, possibly, a disk drive.

The Verdict

The complete package, in the vinyl case, fits compactly in my briefcase while leaving plenty of room for other supplies. Many people from many disciplines can surely benefit from such a portable unit with the ability to prepare its own hardcopy that permits thorough review and analysis of data and computed information.

I can hardly wait for the disk drive!

A Few Programming Tips

Here are a few ideas to assist users of the CE-122 printer.

1. End each program with an automatic paper advance directive so that the operator can simply tear off the finished report without having to manually advance the paper in the printer. This is a nice professional touch. All it takes is the simple directive:

```
FOR Z=1 TO 3:PRINT " ":NEXT Z
```

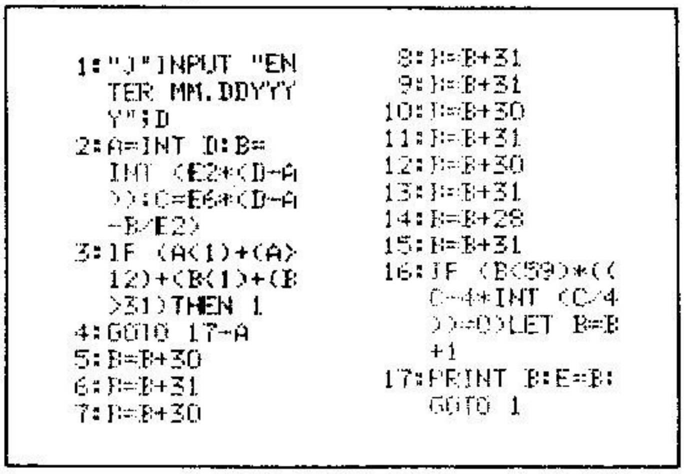
2. If you are using the printer in a program and want to also display information on the PC's LCD display, you can use the trick of using the INPUT statement with a dummy variable. Of course, if the LCD message only needs to be a brief reminder, the PAUSE statement will suffice.

3. You might consider beginning each program that uses the printer with a print statement such as "READY...." so that the user can ensure that the printer is activated. It can be annoying to enter a large amount of data and then discover that output is appearing on the PC's display instead of being printed.

YEAR DAY

The routine presented below calculates the year day. Reference books, such as the *World Almanac and Book of Facts*, often contain a table of year days. Such a table may be used to find the number of days between two dates. Computer systems often use the year day format: YYDDD. In this arrangement YY stands for the last two digits of a year and DDD represents the year day.

Program *Year Day*



```
 1:"J"INPUT "EN
   TER MM.DDYYY
   Y";D
 2:A=INT D:B=
   INT (E2*(D-A
   )):C=E6*(D-A
   -B/E2)
 3:IF (A<1)+(A>
   12)+(B<1)+(B
   >31)THEN 1
 4:GOTO 17-A
 5:B=B+30
 6:B=B+31
 7:B=B+30
 8:B=B+31
 9:B=B+31
10:B=B+30
11:B=B+31
12:B=B+30
13:B=B+31
14:B=B+28
15:B=B+31
16:IF (B<59)*((
   C-4*INT (C/4
   ))=0)LET B=B
   +1
17:PRINT B:E=B:
   GOTO 1
```

To use the program, place the computer in the DEF mode, then type SHIFT J. Key in the date as a two-digit month, a two-digit day-of-the-month, and a four-digit year.

Line 3 in the program does some checking of the input. Line 16 corrects for leap years.

Note: You must subtract 1 for non-leap years ending in 00 if the month is greater than February!

The program returns the year day and stores it in E in the event you want to use the result for further calculations.

Program submitted by: *David Motto, 3639 Roosevelt, Jackson, MI 49203.*

Don't forget to enter
The Hot Steaming Contest
in this issue!

HOT STEAMING CONTEST!

Norlin Rober (who else?) recently wrote about an interesting discovery that came about as a result of his detective work relating to the storage of string variables in the Radio Shack and Sharp PCs. It seems that if a string variable contains words that can be tokenized, then it is possible to store the words as tokens yet display them as complete words. The net result is that it is possible to have string variables that display as more than seven characters!

As Norlin points out, it is first necessary to "tokenize" the character string in the input buffer. Doing so automatically produces an error condition – but you can effectively override this state to end up with a "compacted" string. For example, try entering the following on your PC:

A$=STOPTHENPAUSETOBEEP

Note that no quote marks were used! Press the ENTER key. Immediately the PC will come back with the type 1 error message. Simply ignore the error message by pressing the key that moves the cursor to the left. Now use the cursor to insert quote marks where they belong so that your entry appears as:

A$="STOPTHENPAUSETOBEEP"

Press ENTER again. You have now "tokenized" the message

STOP THEN PAUSE TO BEEP

in the variable A$. However, you won't see it correctly if you attempt to manually display the contents of A$. Typing in A$ followed by the ENTER key results in the string:

STPTBSPAUBEEP

being displayed. However, execution of the programmed directive PRINT A$ as part of a program will cause the entire message:

STOP THEN PAUSE TO BEEP

to be displayed!

The above process of "tokenizing" a character string can be accomplished as part of a numbered statement in a program by following the procedure detailed above, except that with the PC in the PRO mode you will not see an error message when you type in the illegal directive:

10 A$=STOPTHENPAUSETOBEEP

Remember, the directive is illegal because the character string does not have quotes at the beginning and end. Of course, if you were to try and execute the illegal statement, you would get an error message. But you don't do that! What you do is first enter the statement in its illegal form, press the ENTER key, then immediately use the cursor to put quotes around the character string that is to be "tokenized" and press the ENTER key again!

Try playing around with the method on your PC to get the hang of things. Note that a "tokenized" word automatically includes a space at the end. If you include the spaces at the end of each word in the message:

STOP THEN PAUSE TO BEEP

you find you have shoved a 24-character message with room to spare into a string variable that can only hold seven characters. Truth of the matter is, only five characters or "tokens" have actually been stored in the string variable.

Got it?

OK, here is where the fun and the *Hot Steaming Contest!* comes in. Refer to Norlin's table representing what he has discovered about the storage of string variables in the PC. There you can find all of the statement, command and function words that can be tokenized. Now, come up with clever useful messages you might want to display as a programmer. Don't forget, you can combine "tokenized" and untokenized words in one string!

Submit your best messages to the attention of The Editor. To be eligible for consideration your message must display as eight or more characters in length (including spaces) and be stored in just a single string variable. We will judge entries on the basis of originality, usefulness, length, and whatever else

Table *String Variables Storage*

A string variable is stored as eight bytes, the first of which is F5, which identifies the variable as a string. If fewer than seven bytes are contained in the character string that is being stored, the remaining bytes are filled in as 00. The hexadecimal digits are stored in memory in reverse order.

00	Closing "	40	0	80		C0	GRAD
01		41	1	81		C1	PRINT
02		42	2	82	>=	C2	INPUT
03		43	3	83	<=	C3	RADIAN
04		44	4	84	<>	C4	DEGREE
05		45	5	85		C5	CLEAR
06		46	6	86		C6	
07		47	7	87		C7	
08		48	8	88		C8	
09		49	9	89		C9	
0A		4A	.	8A		CA	
0B		4B	E	8B		CB	
0C		4C		8C		CC	
0D		4D		8D		CD	
0E		4E		8E		CE	
0F		4F		8F		CF	
10		50		90	TO	D0	IF
11	SPC	51	A	91	STEP	D1	FOR
12		52	B	92	THEN	D2	LET
13	?	53	C	93		D3	REM
14	!	54	D	94		D4	END
15	#	55	E	95		D5	NEXT
16	%	56	F	96		D6	STOP
17	¥	57	G	97		D7	GOTO
18	$	58	H	98		D8	GOSUB
19	π	59	I	99		D9	CHAIN
1A	√	5A	J	9A		DA	PAUSE
1B	,	5B	K	9B		DB	BEEP
1C	;	5C	L	9C		DC	AREAD
1D	:	5D	M	9D		DD	USING
1E		5E	N	9E		DE	RETURN
1F		5F	σ	9F		DF	
20		60	P	A0	SIN	E0	
21		61	Q	A1	COS	E1	
22		62	R	A2	TAN	E2	
23		63	S	A3	ASN	E3	
24		64	T	A4	ACS	E4	
25		65	U	A5	ATN	E5	
26		66	V	A6	EXP	E6	
27		67	W	A7	LN	E7	
28		68	X	A8	LOG	E8	
29		69	Y	A9	INT	E9	
2A		6A	Z	AA	ABS	EA	
2B		6B		AB	SGN	EB	
2C		6C		AC	DEG	EC	
2D		6D		AD	DMS	ED	
2E		6E		AE		EE	
2F		6F		AF		EF	
30	(	70		B0	RUN	F0	
31	)	71		B1	NEW	F1	
32	>	72		B2	MEM	F2	
33	<	73		B3	LIST	F3	
34	=	74		B4	CONT	F4	
35	+	75		B5	DEBUG	F5	Begin
36	-	76		B6	CSAVE	F6	String
37	*	77		B7	CLOAD	F7	
38	/	78		B8		F8	
39	^	79		B9		F9	
3A		7A		BA		FA	
3B		7B		BB		FB	
3C		7C		BC		FC	
3D		7D		BD		FD	
3E		7E		BE		FE	
3F		7F		BF		FF	

strikes our fancy at the time of judging. The person submitting the best message in the sole opinion of our judges will receive a $25.00 cash honorarium. Two second-place honorariums of $10.00 each will also be awarded. The winning entries and some of the honorable mentions will be published in issue 09 of *PCN*. That gives current subscribers about 45 days to get their thinking caps on and come up with some good ones. The act of submitting an entry grants PCN rights to publish the entry.

When you submit your entry, take a few moments to tell us what kind of programs and materials you like to see the most in *PCN*.

Good luck!

PRINTER PROGRAM CALCULATES SIMPLE INTEREST

The program presented here is specifically designed to utilize the Sharp CE-122 or Radio Shack equivalent printer interface. ~~It calculates and prints out earnings on an investment at fixed,~~ simple interest rates. This program can readily be adapted for a variety of non-compounding investment calculations such as 6-month Treasury-Bills.

This program also serves as an excellent example of a well documented program. Hats off to the program's developer: *Ken Slaughter, 2916 Bangor Avenue, Highland, CA 92346.*

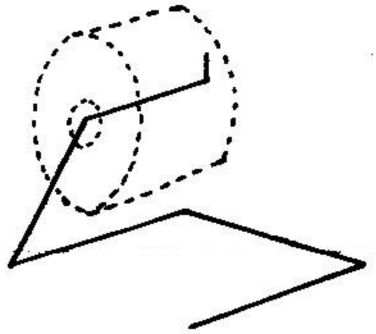
Program *Simple Interest*

```
                    Simple Interest Daily Income Table
          for the TRS-80 Pocket Computer or SHARP Pocket Computer
                              with Printer
                   ----------------------------------------
                    Note: To help in programming numeric
                      images (PRINT USINGs) a reference
                     to the number of characters before
                    and after the decimal point are given
                      as: [IMAGE=8 + 2] where eight "#"
                    preceed the decimal and 2 "#" follow.
                   ----------------------------------------

 1:REM "SIMPLE INTEREST"
 2:REM "<1981> K. SLAUGHTER"
 5:" "CLEAR:
     USING "########.##"              [IMAGE=8 + 2]
10:PAUSE"SIMPLE INTEREST"

[INPUT BLOCK]

20:INPUT"PRINCIPAL? ";P               P=Principal
25:INPUT"% INTEREST? ";I              I=Interest where 16.25=16.25%
26:INPUT"# OF WEEKS? ";W              ~~W=# weeks  Program loops (W*7days)~~
30:INPUT"MONTH? ";M                   M=Month label where 1=Jan, 12=Dec
35:INPUT"DAY? ";D                     D=Day of week
40:INPUT"YEAR? ";Y                    Y=Year

[OUTPUT BLOCK]

50:PRINT"   INVESTMENT"               [3 BLANK LEAD]
52:PRINT"     INCOME"                 [5 BLANK LEAD]
54:PRINT"   PROJECTION"               [3 BLANK LEAD]
56:PRINT" "                           [LINE FEED]
57:PRINT">SIMPLE INTEREST"
58:PRINT"DATA:"
59:PRINT"PRINCIPAL=",P                This section returns data entered
                                      in input block. <Title string>,
                                      <variable> causes numeric values to
                                      be right justified on the paper.

60:PRINT"INTEREST=",I
65:PRINT"DATE: ";M;D;Y                Date printed as MM.DD.YY.
70:PRINT" "                           [LINE FEED]
90:PRINT"DATE      AMOUNT"            [6 BLANK SPACES BETWEEN WORDS]
                                      Column Header

[INITIAL CALCULATIONS]

100: I=I/100/360                      Change I to decimal and divide by
                                      standard 360-day year.
110: L=7W:                            Length of term = 7 (days) x W (weeks)
     A=P*I                            Daily interest = Principal x Interest

115: GOSUB 305                        Find last day of current month
```

CHEAPER PRINTER PAPER

A lot of people quickly latched onto the idea of simply using 1-3/4 inch wide adding machine or cash register tape in place of the small rolls made specifically for the Sharp CE-122. You can either "roll your own" smaller rolls, or, as illustrated by *Allan McCroskery* in the diagram below, you can make a small stand for a whole roll of paper by bending a wire coat hanger.

Figure *Paper Tape Stand*



```
[CALCULATION LOOP: Note, for the most rapid processing possible, do not
 spread-out commands any further than given.]

120: FOR Z=1 TO L                  Loop for <L> days
121: T=T+A:                        T=Final total for month end
     G=G+A                         G=Grand total for term of investment.
122: IF (M>9) * (D>9) GOTO 132     If month and day are 2-digit use format
                                   in line 132
123: IF (M>9) + (D>9) GOTO 130     If month or day are 2-digit use format
                                   in line 130

125: USING:
     PRINT M;D;USING "#########.##";T:  [IMAGE=9+2]
     GOTO 134

130: USING:
     PRINT M;D;USING "########.##";T:   [IMAGE=8+2]
     GOTO 134

132: USING:
     PRINT M;D;USING "#######.##";T     [IMAGE=7+2]

134: D=D+1:IF D>E GOSUB 300            Adds new day.  If D > Last day of
                                       the month change to new month in
                                       gosub 300
135: NEXT Z

[END LOOP]

140: PRINT" ":                     [LINE FEED]
     H$="*******"                  [7 CHARACTERS]
142: GOSUB 400                     Print dividing line for grand total
145: PRINT "GRAND TOTAL=",G        Print grand total
150: GOSUB 400                     Print trailing line for grand total
155: FOR Z=1 TO 3:                 ~~Line feed 3x to advance paper to~~
     PRINT" ":                     tear point in printer.
     NEXT Z

156: END                           ** END **

[SUBROUTINES]

300: M=M+1:
     D=1:
     IF M=13 LET M=1:Y=Y+1         Advance 1 month
301: PRINT" "                      [LINE FEED]
304: T=0                           Reset monthly total accumulator to zero.
305: GOTO 305+M                    Define E as the last day of the month
306: E=31:RETURN                   January
307: E=28:RETURN                   February
308: E=31:RETURN                   March
309: E=30:RETURN                   April
310: E=31:RETURN                   May
311: E=30:RETURN                   June
312: E=31:RETURN                   July
313: E=31:RETURN                   August
314: E=30:RETURN                   September
315: E=31:RETURN                   October
316: E=30:RETURN                   November
317: E=31:RETURN                   December

400: PRINT H$;"  ";H$:             [2 SPACES]
     RETURN

---END PROGRAM LISTING
```

REFINEMENTS

The following correct or update various thoughts, findings and conjectures previously reported in *PCN.*

In Issue 04, page 2, under the heading "MEMORY SURPRISE!!", the second step in the sequence presented should read as follows:

2) Enter this short program with the computer in the PRO mode: 1: OOOOOO (These are letter O's, not zeros.)

When steps 1 through 6 (as corrected herein) are followed, you will find the PC in a mode with apparent expanded memory. Refer to Issue 04 for cautions related to the use of the computer when it has been placed in this mode.

In Issue 05 at the top of the right hand column it was stated that the numeral digits 0 to 9 and the letters A to Z in the accompanying table correspond to ASCII codes. While they follow a pattern similar to ASCII, the PC codes are actually higher in value. Thus, the letter A normally represented as 41 (hexadecimal) in ASCII code, appears in the PC as 51 (hexadecimal).

Also in Issue 05 on page 1, it was reported that the "PC starts looking for labels and line numbers at the beginning of a program." *Norlin Rober* reports that while the PC does begin searching for labels from the start of a program, it does not search for line numbers in the same fashion. He reports that a search for a line number begins at the line containing the GOTO or GOSUB instruction, and then proceeds in the correct direction to find the referenced line. This is confirmed on page 99 of the Radio Shack TRS-80 Pocket Computer Instruction Manual.

Norlin also points out that the use of a GOSUB statement referenced to a program label is not always slower than referring to a line number. It depends on individual cases. A label on the first line of a program is reached immediately. A transfer across a 1400-byte program to a line number can take over 2.5 seconds! Furthermore, it appears that a transfer to a line number only takes about 80 percent as much time to go in the reverse direction (from higher line number to lower) than to go in the forward direction!

Finally, Norlin says a few modifications need to be made to the Tic-Tac-Toe program that appeared on page 5 of Issue 05. (This is in addition to the modifications indicated on page 6 of Issue 06.) Norlin says he has now checked every one of the possible 457 games that can be played. He also says he has had enough of the game, in the development of the program, to last him quite some time!

To put the program in tip-top shape, first change the statement I=23 in line 1 to read: I=27.

Then, alter lines 25 through 27 to read as follows:

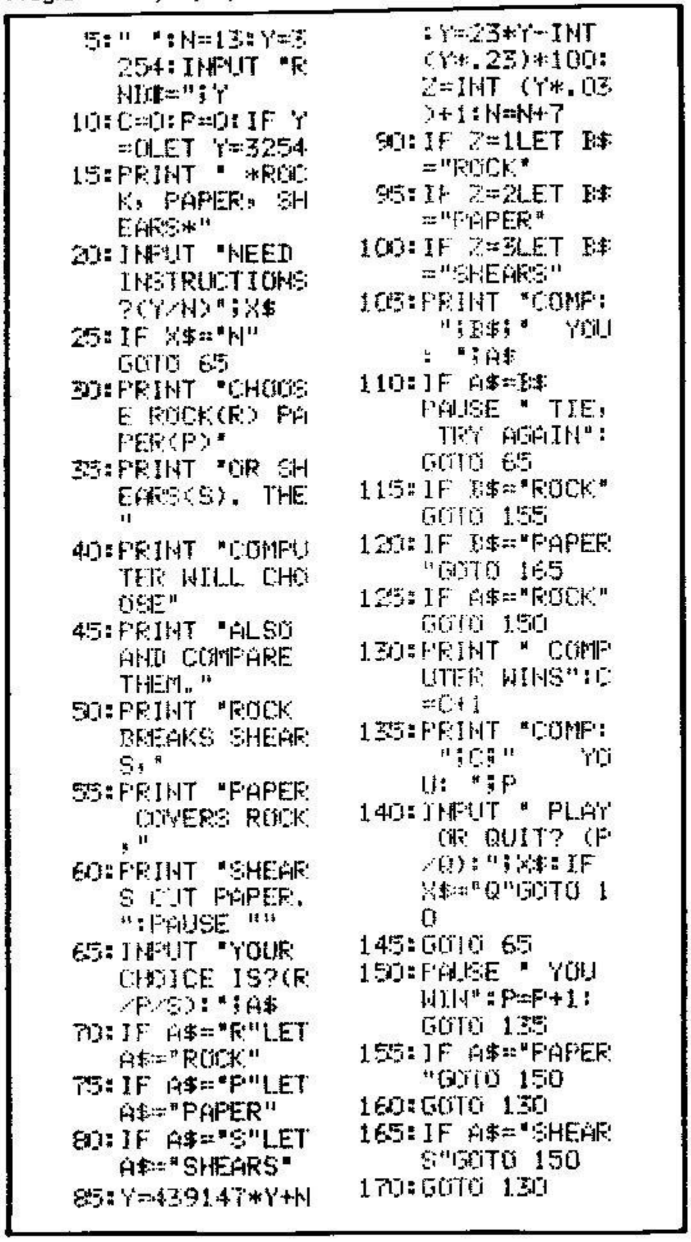
```
25 Y=10-B+(5-A)*(B=9:GOSUB 10:IF ABS(A-B+2)=1 LET
   Y=1+(A+3)*(X=9:GOTO 13
26 IF A+Y=6 IF 1<>ABS(A+X-8 LET Y=2A-1:GOTO 13
27 Y=10-X+(X=9)*(1+ABS(B-3:GOTO 14
```

Don't forget to enter
The Hot Steaming Contest
in this issue!

ROCK, PAPER, SHEARS

Remember the old game you used to play during recess in school? Here is *Gary Heidbrink's* version of the game for a pocket computer. Load it into your PC the next time you need to keep your children occupied for a while.

Program *Rock, Paper, Shears*

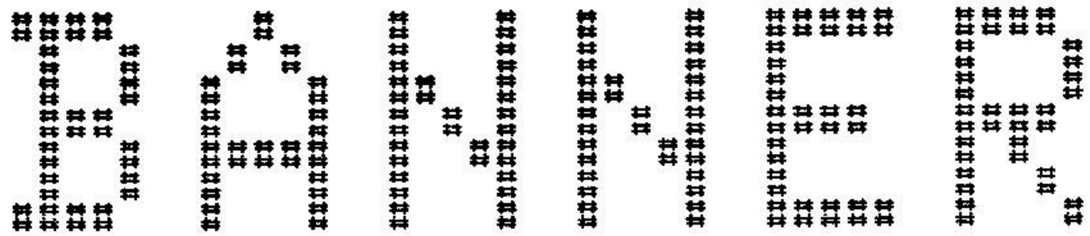


```
  5:"  ":N=13:Y=3
    254:INPUT "R
    NID="; Y
 10:C=0:P=0:IF Y
    =0LET Y=3254
 15:PRINT " *ROC
    K, PAPER, SH
    EARS*"
 20:INPUT "NEED
    INSTRUCTIONS
    ?(Y/N)";X$
 25:IF X$="N"
    GOTO 65
 30:PRINT "CHOOS
    E ROCK(R) PA
    PER(P)"
 35:PRINT "OR SH
    EARS(S). THE
    "
 40:PRINT "COMPU
    TER WILL CHO
    OSE"
 45:PRINT "ALSO
    AND COMPARE
    THEM."
 50:PRINT "ROCK
    BREAKS SHEAR
    S,"
 55:PRINT "PAPER
     COVERS ROCK
    ,"
 60:PRINT "SHEAR
    S CUT PAPER.
    ":PAUSE ""
 65:INPUT "YOUR
    CHOICE IS?(R
    /P/S):";A$
 70:IF A$="R"LET
    A$="ROCK"
 75:IF A$="P"LET
    A$="PAPER"
 80:IF A$="S"LET
    A$="SHEARS"
 85:Y=439147*Y+N
    :Y=23*Y-INT
    (Y*.23)*100:
    Z=INT (Y*.03
    )+1:N=N+7
 90:IF Z=1LET B$
    ="ROCK"
 95:IF Z=2LET B$
    ="PAPER"
100:IF Z=3LET B$
    ="SHEARS"
105:PRINT "COMP:
     ";B$;"  YOU
    : ";A$
110:IF A$=B$
    PAUSE " TIE,
     TRY AGAIN":
    GOTO 65
115:IF B$="ROCK"
    GOTO 155
120:IF B$="PAPER
    "GOTO 165
125:IF A$="ROCK"
    GOTO 150
130:PRINT " COMP
    UTER WINS":C
    =C+1
135:PRINT "COMP:
     ";C;"    YO
    U: ";P
140:INPUT " PLAY
     OR QUIT? (P
    /Q):";X$:IF
    X$="Q"GOTO 1
    0
145:GOTO 65
150:PAUSE " YOU
    WIN":P=P+1:
    GOTO 135
155:IF A$="PAPER
    "GOTO 150
160:GOTO 130
165:IF A$="SHEAR
    S"GOTO 150
170:GOTO 130
```

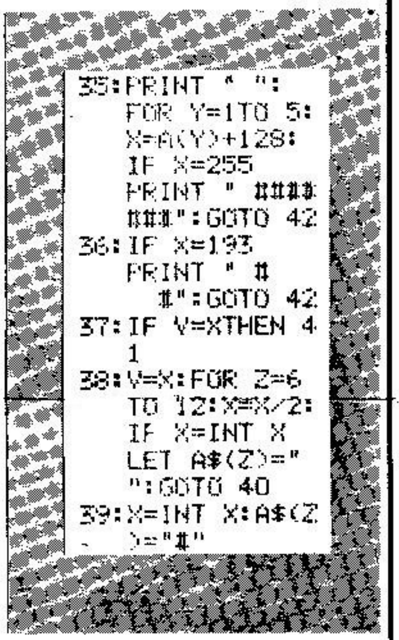
PROGRAM MAKES BANNERS WITH PRINTER

Brian Peterson, 6807 N. Sheridan Road, Apt. 520, Chicago, IL 60626, was one of several readers to submit programs that print banner-type messages. To use his program, execute the RUN command, then input each character *individually*. Press the ENTER key after each input. The program will print characters whenever seven characters have been inputted or when the ENTER key is pressed with no other input.

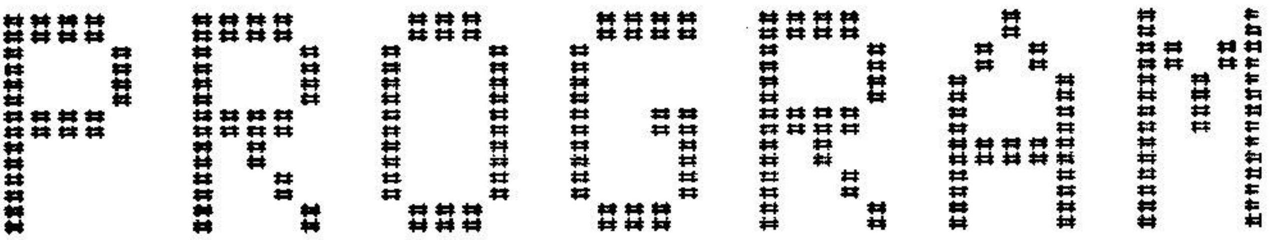
If you want smaller characters use lines 35 through 39 as shown in the shaded inset.



```
 1:CLEAR :FOR T
   =13TO 19:
   INPUT "L>",A
   $(T)
 2:IF A$(T)NEXT
   T
 3:FOR W=13TO T
   :U$=A$(W):
   GOSUB 4:NEXT
   W:BEEP 10:
   GOTO 1
 4:IF U$="A"LET
   A=31:B=36:C=
   68:D=36:E=31
   :GOTO 35
 5:IF U$="B"LET
   A=65:B=127:C
   =73:D=73:E=5
   4:GOTO 35
 6:IF U$="C"LET
   A=62:B=65:C=
   65:D=65:E=34
   :GOTO 35
 7:IF U$="D"LET
   A=65:B=127:C
   =65:D=65:E=6
   2:GOTO 35
 8:IF U$="E"LET
   A=127:B=73:C
   =73:D=73:E=6
   5:GOTO 35
 9:IF U$="F"LET
   A=127:B=72:C
   =72:D=72:E=6
   4:GOTO 35
10:IF U$="G"LET
   A=62:B=65:C=
   65:D=73:E=78
   :GOTO 35
11:IF U$="H"LET
   A=127:B=8:C=
   8:D=8:E=127:
   GOTO 35
12:IF U$="I"LET
   A=65:B=65:C=
   127:D=65:E=6
   5:GOTO 35
13:IF U$="J"LET
   A=7:B=1:C=65
   :D=126:E=64:
   GOTO 35
14:IF U$="K"LET
   A=127:B=8:C=
   20:D=34:E=65
   :GOTO 35
15:IF U$="L"LET
   A=127:B=1:C=
   1:D=1:E=1:
   GOTO 35
16:IF U$="M"LET
   A=127:B=32:C
   =24:D=32:E=1
   27:GOTO 35
17:IF U$="N"LET
   A=127:B=16:C
   =8:D=4:E=127
   :GOTO 35
18:IF U$="O"LET
   A=62:B=65:C=
   65:D=65:E=62
   :GOTO 35
19:IF U$="P"LET
   A=127:B=72:C
   =72:D=72:E=4
   8:GOTO 35
20:IF U$="Q"LET
   A=62:B=65:C=
   69:D=66:E=61
   :GOTO 35
21:IF U$="R"LET
   A=127:B=72:C
   =76:D=74:E=4
   9:GOTO 35
22:IF U$="S"LET
   A=50:B=73:C=
   73:D=73:E=38
   :GOTO 35
23:IF U$="T"LET
   A=64:B=64:C=
   127:D=64:E=6
   4:GOTO 35
24:IF U$="U"LET
   A=126:B=1:C=
   1:D=1:E=126:
   GOTO 35
25:IF U$="V"LET
   A=112:B=12:C
   =3:D=12:E=11
   2:GOTO 35
26:IF U$="W"LET
   A=127:B=2:C=
   12:D=2:E=127
   :GOTO 35
27:IF U$="X"LET
   A=99:B=20:C=
   8:D=20:E=99:
   GOTO 35
28:IF U$="Y"LET
   A=96:B=16:C=
   15:D=16:E=96
   :GOTO 35
29:IF U$="Z"LET
   A=67:B=69:C=
   73:D=81:E=97
   :GOTO 35
30:IF U$="!"LET
   A=0:B=125:C=
   0:D=0:E=0:
   GOTO 35
31:IF U$="¥"LET
   A=¥¥¥:B=¥¥¥:
   C=¥¥¥:D=¥¥¥:
   E=¥¥¥:GOTO 3
   5
32:IF U$=" "FOR
   Y=1TO 6:
   PRINT " ":
   NEXT Y:
   RETURN
33:END
35:PRINT " ":
   FOR Y=1TO 5:
   X=A(Y)+128:
   IF X=255
   PRINT " ####
   #########":
   GOTO 42
36:IF X=193
   PRINT " ##
           ##":
   GOTO 42
37:IF V=XTHEN 4
   1
38:V=X:FOR Z=6
   TO 12:X=X/2:
   IF X=INT X
   LET A$(Z)="
    ":GOTO 40
39:X=INT X:A$(Z
   )="##"
40:NEXT Z
41:PRINT " ";F$
   ;G$;H$;I$;J$
   ;K$;L$
42:NEXT Y
43:PRINT " ":
   RETURN
```



```
35:PRINT " ":
   FOR Y=1TO 5:
   X=A(Y)+128:
   IF X=255
   PRINT " ####
   ###":GOTO 42
36:IF X=193
   PRINT " #
     #":GOTO 42
37:IF V=XTHEN 4
   1
38:V=X:FOR Z=6
   TO 12:X=X/2:
   IF X=INT X
   LET A$(Z)="
   ":GOTO 40
39:X=INT X:A$(Z
   )="#"
```



FROM THE WATCH POCKET

Well, here it is — the end of August — and still no sign of the highly touted Panasonic/Quasar HHC. It looks as though Casio is going to be the next firm to actually begin delivering a HHC.

Rumors are now beginning to circulate that two of the big calculator firms in the U.S., namely Texas Instruments and Hewlett-Packard, have PCs under development. To date, I have not been able to obtain any confirmation of these rumors from representatives of the organizations. However, as we all know, that doesn't mean they aren't working on such units!

Interface Age magazine has started a regular column devoted to the use of pocket computers. It is written by *Bob McElwain.* In the September issue Bob discusses a pseudo-random number generator that is theoretically capable of producing about 500,000 numbers before a pattern of repetition sets in.

One of our readers reported that he was able to make his PC printer (assumed to be a CE-122 or equivalent) skip lines in a listing by temporarily placing the PRINT switch in the off position. When I tried this on my unit, the computer would terminate the listing process. However, I did find that turning the printer unit's POWER switch to the off position resulted in the computer continuing the listing operation. However, since this action apparently interrupts the "handshaking" process, the transmission takes place at a much faster rate during the time the power switch is in the off position. You might try this on your own unit to see what happens. With a little practice, you can operate the switch as desired to skip over sections of a listing. This procedure could save quite a bit of time if you only needed to list the last few lines in a program. Remember, the transmission rate is much faster with the power switch in the off position, so don't leave the switch off more than a few seconds at a time until you get a feel for the faster transmission rate.

A number of people have asked for information related to the transferring of data from/to a bigger computer and a PC. It has recently been reported that an outfit, *Green River Systems, P.O. Box 552, Auburn, WA 98002,* has developed a method whereby data recorded on a cassette tape using a Radio Shack TRS-80 PC (and presumably the Sharp PC-1211) can be read by a Radio Shack TRS-80 Model I or III. Write directly to that company for more information.

Radio Shack has announced that it will now be offering a carrying case that can accomodate its PC mounted on the printer interface or the Radio Shack Minisette-9 cassette tape recorder. The carrying case will also hold several accessories such as paper and ribbons. The case is padded for protection of the equipment and features a handle that can also serve as a loop for carrying on a belt. The current price of the case is $14.95.

If you like to save programs for your PC on cassette tapes, a reliable source has indicated that *AB Computers, 252 Bethlehem Pike, Colmar, PA 18915,* has been selling C-10 cassettes. They are reported to be of good quality and the unit price is just 50 cents each when purchased in lots of 50. You might call (215) 822-7727 to check on availability or write directly to them for a catalog.

Wick Hutchison of Wickenburg, Arizona, reports he has been very successful at restoring the ink on the cartridges used in the Radio Shack or Sharp PC printers. He states that he simply purchased a small tube of *Sanfords No. 638 Black Ink* that is commonly sold for use in numbering machines and check writers. The tube has a very small nozzle with which a minute amount of ink can be applied directly to the ribbon. *It is very important that you only apply a tiny (infinitesimal!) amount of ink to the ribbon!* You can then use a ball point pen or a pencil to turn the tiny roller at the left hand end of the cartridge and thus advance the ribbon in order to distribute the ink throughout its length. (This is done, of course, with the ribbon cartridge removed from the printer unit!) Wick reports that it should be possible to re-ink a cartridge quite a few times — enough to print 20 or more rolls of paper — before the ribbon is likely to start fraying.

The July issue of **80 Microcomputing**, page 243, carries a program for the Radio Shack TRS-80 and Sharp PC-1211 PCs that is entitled *Nag Analysis.* If you are a dedicated handicapper you most likely will want to check this algorithm out!

In a few weeks randomly selected readers of this publication will receive a survey questionaire. I urge those of you who receive it to take the few minutes necessary to respond to the survey. It could lead to improved PC products for all of us in the future. Since the survey will be conducted by an organization that is independent of *PCN,* please make sure that you use the postage-paid return envelope that will accompany the survey questionaire. That way the results will go directly to the polling tabulation center.

With the summer months gone *PCN* will resume publication on a monthly basis. — *Nat Wadsworth, Editor*

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others? Have you developed application programs that you would like to share with fellow pocket computer enthusiasts? Have you had experience with PC products or programs that others might find of value?

If so, write it up! If your submission is accepted for publication you will earn a cash reward as well as have the satisfaction of knowing you are helping other PC users.

We prefer that program listings be supplied on audio cassette tape so that they may be easily reviewed. We can also then make listings of accepted programs directly from the tape to avoid transcription errors. If we cannot use your material, we will endeavor to promptly return it *provided you include a self-addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483

POCKET COMPUTER NEWSLETTER

© Copyright 1981 — Issue 08

THE SHAPE OF THE FUTURE?

The accompanying photograph shows a recently announced programmable calculator produced by Hewlett-Packard. While *PCN* likes to clearly delineate between "calculators" and "pocket computers" and normally does not report on the former, an exception is being made in this case. But, this exception is for a very particular reason.

As anyone who has followed Hewlett-Packard's line of calculators even on a casual basis can testify, the *shape* of the newly announced calculator is what is worthy of mention. It is radically different than the style(s) produced by the firm in the past. And, coincidently, it suddenly takes on a shape that will instantly be recognizable to most current readers of *PCN*.

The fact is, a number of industry speculators believe that world famous HP is shaping things up for a future entry into the pocket computer market. You need to get the shape of a PC into a basically horizontal format so that all of those necessary alphanumeric keys can be handled comfortably. And, it is a good idea to give the display some extra length over that generally allowed by the narrow width of the old style calculators.

It is even a good marketing idea to give customers some time to warm up to (and for the marketing department to "test the waters" around) a new product format before springing something as revolutionary as a *pocket computer* on the general public!

So there may be a good deal of truth to the rumors that these new HP calculators are in the general shape of what will eventually become full-fledged HP pocket computers.

How close to what *PCN* considers a true pocket computer is this new HP calculator? Real close. The unit features a liquid-crystal display. It has continuous memory that is retained even when the power (display) is turned off. It boasts of a number of "programming" capabilities, including the all important program "editing" functions. About all it needs to be classified as a pocket computer is a more sophisticated programming language, such as BASIC.

Indeed, the new programmable calculator has many features associated with currently available pocket computers such as the use of program labels which may be addressed by individual keys, the capability of using indirect addressing in a program, conditional tests on data and calculations, the availability of programming flags, nesting of subroutines up to four levels, and a memory extensive enough to permit some 200 "lines" (well, *short lines*) of program instructions. The editing features enable the insertion and deletion of individual program lines.

The calculator shown in the photograph, the HP-11C, comes with a full set of scientific functions that are key-selectable. In addition to commonly used math, engineering and statistics functions, there are special functions such as hyperbolics and a random-number generator.

The HP-11C is housed in a rugged case that is considered by Hewlett-Packard to be "slim line" and appropriate in size and weight for comfortable carrying in a shirt pocket.

Another somewhat unique feature of the new unit is its ability to test itself. By pressing the appropriate keys, the user can direct the machine to conduct tests of its keyboard and calculating circuits! This capability can be comforting to the frustrated programmer who becomes convinced that the unit is malfunctioning! Incidently, the self-test capability does not interfere with data or programs stored in memory.

In addition to the HP-11C, which is primarily designed for scientists and engineers, there is a second unit, similar in appearance, that caters to financial analysts and businesspersons.

The HP-11C is reported to have a list price of $135.00 and the similar financial model (HP-12C) a slightly higher price of $150.00.

Will this be the shape of a *real* pocket computer produced by Hewlett-Packard? Look for the answer itself to appear during 1982!

Photo *Hewlett-Packard's HP-11C Shirt Pocket Calculator*

SUPER-WUMPUS

In 1973 Gregory Yob developed a computer game called *Wumpus.* The game quickly caught on and became one of the early classic games of personal computer users.

The objective in the game of Wumpus is to search a cave complex that contains twenty rooms connected by tunnels in order to locate and destroy a ferocious Wumpus. The cave complex contains two bats. Enter a cave containing a bat and you will be randomly relocated to another room. Two caves contain pits. Fall into a pit and it is all over! And, of course, there is the Wumpus. Meet the creature in a cave and it is all over for you.

When you are in a cave that is right next to a room containing any of the hazards, you will receive a warning message such as:

YOU FEEL A DRAFT.
YOU HEAR WINGS.
YOU SMELL A WUMPUS!

To destroy the Wumpus you use arrows which have the capability of darting through from one to five caves in a single shot.

The version of the game presented here for the pocket computer has several additional features that increase interest and difficulty.

First of all, you start with one to four arrows depending on which of twenty possible levels of play you select. In levels 1 through 19 you have a chance of finding new arrows when you have used your initial allotment.

Above level 2 earthquakes will move the pits about randomly to different caves. The warning message:

I FEEL AN EARTHQUAKE!

will let you know when a quake has occured.

If you try to shoot an arrow into a cave that is not connected by a tunnel or if you shoot into your own cave, you will, naturally, shoot yourself. What a way to go.

An arrow will only destroy the Wumpus if he is in the final cave of the arrow's flight path.

An arrow that passes through a cave in which the Wumpus is hiding, will quite likely cause the creature to shift caves. It thus may be wise to back up a room when you smell the creature and fire from a distance so that the Wumpus cannot move in on you.

Refer to the accompanying cave diagram to see how the caves are connected by tunnels. You may want to keep track of the progress of a game by drawing your own map as you proceed on a mission.

An accompanying table also shows the chances of finding another arrow when your initial supply is gone as well as the chance that an earthquake will occur, according to the level of play that has been selected.

Variables have been assigned so that those utilized the most have the fastest execution. All of memory is utilized so any user additions or changes will require deletions elsewhere in the program. A likely area for such modifications would be in the PRINT statements.

Line 90 will require a double entry operation in order to fit it all on one line unless you abbreviate statement terms such as GOSUB (use GOS.) and PAUSE (by using PA.). Line 175 also requires a double entry operation. To perform a double entry, first enter as much of the line as possible and press the ENTER key. Then use the cursor-right key to go to the end of the line and enter the rest of the line.

I execute most of my programs with the PC in the DEF mode. When in this mode, the program can be started by using SHIFT S. If you like to execute program with the PC in the RUN mode then you can eliminate the "S" label from line 35 and add the line:

```
1 GOTO 35
```

This modification adds a requirement for two steps of memory. To obtain the necessary space, I suggest you change the PRINT statement in line 85 to read:

"YOU FIND 1 ARROW."

Subroutines were placed at the beginning of the program. Line 5 contains a random number generating routine. You might want to use this routine in other programs. You can get any range of numbers desired for H by defining

H=(G*N/100)

whereby H will represent a random number having a value between 0 and N-1. Adding 1 to this value gives you a range of 1 to N.

Lines 10 through 30 determine the adjacent rooms.

Line 45 asks for the desired level of play and defaults to level 6.

Lines 50 through 75 set up the game and ensure that the player, the Wumpus, bats and pits are initialized to reside in separate caves.

Good luck and happy hunting!

Super-Wumpus was submitted by: *Gary Heidbrink, Box 24, FPO New York, NY 09593.*

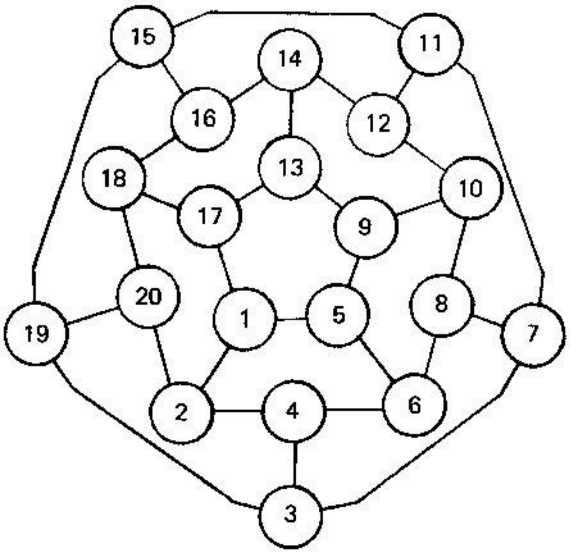
[P.S. I think Mr. Heidbrink has done a super job of programming this game on a PC! — N.W.]

VARIABLES USAGE TABLE

A(1)	First cave being shot
A(2)	Second cave being shot
A(3)	Third cave being shot
A(4)	Fourth cave being shot
A(5)	Fifth cave being shot
F	Used in random number generator, prevents cycling
G	Main random number routine variable
H	Generated random number
I	Number of arrows
J	Number of turns taken
K	Prevents bats from taking you after dropping you off
L$	Holds responses from input statements
M	Location of the Wumpus
N	First pit location
O	Second pit location
P	First bat location
Q	Second bat location
R	Data variable
S	Number of caves being shot
T	FOR/NEXT variable
U	Data variable
V	Level of difficulty
W	Current room being analyzed
X	First adjacent room
Y	Second adjacent room
Z	Third adjacent room

Program *Super-Wumpus*

```
  5:F=F+1:G=4391
    47*G+F:G=23*
    G-INT (G*.23
    )*100:H=INT
    (G*.2)+1:
    RETURN
 10:X=W:Y=W:Z=W:
    IF W/2<>INT
    (W/2)LET X=X
    +3:GOSUB 15
 15:X=X-1:IF X<1
    LET X=X+20
 20:Y=Y+2:IF Y>2
    0LET Y=Y-20
 25:Z=Z-2:IF Z<1
    LET Z=Z+20
 30:RETURN
 35:"S":F=1:G=32
    54:INPUT "RN
    D #: ";G
 40:PRINT "
     SUPER-WUMPU
    S"
 45:V=6:INPUT "L
    EVEL (1-20)
    ";V:IF (V<1)
    +(V>20)GOTO
    45
 50:J=0:K=0:I=4-
    INT (V/6):
    GOSUB 5:M=H
 55:GOSUB 5:N=H:
    IF N=MGOTO 5
    5
 60:GOSUB 5:O=H:
    IF (O=M)+(O=
    N)GOTO 60
 65:GOSUB 5:P=H:
    IF (P=M)+(P=
    N)+(P=O)GOTO
    65
 70:GOSUB 5:Q=H:
    IF (Q=M)+(Q=
    N)+(Q=O)+(Q=
    P)GOTO 70
 75:GOSUB 5:W=H:
    IF (W=M)+(W=
    N)+(W=O)+(W=
    P)+(W=Q)GOTO
    75
 80:BEEP 1:PAUSE
    "YOU ARE IN
    ROOM ";W
 85:GOSUB 5:IF (
    I=0)IF (H>V)
    LET I=1:BEEP
    1:PAUSE "YOU
     FOUND AN AR
    ROW."
 90:IF H<V/2BEEP
    2:PAUSE "YOU
     FEEL AN EAR
    THQUAKE!":
    GOSUB 5:R=N:
    N=H:IF F/2=
    INT (F/2)LET
    O=R
 95:IF N=MBEEP 4
    :PAUSE "WUMP
    US ATE YOU."
    GOTO 40
100:IF (W=N)+(W=
    O)BEEP 4:
    PRINT "YOU F
    ELL INTO A P
    IT.":GOTO 40
105:IF KGOTO 120
110:IF W=PLET P=
    Q:Q=W
115:IF W=QBEEP 2
    :PAUSE "BAT
    GRABBED YOU.
    ":K=1:GOSUB
    5:Q=H:W=H:
    GOTO 80
120:GOSUB 10
125:IF (X=M)+(Y=
    M)+(Z=M)BEEP
    3:PAUSE "YOU
     SMELL A WUM
    PUS!"
130:IF (X=N)+(Y=
    N)+(Z=N)+(X=
    O)+(Y=O)+(Z=
    O)BEEP 1:
    PAUSE "YOU F
    EEL A DRAFT.
    "
135:IF (X=P)+(Y=
    P)+(Z=P)+(X=
    Q)+(Y=Q)+(Z=
    Q)BEEP 1:
    PAUSE "YOU H
    EAR WINGS."
140:PRINT "TUNNE
    LS GO TO:";
    USING "###";
    X;Y;Z
145:INPUT "MOVE
    OR SHOOT? (M
    /S): ";L$:J=
    J+1:IF L$="M
    "GOTO 160
150:IF L$="S"
    GOTO 170
155:J=J-1:GOTO 1
    40
160:INPUT "ENTER
    ING WHICH CA
    VE? ";H:IF (
    H=X)+(H=Y)+(
    H=Z)LET W=H:
    K=0:GOTO 80
165:PAUSE "YOU C
    ANNOT GO THE
    RE.":GOTO 15
    5
170:IF I=0PAUSE
    "SORRY, NO A
    RROWS.":GOTO
    155
175:INPUT "THRU
    HOW MANY CAV
    ES? ";S:IF (
    S>5)+(S<1)
    PRINT "CAN S
    HOOT TO 1-5
    CAVES.":GOTO
    155
180:I=I-1:FOR T=
    1TO S:INPUT
    "THRU WHICH
    CAVE? ";R:A(
    T)=R:NEXT T
185:R=W:FOR T=1
    TO S:U=A(T):
    IF R<>UIF (X
    =U)+(Y=U)+(Z
    =U)LET W=U:
    GOSUB 10:
    NEXT T:GOTO
    195
190:BEEP 4:PRINT
    "MISFIRE! SH
    OT SELF.":
    GOTO 40
195:IF R=MBEEP 5
    :PRINT "SHOT
     WUMPUS IN";
    J;" TURNS.":
    GOTO 40
200:W=M:GOSUB 10
    :GOSUB 5:T=2
    +INT (H/4):
    M=A(T)
205:W=R:GOTO 80
```



LEVEL OF PLAY	STARTING ARROWS	CHANCE TO FIND ARROW	CHANCE OF QUAKE
1	4	95%	— —
2	4	90%	— —
3	4	85%	05%
4	4	80%	05%
5	4	75%	10%
6	3	70%	10%
7	3	65%	15%
8	3	60%	15%
9	3	55%	20%
10	3	50%	20%
11	3	45%	25%
12	2	40%	25%
13	2	35%	30%
14	2	30%	30%
15	2	25%	35%
16	2	20%	35%
17	2	15%	40%
18	1	10%	40%
19	1	05%	45%
20	1	— —	45%

PLOTTER PROGRAM

The program shown in the accompanying listing plots a graph of whatever function is specified in line 50. Note that in line 50, Y is calculated based on the value of X.

The program requests the minimum and maximum values of X and Y that are to be plotted as well as the number of intervals to be used. The values of X used in the program are determined by subdividing the interval between MIN X and MAX X as specified.

The graph is printed lengthwise along the tape. Minimum and maximum values of X and Y are printed at the start of the graph.

If a plotted point lies outside the specified range (because Y goes beyond the MIN or MAX indicated), a blank line is printed in order to maintain proper spacing.

Some examples that give interesting results include:

Sine curve (illustrated)
Line 50: Y=SIN X (DEGree mode!)
MIN X=0; MAX X=360; MIN Y=-1; MAX Y=1; 15 intervals

Normal probability curve
Line 50: Y=EXP(-XX/2)/√2π
MIN X=-3; MAX X=3; MIN Y=0; MAX Y=.4; 16 intervals

Parabola
Line 50: Y=XX
MIN X=-2; MAX X=2; MIN Y=0; MAX Y=4; 8 intervals

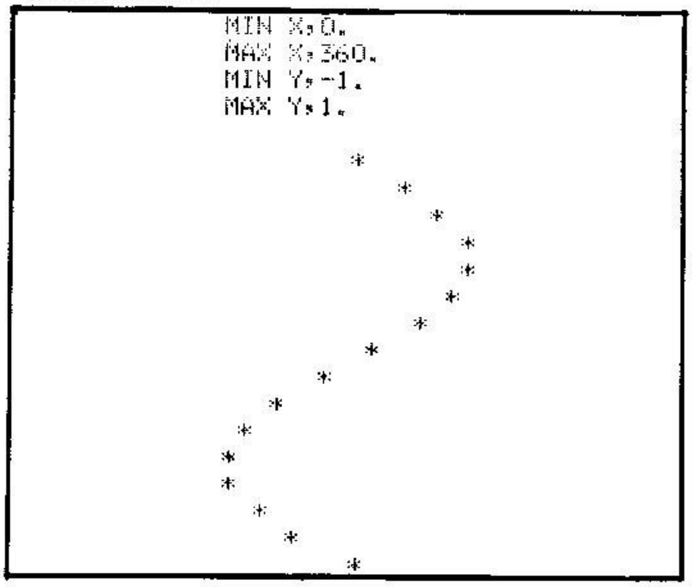
Damped harmonic motion
Line 50: Y=EXP-.2X*COS X (RADian mode!)
MIN X=0; MAX X=12; MIN Y=-1; MAX Y=1; 12 intervals

Semicircle (appears as a semi-ellipse)
Line 50: Y=√(1-XX)
MIN X=-1; MAX X=1; MIN Y=0; MAX Y=1; 12 intervals

Wave with harmonic
Line 50: Y=SIN X + SIN 2X (DEGree mode)
MIN X=0; MAX X=360; MIN Y=-1.75; MAX Y=1.75; 12 intervals

Program submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

Example *Sine curve*



```
MIN X,0.
MAX X,360.
MIN Y,-1.
MAX Y,1.
```

Program *Plotter*

```
10:INPUT "MIN X
   ?";A,"MAX X?
   ";B,"MIN Y?"
   ;C,"MAX Y?";
   D,"NO OF INT
   ERVALS?";N
15:PRINT "MIN X
   ,";A:PRINT "
   MAX X,";B:
   PRINT "MIN Y
   ,";C:PRINT "
   MAX Y,";D:
   PRINT " "
20:FOR Z=0TO N:
   X=A+Z/N*((B-
   A
50:Y=SIN X
75:P=INT (1.5+1
   5*(Y-C)/(D-C
   :GOSUB 80:
   NEXT Z:END
80:R$="        ":
   S$=R$:Q=INT
   .16P:IF PIF
   P<17GOTO 82+
   Q
81:PRINT R$:
   RETURN
82:R=0
83:S=0
84:GOTO 84+P-6Q
85:PRINT R$;S$;
   "*":RETURN
86:PRINT R$;S$;
   " *":RETURN
87:PRINT R$;S$;
   "  *":RETURN
88:PRINT R$;S$;
   "   *":
   RETURN
89:PRINT R$;S$;
   "    *":
   RETURN
90:PRINT R$;S$;
   "     *":
   RETURN
```

MULTIPLE-STRIP PLOTTER PROGRAM

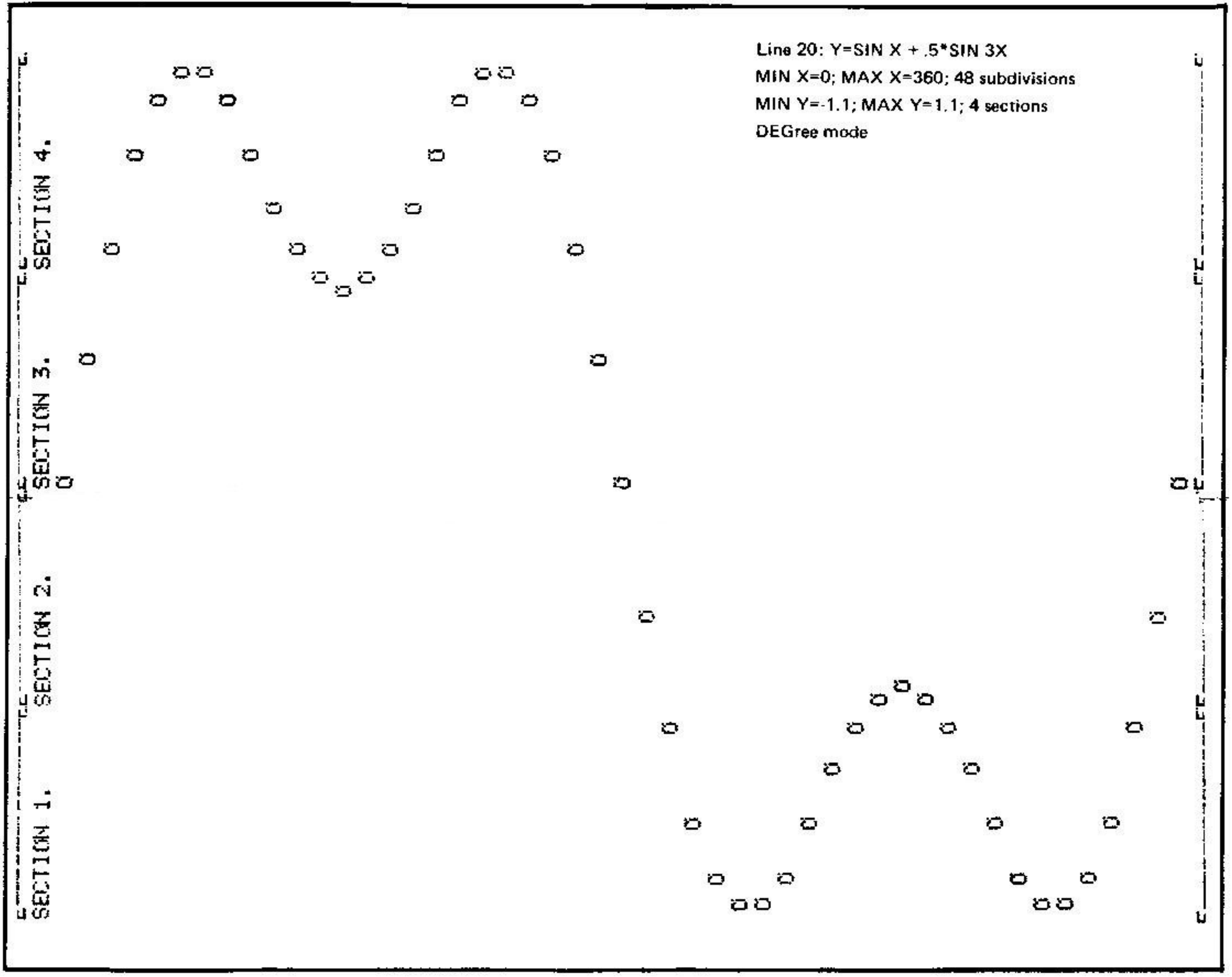
This program produces a larger graph than the previously described plotter program by utilizing several strips of paper. The number of strips or sections that may be utilized are specified by the user. After the individual sections have been plotted, the left edge of each section (except the first) can be trimmed. Then the sections are pasted alongside one another. The program provides markings on the strips to aid in cutting and

A spray adhesive such as *Scotch Spray Mount* (catalog number 6065) considerably simplifies the task of mounting the strips.

Program *Multiple-Strip Plotter*

```
10:INPUT "MIN X
   ?";A,"MAX X?
   ";B,"NO OF S
   UBDIVISIONS?
   ";N,"MIN Y";
   C,"MAX Y";D
11:INPUT "NO OF
    SECTIONS?";
   M:FOR J=1TO
   M+1:PRINT "E
   ----------------
   --E":IF J>M
   END
12:PRINT "SECTI
   ON ";J:FOR I
   =0TO N:X=A+(
   B-A)I/N
65:P=INT (17.5-
   16J+(16M-1)*
   (Y-C)/(D-C:
   GOSUB 90:
   NEXT I:NEXT
   J
90:Q$="        ":
   R$=Q$:S=INT
   .16P:IF PIF
   P<17GOTO 92+
   S
91:PRINT Q$:
   RETURN
92:Q=0
93:R=0
94:GOSUB 94+P-6
   S:PRINT Q$;R
   $;S$;"0":
   RETURN
95:S=0:RETURN
96:S$=" ":
   RETURN
97:S$="  ":
   RETURN
98:S$="   ":
   RETURN
99:S$="    ":
   RETURN
100:S$="     ":
   RETURN
```

Example *Multiple-Section Plot*



Line 20: Y=SIN X + .5*SIN 3X
MIN X=0; MAX X=360; 48 subdivisions
MIN Y=-1.1; MAX Y=1.1; 4 sections
DEGree mode

The user must specify a routine for the calculation of the variable Y as a function of X in lines 20 through 84.

The symbol used for plotting is specified in line 94 of the program. The letter O is used in the example.

A graph measuring approximately 5.5 by 7 inches in size is created, for example, by specifying 4 sections and using 48 subdivisions.

This program is a *Norlin Rober* creation.

STRING PRINTER

To use your CE-122 or Radio Shack PC Printer as a message unit, load this program into your pocket computer:

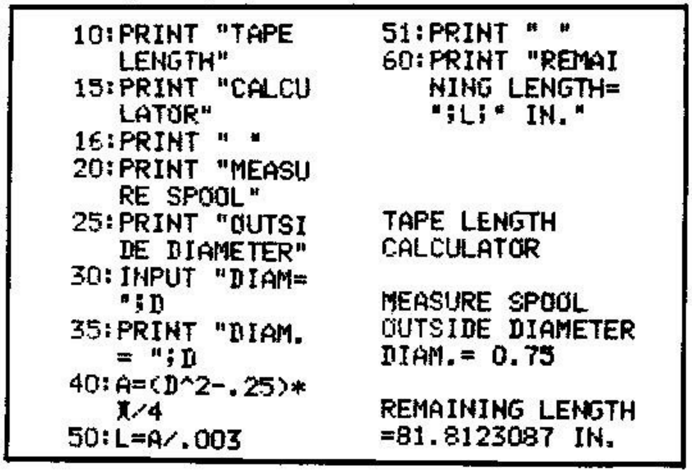
```
100 CLEAR:A=2
110 INPUT "STRING? ";A$(A):A=A+1:GOTO 110
120 PRINT B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;M$;N$
130 GOTO 100
```

Execute RUN. Key in up to 7 characters, press ENTER, repeat as required. Pressing ENTER with no other input causes the message to be printed. Thanks to: *David Motto, 3639 Roosevelt Circle, Jackson, MI 49203.*

PROGRAM CALCULATES REMAINING PRINTER TAPE

Hank Librach, 52 Bulkley Drive, Fairfield, CT 06430, sent in this program that assumes a paper thickness of 0.003 and an inside diameter of 0.5 inches. How much is left on your roll?

Program *Tape Length Calculator*



```
10:PRINT "TAPE              51:PRINT " "
   LENGTH"                  60:PRINT "REMAI
15:PRINT "CALCU                NING LENGTH=
   LATOR"                      ";L;" IN."
16:PRINT " "
20:PRINT "MEASU
   RE SPOOL"
25:PRINT "OUTSI             TAPE LENGTH
   DE DIAMETER"             CALCULATOR
30:INPUT "DIAM=
   ";D                      MEASURE SPOOL
35:PRINT "DIAM.             OUTSIDE DIAMETER
   = ";D                    DIAM.= 0.75
40:A=(D^2-.25)*
   π/4                      REMAINING LENGTH
50:L=A/.003                 =81.8123087 IN.
```

AN ASSORTMENT OF SORTS

Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158, sent in the following assortment of sort routines.

Given a list of N numbers, with N greater than 2, stored in array locations A(27) to A(26+N), the sorting routines shown here rearrange the N numbers in order from smallest to largest. Although more than 160 numbers can be sorted, it should be noted that sorting a long list on a PC takes a long, *long* time. In the routines illustrated, the variable named "N" should contain the number of items to be sorted.

The Bubble Sort

Starting from the beginning of the list, adjacent pairs of numbers are compared; if out of order, they are interchanged. Thus, the larger number is placed lower in the list. The process is repeated until the list is completely ordered. Altogether, (N)*(N-1)/2 comparisons are required.

```
10  FOR X=25+N TO 27 STEP -1:FOR Y=27 TO X
20  IF A(Y)>A(Y+1) LET Z=A(Y):A(Y)=A(Y+1):A(Y+1)=Z
30  NEXT Y:NEXT X
```

The routine uses 75 bytes and variables N, X, Y and Z.

The Exchange Sort

The smallest number in the list is placed into Z, with its location in W; it is then exchanged with the first number in the list. Next, the second smallest number is found and exchanged with the second number in the list; and so on. Although this requires as many comparisons as the bubble sort, fewer stores and recalls are needed, giving a somewhat shorter execution time.

```
10  FOR X=27 TO 25+N:W=X:Z=A(X):FOR Y=X+1 TO
    26+N
20  IF A(Y)<Z LET W=Y:Z=A(Y)
30  NEXT Y:A(W)=A(X):A(X)=Z:NEXT X
```

The routine uses 82 bytes and variables N, W, X, Y and Z.

The Insertion Sort

The list is treated as if it is divided into a sorted part and an unsorted part. Numbers are removed from the unsorted part and inserted into the sorted part, one at a time. A space for the insertion is provided by shifting down all the numbers in the sorted part that are below the insertion position.

If the list starts out completely reversed, (N+2)*(N-1)/2 comparisions are needed. Fewer comparisons are required if the list is initially partially ordered. This sort performs half as many stores and recalls as that required by the exchange sort.

```
10  FOR X=28 TO 26+N:Z=A(X):FOR Y=X-1 TO 26
    STEP -1
20  IF Z<A(Y) LET A(Y+1)=A(Y):NEXT Y
30  A(Y+1)=Z:NEXT X
```

The routine uses 72 bytes and variables N, X Y and Z.

The Shell Sort

A Shell sort is more complex than the methods described previously. Subsets of the list, consisting of the numbers whose locations are separated by multiples of V, ares sorted using an insertion technique. The values of V utilized in the routine illustrated are 121, 40, 13, 4 and 1; starting with a value that is than N. (The initial value for V is determined in line 10 of the program.) In comparison with the other routines presented, there is typically a considerable reduction in the number of comparisons required to accomplish a sort, as well as a reduction in the number of stores and recalls executed.

```
10  V=INT(.5*3ΔINT(LOG 2N/LOG 3
20  FOR W=27 TO 26+V:FOR X=W+V TO 27+N-V
    STEP V:Z=A(X):FOR Y=X-V TO W STEP -V
30  IF Z<A(Y) LET A(V+Y)=A(Y):A(Y)=Z:NEXT Y
40  NEXT X:NEXT W:V=(V-1)/3:IF V GOTO 20
```

The routine uses 124 bytes and the variables N, V, W, X, Y and Z.

Comparisons

Average times for sorting N randomly arranged numbers are tabulated in the accompanying table. The time is noted in minutes and seconds. With the exception of the Shell sort, execution time is approximately proportionate to N^2

Table *Timing Comparisons*

N	Bubble	Exchange	Insertion	Shell
10	0:37	0:28	0:22	0:26
20	2:30	1:40	1:15	1:20
30	5:35	3:35	3:05	2:05
40	10:00	6:10	4:35	2:55
50	15:40	9:30	7:00	4:40

It may be noted that the speed of the Shell sort becomes especially significant as a list lengthens.

The next table tabulates the time required to sort a list of 50 items when it is initially ordered versus being initially completely reversed.

Table *Ordered and Reversed Lists*

	Bubble	Exchange	Insertion	Shell
In order:	10:25	8:35	0:50	2:45
Reversed	20:50	10:25	13:05	3:55

Note, in the case of the Shell sort, that an initially reversed list does not necessarily require more time than when the list is in some other (less than ordered) arrangement. In the other types of sorts, the completely reversed list requires the most amount of execution time.

It is also interesting to note that the insertion type of sort is the fastest when a list is initially ordered or is close to that state.

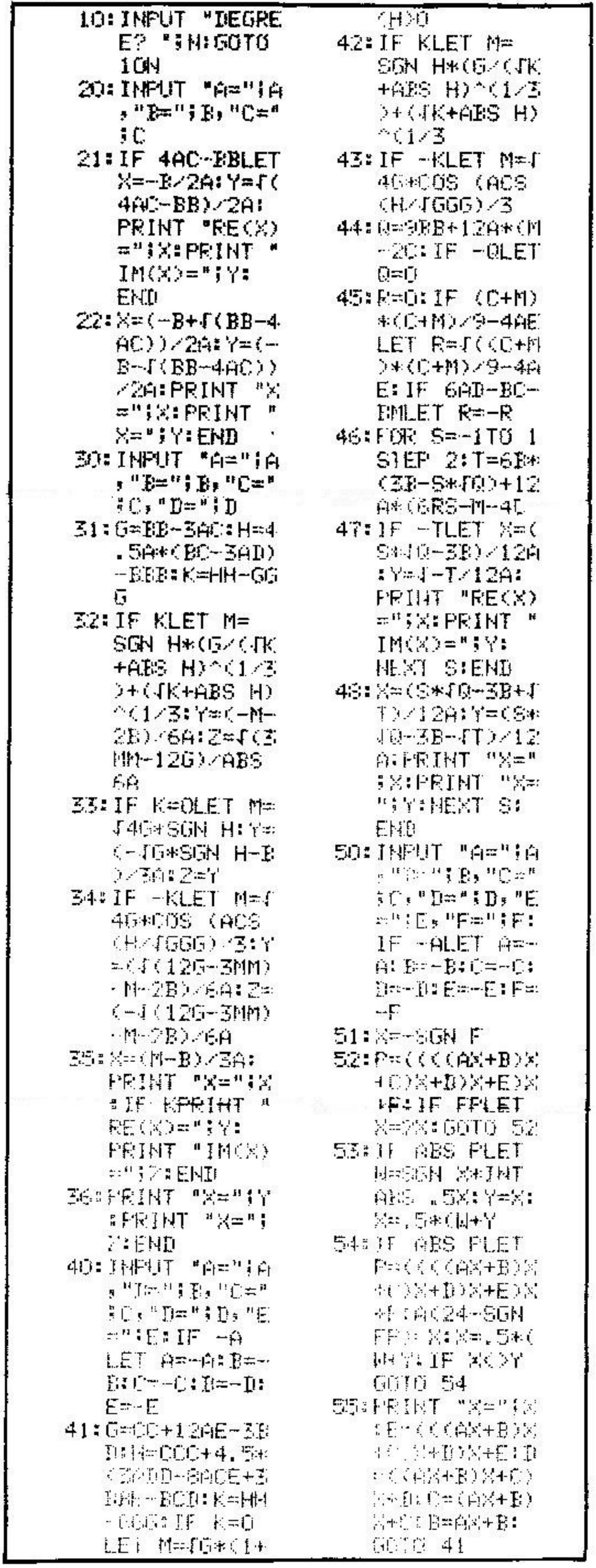
ROOTS OF POLYNOMIALS

The accompanying program, provided by *Norlin Rober,* determines the complex roots of a polynomial having real coefficients, with degree 2 through 5.

To use the program, execute RUN and enter the coefficients as prompted, according to the following format:

Degree 2: $Ax^2+Bx+C=0$, with $A \neq 0$
Degree 3: $Ax^3+Bx^2+Cx+D=0$, with $A \neq 0$

Program *Polynomials*



```
10:INPUT "DEGRE
   E? ";N:GOTO
   10N
20:INPUT "A=";A
   ,"B=";B,"C="
   ;C
21:IF 4AC-BBLET
   X=-B/2A:Y=√(
   4AC-BB)/2A:
   PRINT "RE(X)
   =";X:PRINT "
   IM(X)=";Y:
   END
22:X=(-B+√(BB-4
   AC))/2A:Y=(-
   B-√(BB-4AC))
   /2A:PRINT "X
   =";X:PRINT "
   X=";Y:END
30:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D
31:G=BB-3AC:H=4
   .5A*(BC-3AD)
   -BBB:K=HH-GG
   G
32:IF KLET M=
   SGN H*(G/(√K
   +ABS H)^(1/3
   )+(√K+ABS H)
   ^(1/3:Y=(-M-
   2B)/6A:Z=√(3
   MM-12G)/ABS
   6A
33:IF K=0LET M=
   √4G*SGN H:Y=
   (-√G*SGN H-B
   )/3A:Z=Y
34:IF -KLET M=√
   4G*COS (ACS
   (H/√GGG)/3:Y
   =(√(12G-3MM)
   -M-2B)/6A:Z=
   (-√(12G-3MM)
   -M-2B)/6A
35:X=(M-B)/3A:
   PRINT "X=";X
   :IF KPRINT "
   RE(X)=";Y:
   PRINT "IM(X)
   =";Z:END
36:PRINT "X=";Y
   :PRINT "X=";
   Z:END
40:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D,"E
   =";E:IF -A
   LET A=-A:B=-
   B:C=-C:D=-D:
   E=-E
41:G=CC+12AE-3B
   D:H=CCC+4.5*
   (3ADD-8ACE+3
   BBE-BCD:K=HH
   -GGG:IF K=0
   LET M=√G*(1+
   (H>0
42:IF KLET M=
   SGN H*(G/(√K
   +ABS H)^(1/3
   )+(√K+ABS H)
   ^(1/3
43:IF -KLET M=√
   4G*COS (ACS
   (H/√GGG)/3
44:Q=9BB+12A*(M
   -2C:IF -QLET
   Q=0
45:R=0:IF (C+M)
   *(C+M)/9-4AE
   LET R=√((C+M
   )*(C+M)/9-4A
   E:IF 6AD-BC-
   BMLET R=-R
46:FOR S=-1TO 1
   STEP 2:T=6B*
   (3B-S*√Q)+12
   A*(6RS-M-4C
47:IF -TLET X=(
   S*√Q-3B)/12A
   :Y=√-T/12A:
   PRINT "RE(X)
   =";X:PRINT "
   IM(X)=";Y:
   NEXT S:END
48:X=(S*√Q-3B+√
   T)/12A:Y=(S*
   √Q-3B-√T)/12
   A:PRINT "X="
   ;X:PRINT "X=
   ";Y:NEXT S:
   END
50:INPUT "A=";A
   ,"B=";B,"C="
   ;C,"D=";D,"E
   =";E,"F=";F:
   IF -ALET A=-
   A:B=-B:C=-C:
   D=-D:E=-E:F=
   -F
51:X=-SGN F
52:P=((((AX+B)X
   +C)X+D)X+E)X
   +F:IF FPLET
   X=2X:GOTO 52
53:IF ABS PLET
   W=SGN X*INT
   ABS .5X:Y=X:
   X=.5*(W+Y
54:IF ABS PLET
   P=((((AX+B)X
   +C)X+D)X+E)X
   +F:A(24-SGN
   FP)=X:X=.5*(
   W+Y:IF X<>Y
   GOTO 54
55:PRINT "X=";X
   :E=(((AX+B)X
   +C)X+D)X+E:D
   =((AX+B)X+C)
   X+D:C=(AX+B)
   X+C:B=AX+B:
   GOTO 41
```

Degree 4: $Ax^4+Bx^3+Cx^2+Dx+E=0$, with $A \neq 0$

Degree 5: $Ax^5+Bx^4+Cx^3+Dx^2+Ex+F=0$, with $A \neq 0$

All complex roots will be calculated and displayed. If there is a conjugate pair of complex roots, the real part will be indicated as RE(X) and the imaginary part as IM(X).

A loss of accuracy may occur, particularly in the case of multiple roots. This is unavoidable with floating-point arithmetic.

An example: To solve the equation:

$$8x^5-75x^4+275x^3-485x^2+387x-90=0$$

type RUN and press ENTER. In response to the DEGREE? query, input 5. Enter the coefficients as prompted. In this case A=8, B=-75, C=275, D=-485, E=387 and F=-90. Solutions for this problem will be displayed as:

```
X = 0.375
RE(X) =2    indicating the conju-
IM(X) = 1   gate pair:  2 ± i
X = 3
X = 2
```

Program Notes

Second-degree equations are solved by lines 20 – 22; third-degree by lines 30 – 36; fourth-degree, lines 40 – 48; and fifth-degree by lines 50 – 55 (using half-interval search) with reduction to 4'th degree in line 55, with the resulting quartic equation being solved beginning at line 41.

Determination of the first root of a 5'th degree equation may take up to one minute.

The cubic equation is solved by the use of Cardan's Formulas; the quartic is solved by a modification of Ferrari's method.

PROGRAM CONVERTS FRACTIONS & DECIMALS

The program provided here as submitted by *Norlin Rober*, should be operated with the PC in the DEF mode.

SHIFT F is used to obtain a fraction approximating a given decimal. The input must be a positive number with a non-zero fractional part. A sequence of increasingly accurate approximations is provided. Each approximating fraction given is reduced to lowest terms.

~~Example: Push SHIFT F and~~ use 3.141592654 (π) as input, then press ENTER. The program responds with the approximations:

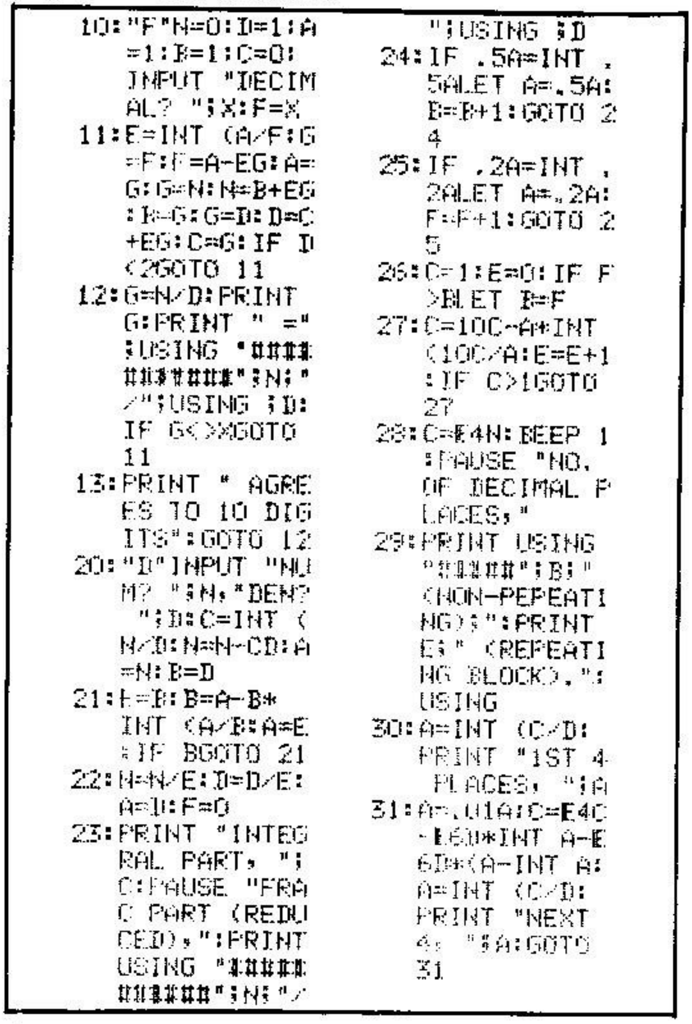
```
3.142857143  =  22/7
3.141509434  =  333/106
3.14159292   =  355/113
3.141592654  =  10438/33215
```

The latter result agrees to 10 significant digits.

SHIFT D produces the decimal value for a given fraction. The fraction is reduced to lowest terms and displayed, then the infinite decimal expansion is determined. Both the numerator and denominator inputted must be positive.

Example: Push SHIFT D, respond with 735 ENTER to the NUM? query and 456 ENTER to the DEN? query. The fraction will initially be reduced and shown as: 2 plus 37/104. Hit the ENTER key again to compute the infinite decimal. The computer displays, for this case, 3 places of non-repeating

Program *Fractions & Decimals*



```
10:"F"N=0:D=1:A
   =1:B=1:C=0:
   INPUT "DECIM
   AL? ";X:F=X
11:E=INT (A/F:G
   =F:F=A-EG:A=
   G:G=N:N=B+EG
   :B=G:G=D:D=C
   +EG:C=G:IF D
   <2GOTO 11
12:G=N/D:PRINT
   G:PRINT " ="
   ;USING "####
   ######";N;"
   /";USING ;D:
   IF G<>XGOTO
   11
13:PRINT " AGRE
   ES TO 10 DIG
   ITS":GOTO 12
20:"D"INPUT "NU
   M? ";N,"DEN?
    ";D:C=INT (
   N/D:N=N-CD:A
   =N:B=D
21:E=B:B=A-B*
   INT (A/B:A=E
   :IF BGOTO 21
22:N=N/E:D=D/E:
   A=D:F=0
23:PRINT "INTEG
   RAL PART, ";
   C:PAUSE "FRA
   C PART (REDU
   CED),":PRINT
   USING "#####
   ######";N;"/
   ";USING ;D
24:IF .5A=INT .
   5ALET A=.5A:
   B=B+1:GOTO 2
   4
25:IF .2A=INT .
   2ALET A=.2A:
   F=F+1:GOTO 2
   5
26:C=1:E=0:IF F
   >BLET B=F
27:C=10C-A*INT
   (10C/A:E=E+1
   :IF C>1GOTO
   27
28:C=E4N:BEEP 1
   :PAUSE "NO.
   OF DECIMAL P
   LACES,"
29:PRINT USING
   "####";B;"
   (NON-PEPEATI
   NG);":PRINT
   E;" (REPEATI
   NG BLOCK),":
   USING
30:A=INT (C/D:
   PRINT "1ST 4
    PLACES, ";A
31:A=.01A:C=E4C
   -E6D*INT A-E
   6D*(A-INT A:
   A=INT (C/D:
   PRINT "NEXT
   4, ";A:GOTO
   31
```

digits, to be followed by a repeating block of digits. Use the ENTER key repeatedly to obtain digits (in blocks of four):

.6118421052631578947З6

Program Notes

In line 11, approximating fractions are determined using the convergents of a continued fraction.

In lines 24 & 25, the length of the non-repeating block of decimal places is determined by counting the 2's and 5's that are removed as factors of the denominator. The larger of the two counts is the number of digits in the non-repeating block.

In line 27, a count is made of the number of divisions required before the remainder repeats itself. The count indicates the size of the repeating block.

Counting the number of divisions in line 27 may take a long time. A denominator of 1087, for example, results in a repeating block of 1086 digits, which takes almost 12 minutes to be determined. If the block size should happen to be a 10-digit number, execution time would be approximately 200 years. (Be sure to start with a fresh set of batteries for this case!)

FROM THE WATCH POCKET

Look for the Panasonic/Quasar HHC to be announced with a lot of fanfare around the end of October.

Meanwhile, the Casio FX-702P unit is being delivered and snapped up. A check with *Markline, Inc., P.O. Box 6800, Waltham, MA 02154,* a distributor of the unit, indicated they quickly sold out their initial supply. They expect more shortly. You can check their availability of the unit by phoning (617) 891-6250.

If you need any Sharp products, including supplies such as paper and ribbons, check with *Atlantic N.E. Marketing, P.O. Box 921, Marblehead, MA 01945.* The phone number is (617) 639-0285.

Sharp is currently selling the PC-1211 at $179.95. However, if you wait a few months you are likely to see Radio Shack make a move for the holiday market at around $169.95.

Bob McElwain continues to do a nice job with his column in **Interface Age.** In the October issue he discusses an application of interest to educators.

George Haller has an automobile trip calculator program designed to run on a Radio Shack TRS-80/Sharp PC-1211 in the September/October issue of **80-U.S. Journal.**

Page 162 of the September, 1981, issue of **BYTE** contains a programming tip of interest to PC users. It explains how to convert a FOR...NEXT loop in the BASIC language to the REPEAT...UNTIL construct that is desirable in some types of applications. The method discussed is easily implemented on a PC so you might want to look it over if you get a chance.

Don't forget to send in your entries for the *Hot Steaming Contest* as described in last month's *PCN.* Submissions must be received by October 20, 1981, in order to be eligible for consideration by the judges. We already have a number of clever messages entered. Don't miss out. Remember, the prizes are *cash* awards!

It is well over a year since I obtained my first PC and I have yet to change the batteries. -- *Nat Wadsworth, Editor*

SHARE YOUR POCKET COMPUTER KNOWLEDGE

We are always interested in hearing about applications for pocket computers, new products and news of the industry. Do you know of any operating time-savers, programming tips, short cuts or technical information that might be of interest to others? Have you developed application programs that you would like to share with fellow pocket computer enthusiasts? Have you had experience with PC products or programs that others might find of value?

If so, write it up! If your submission is accepted for publication you will earn a cash reward as well as have the satisfaction of knowing you are helping other PC users.

We prefer that program listings be supplied on audio cassette tape so that they may be easily reviewed. We can also then make listings of accepted programs directly from the tape to avoid transcription errors. If we cannot use your material, we will endeavor to promptly return it *provided you include a self-addressed stamped envelope for that purpose.* We are careful, but we cannot assume responsibility for submitted materials.

Submissions should be addressed to: The Editor, PC NEWSLETTER, PO Box 232, Seymour, CT 06483

POCKET COMPUTER NEWSLETTER

© Copyright 1981 — Issue 09

November

WORLD'S FIRST COMPUTER FOR LESS THAN $100.00 ALMOST FITS IN A POCKET

The Sinclair ZX81, recently announced as being available in the United States (and previously available in England and elsewhere), measures just 6 x 6.5 x 1.5 inches and weighs a mere 12 ounces. In kit form it is priced at just $99.95, making it the first personal computer to drop below the $100.00 mark. While it does not contain its own built-in display, nor does it quite fit in a pocket, its compactness and features, combined with its trend-setting price, make it worthy of mention in this publication.

The unit is the predecessor to the firm's highly successful ZX80 computer. The improvement in price-performance came about through the reduction in the number of integrated circuits used in the unit, bringing it down to four chips, where previously it was about 20.

The unit comes with an 8K BASIC language in ROM and sports floating-point arithmetic and scientific functions. The software includes these operating features:

— Automatic syntax checking of every statement line. —
— Program editing features. —
— Optional single stroke key-word entry. —

The software drives a television set to provide display of text as well as graphics. Up to 24 lines of text having 32 characters-per-line may be shown on a screen. The software can also drive a companion printer, also priced at less than one hundred dollars, that is expected to be available shortly.

Photo *The Sinclair ZX81 Portable Computer*

The software is actually capable of operating in two modes. In one mode, termed "normal," the unit is able to perform computations while simultaneously presenting a display. (The earlier ZX80 computer blanked out the display while performing computations.) A second mode, termed the "fast" option, executes instructions at a rate four times faster while blanking the screen.

The ZX81 features a 40-key touch sensitive keyboard which operates in several modes to provide the equivalent of 91 functions, keywords and characters.

The unit also boasts a graphics mode that provides 20 extra graphic characters and 54 inverse characters. Or, the display may be divided into a format consisting of 64 by 44 pixels, each of which may be selectively activated or turned off.

The highly portable computer may be directly connected to a regular audio cassette player for the saving or loading of programs. The tape operating system includes the capability to save programs by name and have the computer search through a tape for the program having the name designated by the user.

For those with a scientific/engineering bent, the unit is able to store numbers in scientific notation to an accuracy of 9-1/2 decimal digits. It also features full log, trigonometric and inverse functions. Plus, it is said to be capable of handling multi-dimensional numeric and string arrays.

As shown in the accompanying photograph, a 16K RAM module may be plugged into the back of the ZX81 to expand its memory.

The unit is supplied with a 164 page instruction manual which includes a course in programming using the BASIC language.

The ZX81 is optionally available completely assembled and tested at $149.95.

The firm has indicated it will also be making a line of application programs available for the ZX81. These will be supplied on cassette. Areas being supported include business and well as household management, educational programs and games.

For additional information on the ZX81 you may contact: *Mary E. Reinman, % Agnew, Carter, McCarthy, Inc., 100 Boylston Street, Boston, MA 02116. The phone number is (617) 451-0450.*

PANASONIC/QUASAR UNIT SET TO BOW IN EARLY NOVEMBER

A reliable source indicates that the new RL-H1000 hand-held computer will be released to the general public at a debute to be held in New York City on November 10, 1981. Additional publicity splashes are scheduled thereafter in the Los Angeles, Chicago, Dallas and Atlanta metropolises.

The RL-H1000 has been bandied about in the trade press for over a year as, apparently, the firms involved made numerous design alterations in both the hardware and software.

The significance of the unit is that it is slated to be the first pocket-sized unit that can be connected to a variety of peripherals that will give it the capabilities of most desktop-sized personal computers.

PRODUCT REVIEW
THE CASIO FX-702P HAND-HELD COMPUTER

Produced by: *Casio, Inc.*
List Price: *$199.95*
Availability: *Casio distributors.*
Reviewer: *David G. Motto, 3639 Roosevelt, Jackson, MI 49203.*

The Casio FX-702P arrived in a colorful box, imprinted with the word BASIC in letters 2-1/2 inches high, followed by POCKET COMPUTER in smaller type. The memory capacity of the computer is also shown on the box (1680 steps with 26 variable memories or 226 variable memories with 80 program steps). Below that information on the box is a life-sized picture of the machine and the words PROGRAMMABLE CALCULATOR. Casio seems to have sidestepped the question of whether they are selling a calculator or a computer as it is *both!*

Inside the colorful box was a plastic holder for the FX-702P, a soft carrying case, an 82-page operating manual, a 152-page program library, and the machine.

The unit is 5/8 inches thick by 6-1/2 inches long by 3-1/4 inches wide. That is just a little bigger than the Radio Shack TRS-80/Sharp PC-1211 PCs. The keyboard has 65 keys in various colors. Most keys have more than one function.

Figure *Left Half of FX-702P Keyboard.*

F1	F2	ARC ' FOR	HYP * TO	SIN * STEP	COS : NEXT	TAN ; PRT
LOG A IF	LN B THEN	EXP C GOTO	SQR D GSB	SGN E RET	INT F INP	FRAC G WAIT
ABS H RPC	RND(I PRC	DEG(J DMS	LEN(K SET	CSR L VAC	MID(M STOP	KEY N END
SDX O SAVE	SDY P LOAD	SDXN Q PUT	SDYN R GET	LRA S VER	LRB T DEFM	COR U PASS
EOX V	EOY W	X	Y	Z	=	SPC

Accompanying diagrams show the capabilities of the keyboard. On the left half of the board, the F1 and F2 keys are white, the rest of the keys are light grey. Legends above the keys are color-coded red. These are accessed by the F1 key. Legends below the keys are blue and are accessed by the F2 key.

The keys on the right half of the keyboard come in three colors. Those marked MODE, STAT, STOP, ANS, CONT and the left and right arrows are black. The C and AC keys are orange. The remaining keys are light grey. There are no blue legends under the keys on this side of the board, only above-key, red legends.

Figure *Right Half of FX-702P Keyboard.*

MODE	< (	>)	<= ↑	INS C	AC
P7 7	P8 8	P9 9	>= /	DEL STAT	STOP
P4 4	P5 5	P6 6	<> *	ASTAT ANS	CONT
P1 1	P2 2	P3 3	! -	SAC	HOME left arrow
P0 0	RAN#	pi .	? +	EXE	right arrow

The display can show up to 20 characters at a time and will automatically scroll to the left (unlike the TRS-80/Sharp PC-1211) during printing. However, the scrolling process is very slow. The display buffer will hold only 63 or 64 characters before any previous characters are lost. There are also annunciators in the display for the F1 and F2 shift keys, the ARC and HYP prefixes, RUN and WRT modes, program STOP, DEG or RAD or GRA trig modes, program TRACE and PRT modes. On the right-hand side of the display, four 7-segment LCD digits can show the number of steps of available program space while in the WRT mode. When a program is running, the middle segment of the last digit is on.

Immediately below the display is an abbreviated list of the meanings of the eight possible modes. On the top of the machine, above the display, is a small contrast adjustment dial. When turned all the way to the left, the display almost disappears.

On the left side of the unit, a small plastic cover hides the port for the FA-2 cassette I/O adaptor. This is a 7-pin port, similar to the one found on the Casio FX-502P and FX-602P programmable calculators. The port on these two calculators adapts to the FA-1. The case design of the FX-702P prevents it from being connected to that adaptor.

On the back of the machine is a large door (almost half the size of the unit) that covers an 11-contact port. Perhaps this is where the FP-10 printer will be connected. There is also a battery cover under this door. The FX-702P utilizes two flat lithium batteries.

For some reason the construction is such that it seems to fit better in my coat pocket than the TRS-80 PC.

Functions

Many commands are available to the user, both in the RUN mode as well as under program control. Some of them are available directly on the keyboard. Others have to be typed in. In some cases, the command has to be typed in to place it on a program line, even though it is also available on the keyboard!

Along with the functions, the mathematical operators of addition, subtraction, multiplication, division, raising to a power, and factorial are provided.

Capacity

As mentioned earlier, the FX-702P has the capacity for 1680 program steps with 26 variables or 80 program steps with 226 variables. To allocate memory to variables it is necessary to press the DEFM key and enter a number from 0 to 20. When the program/data mix is altered, the memory which is made available is cleared.

The 26 variables are named A to Z or A$ to Z$. They cannot be addressed as an array. In addition to these variables, there is a special string variable called "$" which can hold 30 characters. The MID function uses this variable. Standard string variables can hold up to 7 characters and can be appended to one another.

A group of separate statistical registers are always available.

If array space or simply additional variables are needed, the space must be reserved as indicated above. The area needed may be named in three separate ways:

1. A0 to A9, B0 to B9, C0 to C9, . . . T0 to T9 (adding a $ sign to signify strings).
2. A(n) or A$(n) where n ranges from 0 to 199.
3. A(l,m) or A$(l,m) where l ranges from 0 to 19 and m from 0 to 9.

Thus, the variables S7, A(187) and A(18,7) are all the same location.

If memory is not being used to store data it can hold program steps. Casio refers to each byte as a step. It takes 8 such steps to store a variable. Thus, when you allocate 10 variables you lose 80 steps.

Program lines may be up to 62 characters long including the line number. The line number may range from 1 to 9999.

Each program in memory is accessed by a program number. There may be ten separate programs. These are labeled P0 through P9. Each program is executed by pressing the F1 key followed by the program number.

Lines are stored in memory in a compressed format. Each keyword is stored as one step. All other symbols are one step each. The line number (and possibly the end-of-line token) takes up three steps. Unlike the TRS-80 PC, the keywords may be changed letter-by-letter. So, to change PRT to PUT it is only necessary to position the blinking underline cursor to the letter R in PRT and type the letter U. Pressing EXE enters the altered line.

An accompanying table shows the large number of trigonometric, hyperbolic and statistical functions available. Calculations are performed to 12-digit accuracy, then rounded to 10 digits for display.

Let's take a quick look at the speed of the FX-702P. The best way to get an estimate of the relative speeds of computers is to run the same

Table *FX-702P Commands, Statements and Functions.*

Command	Parameters	Programmable	Notes
ABS	x	yes	absolute value
ACS	x	yes	arc cosine
AHC	x	yes	arc hyperbolic cosine
AHS	x	yes	arc hyperbolic sine
AHT	x	yes	arc hyperbolic tangent
ASN	x	yes	arc sine
ASTAT			displays statistical values - sums of X, Y, X↑2, Y↑2, X*Y and number of entries
ATN	x	yes	arc tangent
CLR			clears current program
CLR ALL			clears all programs
CNT		yes	number of entries
COR		yes	correlation coefficent
COS	x	yes	cosine
CSR	x	yes-only	position cursor (0-19)
DEFM	x		reserve 10*x memories (0-20 for 0 to 200)
DEG(	d,m,s)	yes	degrees, minutes, seconds to decimal degrees
DEL	x,y;count	yes	subtract from statistical registers
DMS	x	yes	decimal degrees to degrees, minutes, seconds
END		yes-only	end program
EOX	x	yes	estimate of X
EOY	x	yes	estimate of Y
EXP	x	yes	e↑x
FOR	x=	yes-only	for-next loop - x can range to the capacity of the machine
FRAC	x	yes	fractional part of x
GET	x	?	read a range of variables from tape
GOTO	x	yes-only	jump to a line number
GSB	x	yes-only	jump to a subroutine either at a line number, or in another program
HCS	x	yes	hyperbolic cosine
HSN	x	yes	hyperbolic sine
HTN	x	yes	hyperbolic tangent
IF	x	yes-only	single condition test
INP	'prompt',x	yes-only	input a variable
INT	x	yes	integer part of x
KEY		yes-only	key pressed when executed
LEN(	$)	yes	length of string
LIST	x		list line number or current program
LIST ALL			list all programs (with step counts) and variables
LIST V			list all variables
LN	x	yes	natural logarithm
LOAD	x	yes	load program (if used in a program, begins execution)
LOAD ALL			load all programs and variables
LOG	x	yes	common logarithm (base 10)
LRA		yes	linear regression constant
LRB		yes	linear regression coefficient
MID(	b,l)	yes	part of string $ starting at position b, length l
MODE	x	yes	change to mode #n (0 to 8)
MX		yes	mean of X values
MY		yes	mean of Y values
NEXT	x	yes-only	end of FOR-NEXT loop
PASS	'password'		1 to 8 characters
PRC	r,theta	yes	converts polar coordinates to rectangular (x,y in X,Y)
PRT	x,y;z	yes-only	prints on display or on printer (depending on MODE) - if following a ';', next to the last character displayed. if no WAIT command, stops the program
PUT	x	?	outputs a range of variables to tape
RAN#		yes	a random number between 0, 1
RET		yes-only	return from subroutine
RND(	x,place)	yes	rounds x to decimal place
RPC	x,y	yes	converts rectangular x,y to polar r, theta in X,Y
RUN	x		runs current program from line number x
SAC	x	yes	clears all statistical values
SAVE	x		saves program to tape
~~SAVE ALL~~	~~x~~		~~saves all programs and data~~ variables to tape
SDX		yes	standard deviation of X
SDXN		yes	normal standard deviation of X
SDY		yes	standard deviation of Y
SDYN		yes	normal standard deviation of Y
SET	x	yes	set display digits, rounding
SGN	x	yes	sign of the number (-1, 0, 1)
SIN	x	yes	sine
SQR	x	yes	square root
STAT	x,y;count	yes	accumulate X and Y
STEP	x	yes-only	value to change index in FOR-NEXT loop - can be anything in range of machine
STOP		yes-only	halts program
SX		yes	sum of X values
SX2		yes	sum of X↑2 values
SXY		yes	sum of X*Y values
SY		yes	sum of Y values
SY2		yes	sum of Y↑2 values
TAN	x	yes	tangent
THEN	x	yes-only	branch to line number in IF
TO	x	yes-only	limit in FOR-NEXT loop
VAC	x	yes	clear all variables
VER	x		verify program, data output to tape correctly
WAIT	x	yes-only	set PRT display time to x * .05 seconds (0 - 999)

or similar tasks on them, then examine the time it takes to perform the work. Perhaps one of the easiest benchmarks to run is a simple counting algorithm. For comparison I ran the following on a TRS-80 PC:

```
10 FOR I=1 TO 100:NEXT I
```

It finished in 25 seconds. I then ran this on the FX-702P:

```
10 FOR I=1 TO 1000:NEXT I
```

It finished in 16 seconds. That may not sound impressive if you have not noticed the extra zero in the second program! The conclusion is evident: the FX-702P is about 15 times faster than the TRS-80 PC, at least in a straightforward looping test such as this. Of course, some functions may be slower.

The FX-702P has another ability which the PC-1211 does not provide: full comparison of strings. With this capability I was able to write a simple sort utility which gave me the order of the characters used in the FX-702P as illustrated in the accompanying table. As you may observe, there are 60 characters in all. In addition to these, the unit is also capable of displaying a degree symbol and a single quote mark (for the DEG function), though these are apparently not readily available to the user.

FX-702P users can operate the unit in various modes. When the power switch is turn on the machine comes up in the RUN mode, displaying READY P0. (Most of the time it does, anyway. Twice, I have powered up with garbage appearing in the display, but a quick off-and-on again has remedied such situations.) To alter the mode, you press the MODE key and then a digit key. The mode of operation can also ~~be controlled by a program with the restrictions that only~~ MODE 4 through MODE 8 may be specified. An accompanying table illustrates the various operating modes.

Programming

Let's face it: the reason we buy a programmable device is for the capability to make it our own unique machine by buying or writing programs for it. The best way to learn how this machine programs is to try it.

First, a word about programmability. The FX-702P lacks some of the niceties of the TRS-80/PC-1211's keyboard. One of the keys that I sorely miss is the back-a-line key. To look at the previous line on a FX-702P you have to list it by line number. I also miss the cursor recovery feature of the two cursor movement keys on the TRS-80 PC, whereby, if the display shows a program line, but the cursor is not blinking (such as when you have just hit ENTER), you can recover the cursor just by hitting either of the cursor movement keys. I also prefer the nice large size of the ENTER key on the TRS-80 and its QWERTY keyboard layout. Casio did try to save keystrokes by putting some of the functions above and below the keys, but finding them is sometimes more trouble than it is worth. I just found out that the SET command is on the keyboard!

Here are a few programs I have written for the FX-702P. The first one is named REFLEX. It is a reaction time program. The rules are simple. A bunch of letters are displayed. When you see the letter X, press any alphanumeric key. The time you took to spot the character and press a key is displayed. I haven't calibrated the program, but my record is 3 thingums.

Here is a line-by-line analysis of the REFLEX program written for the Casio FX-702P.

Line 100 clears all variables (A through Z) to zero.

Line 110 asks the user to choose the mode of play.

Line 120 grabs the first letter of the reply for testing and sets up variable B.

Line 130 tests for the "easy" mode. If true, it sets B to the line number of the EASY display and branches to 150. An IF condition must be followed either by a THEN and a line number or by a semicolon and a BASIC keyword.

Line 140 tests for an invalid condition. If true, it branches back to 110.

Line 150 sets up the alphabet in string variable $. It also adds one to the number of trials and initializes the time counter to zero. Then it clears the display.

Line 160 grabs a random letter from the string $ and branches to either line 170 (HARD) or 180 (EASY).

Line 170 displays the randomly selected letter anywhere within the first eight positions of the display, without clearing the display.

The semicolon following the variable name sets this up. Then, the routine branches to line 190.

Line 180 displays the character in the fifth position.

Line 190 tests the character. If it is not an X, it branches back to display another character.

Line 200 waits for you to press a key. Each time it tests and finds that no key has been activated, it adds one to the timer value and loops back to itself.

Line 210 clears the display, then displays the reaction time for about one second. It also adds the reaction time to the total.

Line 220 asks you to play again.

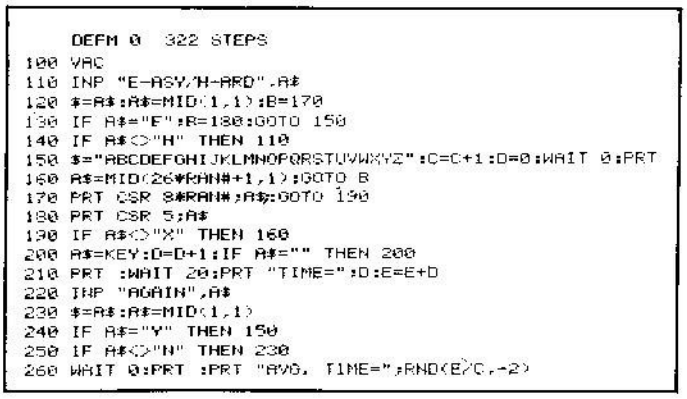
Line 230 grabs the first letter of your answer.

Line 240 tests for a YES answer. If so, it branches to 150.

Line 250 tests for an invalid answer. If found, it branches to 230.

Line 260 clears the display. It then displays the average score, rounded to the nearest tenth.

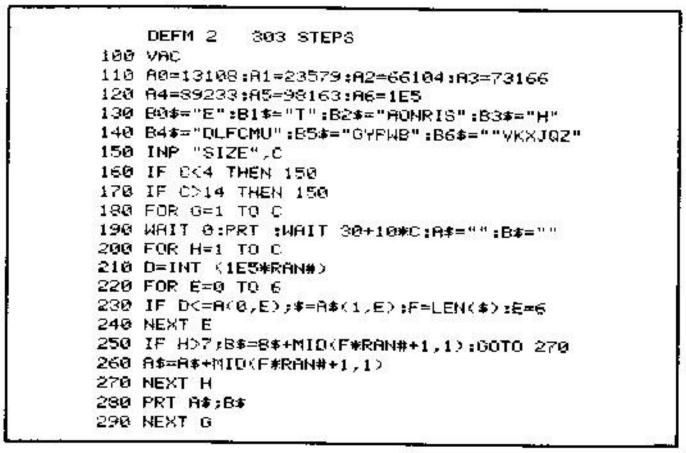
Program *Reflex for the Casio FX-702P*



```
     DEFM 0   322 STEPS
100 VAC
110 INP "E-ASY/H-ARD",A$
120 $=A$:A$=MID(1,1):B=170
130 IF A$="E";B=180:GOTO 150
140 IF A$<>"H" THEN 110
150 $="ABCDEFGHIJKLMNOPQRSTUVWXYZ":C=C+1:D=0:WAIT 0:PRT
160 A$=MID(26*RAN#+1,1):GOTO B
170 PRT CSR 8*RAN#;A$;GOTO 190
180 PRT CSR 5;A$
190 IF A$<>"X" THEN 160
200 A$=KEY:D=D+1:IF A$="" THEN 200
210 PRT :WAIT 20:PRT "TIME=";D:E=E+D
220 INP "AGAIN",A$
230 $=A$:A$=MID(1,1)
240 IF A$="Y" THEN 150
250 IF A$<>"N" THEN 230
260 WAIT 0:PRT :PRT "AVG. TIME=";RND(E/C,-2)
```

The second program I am including is a version of the game *Word Squares*. (The version for the TRS-80/PC-1211 PC was originally published in Issue 06 of *PCN*.) To use this program simply start it. When asked for the size of the square, type in any number from 4 to 14.

Program *Word Squares for the Casio FX-702P*



```
          DEFM 2    303 STEPS
      100 VAC
      110 A0=13108:A1=23579:A2=66104:A3=73166
      120 A4=89233:A5=98163:A6=1E5
      130 B0$="E":B1$="T":B2$="AONRIS":B3$="H"
      140 B4$="DLFCMU":B5$="GYPWB":B6$=""VKXJQZ"
      150 INP "SIZE",C
      160 IF C<4 THEN 150
      170 IF C>14 THEN 150
      180 FOR G=1 TO C
      190 WAIT 0:PRT :WAIT 30+10*C:A$="":B$=""
      200 FOR H=1 TO C
      210 D=INT (1E5*RAN#)
      220 FOR E=0 TO 6
      230 IF D<=A(0,E);$=A$(1,E):F=LEN($):E=6
      240 NEXT E
      250 IF H>7;B$=B$+MID(F*RAN#+1,1):GOTO 270
      260 A$=A$+MID(F*RAN#+1,1)
      270 NEXT H
      280 PRT A$;B$
      290 NEXT G
```

The letters in each line will be displayed long enough to copy them down. The lines in the program perform as indicated here:

Line 100 clears all variables to zero (A-Z, A0-A9 and B0-B9).

Lines 110 - 140 set up the array values for the letter frequencies.

Line 150 asks for the square size.

Lines 160 & 170 test the square size for proper limits. Note that I did not combine the two tests on one line using an asterisk as may be done on a TRS-80 PC. The FX-702P will not accomodate that type of logic operation.

Line 180 sets up the outer loop (number of lines to display).

Line 190 clears the display, sets up the display time (which depends upon the size of the square), and initializes the line.

Line 200 sets up the inner loop (number of letters per line).

Line 210 finds a random number from 0 to 100,000.

Lines 220 - 240 search the array for a set of letters.

Line 250 adds a random letter from the chosen set to the second part of the line, if the first part is full.

Line 260 adds a random letter from the chosen set to the first part of the line, if it is not full.

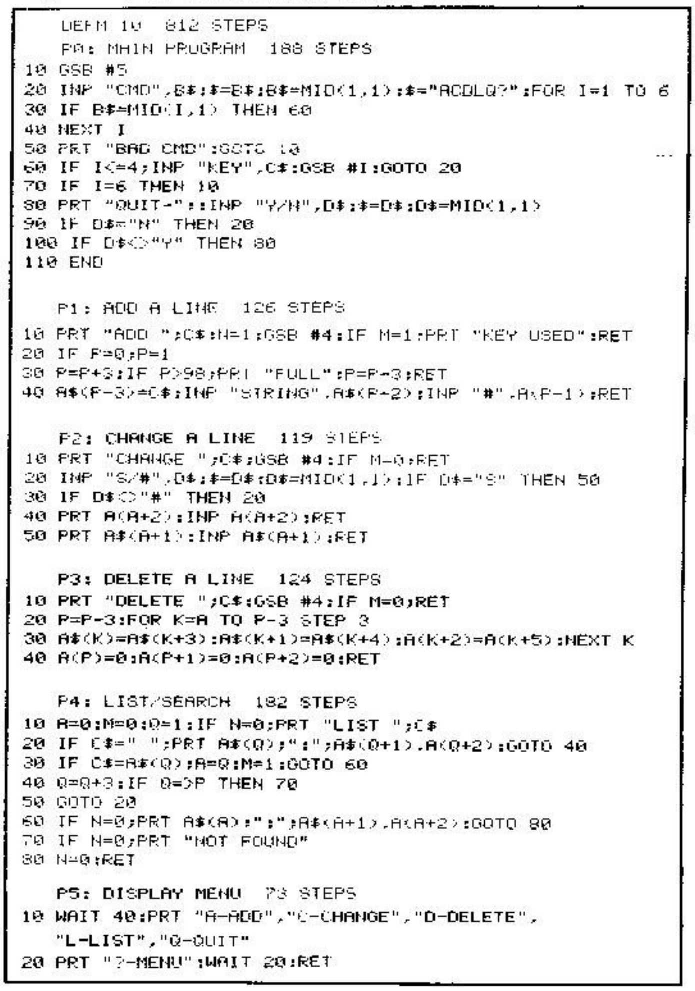
Line 270 is the end of the inner loop.

Line 280 prints the line.

Line 290 is the end of the outer loop.

The third program commemorates Issue 01 of *PCN*. The function of this program is to provide a 32-page electronic notebook in which you can write a key, an alphanumeric entry, and a number. You also are provided with the ability to review, change or delete entries.

Program *Pocket Notebook for the Casio FX-702P*



```
    DEFM 10   812 STEPS
    P0: MAIN PROGRAM  188 STEPS
10 GSB #5
20 INP "CMD",B$:$=B$:B$=MID(1,1):$="ACDLQ?":FOR I=1 TO 6
30 IF B$=MID(I,1) THEN 60
40 NEXT I
50 PRT "BAD CMD":GOTO 10
60 IF I<=4;INP "KEY",C$:GSB #I:GOTO 20
70 IF I=6 THEN 10
80 PRT "QUIT-";:INP "Y/N",D$:$=D$:D$=MID(1,1)
90 IF D$="N" THEN 20
100 IF D$<>"Y" THEN 80
110 END

    P1: ADD A LINE  126 STEPS
10 PRT "ADD ";C$:N=1:GSB #4:IF M=1;PRT "KEY USED":RET
20 IF P=0;P=1
30 P=P+3:IF P>98;PRT "FULL":P=P-3:RET
40 A$(P-3)=C$:INP "STRING",A$(P-2):INP "#",A(P-1):RET

    P2: CHANGE A LINE  119 STEPS
10 PRT "CHANGE ";C$:GSB #4:IF M=0;RET
20 INP "S/#",D$:$=D$:D$=MID(1,1):IF D$="S" THEN 50
30 IF D$<>"#" THEN 20
40 PRT A(A+2):INP A(A+2):RET
50 PRT A$(A+1):INP A$(A+1):RET

    P3: DELETE A LINE  124 STEPS
10 PRT "DELETE ";C$:GSB #4:IF M=0;RET
20 P=P-3:FOR K=A TO P-3 STEP 3
30 A$(K)=A$(K+3):A$(K+1)=A$(K+4):A(K+2)=A(K+5):NEXT K
40 A(P)=0:A(P+1)=0:A(P+2)=0:RET

    P4: LIST/SEARCH  182 STEPS
10 A=0:M=0:Q=1:IF N=0;PRT "LIST ";C$
20 IF C$=" ";PRT A$(Q);":";A$(Q+1),A(Q+2):GOTO 40
30 IF C$=A$(Q);A=Q:M=1:GOTO 60
40 Q=Q+3:IF Q=>P THEN 70
50 GOTO 20
60 IF N=0;PRT A$(A);":";A$(A+1),A(A+2):GOTO 80
70 IF N=0;PRT "NOT FOUND"
80 N=0:RET

    P5: DISPLAY MENU  73 STEPS
10 WAIT 40:PRT "A-ADD","C-CHANGE","D-DELETE",
   "L-LIST","Q-QUIT"
20 PRT "?-MENU":WAIT 20:RET
```

To use the program, press F1 P0. The display will show the list of possible commands (section P5 of the program) then prompts for a command from the user.

If you want to ADD to the notebook, press A and then the EXE key. You will be asked for the key for this entry. The contents of the notebook are searched for that key. If it has already been used, you are asked for another command. If the key has not been used and the notebook is not full, you will be asked to key in the string and the number that you wish to store. Because of the manner in which the INP statement functions, you must key in something at each prompt or the program stops.

If you want to CHANGE an entry, type C (and the EXE key). The program then asks for the key and whether you wish to change the string (S) portion of the entry or the number (#) portion. The old contents are displayed. To DELETE an entry, type D (EXE). Enter the key. The entry is displayed and then deleted. All entries in the notebook beyond that point are shifted down to fill the slot. If you want to LIST an entry, type L (EXE). To look at a single entry, enter the key. To review the entire notebook, enter a space as the key.

To review the commands available in the program, type ? (EXE).

Odds and Ends

Documentation can be a very bad word to some programmers. Writing it is boring, reading it is more boring. The people at Casio must be like some programmers. The owners manual is not bad. It has a lot of examples. Neither is it very good. The examples are incomplete; not linked together. Some of the English is wierd. Any company which sells an item in another country should have someone on their staff to write manuals in that country's language.

The Casio FX-702P can round numbers and display them at 1 to 10 decimal places. The SET command is used to control the number of digits displayed as well as whether the last digit is to be rounded up or down.

The machine does not have any audio capability. Perhaps the FA-2 adaptor will remedy this deficiency.

The error detecting capabilities are similar to those of the TRS-80 PC. If an error condition is found, the display presents an error number (and the line number if appropriate). After an error condition the display must be cleared. The machine does not show the place in a line where an error occurs as the TRS-80 does.

The control keys (such as STOP, CONT, HOME, EXE, ANS, etc.) are very helpful. The ANS key, for instance, displays the result of the last calculation. The C key may be used to delete the character to the left of the current cursor position. The HOME key positions the cursor to the character that is the furthest to the left in the display.

It would be nice to be able to access the four digits on the right side of the display. One could then show, for example, what program was being executed.

The machine has a password feature that prevents those who do not know the password from examining, tracing or copying a program. Does anyone really need that kind of security in a pocket computer?

Casio chose the alphabetic arrangement for their keyboard. They also made all of them the same size. I am not sure that either of these choices were the best ones to make.

In designing the FX-702P they apparently took a lot from some of their better programmable calculators. Most of the documentation is similar, including the book of over 70 programs. Perhaps they don't realize how much can be done with the machine.

The statistical functions are very nice. However, the statistical registers are not accessible by the user except through the STAT and DEL commands. If users could get at those registers they might be able to save some time and steps by utilizing the built-in functions. Thus, for example, programs that require division of a number might utilize the MX and MY functions.

Table *FX-702P Error Codes.*

ERROR	MEANING
1	Memory or system stack overflow.
2	Syntax error . (What are you trying to say?)
3	Mathematical error (out of range, indefinite).
4	Undefined line number.
5	Argument error (array subscript too big, etc.).
6	Variable error (not reserved by DEFM).
7	Nesting error (FOR-NEXT or GSB-RET not correct).
8	Password error (invalid or absent).
9	Option error (printer or cassette not attached).

If a variable contains a number and you access it as a string, nothing is displayed. On the other hand, if it contains a string and you access it as a number, an error condition results. The accompanying table shows the error codes displayed by the machine.

The user's manual does not provide a list of the commands that are available in table form. I made up such a list which is included as part of this review. The manual also does not have an index.

Conclusion

How does the TRS-80/Sharp PC-1211 compare with the FX-702P? The memory sizes are similar. The number of nested loops on the FX-702P is 8; twice as many as that of the PC-1211. The number of nested subroutines possible is 10 on the FX-702P. That is 2-1/2 times as much as that of the TRS-80 PC. The FX-702P has more functions, a variable-length display, the ability to look at the keyboard without stopping the program, and greater string-handling features. The TRS-80/Sharp PC-1211 has labels, rudimentary sound, a sturdier case, plus the printer and cassette interfaces are available *now.*

The BASIC interpreter is simple and straightforward in both units. If you know how to type, the keyboard arrangement on the TRS-80 PC seems a little easier to use. On the PC-1211 you can reserve a key to perform a whole bunch of keystrokes. On the FX-702P many commands are also just a shift away.

The FX-702P has it over the others on speed.

The Casio unit uses lithium batteries. These may not be as readily available as the batteries used in the TRS-80 Pocket Computer.

The FX-702P is currently less expensive than the Radio Shack PC. But, can it be repaired or replaced in as many places?

If I had to leave one at home, I would leave the TRS-80/PC-1211. While the FX-702P is not generation 2 in the pocket computer race, it is at least generation 1.5. Will Panasonic have something that is really a full step ahead?

PRODUCT REVIEW
TRAVEL/BUSINESS EXPENSES PROGRAM

Produced by: *Blanton Software Service*
List price: *$7.95*
Availability: *Directly from the publisher – Blanton Software Service, 4522 Briar Forrest, San Antonio, TX 78217, Phone (512) 657-0766.*
Reviewer: *Nat Wadsworth, Editor.*

Blanton Software Service produces a number of PC software packages. Several others will be mentioned later. The program primarily being reviewed here is called the *TRS-80 Pocket Computer Travel/Business Expenses Program* and it is understood that this package is for the *Radio Shack* TRS-80 Pocket Computer or its equivalent. Essentially this is a program similar in capability to the *Trip Expenses* program published in Issue 06 of *PCN.* You input expenses during a trip, such as those incurred for gasoline, servicing your car, lodging, food, entertainment and the catch-all "miscellaneous." The program maintains a running total in all categories. It also produces a few statistics such as miles-per-gallon, average price of gasoline purchased and so forth.

The program is nicely written and executes well. You always have the options of entering an expense, reviewing expenses, *printing* a summary of expenses for an entire trip (yes, this program is designed to make *optional* use of a PC printer), initializing all expense categories to zero (at the start of a trip) or taking the option of exiting the program. The package uses a menu presentation/selection technique which makes it easy to use. For instance, to enter a current expense you just press SHIFT A then repeatedly press the ENTER key until the expense category you are interested in appears on the display. Once you input an expense, the program immediately shows you the accumulated expenditures in that category.

A particularly nice feature of the program is the option of having a summary of expenses printed out. The total miles traveled, expenses by category and statistical data are all neatly tabulated by the printer for clipping to your expense report if desired.

The price listed includes the program supplied on a cassette, a listing of the program, five pages of operating instructions and a listing of the variables used and their assignments.

My conclusion: a good software package at a fair price. The cassette saves a lot of keying as this program takes almost all available memory.

DOUBLE-PRECISION LOG, ANTILOG, POWERS & ROOTS

This program calculates base 10 logarithms and antilogarithms. Logarithms are calculated to 24 decimal places. Antilogarithms are determined to 24 significant digits. In both cases the results are displayed correctly rounded to 20 digits or decimal places. The full 24 digits may be obtained by manually displaying variables A, B and C. The 24th character will be correct or nearly correct.

The routines should be executed with the PC in the DEF mode.

Logarithm Calculation

Press SHIFT L. Input using scientific notation. Include a properly located decimal point after the first digit of the mantissa. Output is in fixed-point. Calculation time takes 30 seconds to as much as 5 minutes.

For instance, to determine LOG 658.7123047611, press SHIFT L then enter inputs as prompted:

1ST 8 DIGITS?	6.5871230
NEXT 8?	47611000
NEXT 8?	0
EXP?	2

The result will be:

INT PART,	2
	8186957761 4826121739

to 20 decimal places.

Antilogarithm Calculation

Press SHIFT A. Input using fixed-point, including a decimal point preceeding the first block of 8 decimal places. Output is in scientific notation with a 20-digit mantissa displayed. The input data may be either positive or negative. Calculation time ranges from 0.4 to about 5.7 minutes.

Thus, to determine the antilog of 2.753 (equivalent to $10^{2.753}$); press SHIFT A, enter the data as prompted:

SIGN?	+
INT PART?	2
8 PLACES?	.75300000

Program *Double-Precision Log, Antilog, Powers & Roots*

```
  1:"L"B=0:C=0:D
    =0:INPUT "1S
    T 8 DIGITS?
    ";A,"NEXT 8?
     ";B,"NEXT 8
    ? ";C,"EXP?
    ";D
  2:A=E7A:B=BO:C
    =CO:E=0:F=0:
    G=0:J=1:FOR
    I=16TO 40
    STEP 2:H=INT
    (N*LOG (J+1
  3:IF I-33LET M
    =(A-N+B)JN+C
    J+C:L=B+JN:
    GOTO 5
  4:L=BJN:M=L-
    INT L+CJ+C:L
    =AJ-INT AJ+B
    +INT L*O
  5:K=A+INT AJ:
    IF K-N+L+MO
    LET J=.1J:
    NEXT I:B=(N-
    A-B)N-1:GOTO
    7
  6:C=M-INT M:B=
    L+INT M*O:A=
    K+INT B:B=B-
    INT B:G=G+A(
    I+1):F=F+A(I
    )+INT G:E=E+
    H:G=G-INT G:
    GOTO 3
  7:H=INT ((1-C+
    B)/LN 10:C=(
    1-C+B)/LN 10
    -H+G:B=F+H+
    INT C:A=E+
    INT BO:B=B-
    INT BO*N:C=C
    -INT C
  8:H=-1:IF D>=0
    LET H=1:B=B+
    SGN C:C=SGN
    C-C:A=N-A-
    SGN B:B=SGN
    B*N-B
  9:D=ABS D-(D<0
    :GOSUB 20:K=
    KH+HO:PRINT
    USING "#####
    #######";"INT
    PART,";K:
    PRINT (L<>J)
    L;M
 10:M=1:INPUT "N
    UM? ";L,"DEN
    OM? ";M
 11:H=SGN HLM:L=
    ABS L:M=ABS
    M:G=INT CL+B
    L:B=INT GO:F
    =AL+B:A=INT
    FO:E=DL+A
 12:D=INT (E/M:K
    =(E-DM-A)N+F
    :A=INT (K/M:
    K=(K-AM-B)N+
    G:B=INT (K/M
    :C=(K-BM+CL-
    INT CL)/M:
    GOTO 32
 20:J=E10:L=E-6B
    :M=L-INT L+
    INT (E4C+.5)
    /J:L=E2A+INT
    L+INT M:M=(M
    -INT M)J:K=D
    +(L=J:BEEP 1
    :RETURN
 30:"A"A=0:B=0:C
    =0:INPUT "SI
    GN? ";G$,"
    INT PART? ";
    D,"8 PLACES?
     ";A,"NEXT 8
    ? ";B,"NEXT
    8? ";C
 31:A=AN:C=CO:H=
    1:IF G$="-"
    LET H=-1
 32:IF -HLET A=N
    -A-SGN B:B=
    SGN B*N-B-
    SGN C:C=SGN
    C-C:D=-D-1
 33:E=E7:F=0:G=0
    :J=1:FOR I=1
    6TO 40STEP 2
    :H=INT (N*
    LOG (J+1
 34:K=A-H:L=B-A(
    I):M=C-A(I+1
    ):IF -KN-L-M
    LET J=.1J:
    NEXT I:GOTO
    38
 35:C=M+(M<0:B=L
    -(M<0:A=K-(B
    <0:B=B+(B<0)
    N:IF I-33LET
    K=0:L=EJN:M=
    FJN:GOTO 37
 36:K=EJ:L=FJN:M
    =0
 37:G=L-INT L+GJ
    +G+M:F=K-INT
    K+F+(INT G+
    INT L)O:E=E+
    INT F+INT K:
    F=F-INT F:G=
    G-INT G:GOTO
    34
 38:H=INT (LN 10
    *(B+C)O+(E+F
    :C=LN 10*(B+
    C)O+(E+F)-H+
    G:B=FN+H+INT
    C:A=E+INT BO
    :B=B-INT BO*
    N
 39:C=C-INT C:
    GOSUB 20:L=E
    -9L/(1+9*(L=
    J:PRINT
    USING "##.##
    ########";L;
    USING "#####
    #######";M:
    PRINT USING
    ;"EXP ";K
```

```
A(14.)=100000000
.
A(15.)=0.0000000
1
A(16.)=56639811.
A(17.)=9.5213738
89E-01
A(18.)=51582250.
A(19.)=4.0750199
97E-01
A(20.)=37826425.
A(21.)=7.4275188
18E-01
A(22.)=74793186.
A(23.)=4.0668921
39E-01
A(24.)=72768626.
A(25.)=6.9637313
53E-01
A(26.)=29231044.
A(27.)=5.3186855
49E-01
A(28.)=42942647.
A(29.)=5.6155640
74E-01
A(30.)=34294460.
A(31.)=1.8852918
01E-01
A(32.)=43429447.
A(33.)=9.7317794
33E-01
A(34.)=4342944.
A(35.)=8.1686104
59E-01
A(36.)=434294.
A(37.)=4.8188153
71E-01
A(38.)=43429.
A(39.)=0.4481901
08
A(40.)=4342.
A(41.)=9.4481903
03E-01
```

```
        NEXT 8?               0
        NEXT 8?               0
```

The result is presented as;

```
        5.662392890 3825330275    EXP 2
```

Powers and Roots

Double-precision powers and roots may be calculated, with the limitation that the numerator and denominator of the power must be one- or two-digit integers. Thus, a number may be raised to the 1/8, 23/46 or 17/1 powers, but not to powers such as 150/1 or 37/119. Violating the limitation will cause incorrect results.

The double-precision logarithm of the base must be calculated first; then the numerator and denominator of the power being taken may be entered.

As an example: to calculate the cube root of 7, set the entry up as $7^{1/3}$. First LOG 7 must be determined. The program is entered by pressing SHIFT L, then entering data as prompted:

```
        1ST 8 DIGITS?         7
        NEXT 8?               ENTER (zeros may be skipped)
```

The display will show:

```
        INT PART,             0
                              8450980400 1425683071
```

which is LOG 7. Now press the ENTER key:

```
        NUM?                  1
        DENOM?                3
```

The previously determined logarithm of 7 is multiplied by 1 and then divided by 3. Then the antilogarithm of the result is calculated. All 24 of the calculated decimal places of the logarithm are utilized, insuring that the final 20-digit result will be correct as displayed. The result is then shown as:

```
        1.912931182 7723891012    EXP 0
```

Comments

The method used by the program is simple in principle, though complex in details. The calculation of the logarithm is accomplished by a procedure essentially equivalent to the following: With X initially taken as being between 0.1 and 1.0, X is replaced by 2X, 1.1X, 1.01X, 1.001X, etc., as often as possible in each case, without the result exceeding 1. Each time this is done, LOG 2 (or LOG 1.1, LOG 1.01, etc., as the case may be) is added to an accumulation that ultimately equals LOG X. The required logarithms of 2, 1.1, etc., are furnished (in part) by the pre-stored data that resides in variables A(16) through A(41). The real beauty of this approach lies in the fact that multiplication by 2, 1.1, 1.01, etc., requires only a shift and an add operation. Incidentally, the same approach was apparently used by Sharp in designing the algorithm used in the PC to calculate LOG X. (This observation is based on otherwise unexplainable results of timing measurements.)

The antilogarithm routine essentially reverses the procedure just stated, with some variations.

A Note

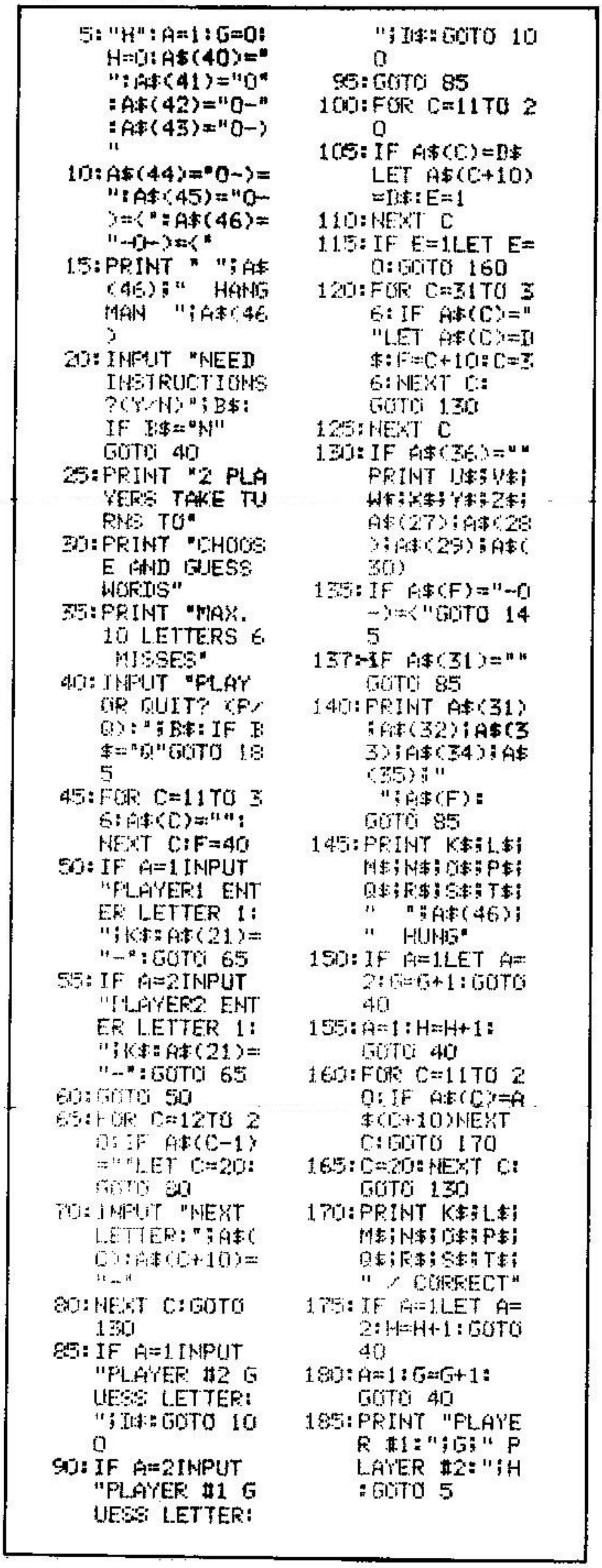
Be sure to include the data for variables A(14) through A(41) as shown in the listing!

This program submitted by: *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.*

STUMP YOUR OPPONENT WITH THIS VERSION OF HANGMAN

Gary Heidbrink, Box 24, FPO New York, NY 09593, has provided this version of the classic game *Hangman*. It is designed to be played with two people. One player thinks of a word (having up to 10 letters) and feeds it into the PC one character at a time. Pressing the ENTER key without inputting a character terminates the setting up process. A second player then tries to determine the word by guessing the letters contained in it. Each time a letter is selected, the computer will show its position(s) in the word, if it is indeed utilized. If the letter is not used, then a segment is added to the dreaded "hanging man." To help out, the computer also shows all the incorrect letters guessed each time a miss occurs. It only takes six wrong guesses to be hung. Watch out!

Program *Hangman*



```
  5:"H":A=1:G=0:
    H=0:A$(40)="
    ":A$(41)="O"
    :A$(42)="O-"
    :A$(43)="O-)
    "
 10:A$(44)="O-)=
    ":A$(45)="O-
    )=<":A$(46)=
    "-O-)=<"
 15:PRINT " ";A$
    (46);" HANG
    MAN  ";A$(46
    )
 20:INPUT "NEED
    INSTRUCTIONS
    ?(Y/N)";B$:
    IF B$="N"
    GOTO 40
 25:PRINT "2 PLA
    YERS TAKE TU
    RNS TO"
 30:PRINT "CHOOS
    E AND GUESS
    WORDS"
 35:PRINT "MAX.
    10 LETTERS 6
     MISSES"
 40:INPUT "PLAY
    OR QUIT? (P/
    Q):";B$:IF B
    $="Q"GOTO 18
    5
 45:FOR C=11TO 3
    6:A$(C)="":
    NEXT C:F=40
 50:IF A=1INPUT
    "PLAYER1 ENT
    ER LETTER 1:
    ";K$:A$(21)=
    "-":GOTO 65
 55:IF A=2INPUT
    "PLAYER2 ENT
    ER LETTER 1:
    ";K$:A$(21)=
    "-":GOTO 65
 60:GOTO 50
 65:FOR C=12TO 2
    0:IF A$(C-1)
    =""LET C=20:
    GOTO 80
 70:INPUT "NEXT
    LETTER:";A$(
    C):A$(C+10)=
    "-"
 80:NEXT C:GOTO
    130
 85:IF A=1INPUT
    "PLAYER #2 G
    UESS LETTER:
    ";D$:GOTO 10
    0
 90:IF A=2INPUT
    "PLAYER #1 G
    UESS LETTER:
    ";D$:GOTO 10
    0
 95:GOTO 85
100:FOR C=11TO 2
    0
105:IF A$(C)=D$
    LET A$(C+10)
    =D$:E=1
110:NEXT C
115:IF E=1LET E=
    0:GOTO 160
120:FOR C=31TO 3
    6:IF A$(C)="
    "LET A$(C)=D
    $:F=C+10:C=3
    6:NEXT C:
    GOTO 130
125:NEXT C
130:IF A$(36)=""
    PRINT U$;V$;
    W$;X$;Y$;Z$;
    A$(27);A$(28
    );A$(29);A$(
    30)
135:IF A$(F)="-O
    -)=<"GOTO 14
    5
137:IF A$(31)=""
    GOTO 85
140:PRINT A$(31)
    ;A$(32);A$(3
    3);A$(34);A$
    (35);"
     ";A$(F):
    GOTO 85
145:PRINT K$;L$;
    M$;N$;O$;P$;
    Q$;R$;S$;T$;
    "  ";A$(46);
    "  HUNG"
150:IF A=1LET A=
    2:G=G+1:GOTO
    40
155:A=1:H=H+1:
    GOTO 40
160:FOR C=11TO 2
    0:IF A$(C)=A
    $(C+10)NEXT
    C:GOTO 170
165:C=20:NEXT C:
    GOTO 130
170:PRINT K$;L$;
    M$;N$;O$;P$;
    Q$;R$;S$;T$;
    " / CORRECT"
175:IF A=1LET A=
    2:H=H+1:GOTO
    40
180:A=1:G=G+1:
    GOTO 40
185:PRINT "PLAYE
    R #1:";G;" P
    LAYER #2:";H
    :GOTO 5
```

AND THE WINNER IS . . . .

A big winner! Yes, *David Motto, 3639 Roosevelt Circle, Jackson, MI 49203* took all three prizes in our *Hot Steaming Contest* announced in Issue 07 of *PCN.* His winning entries, in order of selection, were:

INPUT C TO CLEAR
RETURN FOR USING PRINT ER
SIN ↑COS SIN AINT NEW

David earned a check for $45.00 with these winning entries. Now, aren't you sorry you didn't send in those clever lines you created?

HAVING PROGRAM LOADING PROBLEMS?

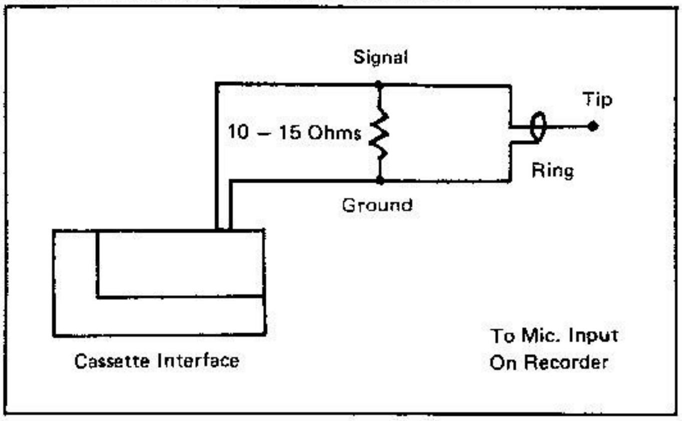
Peter Korda, % Price Systems, 6363 Sunset Boulevard, Suite 529, Hollywood, CA 90028, forwarded the following information for users of the Sharp PC-1211 and Radio Shack TRS-80 PCs.

If, after successfully performing a CSAVE operation to record a program on a cassette, you have not been able to load the program using CLOAD because the PC does not recognize the end of the program on the tape, you may have a problem with the record signal overloading the tape recorder.

After experimenting with a variety of different brands of cassette recorders, including those by Sony, Panasonic, JVC and Radio Shack, Peter believes the problem is caused by excessive amplitude from the computer interface to the recorder. The signal then overloads the automatic gain control of the cassette recorder. He suggests a simple solution to the problem. Connect a 10 to 15 ohm resistor between the signal line and the ground line of the output from the interface. Peter notes that the resistor, in addition to attenuating the signal, provides a load to the interface output which attenuates spurious noise spikes.

As indicated by the accompanying diagram, the resistor can be installed in a patch cord that runs between the red jack of the interface and the microphone input of the tape recorder. Fifteen minutes of effort and an 8-cent resistor might make PC recordings easier if you are experiencing this type of problem.

Figure *Schematic of Cassette Interface Attenuator*



DO YOU WANT AN INDEX?

We are considering putting an index for Issues 01 – 10 of *PCN* in the next issue. Would you like to see such an index or do you think it would be a waste of valuable space? Your opinion counts, so let us know if you feel an index is desirable. Hurry!

P.O. Box 232, Seymour, CT 06483

FROM THE WATCH POCKET

Would you believe about 250,000 PCs sold to date? That is the consensus these days. And now with Casio on board and others soon to join . . . .

Blanton Software Service offers a number of PC programs in addition to the one I reviewed elsewhere in this issue. All of their programs are optionally available on cassette. The prices generally range from $5 to $8. Write to the firm at the address given in the review for a current catalog. Of the programs I examined, I was favorably impressed with *Electronic Checkbook, Appointment Calendar, Countdown Timer/Clock/Alarm,* and the *Travel/Business Expenses* program. I was not impressed by Blanton's version of the card game *Blackjack.* Neither the display format nor speed of execution was to my liking. (In other words, I have seen better on a PC.) This is purely a subjective opinion. I saw nothing wrong with the technical execution of the program. All-in-all, I feel this is a software firm worth consideration if you are in the market for programs that can be supplied on cassettes.

If you are a beginner to the use of a PC and programming using the BASIC language, write to **XCEL**, *13763 Polk Street, Sylmar, CA 91342* and ask for information on their new tutorial *Beginner's Guide to Programming.* This consist of a book in combination with several packs of "program" and "task" cards. These have been integrated to form a do-it-yourself course in programming. The price is $19.00.

Another new outfit under the name **PersaSoft, Inc.**, *9609 S.W. Kelly, Portland, OR 97219,* has announced a line of software for the Sharp PC-1211/Radio Shack TRS-80 Pocket Computer. They have announced the availability of a package called *Beat-The-Market* that is aimed at stock market investors. Total price of the package on six cassettes is $75.00. Individual parts of the package are available separately. Other packages, including a series named *Creative Real Estate Financing* are reportedly under development. Write to the company for further information.

The September 22 issue of **Electronics** carries another pocket computer program by *Cass R. Lewart.* This one analyzes problems in classical queuing theory. Looks useful for engineers/scientists/statisticians. See page 157.

November **Interface Age** has *two* (!) articles related to PCs. One is the regular *Bob McElwain* column in which he presents an inventory estimating program. The other is an article by *Alan R. Miller* in which he introduces the printer for the Radio Shack TRS-80 PC. (Do you remember when *PCN* introduced the printer? See the difference between a newsletter and a magazine?) That same article presents a Shell-Metzner sorting routine for the PC. The significant point is that at least a magazine devoted two whole articles to PCs. Think they could be catching on?

Not to be left out of the field, **80 Microcomputing** has an article in the October issue that presents a nice three-part program. The first part finds the term of a loan. The second determines the periodic payment. The last section will determine the true annual rate of interest. This is a nicely written package by *Walter J. Atkins, Jr. Ph.D.* Nicely presented, too!

David Motto, whose review of the Casio FX-702P graces these pages, seems basically pleased with the new machine. He does note, however, in a recent communication, that the unit has failed to power-up correctly a number of times. This is cleared by recycling the ON-OFF switch. But, he has asked the firm for comment. Several other users have not mentioned the problem. The most common statement heard concerns satisfaction with the speed of the unit. Others have complained they cannot obtain the unit.

Try **Atlantic N.E. Marketing** if you are interested in a Casio. The phone number is (617) 639-0285.

Have you heard the rumor about Sharp Electronics running field trials of a 16K version of its PC? One international traveler is reported as actually seeing the unit — complete with a considerably enhanced printer that has multiple-color capability. Now a 16K unit would be nice. Yes, that would be nice.

Expect a surprise from *PCN* during November. I hope you like it.

— Nat Wadsworth, Editor

POCKET COMPUTER NEWSLETTER

© Copyright 1981 — Issue 10

December

PANASONIC TO MARKET A HAND HELD COMPUTER

At a press conference held in New York City on November 10, 1981, Panasonic Company announced that it was ready to begin U.S. marketing of a hand held computer.

The unit is described by Panasonic as being approximately the size of a paperback book. Its actual measurements are 8-5/16 by 1-3/16 by 3-3/4 inches. Its weight is listed at 14 ounces.

The hand held computer sports a 65 key typewriter style keyboard featuring two-key rollover operation. The keys may be redefined using software. The single line display is made up of 159 columns of dots arranged as 8 rows which allows up to 26 characters to appear at one time. Special characters can be created with software.

Two versions of the hand held computer (HHC) have been announced. The RL-H1000 contains 2K of user memory. The RL-H1400 has 4K of memory that is programmable. In addition to the RAM elements, both units are capable of holding up to three ROM elements that can contain high level languages, operating systems and customized programs.

The new HHCs are claimed to be exceedingly powerful and the firm made it a point to state that the unit should not be confused with so-called *pocket computers* now available on the market. Panasonic representatives pointed out that their unit was capable of addressing hundreds of thousands — even megabytes — of memory. They also stated that a whole host of peripheral devices could be connected to the device, including in the near future, a floppy disk unit capable of running under the popular CP/M operating system.

Initially, however, the company stated that it would be providing six basic peripheral devices:

A video/rf adaptor capable of driving a color television display in a text mode of 16 lines by 32 characters or a graphics mode having 48 by 64 picture elements in eight colors.

A thermal printer that prints 15 characters per line.

An RS-232c serial interface through which one may connect various types of ~~communications devices.~~

A programmable accoustic modem.

RAM expansion units allowing the addition of user memory in 4K or 8K blocks.

An I/O adaptor through which up to six other modules may be connected to the HHC.

All of the peripherals announced are small in size so that a complete system may be connected together while residing in a briefcase. Additionally, all of the units except the television interface operate off of the HHC's built-in battery supply. The TV interface utilizes an AC acaptor that powers the unit while recharging the HHC's batteries. The 44 pin bus system used to interconnect units has been specially designed so that units may be attached/detached without harming data in other units. Loose connections are signaled by an audio beeper.

Software modules announced for the system include a four-function calculator, an editor and a free-form file system. The unit also features an internal real-time clock with an accuracy of 1/256 of a second. Versions of BASIC and SNAP (a variation of FORTH) will also be available for the system.

The RL-H1000 unit having 2K of user memory has a list price of $500.00. The 4K RL-H1400 lists at $600.00. Peripheral units range from a list price of $158.00 for the I/O adaptor to $349.00 for the video/rf adaptor. A typical system price is likely to be in the range of $1500.00 to $2000.00.

For additional information contact: *Panasonic Company, One Panasonic Way, Secaucus, NJ 07094.*

Photo *The Panasonic Hand Held Computer*

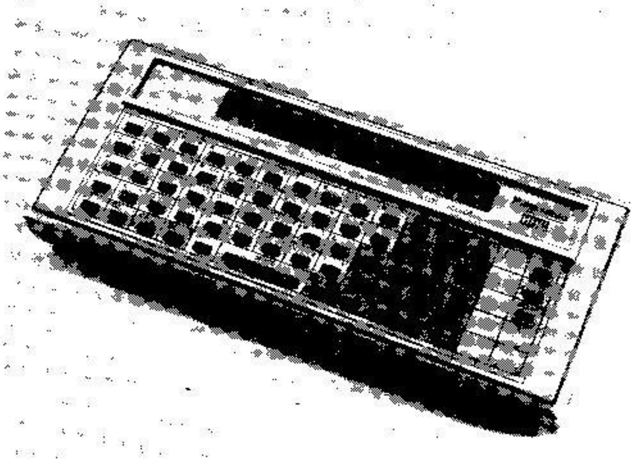
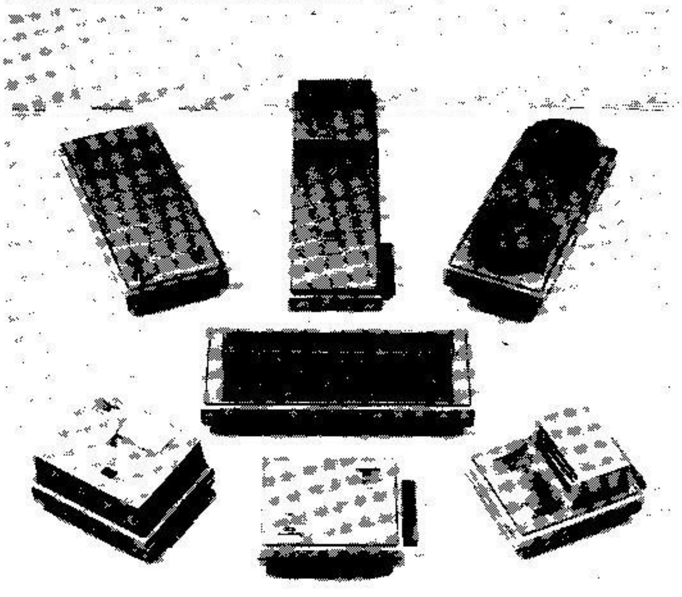


Photo *The Panasonic HHC Surrounded by Peripherals*



Another *Personal Information*™ product from SCELBI Publications.

PRODUCT REVIEW
MATHEMATICS DRILL PACKAGE

Produced by: *Radio Shack for the TRS-80 Pocket Computer.*
List price: *$14.95*
Availability: *Radio Shack Computer Centers and selected Radio Shack stores.*
Reviewer: *C. C. Vans Evers II, 3607 Oklahoma, Tampa, FL 33611.*

Overview

The mathematics exercises contained in the three programs that make up the package *Mathematics Drill* are similar to those found in such single-purpose units as Texas Instruments' *Little Professor* and *Dataman.* That is, the user can specify the kind of operation (addition, subtraction, multiplication or division) and level of difficulty desired.

Although the Radio Shack programs provide more prompting, the execution speed of the PC software versus firmware is quite evident to those who have used a *Dataman* or similar unit. Inherent in the PC is a certain slowness that only careful programming can minimize.

The programming of the three programs (ED1, ED2 and ED3) is straightforward, using subroutine calls which account for quite a bit of the delay between inputs and displays. (As a matter of fact, the slow execution got boring to both my daughters, ages 7-1/2 and 12.)

One game is included. More on that later.

The programs use virtually all of memory. Three-digit line numbers predominate. Either initial use of, or conversion to, two-digit line numbers on final debug would have helped, as well as attempts to clean up some sections of code. For example, there is a GOTO in ED1 to a nonexistent line number. Fortunately, the statement will never be executed in normal operation.

ED1

The first program is composed of three parts, each of which is subdivided so that addition, subtraction, multiplication or division and one of four levels of difficulty may be selected. (This is standard format throughout the package.) *Math Test* prompts the user to answer 10 items using the chosen operation. Three tries are allowed in which to input a correct answer, after which the program supplies the result.

Math Tables gives the user a choice of operators and a base number. The program then displays a number from 0 to 12, the selected operator, and the base value. In this mode, the correct answer is provided by the computer.

Tables Test is identical to *Math Tables* except that the user is expected to enter the correct answer.

One confusing point for adult users who are familiar with computers is that the ENTER key is not used after inputting answers. Instead the SHIFT = key is utilized. This didn't bother my children, but I was simply used to using ENTER.

This program, ED1, is no more and no less than what is available with single-purpose hand-helds, except that it is noticeably slower.

ED2

In its hurry to get software on the market, Radio Shack chose to substitute the well known *High-Low* game in place of the *Math Test* code of ED1. The remainder of ED2 is identical to ED1. For the most part, *High-Low* executes fairly quickly because the use of subroutine calls is limited.

The user has eight tries to guess a number between 0 and 100. If the (also!) well known strategy of what one might call a quasi-binary search is employed, the user can't lose. My 7-1/2 year old had to be told. My 12 year old figured it out fairly quickly. That's typical for those ages.

It is a shame that another numbers-oriented game wasn't paired with *High-Low* rather than what was done (or rather, *not done,* in this case). Perhaps *High-Low* could be used as a reward for doing all math problems correctly.

ED3

The third program in the package permits the user to specify two operands and an operator (+, -, * or /) for up to twelve problems. The problems are then displayed for the user to supply the proper answer.

Summary

With more compact and cleaner coding to eliminate what I would call bugs or inelegant techniques, the package would have greater execution speed and improved appeal.

For example, if division problems are chosen, the program prompts for the remainder. The remainder must then be entered, even if it is zero. Minor code modification could take care of that situation.

As is the case with most Radio Shack PC software available so far, the PC printer cannot be used with the package unless there is extensive program editing. Menus are displayed for each program. Since the menus are composed of PRINT statements, an endless loop occurs when the PC printer is activated.

If your children already have stand-alone math drill units, they already have what this package offers. However, if you are interested in studying the coding for this type of program or you really want a *High-Low* game, then consider this package.

PRODUCT REVIEW
BEGINNERS GUIDE TO PROGRAMMING

Author: *Bob McElwain*
Publisher: *XCEL, 13763 Polk Street, Sylmar, CA 91324*
List price: *$19.00*
Availability: *Directly from the publisher.*
Reviewer: *Nat Wadsworth, Editor.*

The full title of this educational package is the *Beginners Guide to Programming for the Sharp PC-1211 and Radio Shack TRS-80 PC.* The package is a book in combination with several packs of cards called "Program Cards" and "Task Cards." These are integrated to form a do-it-yourself course in programming.

The book contains 128 pages, is 8-1/2 by 11 inches and softcover. The text appears to have been set using a high quality typewriter at 1-1/2 line spacing. While this may not seem impressive, I actually found the format highly readable. The extra line spacing makes it easy to keep your place when working on problems.

This is definitely a "hands-on" book. The bulk of it is devoted to "lessons" that fill you in on specific aspects of developing programs. For instance, in Lesson 1 you learn about the PAUSE and PRINT directives. Lesson 2 takes you into "counting." Lesson 3 into "testing" and so forth. These lessons are designed to teach fundamental programming concepts while drilling a student on the practical aspects of implementing the concepts.

Thus, in the section on counting, concepts such as J=J+1 are integrated with the practical matters of assigning variables (in which to maintain the counts), the use of flowcharts, identifiers, the use of commas and semicolons in a PRINT statement, the importance of properly identifying results to the operator, the definition of an assignment statement, and so forth.

As each lesson is worked on, the student refers to separate "Program Cards" and performs the operations detailed. Each card presents and demonstrates one or two programming steps. The cards build upon one another in sequence.

For instance, the cards used in Lesson 1 start with a two-line program that displays a person's name. Other cards lead the user through the process of editing a program, correcting errors, inserting and deleting lines, and illustrate how a program can be made to "branch" to another section of itself.

The use of these cards definitely spices up the learning process.

A second group of cards for each lesson are referred to as "Task Cards." These cards can really be considered as mini-quizzes. They challenge the user to program the computer to perform a specific task or function. The reverse side of each card provides hints on how to approach the problem in case the student is stumped. The level of complexity of these cards relates directly to the progress made previously.

There are approximately 40 two-sided "Program Cards" and 20 "Task Cards" spread over seven lessons. These are in turn backed by about 40 pages of text in the book.

The next approximately 55 pages in the book build upon the basic concepts and principles developed earlier to work up three complete programs: the game HILO, a program called CAR that keeps track of travel expenses, and a rather substantial CHECKBOOK program. Each program is developed in a step-by-step expansion fashion. Fundamental capabilities are developed and then amplified by inserting additional routines.

The book is rounded out with a list of approximately 50 programming problems for those that want additional programming practice. While solutions to these problems are *not* provided in the book, they can be obtained from the publisher at a current charge of $4.00. (It appears as though the premise here is that the book may be used as a school text and that the solutions to these problems are supplied separately, if they are desired, as a teacher's reference.)

Finally, about 20 pages of the text book are devoted to illustrating solutions to the problems presented on the "Task Cards."

Conclusion

Many people have written *PCN* asking if anyone produced a good introductory text on programming using BASIC on a PC. This package may not be fancy in appearance, but its contents and methods are good in application. If you are a beginner to the art of programming a pocket computer, I would recommend this package. School teachers might be wise to examine this package as a possible text or teaching supplement, too. It takes patience for an expert to develop and write this type of course. I hope students of this package will recognize the talent of its author.

PRODUCT REVIEW
101 POCKET COMPUTER PROGRAMMING TIPS & TRICKS

Author: *Jim Cole*
Publisher: *ARCsoft Publishers, P.O. Box 132, Woodsboro, MD 21798*
List Price: *$7.95*
Availability: *Directly from the publisher.*
Reviewer: *Nat Wadsworth, Editor.*

An advertising blurb on the back of this publication, after listing a number of the routines contained therein, states that ". . . . dozens more are in this handy sourcebook of notes and ideas from a programming pro, for the newcomer, novice programmer, layman and general consumer user." This certainly appears to be a truthful statement. This book is not intended for expert programmers. It is, however, full of tips and ideas for beginning programmers.

Some tips appear rather trivial. For instance, almost a page of text is devoted to touting the excitement of a flashing "billboard" which leads to the presentation of the one line statement:

```
100 FOR L=1 TO 3:PAUSE " *RED ALERT *":NEXT L
```

Few *PCN* readers are likely to find this a new or startling "tip." It is granted, however, that a novice programmer who had missed the joy of discovering such operation on his/her own, would certainly be able to pick up on the idea after the build up it receives in this manual!

Other examples are introduced with less build up and are, indeed, less trivial. Jim Cole illustrates how to program a look-up table, methods of generating pseudorandom numbers, isolating the decimal portion of a number, various ways to round off numbers, error trapping, jumping to labels, ways of exiting loops, and other operations. These examples present important concepts in the context of practical applications and are of particular value to beginning programmers.

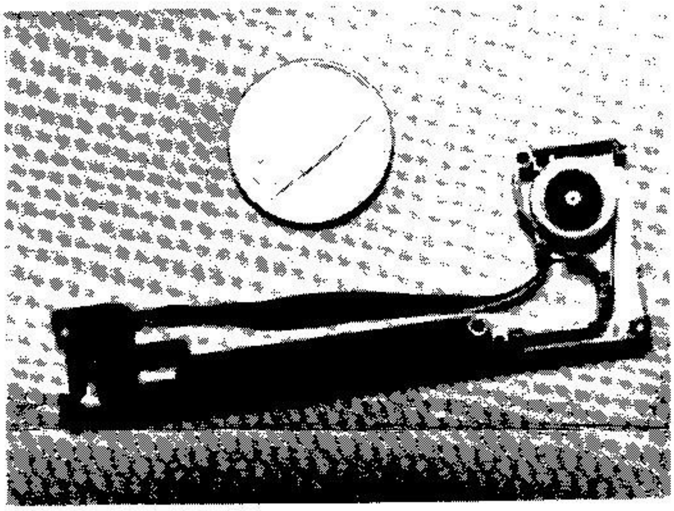
No, this is not a book for programming experts who think they may have missed little goodies in their PC's operating manuals or those who are seeking esoteric capabilities. It is a book for programming novices — and possibly intermediates — who need to explore PC programming ideas and concepts. At a price of less than eight cents an idea, this publication is certainly worthy of consideration.

Read any good books on the use of pocket computers? Be sure to tell us about them so that we can spread the word. Thanks!

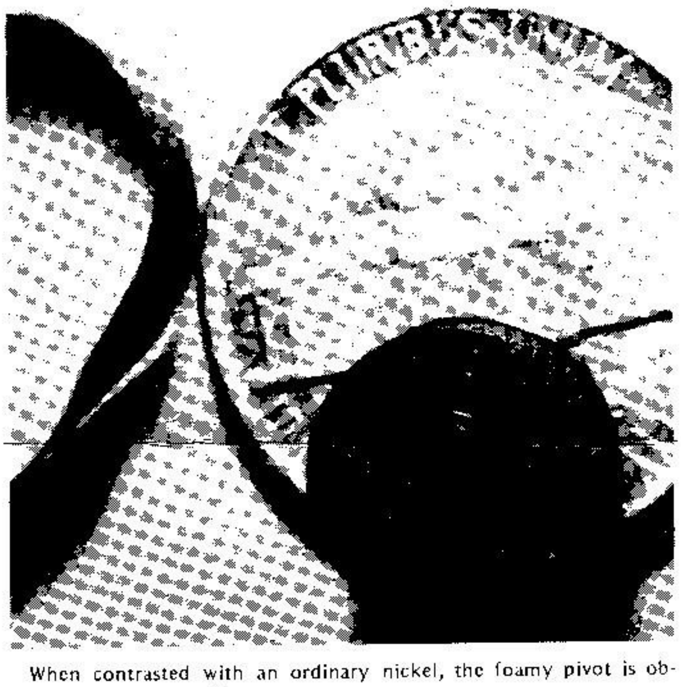
INSIDE THE SHARP CE-122 PRINTER RIBBON CARTRIDGE

The printer for the Radio Shack and Sharp pocket computers uses a miniscule ribbon cartridge employing a continuous loop of ribbon.

A close look at the underside of the cartridge shows the trademark *Epson*. This is undoubtably the same company that manufactures the well respected MX series printers for desktop computers.



The inside of the cartridge contains an endless loop of ribbon that is continually re-inked by a foamy pivot. This pivot may be seen in the upper right portion of the cartridge as shown in the accompanying photograph. This picture shows the cartridge with its bottom cover removed.



When contrasted with an ordinary nickel, the foamy pivot is observed to be quite small.

At around $3.00 for a ribbon cartridge, many people have been opting to re-ink the ribbon directly or by placing a drop of ink on the foam element. The manufacturer does not recommend this because the ribbon will eventually fray. A frayed ribbon may eventually snag the printing pins and cause severe damage to the printer.

Perhaps someone will put together a package with a small vial of ink and a supply of ribbon loops at an inexpensive price?

This photo-story by: *Ken Slaughter, 2916 Bangor Avenue, Highland, CA 92346.*

NUMERICAL INTEGRATION ROUTINES

The routines presented here were provided by *Norlin Rober, 407 North 1st Avenue, Marshalltown, IA 50158.* They may be used to approximate the integral from a to b of f(x) dx. Each program calls upon a subroutine that is assumed to start at line 90. This subroutine, provided by the user, must compute Y as a value of X. The variables A, B, N, S, W and Z should not be altered by the user when interfacing with the routines.

(1) Trapezoidal Rule

```
10 INPUT "A?";A,"B?";B
11 INPUT "N?";N
12 X=A:GOSUB 90:S=.5Y:X=B:GOSUB 90:S=S+.5Y:FOR Z=1 TO
   N-1
13 X=A+(B-A)Z/N:GOSUB 90:S=S+Y:NEXT Z:S=(B-A)S/N:BEEP 1
   :PRINT S:END
```

(2) Simpson's Rule

```
20 INPUT "A?";A,"B?";B
21 INPUT "N?";N
22 X=A:GOSUB 90:S=Y:X=B:GOSUB 90:S=S+Y:W=2:FOR Z=1 TO
   N-1
23 X=A+(B-A)Z/N:GOSUB 90:W=6-W:S=S+WY:NEXT Z
24 S=(B-A)S/3N:BEEP 1:PRINT S:END
```

(3) Three-eighths Rule

```
30 INPUT "A?";A,"B?";B
31 INPUT "N?";N
32 X=A:GOSUB 90:S=Y:X=B:GOSUB 90:S=S+Y:Z=N-1
33 GOSUB 35:S=S+3Y:GOSUB 35:S=S+3Y:IF Z GOSUB 35:S=S+2Y
   :GOTO 33
34 S=3S/8N*(B-A:BEEP 1:PRINT S:END
35 X=A+(B-A)Z/N:Z=Z-1:GOTO 90
```

(4) Bode's Rule

```
40 INPUT "A?";A,"B?";B
41 INPUT "N?";N
42 X=A:GOSUB 90:S=7Y:X=B:GOSUB 90:S=S+7Y:Z=N-1
43 GOSUB 45:S=S+32Y:GOSUB 45:S=S+12Y:GOSUB 45:S=S+32Y
   :IF Z GOSUB 45:S=S+14Y:GOTO 43
44 S=(B-A)S/22.5N:BEEP 1:PRINT S:END
45 X=A+(B-A)Z/N:Z=Z-1:GOTO 90
```

(5) Weddle's Rule

```
50 INPUT "A?";A,"B?";B
51 INPUT "N?";N
52 X=A:GOSUB 90:S=Y:X=B:GOSUB 90:S=S+Y:Z=N-1
53 GOSUB 56:S=S+5Y:GOSUB 56:S=S+Y:GOSUB 56:S=S+6Y
   GOSUB 56:S=S+Y:GOSUB 56:S=S+5Y
54 IF Z GOSUB 56:S=S+2Y:GOTO 53
55 S=.3S/N*(B-A:BEEP 1:PRINT S:END
56 X-A+(B-A)Z/N:Z=Z-1:GOTO 90
```

(6) Midpoint Rule

```
60 INPUT "A?";A,"B?";B
61 INPUT "N?";N
62 S=0:FOR Z=1 TO N
63 X=A+(B-A)/N*(Z-.5:GOSUB 90:S=S+Y:NEXT Z:S=(B-A)S/N
   :BEEP 1:PRINT S:END
```

Choosing Subintervals

The number of subintervals used, N, must be chosen as follows:

Trapezoidal Rule: Any integer, 2 or greater.
Simpson's Rule: Any even integer, 2 or greater.
Three-eighths Rule: Any positive integral multiple of 3.
Bode's Rule: Any positive integral multiple of 4.
Weddle's Rule: Any positive integral multiple of 6.
Midpoint Rule: Any positive integer.

Example Solutions

As an illustration, consider the approximation of the integral over the interval 0 to 1 of (1/1+X^{2})dx. Analytical methods show that the exact value of the integral is pi/4.

In performing the tests, line 90 contained the subroutine:

```
90 Y=1/(1+XX:RETURN
```

The values of A and B for the desired interval are 0 and 1 respectively. If the value of N is selected as 12, the various approximation routines yield the following results:

Trapezoidal Rule:	.7851088118
Simpson's Rule:	.7853981600
Three-eighths Rule:	.7853981484
Bode's Rule:	.7853981744
Weddle's Rule:	.7853981693
Midpoint Rule:	.7855428391
Pi/4 actually equals:	.7853981634

Bonus Routine: Gaussian Quadrature

The following routine may be used to approximate an integral using a 16-point Gaussian quadrature. An advantage of this method over those previously described is that the function need not be defined at the end points of the interval. As in the previous routines, a subroutine beginning at line 90 should calculate Y as a function of X.

```
80 INPUT "A?";A,"B?";B
81 S=0:W=.09501250984:Z=.1894506105:GOSUB 86:W=.281603
   5508:Z=.182603415:GOSUB 86
82 W=.4580167777:Z=.1691565194:GOSUB 86:W=.6178762444
   :Z=.1495959888:GOSUB 86
83 W=.7554044084:Z=.1246289713:GOSUB 86:W=.8656312024
   :Z=.09515851168:GOSUB 86
84 W=.9445750231:Z=.06225352394:GOSUB 86:W=.989400935
   :Z=.02715245941:GOSUB 86
85 S=.5S*(B-A:BEEP 1:PRINT S:END
86 X=.5*((B-A)W+A+B:GOSUB 90:X=.5*((A-B)W+A+B:W=Y
   :GOSUB 90:S=S+Z*(W+Y:RETURN
```

The example used earlier yields .7853981635 by this method.

PROBABILITY FUNCTIONS TO 10 PLACES

Norlin Rober presents the following routines that calculate the probability that a random variable having the stated probability distribution will exceed a given value. The routines calculate ten digits following the decimal point, with the tenth digit being considered unreliable. Parameter values are assumed to be stored in the appropriate variable locations when the routines are entered.

Normal Distribution

Given Z, the probability P of exceeding Z is calculated. Any value of Z may be used. Execution time varies to a maximum of 30 seconds.

```
10 Y=ZZ:IF Y>41 LET P=SGN Z/2:GOTO 13
11 P=1:FOR X=9+2*INT 8Z to 3 STEP -2
12 P=PY/X+1:NEXT X:P=PZ/√(2π*EXP Y
13 P=.5-P
```

The routine uses 82 steps and variables P, X, Y and Z.

Student's T-Distribution

Given T and N, the number of degrees of freedom, the probability of exceeding T is calculated. Any value of T may be used. Execution time depends on N. It is approximately N/3 seconds. The computer must be in the DEGree mode.

```
20 P=(N>1:X=N/(N+TT:Y-P
21 IF N>2 FOR Y=N-3 TO 2 STEP -2:P=1+PXY/(Y+1:NEXT Y
22 IF Y-1 LET P=.5PT/√(N+TT:GOTO 24
23 P=PTX/√N/π + ATN(T/√N)/180
24 P=.5-P
```

The routines takes up 116 steps and uses variables N, P, T, X, Y and Z.

Chi-Square Distribution

Given C (Chi-squared) and N (the number of degrees of freedom), the probability P of exceeding C is calculated. The maximum usable value of N (to prevent overflow) is 120. Execution time varies considerably. It becomes long if N is large or if C is large compared to N.

```
30 P=0:X=N:Y=1:IF C=0 GOTO 34
31 Z=P:P=P+Y:X=X+2:Y=CY/X:IF P >Z GOTO 31
32 Y=1:FOR X=N TO 2 STEP -2
33 Y=XY:NEXT X:P=P/Y*EXP(.5N*LN C-.5C: IF X=1 LET P=P/
   √.5 π
34 P=1-P
```

The routine requires 122 steps and uses variables C, N, P, X, Y and Z.

Notes

The routines illustrated may be incorporated into programs involving statistical analysis in which the level of significance is sought. The need for tables is thus eliminated while achieving increases in accuracy and versatility.

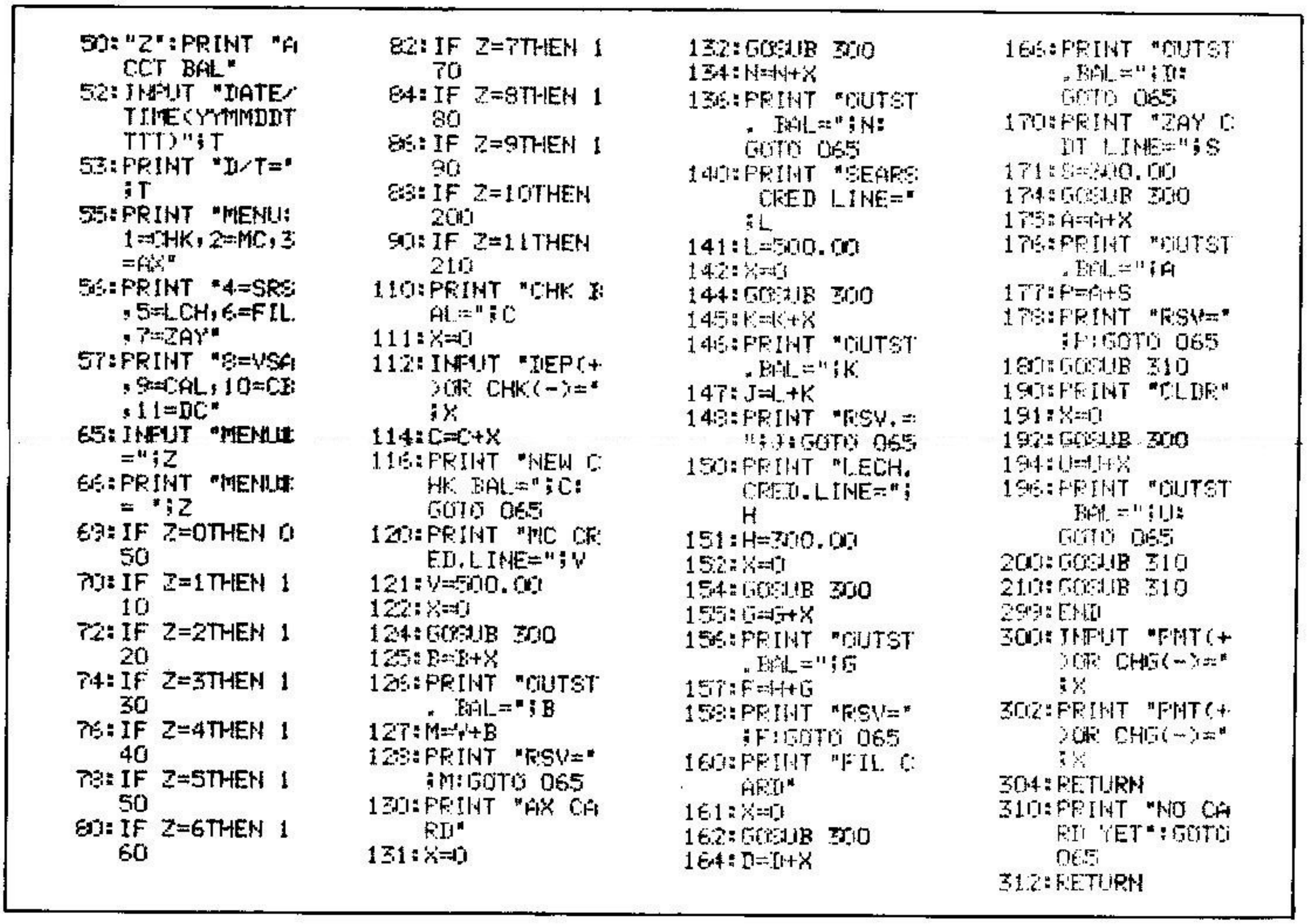
Test Examples

Normal probability: Z=2: Calculated value of P=.022750132
Student's T: N=1: T=2: Calculated value of P=.1475836176
N=4: T=5: Calculated value of P=.0037452171
Chi-square: N=1: C=4: Calculated value of P=.0455002649
N=2: C=5: Calculated value of P=.0820849981
N=25: C=42: Calculated value of P=.017972808

PROGRAM KEEPS TRACK OF BANK & CREDIT CARD BALANCES

Arthur C. LeFort, Jr., 174 Colburn Street, Dedham, MA 02026, provides this program for Radio Shack and Sharp PCs. Now you won't have any excuse for overdrawing your account or exceeding your credit limit! By the way, Art specifically designed this program so that you can use a printer if you need to have a permanent record of your account balances.

Program *Account Balances*



```
50:"Z":PRINT "A
    CCT BAL"
52:INPUT "DATE/
    TIME(YYMMDDT
    TTT)";T
53:PRINT "D/T="
    ;T
55:PRINT "MENU:
    1=CHK,2=MC,3
    =AX"
56:PRINT "4=SRS
    ,5=LCH,6=FIL
    ,7=ZAY"
57:PRINT "8=VSA
    ,9=CAL,10=CB
    ,11=DC"
65:INPUT "MENU#
    =";Z
66:PRINT "MENU#
    = ";Z
69:IF Z=0THEN 0
    50
70:IF Z=1THEN 1
    10
72:IF Z=2THEN 1
    20
74:IF Z=3THEN 1
    30
76:IF Z=4THEN 1
    40
78:IF Z=5THEN 1
    50
80:IF Z=6THEN 1
    60
 82:IF Z=7THEN 1
    70
 84:IF Z=8THEN 1
    80
 86:IF Z=9THEN 1
    90
 88:IF Z=10THEN
    200
 90:IF Z=11THEN
    210
110:PRINT "CHK B
    AL=";C
111:X=0
112:INPUT "DEP(+
    )OR CHK(-)="
    ;X
114:C=C+X
116:PRINT "NEW C
    HK BAL=";C:
    GOTO 065
120:PRINT "MC CR
    ED.LINE=";V
121:V=500.00
122:X=0
124:GOSUB 300
125:B=B+X
126:PRINT "OUTST
    . BAL=";B
127:M=V+B
128:PRINT "RSV="
    ;M:GOTO 065
130:PRINT "AX CA
    RD"
131:X=0
132:GOSUB 300
134:N=N+X
136:PRINT "OUTST
    . BAL=";N:
    GOTO 065
140:PRINT "SEARS
     CRED LINE="
    ;L
141:L=500.00
142:X=0
144:GOSUB 300
145:K=K+X
146:PRINT "OUTST
    .BAL=";K
147:J=L+K
148:PRINT "RSV.=
    ";J:GOTO 065
150:PRINT "LECH.
    CRED.LINE=";
    H
151:H=300.00
152:X=0
154:GOSUB 300
155:G=G+X
156:PRINT "OUTST
    .BAL=";G
157:F=H+G
158:PRINT "RSV="
    ;F:GOTO 065
160:PRINT "FIL C
    ARD"
161:X=0
162:GOSUB 300
164:D=D+X
166:PRINT "OUTST
    .BAL=";D:
    GOTO 065
170:PRINT "ZAY C
    DT LINE=";S
171:S=200.00
174:GOSUB 300
175:A=A+X
176:PRINT "OUTST
    .BAL=";A
177:P=A+S
178:PRINT "RSV="
    ;P:GOTO 065
180:GOSUB 310
190:PRINT "CLDR"
191:X=0
192:GOSUB 300
194:U=U+X
196:PRINT "OUTST
     BAL=";U:
    GOTO 065
200:GOSUB 310
210:GOSUB 310
299:END
300:INPUT "PMT(+
    )OR CHG(-)="
    ;X
302:PRINT "PMT(+
    )OR CHG(-)="
    ;X
304:RETURN
310:PRINT "NO CA
    RD YET":GOTO
    065
312:RETURN
```

UNLIMITED NESTING OF SUBROUTINES

Several readers have noted the fact that variables may be used in the Radio Shack/Sharp PCs to facilitate nesting of subroutines to more than the four levels permitted by the unit's internal subroutine stack. The procedure to use is simply assign the value of the next line number to be executed after returning from a subroutine call to a variable name. Use a GOTO directive to jump to the start of the desired subroutine. Use the directive GOTO VARIABLE to terminate the subroutine instead of the usual RETURN statement. An example:

```
100  Z=110:GOTO 900
110  .... main program continues from here ....
.....
900  .... subroutine starts here ....
.....
990  GOTO Z
```

Directed GOTOs Also Possible

Along these same lines of using a variable as a label, you can create directed GOTO statements. Say, for instance, that a variable can only take on integer values from 1 to 5. A different operation is to be performed for each value that the variable assumes. The following arrangement permits five different routines to be established starting at lines 100, 200, . . . etc., and gives the program capability to branch immediately to the proper segment:

```
  10  .... main program begins ....
  50  .... variable Z set to value 1 to 5 by user input here ....
  60  Z=Z*100:GOTO Z
.....
100  .... routine executed when Z = 1 ....
200  .... routine executed when Z = 2 ....      etc.
```

TIME ADDITION

Submitted by: *David G. Motto, 3639 Roosevelt Circle, Jackson, MI 49203.*

This routine was developed for a Radio Shack TRS-80 PC and may be used with or without the printer unit. The routine accepts inputs in the form of MM.SS, where MM stands for minutes and SS for seconds. Use the ENTER key after each timing input. To obtain a tally of all times, press the ENTER key again.

Program *Time Addition*

```
100 B=0
110 A=0:INPUT "TIME? ";A
120 IF A=0 GOTO 150
130 B=B+DEG A
140 GOTO 110
150 B=1000*DMS B
160 C=B/100:IF (C-INT C)>.3
    LET C=C+1
170 B=INT C/100
180 PRINT "TOTAL: ";B:GOTO 100
```

BLACKJACK

As submitted by: *Brian Peterson, 6807 N. Sheridan Rd., Apt. 520, Chicago, IL 60626.*

```
  1:INPUT "SEED=
    ";S:X=E2:W=0
    :R=-1
  2:BEEP 2:IF -R
    PAUSE "SHUFF
    LING...":FOR
    R=27TO 39:A(
    R)=5:NEXT R:
    IF W=0GOTO 2
    5
  3:PRINT "YOU:"
    ;A$;B$;C$;D$
    ;E$;F$;"DEAL
    ER:";G$;H$;I
    $;J$;K$;L$;M
    $:RETURN
  4:W=ABS (43914
    7+W+S:W=23W-
    E8*INT (23W/
    E8:N=26+INT
    (13W/E8+1:A(
    N)=A(N)-1:IF
    A(N)LET R=R-
    1:GOTO N-21
  5:GOTO 4
  6:A$(P)="A":V=
    1:O=10:
    RETURN
  7:A$(P)="2":V=
    2:RETURN
  8:A$(P)="3":V=
    3:RETURN
  9:A$(P)="4":V=
    4:RETURN
 10:A$(P)="5":V=
    5:RETURN
 11:A$(P)="6":V=
    6:RETURN
 12:A$(P)="7":V=
    7:RETURN
 13:A$(P)="8":V=
    8:RETURN
 14:A$(P)="9":V=
    9:RETURN
 15:A$(P)="T":V=
    10:RETURN
 16:A$(P)="J":V=
    10:RETURN
 17:A$(P)="Q":V=
    10:RETURN
 18:A$(P)="K":V=
    10:RETURN
 19:INPUT "INSUR
    ANCE?";N$:IF
    N$="N"RETURN
 20:N=INT .5Y:X=
    X-N:PAUSE "Y
    OU LOSE $";N
    :IF U+O<>21
    LET V=0:
    GOSUB 31:
    RETURN
 21:GOSUB 2:U=0:
    RETURN
 25:FOR Y=1TO 13
    :A$(Y)=" ":
    NEXT Y:O=0:T
    =0:U=0
 26:BEEP 1:PAUSE
    "BANK $";X:
    IF XINPUT "B
    ET=";Y:IF Y>
    XGOTO 26
 27:IF X<=0INPUT
    "AGAIN?";N$:
    IF N$="Y"LET
    X=E2:GOTO 26
 29:IF X<=0END
 30:PAUSE "DEALI
    NG...":P=7:
    GOSUB 4:U=U+
    V:P=8:GOSUB
    4:Z=V:U=U+V:
    P=1:GOSUB 4:
    T=T+V:P=2:
    GOSUB 4:T=T+
    V:P=3
 31:N$=G$:G$="^"
    :GOSUB 2:G$=
    N$:IF V=0
    RETURN
 35:IF P=3IF O=1
    0IF T=11
    PAUSE "BLACK
    JACK!":Y=1.5
    Y:GOTO 63
 40:IF P=3IF (Z=
    1)+(Z=10
    GOSUB 19:IF
    U=0GOTO 25
 42:IF P=3IF O=0
    IF (T=10)+(T
    =11INPUT "DO
    UBLE?";N$:IF
    N$="Y"GOSUB
    4:Y=2Y:T=T+V
    :GOTO 50
 45:IF P<7IF T<=
    21INPUT "HIT
    ?";N$:IF N$=
    "Y"GOSUB 4:T
    =T+V:P=P+1:
    GOTO 31
 48:IF T>21PAUSE
    "YOU BUST WI
    TH ";T:GOTO
    65
 50:GOSUB 2:O=0:
    P=8:IF G$="A
    "LET O=10
 51:IF H$="A"LET
    O=10
 53:N=U+O:IF (U
    <>6)+(O<>10
    IF N<=21IF N
    >=17+(P>8LET
    U=N:GOTO 60
 54:IF U>21PAUSE
    "DEALER BUST
    S WITH ";U:
    GOTO 63
 55:IF U>=17+(P>
    8GOTO 60
 57:P=P+1:GOSUB
    4:U=U+V:IF V
    =1LET O=10
 59:GOSUB 2:GOTO
    54
 60:PAUSE "DEALE
    R STAYS WITH
     ";U:FOR N=1
    TO 6:IF A$(N
    )="A"LET O=1
    0:N=6
 61:NEXT N:IF U
    >=T+O*((T+O)
    <=21GOTO 65
 63:PAUSE "YOU W
    IN $";Y:X=X+
    Y:IF X<E4
    GOTO 25
 64:PRINT "YOU W
    ON $";X:END
 65:PAUSE "YOU L
    OSE $";Y:X=X
    -Y:GOTO 25
```

YES, THERE IS A DIFFERENCE

Dan Pinko, of Parksville, B.C., Canada, reports that there is a difference between the Sharp PC-1211 and the Radio Shack TRS-80 PC. Namely, the PC-1211 (or at least the one he has!) does not execute the special CLOAD 1 directive. This command, explained on page 96 of the manual for the TRS-80 PC, allows a user to load multiple programs into the PC. It is used to bring in a program from a cassette tape without wiping out the current contents of memory. (All of memory is effectively erased if an ordinary CLOAD directive is used to load a program from tape.)

CORRECTION TO SUPER-WUMPUS

Joseph P. Jones from Danville, IL, makes the following recommendations concerning the program listed on page 3 of Issue 08 of *PCN*: First, either line 5 should use the statement H=INT (G*.12)+1 or line 200 should contain T=23 + INT (H/6). This is needed to prevent M from exceeding 26 in line 200. Second, line 105 should use IF K=1 to prevent bats from taking you twice. Finally, in line 95 there should be a colon (:) before the statement GOTO 40.

Thanks for the improvements, Joseph.

FROM THE WATCH POCKET

Here it is, the end of our first year of publication. Time to recap and look ahead.

1981 was the first full year of the existence of pocket computers after their introduction in the latter part of 1980. In retrospect, relatively little (in comparison with what is likely to happen in 1982!) happened to advance the state of the art. True, users were provided with the ability to get a printed listing of results and programs. And, just a few months ago a second company (Casio) began offering competition to the original Sharp/Radio Shack PC. However, though it does offer some increased operating speed, the Casio certainly does not represent a significant advancement in terms of PC capability. The newly announced Panasonic HHC, which the firm does not want misconstrued as a pocket computer (and about which I will have more to say shortly), really is not yet available in the U.S. A serious shortcoming of PCs, in my opinion, has yet to be overcome. This is plain and simply the relatively long length of time and general awkwardness involved in loading a new program into the machine. It is certainly a problem that could readily be overcome with today's technology if somebody in the marketing department ever tuned into the fact that it is a serious handicap. No, 1981, was not a year of great advancement. It might perhaps best be characterized as the initial year of assimilation by the advanced echelons of our technological society. I have seen few layman caught by any desire to own or utilize a PC.

The Year Ahead

1982 will undoubtably be a year of increasing momentum in the field. While there was only one new firm actually entering the field and delivering product in 1981, there are likely to be 10 to 20 firms coming on line in 1982. Some of these will be biggies. I expect Texas Instruments to pop something in this area, though I doubt that it will be anything earth shaking. Hewlett-Packard is also expected to join the fray and I have high hopes that they can at least advance the field in terms of general capability and the program loading area. Some large Japanese firms will have additional entries.

Of course, Sharp Electronics, the originator of the PC as we know it, will continue to be a contender. Having a great, almost two year jump on the competition, and that extended length of time in which to conduct market research and determine user's desires, I expect that they will come up with another winner in 1982. It is likely that they will arrive with increased memory capacity, improved mass storage (hopefully fast loading and convenient to use). I believe they will quite likely incorporate the feature most sought after by many PC users the ability to connect (perform input/output operations) with other devices.

The general consensus appears to be that something over a quarter of a million PCs were sold in 1981. 1982 could see this figure easily approach 2 million units amongst the various vendors. At an average retail price of $200.00 we are talking about a market approaching four hundred million dollars. Nothing to sneeze about.

While the average price of new units, because they offer improved capability, is likely to stay around $200.00, there will be steady price reductions just as there were in 1981. For instance, the introductory $249.95 price of the Sharp PC-1211 is now down to $179.95. Radio Shack offers a holiday sale price on their original $249.95 unit of just $169.95. A unit having the capability of a PC-1211 will likely be under $100.00 by this time next year, possibly reaching as low as the $80 mark.

The Panasonic HHC

After approximately a year of dropping repeated hints as to its capabilities, Panasonic and Friends Amis (the designers) unveiled their Hand Held Computer. There are still a lot of unanswered questions about the unit, including when will it actually be available in the U.S. At this point it appears that for $500.00 you get a 2K portable computer that can also act as a terminal. For another $100.00 you can get an added 2K of memory. That is quite a bit less than the 16K worth that Clive Sinclair delivers as an add on to his ZX81 for the same price! In all fairness, it is still too early to talk in real specifics about this device because we cannot get our hands on an actual unit. It appears that it may be several months before they are generally available in this country, though the firm claims they have already begun supplying the units in Europe.

This and That

There is an outfit, **Conlen's Custom Cases**, *Box 306, Norristown, PA 19404*, that provides custom-built briefcases for Radio Shack and Sharp PCs. The price is $72.50 plus shipping/handling. You can phone (215) 279-5334 for more information.

If you are into using your PC for gambling activities, write to **Joe Computer**, *22713 Ventura Blvd, Suite F, Woodland Hills, CA 91364*, for a copy of their catalog.

Bob McElwain is putting out a newsletter for PC users entitled **Compunotes**. A sample issue is $4.00. For information write to **XCEL**, *13763 Polk Street, Sylmar, CA 91342*.

There hasn't been much in the computer trade magazines lately for PC users. Look for this to change in 1982 as more powerful PCs force magazine editors to pay attention to this area.

If you wish additional information on the Sinclair ZX81 announced in Issue 09 of *PCN*, please write directly to the U.S. supplier, **Sinclair Research Ltd.**, *One Sinclair Plaza, Nashua, NH 03061.*

An outfit by the name of **H L Enterprises**, *22 Ellsworth Avenue, Morristown, NJ 07960*, puts out a clever little gadget called the *Line-Minder* (TM) that may be of interest to *PCN* readers. It consists of a clamp strip that will hold onto the side of a single sheet of paper. A line guide travels along this clamp strip making it easy to keep your place when, for instance, you are loading a program. It works on either side of a sheet of paper and is thus ideal for *PCN* readers who frequently lose their place while trying to load in a program from a listing. The price of the clever little device is $2.95. The phone number of H L Enterprises is (201) 539-8058 if you need more information.

Technical manuals on the Radio Shack TRS-80 Pocket Computer and the companion printer/cassette interface are reportedly available now through special order. Tell your local dealer what you are looking for and ask him/her to order it for you. Expect to wait a few weeks for delivery. Several readers have indicated it is well worth the price and the wait if you are interested in the electronic details (schematics, etc.) of your unit.

My personal thanks to all of you who participated in the recent survey of pocket computer users. Your answers will influence the future design of pocket computers.

If you are a serious user of personal computers and feel it is important to have access to a large historical reference on the field, be sure to obtain a copy of the December, 1981, issue of **BYTE Magazine**. This edition features a comprehensive, multiple-key index to all articles published in the magazine since its inception.

If you are thinking of submitting material to *PCN*, please be advised that readers are expressing particular interest in job-related applications of their pocket computers. Please, do not submit versions of popular games. We have quite a sufficient backlog of entertainment programs to last for some time. The chances of a game submission being accepted at this time are virtually nil. Good, practical applications programs will receive careful consideration. Please remember to include a cassette copy of the program being submitted if it is more than a few lines in length. Allow 4 to 8 weeks for an article or program to be reviewed. Include a stamped, self-addressed mailer if you want submitted materials that are not accepted for publication returned.

A New Mailer

Beginning with the next issue, *PCN* will go to a lighter weight paper and use a new mailer. The primary purpose of this change is to reduce the cost of postage so that we may continue to serve you by first class mail despite the fact that the USPS has raised the rate of first class mail by 33 percent during the past year.

PCN plans to continue bringing you the latest information about pocket computers, to provide timely and informative reviews and practical, useful programs. We plan to continue doing this in a speedy, compact manner so that you may quickly glean information that is of value to you and get on to the business of your day.

Thank you for your support. Best wishes in the coming New Year!

1981 INDEX TO ARTICLES AND PROGRAMS

BUSINESS PROGRAMS

ENTERTAINMENT PROGRAMS

GENERAL APPLICATIONS

OPERATING INFORMATION

35 Old State Rd, Oxford, CT 06483

PRODUCT ANNOUNCEMENTS

PRODUCT REVIEWS

SCIENTIFIC & ENGINEERING PROGRAMS