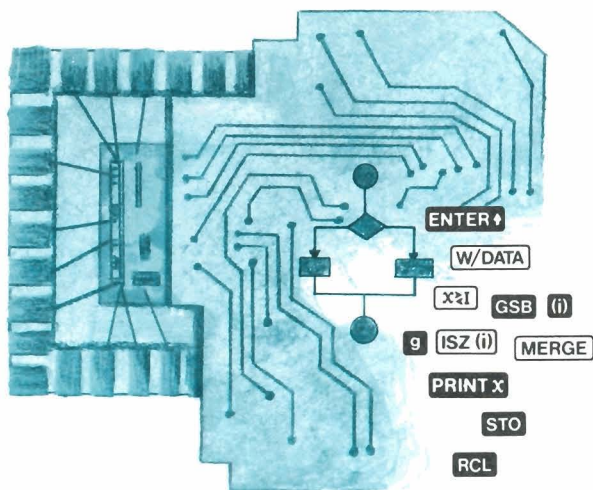


Featuring, this issue:

Library Corner	2
HP-41C Tips and Techniques	3
(67) Twenty-Element 4×5 Matrix	4
Book Reviews	4
HP-41C Tips From an Owner	5
Randomly Yours	6
We Get Letters	7
"25 Words" (More or Less!)	7
HP-41C Owner's Handbook Addendum	12



November 1979 Vol. 3 No. 4



HP Key Notes

What Hath HP Wrought?

Can you believe that it was only 7 years ago that the HP-35 created a worldwide revolution in handheld personal calculators? Almost overnight, it made the venerable slide rule a computational antique. Plus, it started a very large new industry and caused most of us to restructure our lives—at least as far as numeric calculations were concerned. And, although all HP-35 calculators are now 5 to 7 years old, *most* of them are still going strong even today. It was—and still is—a marvelous product.

However, most of you probably consider the HP-65 a more significant contribution to the

way that calculators have changed your life. It caused a quantum jump in personal calculator technology and forever banished the tedium of repetitious numeric calculations. It was responsible not only for putting the "programmable" in personal calculators but also for making this newsletter possible.

Today, programmable calculators are a relatively common tool in every walk of life. They save time, save money, make life easier, and increase the scope of our knowledge. We almost take them for granted, just 5 short years after they were introduced.

Now, you might ask, "What's next?" Well, the HP-41C system certainly shows you that

Hewlett-Packard takes this business very seriously and always tries to make a significant contribution to the state of the art. So you can bet that we will continue to develop even better and more significant products.

With the Christmas season upon us, we thought you might enjoy seeing all the calculators that Hewlett-Packard has produced since the HP-35 started this revolution. In the photo are 27 calculators, with the HP-35 at bottom center and the HP-41C at top center. And if Santa Claus were to bring you *every one* of these, he would have to have a check for \$7,725 (total of original list prices)!



Editorial

Harden Schaeffer of Goldthwaite, Texas, saw a weird mistake in the May issue (Vol. 3 No. 2, page 10, col. 1) and, for that matter, so did **Dr. Hans Stöcklmair**. It seems I somehow moved Dr. Stöcklmair's hometown of Klagenfurt from Austria to Australia!! Sorry about that. I would fire the proofreader, except that I am the proofreader.

And while I'm confessing, here is another admission of guilt. In the August issue (Vol. 3 No. 3, page 4, col. 2), **Richard Nelson** said that functions cannot be assigned to the "correction key" ☐. That's not true. I missed it when I checked his copy. Thank you, **Kjeld Hvatum** (Cambridge, Massachusetts) for bringing that to my attention. On the HP-41C, you can reassign any key but the gold key.

Some of you have asked, "Why don't you publish an index for KEY NOTES?" Well, first there's the added costs; then there's the unavailability of back issues, which everyone would want; and finally, it would take up a lot of space that the majority of you prefer to see in routines, tips, and techniques.

You often ask why we do not print programs in KEY NOTES. Okay, I'll answer that for you. Primarily, the main reason is that we do not want to compete with the Library, nor do we want to compete with our dealers. Also, to make absolutely sure that all readers, worldwide, could use any program printed in KEY NOTES, it would require extensive and immaculate instructions and examples, which translates into a great deal of space in the newsletter. Furthermore, with only four issues a year and, say, one program an issue, imagine the problem of choosing only one from the many thousands that would be sent in. In simple terms, it just isn't practical, so we use the space for tips, techniques, routines, and applications, which is what most of you ask for.

Beginning with the February 1980 issue, all copies of KEY NOTES will be printed in the U.S.A. and then distributed from here throughout the world in bulk quantities for mailing within several large geographical areas. This will enable most of our readers, anywhere in the world, to see the same issue at approximately the same time, instead of many months after the U.S. version is printed. This will also enable us to print items of interest to various parts of the world and not just U.S. news and views.

Under this new plan, there may be a few problems to iron out, so please be patient during the transition period. KEY NOTES has grown enormously in the past 3 years, so it takes a lot of careful, trouble-free planning to get it to all of you on time. To give you some idea of the magnitude of KEY NOTES, consider this: someone—somewhere in the world—at this very moment—is reading the one-millionth printed copy of KEY NOTES!! And just think, unlike the "rock" stars, I didn't even get a "gold" record! Oh, well, I enjoy doing this, and your enjoyment of KEY

NOTES is enough reward.

Letters to the editor should be addressed to:

Henry Horn, Editor
HP KEY NOTES
Hewlett-Packard Co.
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330 U.S.A.

We cannot guarantee a reply to every letter, but we will guarantee that every letter received will be read by the editor, and as many as possible will be answered either in KEY NOTES or in a personal response. Please be sure to put your return address on the face of your letter. Letters sometimes get separated from envelopes.

And, at this time of year, once again I want to thank all of you for your support of KEY NOTES, for you letters, and for your nice comments about the newsletter. I wish all of you a happy and safe holiday season and many hours of happy programming. And may all your random numbers be random!

Library Corner

CATALOG UPDATE

Can you believe it's already time for another update to the *Users' Library Catalog*? The Library is very excited for the update to be mailed (it's on the presses right now) so they can hear your reactions to their new format. The *Catalog* update will go into the mails the first few weeks of December (bulk rate, as usual). With the Christmas season upon us, please allow through the month of January to receive your copy before you contact the Library. Besides including HP-41C, HP-67, and HP-97 programs in the *Catalog*, we are going to send all Library subscribers a completely bound *Catalog—no updating required!*

In order to bring you better programs, we have implemented a new plan for eliminating non-selling programs from the *Catalog*. This is an effort to encourage good program submissions: well-documented programs that potentially have some use for another user. Once a program is accepted in the Library it will remain available from older editions of the *Catalog*. However, if a program has appeared in print for approximately 1 year and has not sold or has sold poorly, it will be eliminated from future editions of the *Catalog*. If a program picks up and begins selling well, it can always be reinstated into the *Catalog*.

HP-41C PROGRAMS

As of the end of November, there were 300 HP-41C programs in the Library. We encourage you to document and submit any HP-41C programs you have written. Remember, as we mentioned in our last issue of KEY NOTES, Library prices have gone up slightly. All programs include documentation and a recorded magnetic card(s) and are \$6* each. *Documentation alone is no longer available.*

* U.S. dollars. See note at bottom edge of cover.

ORDERING PROGRAMS

Library programs are \$6* each. Whenever possible, use the Program Order Forms in your *Catalog of Contributed Programs* to place Library orders. If you see a program you want in this issue of KEY NOTES, and you can't find your order form, a plain piece of paper with your name and address and the program numbers you desire is certainly adequate. Mail your order and a check or money order to the address listed on the back cover of this issue. Don't forget to include your state and local taxes. Or you can place your order by calling toll-free: 800-648-4711 (in Nevada 800-992-5710).

SOLUTIONS BOOKS

The *Users' Library Solutions Books* are a very good value. Subjects of the books range from Real Estate to Solar Engineering. The 25 HP-41C books are \$12.50* each and the 40 HP-67/97 books are \$10.00* each. But, don't forget that if you don't require an entire book of programs on one subject you can always purchase the individual programs directly from the *Users' Library*. Appendix A of your *Catalog* lists the corresponding program numbers for each program in every Solutions Book.

NEW PROGRAMS

Here are some recent submittals you might like. They are too new to appear in the November *Catalog* update. None of the programs in this issue are available in Europe at this time. (See "Editorial" column.)

(41C) 2AXCON Concrete Column/ Biaxial Bending (#00230C)

2AXCON computes the ultimate capacity for a given concrete section, subject to a given axial compression load and moments about two perpendicular axes. The concrete section is either square or rectangular, with up to \$4 reinforcing bars. The method of analysis is based on ultimate strength design, following A.C.I. 1977 reinforced concrete building code requirements. Requires three Memory Modules. (621 lines)

Author: **Charles I. Dinsmore, P.E.**
Seattle, Washington

(97) Cash Flow Analysis—Return on Investment & IRR—8 Years (#04085D)

A two-card program. Card #1 program prints a maximum of 8 years of after-tax cash flow, with variable or constant rents and expenses. The program will calculate debt service, amortization interest, and changing rents or expenses. It will operate with 0, 1, or 2 mortgages. Card #2 program prints final year data, resale price (appreciation (+)% or depreciation (-)%), sales expense, pretax proceeds, and LTCG tax. After-tax cash from sale is added to final year ATC from ordinary in-

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

come. The program also will calculate and print the internal rate of return. (368 lines)

Author: **Donald Bazinet**
Palm Beach, Florida

(67) The 6174 Problem (Kaprekar's Constant) (#04001D)

This program demonstrates an interesting mathematical curiosity. Any four-digit number, when re-arranged in a certain order and then processed in a certain manner, will always yield "6174" (Kaprekar's Constant) after a few iterations. This program is much better than previously submitted versions because it displays all intermediate steps used in the process and allows you the option of printing or pausing the display. (109 lines)

Author: **Richard S. Altman**
Richmond, California

(97) Industrial Steam Trap Selection (#03942D)

Given pertinent data for input, this program outputs the amount of condensate formed and the required steam trap capacity for any of six different types of heating equipment, enabling the designer to select a steam trap of adequate capacity. (165 lines)

Author: **Allan T. Seidcheck**
Honolulu, Hawaii

(67) Engineering Investment Payback Period (#03908D)

This is a program to solve the problem of how many years are required for an investment that saves energy, water, sewer charges, etc. to pay for itself with the net savings. It includes effects of interest, inflation, running costs, depreciation allowance, and taxes. (112 lines)

Author: **James E Rich, P.E.**
Sherwood, Oregon

(67) Annuities and Amounts With Interest Compounded Instantly (#03915D)

This program solves compound interest and annuity problems where interest is compounded instantaneously or continuously. Flag 3 is used to enable interchangeable solutions between all variables. The program has the same problem-solving capability as programSD-05A from the Standard Pac. (196 lines)

Author: **George E. Bahrman**
Chassell, Michigan

Now, here is a treat for those of you who like to play card games. This "special" program is 47 pages long, uses 7 magnetic cards, and contains 1,147 lines. To say the least, it is a fascinating work of art. The calculator actually plays *against* the user and must make all of its own decisions; that is why seven magnetic cards are necessary. It is a masterpiece of documentation; even the flow diagram is cross-referenced. The author, **John C. Nelson**, de-

serves congratulations for this one!

This "special" program is titled, **Full Poker—5-Card Draw #67000-99973**, and the price is \$22.50.* Here is the abstract:

(67/97) This program is an almost completely full-scale poker playing program. The program deals the user and itself five cards each, from a 52-card deck without duplicates. It then evaluates each hand; decides whether to stay, fold, or raise; allows a draw by the user as well as by itself; and goes into the showdown making its own decisions all the way. It also knows who won each hand, and it keeps the accounts.

Author: **John C. Nelson**
Des Moines, Iowa

(Here is a good challenge for someone with an HP-41C. This would become quite a fascinating program with symbols, alpha prompts and so forth. Ed.)

* U.S. dollars. See note at bottom edge of cover.

HP-41C Tips and Techniques

Here is a collection of tips and techniques for HP-41C users. You will find them useful and, because they are described in detail, you may learn a little more about your HP-41C.

One of the most obvious "new" features of the HP-41C is the display. In addition to being larger than on previous models, it has an alpha capability and a new instruction—VIEW. When you think of the HP-41C display, you can think of it as a window to VIEW various parts of the calculator. The display is *not* a register and you can't store or recall from it. One of the most interesting uses of the VIEW instruction is to VIEW the alpha register.

It is possible for the display to be giving you information *while* the calculator is running. A routine for demonstrating that is given below. The routine will count to N and produce a tone for each count while it is counting. Thus, the display tells you what the calculator is doing while it is doing it. You know the calculator is "busy" because it is "talking" to you with a tone while it counts up to the number, N, which you key into the machine. This routine also uses a number of the HP-41C features, so we will explore it in detail. Key it in and try it.

```
01+LBL A      08+LBL 01
02 " COUNTING" 09 TONE 8
03 AVIEW      10 ISG X
04 1           11 GTO 01
05 -          12 " DONE"
06 1000       13 AVIEW
07 /          14 STOP
              15 .END.
```

Here are a few tips to help you if you are still unfamiliar with your machine. When you enter line 02, notice the two blank spaces preceding the word COUNTING. The display is 12 characters across, but COUNTING is an eight-character word, so it looks better if it is centered. Keying in two spaces moves it to the center. Tip: *If memory space is adequate, center display messages with preceding blanks for a more attractive display.* After you finish the "G" in COUNTING, you can immediately press **VIEW** because the HP-41C "knows" that you are finished with your alpha entry when you press the **VIEW** sequence for line 03, "AVIEW." Then press **ALPHA** and continue. Line 08 is LBL 01. Did you press **LBL** 01 or have you become familiar enough with your HP-41C to press **LBL** **EX**? Tip: *Use the top two rows of keys for the numbers 01 through 10 for XEQ, GTO, and LBL instructions.* Line 09 is TONE 8. There are no quote marks around the TONE or the 8 so this must be an instruction to execute. Press **XEQ** **ALPHA** TONE **ALPHA**, then 8. Step 11 is the ISG x instruction. Don't forget that X is the stack, and the sequence is: **ISG**, **X**, **X**. Line 11 lets us use the "01" key again. Notice the four leading spaces in line 12 to center "DONE" in the display.

Check the routine by pressing **RTN**, **PRGM** and then **SST** through the 14 lines. To test the routine press **USER**, 10, and **A**. The display should indicate a centered "COUNTING" and produce 10 tones and then change to "DONE." Technique: *AVIEW a display message telling the user what is happening during calculations.* This is useful for many parts of the program and makes the HP-41C "friendly" and more fun to use.

The AVIEW instruction can be used to show "nothing" as well as a message. If you don't want to look at the program execution symbol (the "flying goose") while the calculator is busy, you can place a single space, and AVIEW, at the beginning of the program. To illustrate that you can do what you want with the display, try routine E and have fun with the "goose."

```
01+LBL E      09+LBL 05
02+LBL 05     10+LBL 05
03+LBL 05     11+LBL 05
04+LBL 05     12+LBL 05
05+LBL 05     13 TONE 3
06+LBL 05     14 GTO 05
07+LBL 05     15 .END.
08+LBL 05
```

The 11 label 05's advance the "goose" forward and around "one position short" of a full trip, then the "goose" honks and hesitates and repeats, and only appears to be flying backwards.

Did you notice how LBL A was used? Tip: *Use LBL A thru LBL J for simple, short routines.* This saves memory and you get an "automatic" key assignment using these single letter labels. Labels a thru e also work the same way.

(Continued)

However, if you pressed **GTO** **□** **□** to key the second routine, LBL E, you will lose the LBL A key assignment, because pressing **GTO** **□** **□** places an END in the program and moves the program pointer below the END in program memory. And, before we go any further, let's make sure you understand what is occurring at this point.

Those 15 local ALPHA labels mentioned above are called "local" labels because they can be accessed only if the calculator is already positioned to the program that contains them. They DO NOT appear in CATALOG 1, which lists only the user-programmed "global" labels.

This can cause a problem. If you have a program containing only local labels (for example, an HP-67/97 card-reader-translated program), you may "lose" the program in the HP-41C memory by adding subsequent programs. You will not be able to use GTO or XEQ to get to the program because these functions require a global label.

Well, this really isn't as complex as it seems. Technique: *To cure this problem, first find the program (or routine) by using CATALOG 1. Stop the catalog when you see two END statements in a row. (Stop at the second END.) This indicates a program space with no ALPHA labels. Switch to PROGRAM mode and key in an ALPHA label, which will "title" the program. From then on, you will be able to use CATALOG, GTO, or XEQ to access the program, and it will run just as before!*

Routine A uses the new and powerful ISG instruction in line 10. The X-register is specified as the counter register, and lines 04 thru 07 prepare your number, N, so it will be in the proper format. If "1" (lines 04 and 05) is not subtracted from the number you key in, the ISG loop (lines 08 thru 11) will produce N + 1 tones. You can see this taking place by adding a VIEW after line 08. Do this by pressing **■** **GTO** **□** **□** **VIEW** **□** **□** **PRGM**. Now press **■** **VIEW** **□** **□** **PRGM**. They key the number 5 and press **■** **A**. Observe the display carefully and count each tone. The ISG counter counts from 0 to 4. When the number 5 is in the X-register and has 1 subtracted from it and is divided by 1,000, it looks like 0.0040. The number to the left of the decimal increments by "1" until it is GREATER THAN the number 4. When this happens, 5.0040 is in the counter (X-register in this routine) and the ISG instruction skips the GTO 01 and AVIEWS "DONE" and stops.

The counter used for ISG (and in a similar manner for DSE) can have the initial value of: .004 or 1.005

Both of these initial values will count to five. You may verify this for yourself by testing the ISG loop. Key .004 **XEQ** **□** **□** (01). Repeat with 1.005 **XEQ** **□** **□**. Which number required the fewest keystrokes? It was .004, of course. If the initial number is to be stored by the program, the .004 would be preferred. Tip: *Often it is possible to save a byte of memory by using an alternate initial form of an ISG or DSE counter.*

The VIEW instruction will command the display unless another instruction overrides the previous VIEW instruction. Try this: .004 **XEQ** **□** **□**. The display showed 4.0040 just before DONE appeared. Turn the calculator OFF, then ON. Now what do you see? 5.0050? Why? The display is now showing the X-register. The previous VIEW was showing the X-register prior to the last increment of the ISG instruction. As you can see, it is important to understand what is in the display. Repeat the .004 **XEQ** **□** **□** sequence. Now press **□** **□** twice. In each case you have changed the status of the calculator since the last VIEW, and the display automatically shows the X-register.

The VIEW and AVIEW instructions also have another useful feature. They cause the display to be printed if the printer is plugged in and turned on. This is important to know if your programs will be run with a printer some day. Tip: *Instead of using PROMPT to stop for an input after an alpha message, use AVIEW, STOP. This method used an extra line, but makes the program run really well if the printer is plugged in. If you want to print the input data, use the following techniques:*

AVIEW	}	Stops for input. Prints alpha display.
STOP		
VIEW		Prints input (right justified).

If you use the ADV instruction to space-out or group your inputs and outputs, you can have your programs "printer compatible" without a printer. The VIEW and ADV instructions were designed with this purpose in mind. If you want to be "extra neat" you could use the sequence:

AVIEW	}	Stops for input. Prints alpha message.
STOP		
CLA	}	Prints input (left justified).
ARCLX		
AVIEW		

The HP-41C has a number of new instructions and VIEW and AVIEW are only two of the most obvious. Routine "A" demonstrated AVIEW and the new ISG instruction. The X-register was used as the counter register for the ISG, and this capability is also new on the HP-41C. Probably one of the most powerful new instructions is the full indirect addressing capability of the HP-41C. This may be demonstrated by changing one instruction in Routine A. Key **■** **GTO** **□** **□** **PRGM**, then **■** **SST** to the TONE 8 instruction. Delete with **■** and key **■** **XEQ** **□** **□** **ALPHA** **■** **□** **□** **PRGM**. Now key 5 and press **■** **A**. This time the tone corresponding to the "count" number is heard. Try 1.005 **XEQ** **□** **□**. The count started at 1 and went to 5. Try 4.008 **XEQ** **□** **□**.

This "Tips and Techniques" article was intended to illustrate a few tips and techniques in controlling the display. ISG loops and indirect addressing were briefly mentioned. There will be more to discuss on these topics in future issues of KEY NOTES.

(67) Twenty-Element 4 × 5 Matrix

In line with our efforts to provide something for everyone, here is a routine for those of you who might wonder how to store a two-dimensional matrix in a one-dimensional register space. It is the contribution of **William R. Lindahl** of Leesburg, Florida.

This matrix (array) can be used with subscripts, taking all 38 lines or, if the matrix is loaded manually, only the last 13 lines are required. Many higher level programming languages use subscripts in programs and flow charts, and this routine will provide easier conversion for small arrays. The user's program must control the subscripts and their limits for data manipulation.

1. Labels used: d, e, and C.
2. Registers used: D, E, I, and 0 thru 19.
3. Register D is J and E is K.
4. Use **■** **GSB** **□** to fill array. The display will stop and show J,K, then press **■** **ENTER** **■** **a_{J,K}** and **■** **R/S**.
5. Use **■** **GSB** **□** to retrieve a_{J,K} in the X-register of user's program.
6. For manual input, use: STO 0 thru 9 **■** **P/S** STO 0 thru 9 **■** **P/S**. Caution: be careful of array notation:
Reg. 0 thru 4 = a_{1,1} thru a_{1,5}
Reg. 5 thru 9 = a_{2,1} thru a_{2,5}
Reg. 10 thru 14 = a_{3,1} thru a_{3,5}
Reg. 15 thru 19 = a_{4,1} thru a_{4,5}

The steps of the entire routine are:

001	*LBL d	020	+
002	CLX	021	+
003	STOI	022	R/S
004	*LBL C	023	STOI
005	4	024	ISZ I
006	RCL I	025	GTCC
007	5	026	*LBL e
008	=	027	RCL D
009	X=Y?	028	1
010	RTN	029	-
011	ENT↑	030	5
012	INT	031	x
013	X≠Y	032	1
014	FRC	033	-
015	2	034	RCL E
016	÷	035	+
017	1	036	STOI
018	.	037	RCL I
019	1	038	RTN

Book Reviews

From time to time we have valiantly tried to keep you informed of additional literature about personal programmable calculators. But it is a difficult task. Books started into production are often held up for a multitude of reasons, such as strikes, paper shortages, and so on. And these headaches give us headaches, because they always happen after the date on which we have told you a book will be avail-

able. But we do try our best ... and we do it for your benefit. So please bear with us.

Here is one that was slated for the last KEY NOTES but was left out because of a lack of space. It is a "paper" titled *Calculation of Flue Losses for High-Efficiency Furnaces and Appliances*, written by **Richard S. Brokaw**, a consultant for A.G.A. Laboratories in Cleveland, Ohio. It originally appeared in the January, 1979, *ASHRA Journal* (American Society of Heating, Refrigerating, and Air-Conditioning Engineers).

The paper presents equations for calculating flue losses under conditions where water in the flue gases is fully vaporized or partially condensed. Required input information includes room and flue temperatures, barometric pressure, air humidity, and flue carbon dioxide concentration, as well as combustion constants characterizing the fuel. The equations have been programmed for the HP-97 calculator and the program is listed in the paper.

Mr. Brokaw has a limited supply of reprints of this article *for those who could use this information*. To obtain a reprint from a U.S. address, send a self-addressed business-size envelope containing a 15¢ stamp. If you also desire a direct copy of the program, include two blank magnetic cards. From outside the U.S., send an envelope at least 22 cm long and include U.S. postage to cover 20 grams. Send requests to:

Mr. Richard S. Brokaw, Consultant
American Gas Association Laboratories
8501 Pleasant Valley Road
Cleveland, Ohio, U.S.A. 44131

Now, here is a book co-authored by **Kenneth J. Rothman** of the Harvard University School of Public Health, Department of Epidemiology and **John D. Boice, Jr.** of the Environmental Epidemiology Branch of the National Cancer Institute in Bethesda, Maryland. The book is titled "Epidemiologic Analysis With a Programmable Calculator." The book is a collection of 22 pocket calculator (HP-67) programs developed to handle the range of analyses that most epidemiologists face routinely. Most of the examples are drawn from the area of chronic disease epidemiology and are meant to exemplify the application of calculator programs and not epidemiologic findings. To obtain a copy of this 142-page hardbound book, order NIH Publication number 79-1649 (stock #017-042-00143-9) from:

Superintendent of Documents
Government Printing Office
Washington, D.C., U.S.A. 20402

In the U.S. it is \$8* postpaid. Overseas the price is \$10* postpaid.

For a change of pace—and subject—here is a letter from **Donald A. McIntyre**, who happens to be the author of the following literature.

"I have written a number of programs for the HP-67/97, to calculate environmental indices of comfort and heat stress. These indices are used by people concerned with health and safe-

ty in the work place—both industrial and military. The comfort indices and the physical programs are also of interest to the air conditioning engineer. The programs have been issued by the Electricity Council Research Centre as document ECRC/MM31. Copies are free of charge on application to the Electricity Council Research Centre; Copenhurst; Chester; CH1 6ES United Kingdom. We cannot, of course, commit ourselves to an unlimited number of free copies. If the demand exceeds our expectations, we would have to reconsider the position."

Now, as a final entry, this review is about a small booklet produced by GTE Lenkurt. It is called the *GTE Lenkurt DEMODULATOR*. The March/April issue of this year featured—and was titled—"Path Profiling With a Programmable Calculator." The pamphlet presents a brief amount of theory and usage material on two programs that can take away some of the boring calculations involved in Microwave Path Profiling. Both programs were written for the HP-97, and each contains 224 lines. They are the work of **John E. Hendricks, Jr.**, a Staff Engineer at Lenkurt.

A limited number of pamphlets are available to people with a professional interest, and they are free on request by writing to:

John B. Birge, Editor
GTE Lenkurt DEMODULATOR
1105 County Road
San Carlos, California, U.S.A. 94070

KEY NOTES thanks Mr. Hendricks and Mr. Birge for their generous offer. Please do not request the pamphlet if you do not have a genuine use for it.

The Case for HP-41C

Although a carrying case is provided for the HP-41C, some owners have asked if they can use another case when they don't have a card reader on their calculator. Yes, you can use the HP 82017A soft leather case that was made for the HP-67 and HP-65. It provides the best fit and has a pouch where extra modules can be kept. The list price is \$10.*

As for hard leather cases, both of the available models can be used. The "Classic" hard case (HP 82006A) provides a tight but acceptable fit. List price is \$30.* If you do not mind a slightly bulkier case, you can use the one (HP 82016A) for the HP-67/65 that has an inside pocket. It will hold the HP-41C somewhat loosely, plus several modules and the quick reference guide. It lists for \$35.*

If you own the HP-41C and all of its peripherals, and you do not have an attache case or briefcase, perhaps you might like to use the HP-97 carrying case (HP 82035A). List price is \$10.* This case will accept the HP-41C, the card reader, the printer, several books, and even a Memory Module Holder.

* U.S. dollars. See note at bottom edge of cover.

* U.S. dollars. See note at bottom edge of cover.

HP-41C Tips From an Owner

The world is full of collectors: stamp collectors, coin collectors, art collectors, antique collectors, car collectors ... virtually anything you can name. So it doesn't surprise us to know that **Craig Pearce** of Berwyn, Illinois, "collects" our programmable calculators. And, knowing that, it follows that he is already an HP-41C enthusiast. So here are some of his tips for other HP-41C owners.

If you want to know how many registers have been "SIZED" and you don't own a printer or don't have one with you, this routine is for you. It makes use of the error ignore flag (F25) and will cause the loss of the Z- and T-register contents. However, it has been written to maintain (leave unaltered) all of the data registers contents in the machine. This routine requires one local label (in this case LBL 00) and resets the display to 'FIX 0' upon termination.

LBL "SIZE" Assigned to: ☒ key
44 Bytes Exits USER mode upon execu-
20 Lines tion? Yes.
07 Registers System Requirements: Std. 41C

01*LBL "SIZE"	11 FS? 25
02 FIX 0	12 GTO 00
03 SF 25	13 RDN
04 CF 27	14 1
05 .5	15 -
06 ENTER↑	16 INT
07*LBL 00	17 "SIZE="
08 CLX	18 ARCL X
09 RCL IND Y	19 RVIEW
10 ISG Y	20 .END.

INSTRUCTIONS:

1. Load routine and assign it to a key, if desired. (These instructions assume the routine is assigned to the ☒ key.)
1. To find the SIZE that has been allocated, press ☒ then ☒. The program will exit the USER mode immediately. (Running time increases with more data registers allocated.)
3. When completed, the display will show: "SIZE = xx." (where 'xx' is the SIZE allocated).

This routine was written to duplicate the 'recall-summation' function of the HP-67/97 series. THIS ROUTINE ASSUMES THAT THE HP-41C SUMMATION REGISTERS HAVE NOT BEEN REASSIGNED. If so, change the register RCL in line 05 to reflect the new "sums of Y's" register and the RCL of line 06 to reflect the new "sums of X's" register.

Stack contents are treated as they are on the HP-67/97 units when a 'recall-summation' is executed, in that the current value of the X-register is stored in LAST X, the value in 'Y' is overwritten, and the 'Z' and 'T' values remain unchanged.

LBL "RCLΣ" Assigned to: ☒ key.
17 Bytes Exits USER mode upon execu-
07 Lines tion? Yes.
03 Registers System Requirements: Std. 41C

```

01*LBL "RCLZ"      05 RCL 13
02 CF 27           06 RCL 11
03 +               07 .END.
04 RDN

```

INSTRUCTIONS:

1. Load program and assign to a key if desired.
2. Press the **USER** key and then the assigned key. The sums of all the X-register values will be placed in the X-register. The "y" value sums will be placed in the Y-register, while the contents of 'Z' and 'T' remain unchanged. The previous 'Y' value has been overwritten (lost) and the previous 'X' value is now in LAST X.

This routine is very similar to the routine labeled "SIZE" with the difference that it quickly flashes the contents of all the available registers while checking for the SIZE of the machine. It was suggested by **Bill Danby** (Chicago, Illinois).

Stack contents are pushed up, so the T value is lost, but the previous X, Y, and Z values are retained. NOTE: This program runs about 50% slower than "SIZE".

LBL "VIEWSIZE" Assigned to: **☐** key.
 46 Bytes Exits USER mode upon
 15 Lines execution? Yes.
 08 Registers System Requirements: Std.
 41C

```

01*LBL "VIEWSIZE" 09 FS? 25
02 FIX 4          10 GTO 00
03 CF 27          11 FIX 0
04 SF 25          12 "SIZE="
05 -1.4           13 ARCL X
06*LBL 00         14 AVIEW
07 ISG X          15 .END.
08 VIEW IND X

```

INSTRUCTIONS:

1. Load program and assign it to a key if desired. (These instructions assume the routine has been assigned to the **☐** key.)
2. To find the SIZE the HP-41C is adjusted to, press **USER** and then **☐**. The program will immediately exit the USER mode upon execution.
3. Each register will be displayed very briefly, beginning with R00, until a limit is found. The display will then show:
 "SIZE = xx." (where 'xx.' is the SIZE).

This routine has been written for the user who doesn't have a printer (or when a printer is unavailable), to locate where the summation-registers block is located. The stack contents are altered, but no data registers are altered, nor is the summation-register block changed in any way. This routine requires three local labels (in this case, LBL 00 to LBL 02). The display is left in the 'FIX 0' mode when the program finishes execution.

Because of the method in which this routine searches for the sigma registers, the search can take some time. The routine has been designed for use when there is no other way for the user to locate where the summation-registers block has been located, but where altering of any data in

the registers cannot be allowed.

LBL "ΣFIND" Assigned to: **LOG** key.
 67 Bytes Exits USER mode upon execu-
 33 Lines tion? Yes.
 10 Registers System Requirements: Std. 41C

```

01*LBL "ΣFIND"    17*LBL 01
02 CF 27          18 XEQ 02
03 5.4            19 5
04*LBL 00         20 -
05 RCL IND X      21 INT
06 0              22 FIX 0
07 ENTER↑        23 "Σ REG="
08 Σ+            24 ARCL X
09 RDN           25 AVIEW
10 RDN           26 RTN
11 RCL IND Y      27*LBL 02
12 X<>Y          28 XEQ "X<>Z"
13 GTO 01         29 0
14 XEQ 02         30 ENTER↑
15 ISG X          31 Σ-
16 GTO 00         32 XEQ "X<>Z"
                  33 .END.

```

INSTRUCTIONS:

1. Load program and assign it to a key, if desired. (These instructions assume the routine is assigned to the **LOG** key.)
2. To locate the summation registers, press **USER** and then **LOG**. The program immediately exits the USER mode upon execution.
3. At program completion, the display will read:
 "Σ REG = xx." (where 'xx.' is the start of the six-register summation block).

(41C) Software Changes

If you have early copies of the following two pacs, you should make these corrections in your copy. Later copies have an addendum card, or the corrections have been incorporated in the printed copy.

HP-41C STANDARD APPLICATIONS BOOK

(1) Add a final paragraph for the description of "Root Finder" (page 38) as follows:

This program will calculate the closest obtainable approximations to a root, but may continue to iterate when the magnitude of the function evaluated at these approximations exceeds the tolerance. You can check the progress of the solution by inspecting the current guesses in registers 1 and 2 using the VIEW function. You may find it convenient to assign VIEW to some key.

(2) For some combinations of values, "Financial Calculations" (page 32) fails to converge to a solution for periodic interest i. This effect may be avoided by using a different initial value for i. You may try your own non-zero initial value by storing it in register 09 and executing label 06. The value you store should be expressed as a decimal fraction.

Example: Try a guess of 1%.

.01 **STO** 09 **XEQ** 06

(3) The program "Vector Operations" requires the instruction $x \Rightarrow y$ between steps 04 and 05 (page 52) in order to subtract vectors correctly.

MATHEMATICS PAC

This correction does not affect the module. It is only an operational correction.

When the order of the first coefficient is zero, the "Fourier Series" program (page 34) does not compute correct values of the series at user-specified points (t). After the coefficients have been calculated, use the following procedure to determine f(t) for a given t.

Keystrokes	Display	
RCL 22	(R)	Register number
2 STO ☐ (R)		
STO ☐ (R-1)		
USER		Set USER mode
t E	f(t)/2	
2 X	f(t)	

Alternatively, the program could be copied into program memory and line 160 changed to FS? 02. (However, in that case, one Memory Module is required to execute the program.)

Randomly Yours

That's a unique title, isn't it? It also happens to be the way that **Donovan E. Smith** of El Cerrito, California, closed the following letter to us.

This all started, innocently enough, with **Richard Cosway's** random number generator (RNG) that we printed in Vol. 3 No. 2 on page 11, column 3. It seems that the generator was not as random as we thought. First there was a letter from **Edgar N. Gilbert** of Whippany, New Jersey, stating that a seed of .657428 caused some very weird problems, and that there were other funny things amiss with that RNG. Then other letters started to arrive, and finally Mr. Smith's letter settled the matter: We should have checked Mr. Cosway's routine a bit more carefully. Anyway, let's let Mr. Smith have his say:

Richard Cosway's RNG reminded me of the early HP-65 days, when I was one of the users who pointed out the unsatisfactory short-cycle results of HP's original 8th-power RNG. In fact, I then became involved in considerable correspondence with the HP staff at Cupertino concerning possible replacements, culminating in rejection of several power functions and adoption of the still-popular congruential RNG, $x_{i+1} = 997x_i \pmod{10^7}$. Cosway's RNG therefore immediately struck me as likely to develop a short cycle within the first few thousand generated numbers, as with other power functions.

Consequently, I dug out my old HP-65 "Test Random Numbers" program (02356A) and made a quick-and-dirty conversion to a simpler HP-67 program. I'm sure you will agree that the following results indicate that your readers should be warned that Cosway's RNG, while ad-

mirably short, is dangerous to the user's statistical health and randomized games:

Seed	Numbers Per Cycle	Cycle Begins Somewhere Between
Zero	3,819	2,000 & 4,000
0.1	92	1,500 & 2,000
0.2	3,819	6,000 & 10,000
0.3	92	2,800 & 3,000
0.4	3,797	8,000 & 12,500
0.5	3,797	3,000 & 6,000
0.6	3,797	6,000 & 10,000
0.7	92	8,000 & 10,000
0.8	3,797	4,000 & 8,000
0.9	92	2,000 & 4,000

Cosway may have been deceived by the fact that his RNG often generates numbers with a mean and standard deviation close to the theoretical $\frac{1}{2}$ and $\sqrt{1/12}$, respectively. But the mean of the 92-number cycle is 0.602, which certainly would "tilt" most games, even if the repetition didn't matter.

P.S. A few "trial runs" with Victor Heyman's "Rectilinear Random Walk Test" program (02461D) would have alerted Cosway to the early appearance of short cycles. In fact, that's how I strengthened my initial suspicion before undertaking a search for the specific cycles with my own program. (Which reminds me to suggest that some KEY NOTES publicity for Heyman's numerous and excellent RNG test programs might help other RNG experimenters avoid embarrassment.)

(Ouch! I really asked for that one! It proves once again that you can never be too careful where RNG's are concerned. Mr. Cosway's routine is, however, a nice, neat, short routine, and I must give him credit for what he did. Perhaps we all learned something, and that's what it's all about. Ed.)

We Get Letters . . .

Since everyone reading this newsletter already has an HP calculator, we don't have to go out of our way to convince you they are the very best in their class. And you already know that "HP's" have survived everything from treacherous seas to crushing to Mt. Everest to outer space. So the following letter should not surprise you. It will, however, make you feel even happier about having chosen an "HP." And, we have to admit, it is an amazing story.

Gentlemen:
This summer, in central Alaska, I had the misadventure of being caught in a 50- to 70-knot windstorm, with snow and rain that lasted for three days. During the first night, our steel-framed office tent blew away, traveling 180 feet (5 meters) in the air before hitting the ground and completely destroying itself. My HP-19C that was connected to the battery charger went with the tent.

The next day, all that could be found of the unit was the recharger, with the plug-in wire torn away. Four days later, after the snow had melted, the calculator was found with the face torn open, exposing the circuitry. Naturally, I presumed the unit to be destroyed and was very distraught at the thought of proceeding with lengthy gravity

calculations, minus the programmable HP-19C, during the time in which a new unit could be ordered.

Though not really believing the calculator would be operable, I attempted to dry out the unit over a small heater. After 20 minutes, the display would light up and the normal functions became operable; this at least gave me straight key capabilities. Fifteen minutes later, the display would light up on program but would not "run." At this point I was ecstatic and continued with the drying process. Shortly thereafter, all capabilities returned, so I snapped the faces together. By checking polarity and hot-wiring the charger, the calculator was fully functional; the plug-in has since been replaced.

After this incident my respect, plus that of the other 23 people in camp, has greatly increased for HP calculators. Many thanks for producing such a fine unit that can withstand Alaska's worst spring weather conditions, lying outside, broken open, during three days of wind, snow, and rain.

I am considering the purchase of an HP-67 and was wondering if you would be introducing a more powerful version of the HP-67 within the next few months. Also, could you send me some information on the HP-41C? Buyer's guide information is hard to come by in Alaska.

Yours with thanks,
John F. Hendrick (M.E.)
College Station (Fairbanks), Alaska

"25 Words" (More or Less!)

It is no secret that most of you enjoy—and make use of—this column. We wish we could expand it to include all of the ideas you send. However, that is not possible, so we will continue to bring you those contributions that benefit the majority of our readers.

First contribution this month is from **Arvin Chaikin**, who lives in Rosendale, New York. If you've ever temporarily "lost" data, here is a neat idea.

(97) In many of the programs I write there is a need to enter numerous inputs in proper sequence. A simple five-instruction set, at each data entry point, allows me to keep track of where I am during entries by displaying a prompting index number.

DSP 0
1 (or any data index number)
R/S (Enter required data, then press R/S.)
 $x \Rightarrow y$
R↓

The index number, without zeros, is less likely to be confused with data, hence the DSP0. The $x \Rightarrow y$, R↓ places the used index number into the stack's T-register and the last entry back into the X-register.

Up to three data entries may be stored in X, Y, and Z-registers at any one time, if required. Remember, the T-register will contain the last index number displayed, not any data entry.

If your written data input list has corresponding index numbers, it is easy to take a coffee break, answer the telephone, etc., and still know where you were when you return to your calculator.

(Thanks for the nice comment about the

newsletter, Mr. Chaikin. See my "Editorial" for an answer to your Index question. Ed.)

Let's swing down to Litchfield Park, Arizona, for our next contribution, which was donated by **Martin C. Koenig, M.D.**

(67) **Y Modulo X**. This routine uses only the stack, does not require the use of registers, and may be executed manually. Used as a program, it consists of only four lines. When incorporated in a program in which X and Y are already in storage registers, it makes a subroutine superfluous. **Example:** 1309° 23' Module 360.

Program:	LBLA	Input:	1309.23
	÷		ENTER
	INT		360
	×		ENTER
	−		1309.23
	RTN		ENTER
			360

After keying in the program and the inputs, press \square and the display will show the result: 229° 23'.

How about another similar routine from **Dr. Martin C. Koenig**? And, below his routine, he has an interesting comment about KEY NOTES.

(67) The following routine may be used (1) in divisions when the remainder, if any, rather than a fraction is desired and (2) for X Modulo Y, in which case, line 9 may be omitted. Note that no register has been used.

001	LBLA	Dividend	008	INT
002	ENTER		009	-x- Quotient
003	ENTER		010	X
004	R/S	Divisor	001	−
005	÷		012	-x- Remainder
006	LSTx		013	RTN
007	$X \Rightarrow Y$			

I enjoy KEY NOTES very much and feel that "25 Words" is very helpful and should be expanded. Any even minor contribution like this may be of help to someone. After all, the majority of users are probably not mathematicians and need all the assistance they can get. It is for this reason that I consider "25 Words" the most interesting part of KEY NOTES.

(Thank you for the comment, Dr. Koenig. You will notice that this issue is a step in the right direction. Many readers feel the same as you do about KEY NOTES, so we'll try to please you. Ed.)

Next, a reference to an article in a rather old issue of KEY NOTES. But this routine does the job, and is the contribution of **Walter W. Steffen** of Indianapolis, Indiana.

(97) Here is a routine I have used for eliminating the congestion in the I-register. This subject was also covered in Vol. 2 No. 2 (page 5).

I have used this routine for preparing Model Office Tables, where 20 values are stored on various magnetic cards. It is then desirable to combine the values. Ten registers of two different magnetic cards are entered into the calculator, and then this routine combines them for storage onto a third magnetic card. Then the second set of 10 registers is entered, combined, and stored on the other side of the third magnetic card.

In this routine the function performed was addition, but any storage register arithmetic function

can be performed.

```

001 *LBLA      012 X=Y
002 EEX        013 ISZI
003 1          014 X=I
004 STOI       015 ISZI
005 CLX        016 EEX
006 *LBL5      017 1
007 RCL I      018 X=Y?
008 X=Y        019 RTN
009 X=I        020 R↓
010 X=Y        021 GTOS
011 ST+I      022 R/S

```

Now, a treat for HP-41C owners. This routine is from an old friend of KEY NOTES, who hails from the days of the original HP-65 KEY NOTE, and who is none other than Dr. Gus W. Weiss, Jr. of Washington, D.C.

(41C) Have some FUNction Analysis. This program, FUN, lets you evaluate any function where x returns $f(x)$ with both values being printed out of the print buffer; PRPLOT can then graph values as required. Key-in FUN (51 bytes) with minimum SIZE 013. R_{12} serves as the working register for STO and RCL of x (R_{11} services PRPLOT). To compute $f(x)$ start with x and XEQ FUN. As written, FUN increments x by 1, but the increment can be changed by deleting step 14 and keying your choice. The function is called up as AA and should be keyed as a subroutine, lead by SF 25 to avoid being stopped by DATA ERROR. After SF 25, key RCL 12 if you want to print numeric values by FUN. After scanning the numbers, delete RCL 12 under AA and insert STO 12; then XEQ PRPLOT in the way described in the *Printer Owner's Handbook*. If $AXIS=0$, the location of roots is made easier. FUN will generate $x, f(x)$ pairs until you stop it with R/S, while PRPLOT stops on x MAX. With flag 25 repeatedly set, be careful of improper or nondefined results that come about from reciprocals containing x at $x=0$, or roots or logs that might be negative.

MAIN PROGRAM:

```

01 *LBL "FUN"      10 SKPCHR
02 CLRG            11 XEQ "AA"
03 *FUNCTION VALUES 12 ACX
04 AVIEW          13 PRBUF
05 STO 12         14 1
06 *LBL 10        15 ST+ 12
07 RCL 12         16 GTO 10
08 ACX            17 END
09 3

```

FUNCTION VALUES

-3.00	4.83
-2.00	2.50
-1.00	1.50
0.00	0.00
1.00	-0.50
2.00	1.50
3.00	4.17
4.00	7.75
5.00	12.30

$$F(x) = x^2 - x^2/2 - 1/x$$

AA set for FUNCTION VALUES

```

01 *LBL "AA"      08 /
02 SF 25          09 -
03 RCL 12         10 RCL 12
04 X↑2            11 1/X
05 RCL 12         12 -
06 X↑2            13 END
07 2

```

AA set for PRPLOT

```

01 *LBL "AA"      08 /
02 SF 25          09 -
03 STO 12         10 RCL 12
04 X↑2            11 1/X
05 RCL 12         12 -
06 X↑2            13 END
07 2

```

Wide range PRPLOT

```

PLOT OF AA
X <UNITS= 1. > ↓
Y <UNITS= 1. > ↓
-15.0      15.0
          0.0
-----|-----
-4.00      | x
-3.00      | x
-2.00      | x
-1.00      | x
0.00       | x
1.00       | x
2.00       | x
3.00       | x
4.00       | x

```

Narrower range PRPLOT

```

PLOT OF AA
X <UNITS= 1. > ↓
Y <UNITS= 1. > ↓
-4.00      4.00
          0.00
-----|-----
-1.00      | x
-0.50      | x
0.00       | x
0.50       | x
1.00       | x
1.50       | x
2.00       | x

```

Although the next routine is not a sophisticated technique, it is a very handy thing for group lunches. It was donated by William L. Slayton, of Washington, D.C.

(67/97) Have you ever gone to lunch or dinner with a group where there is but one check and the

group has agreed that each will pay his/her exact portion? The spread sheet calculations: taking sales tax, tip, and each individual check into account are considerable. This "25 Words" routine does the job.

Store sales tax % in A; store tip % in B; store number of people (six or less) in C. Store, member one's bill in 1; member two's bill in 2; etc. Press [A].

The answer will show each member's original bill and his/her bill plus tax and tip. The total amount ordered is stored in 7, the total tax in 8, the total tip in 9, and the final tab in 0. And, of course, each member's total tab is retrievable by recalling 1, 2, 3, etc.

```

001 *LBLA      014 %
002 ISZI      015 ST+I
003 RCLC      016 ST+8
004 RCL I     017 RCLB
005 X>Y?      018 LSTX
006 RTN       019 %
007 DSP0      020 ST+I
008 PSE       021 ST+9
009 DSP2      022 RCL I
010 RCLA      023 PRTX
011 RCL I     024 ST+0
012 PSE       025 GTOA
013 ST+7      026 RTN

```

(There are many versions of this routine, but Mr. Slayton was first to contribute one. Besides, we thought it might come in handy over the holidays. Ed.)

We've had only one other woman contribute to this column before, so the following donation is a welcome change of pace. Our second contributor is Janice Vaillant of Rego Park, New York, and, at age 16, she already knows quite a bit about programmable calculators.

It is easy to count program lines in determining whether or not it pays "to save program lines by writing a subroutine." But after many such counting episodes, it was clear that I needed a simple, general expression for the number of lines saved. The article by C. Close (August '79) prompted this letter for your section "25 Words" (More or Less!).

If there are k identical lines repeated N times throughout a program, the number of lines saved by a subroutine for the k steps is

$$(N-1)(k-1)-3$$

For example, for 3 repetitions of 5 lines each, a subroutine saves 5 lines. (This is less than the $15-8+1$ implied by the article in your issue of August '79.) The symmetric form of this expression makes it easy to remember.

Here is another contribution from a doctor. It seems that doctors make good use of programmable calculators! This HP-41C routine is from David Feigal, M.D., of Sacramento, California.

(41C) Since I haven't yet received my Statistical Module for my HP-41C, this program may be inelegant or redundant, but since correlations are used so often it may be a helpful routine for

your "25 Words" section.

01 *LBL "PPM"	13 RCL 01
02 SDEV	14 ENTER↑
03 *	15 RCL 03
04 RCL 06	16 *
05 RCL 06	17 -
06 1	18 X<Y
07 -	19 /
08 *	20 "R="
09 *	21 RCL X
10 RCL 06	22 RVIEW
11 RCL 05	23 END
12 *	

This program takes advantage of the SDEV function built into the HP-41C to short-cut some of the computations and to save program lines. I also took advantage of another HP-41C option, of moving the statistic registers to 01 thru 06.

Now, let's see what Ramon Santoyo does with his calculator in Mexico City, Mexico.

(67) In reference to the three "Slow Register Review" routines contributed by C. J. Caldwell in the May 1979 issue of KEY NOTES, I suggest another routine for the same purpose. It is also 15 steps long and it ends in a "normal" way:

001 *LBLA	009 XZY
002 0	010 2
003 STO1	011 5
004 *LBL1	012 X=Y?
005 RCL1	013 RTN
006 PSE	014 ISZ1
007 RCL1	015 GT01
008 PRTX	016 R/S

For a faster sequential register review, change line 8 to PAUSE, and for a better register identification, try DSP 0 between lines 5 and 6, and DSP 2 between lines 7 and 8, making the routine 17 lines long.

Do you know if in the U.S.S.R. they have a programmable calculator like the HP-67? Or do they use HP's?

(Your routine works okay, Mr. Santoyo, but watch that I-register. Yes, they use HP calculators in Russia. However, I do not know of any card-programmable calculator that is made in Russia. Perhaps one of our readers can elaborate on this? Ed.)

From Mexico, let's return to Bedford, Virginia, for a contribution from Arthur N. Ogden, and an answer to a previous article in KEY NOTES.

(97) This letter is in response to C. J. Caldwell's letter which appeared in the May 1979 KEY NOTES. I spend considerable time with navigation problems and have always used the "Mod" function for the purpose he describes. This function yields the remainder of the division x/y , when the quotient is constrained to be an integral value. In terms of the intrinsic functions

available on the HP-97 or HP-67, it is most simply expressed as: $\text{Mod}(x,y) = (\text{FRC}(x/y))y$. I am enclosing several programs which I have used for this purpose. They are not limited to any maximum or minimum values except by the capacity of the calculator. Program I returns the positive equivalent of any angle and does not use any storage registers. However, the contents of the stack are lost. If negative angles are acceptable (trig functions are the same for negative angles as for their positive equivalent), then program II saves a couple of program lines.

PRGM I	PRGM II	PRGM III	PRGM IV
001 *LBLA	001 *LBLA	001 *LBLA	001 *LBLA
002 3	002 3	002 RCL0	002 RCL0
003 6	003 6	003 ÷	003 ÷
004 0	004 0	004 FRC	004 FRC
005 ENT↑	005 ENT↑	005 RCL0	005 RCL0
006 ENT↑	006 ENT↑	006 X	006 X
007 R↑	007 R↑	007 X<0?	007 RTN
008 XZY	008 XZY	008 GT00	
009 ÷	009 ÷	009 RTN	
010 FRC	010 FRC	010 *LBL0	
011 X	011 X	011 RCL0	
012 X<0?	012 RTN	012 +	
013 +		013 RTN	
014 RTN			

And while we are on the subject, here is another reply to C. J. Caldwell's article. It is from Willem L. C. Brunings of Bilthoven, Holland. Notice the similarity to the routines above, yet it is different.

(67) In HP KEY NOTES, Vol. 3 No. 2, page 9, Mr. Caldwell hit some jackpot to have solution angles bounded between 000 and 360. Some shorter solutions, without any upper or lower limits appear below.

$0 \leq \theta < 360$

001 *LBLA
002 3
003 6
004 0
005 +R
006 +P
007 XZY
008 X<0?
009 +
010 RTN

Examples: $3603^\circ = 3^\circ$
 $-896^\circ = 184^\circ$

$-180 < \theta \leq 180$

001 *LBLA
002 1
003 +R
004 +P
005 XZY
006 RTN

Example: $182^\circ = -179^\circ$

In my experience, the value "360" is used so frequently that I find it economical of program lines to have it stored for recall when needed. Program III assumes that 360 is stored in register 0. Program IV is the equivalent program if negative values are acceptable. It is the most economical of program lines, and is the one I most frequently use.

I hope that these comments will be of interest to Mr. Caldwell, and perhaps to others. I do enjoy KEY NOTES very much, and find much interesting and useful material in it. Keep it coming.

The second routine was designed to be used in navigation programs when longitude is between -180 and $+180$.

I also use in some of my navigational programs, for latitudes between -90° (south) and $+90^\circ$ (north), a variation of the first program, as follows:

001 *LBLA	012 1
002 STO0	013 +R
003 9	014 +P
004 0	015 R↓
005 RCL0	016 2
006 ABS	017 ÷
007 XZY?	018 CHS
008 GT00	019 STO0
009 RCL0	020 *LBL0
010 2	021 RCL0
011 X	022 RTN

001 *LBLA	011 CHS
002 STO0	012 +R
003 9	013 +P
004 0	014 R↓
005 RCL0	015 CHS
006 ABS	016 STO0
007 XZY?	017 *LBL0
008 GT00	018 RCL0
009 RCL0	019 RTN
010 1	

There are differences between the two routines, but they give the same results between the limits -180 and $+180$.

(Continued)

Let's go back to the U.S., now, and hear from **Rand E. Gerald** of Deerfield, Illinois, who has some further ideas about improving SD-12B ("English-SI Conversions").

(67) While reading the Vol. 3 No. 2 issue of KEY NOTES, I discovered that **Frank A. D'Amico's** idea in "25 Words" on page 6 could be carried one step further. Add a RTN immediately following lines 001 LBL1 and 082 LBL2. Then replace the RTN's at lines 014, 027, 046, 065, and 081 with GTO1's and the RTN's at lines 032, 102, 121, 138, 155, and 173 with GTO2's. The program will then reset itself to the proper point after each calculation, and the GTO1 or GTO2 need only be keyed in when "Changing Sides."

How about another tip? We'll bet that a lot of people never thought of this. It is from **T. R. Bainbridge** of Kingsport, Tennessee.

(97) The "print stack" key of the HP-97 is an easy way to display the input to a program. When four items are entered with the aid of the stack, merely call for "print stack." If only three items are entered with the aid of the stack, use the following:

	001	*LBLA
The equivalent operation is	002	0
also possible with the	003	R↓
HP-67.	004	PRST
	005	R/S

We haven't heard from Germany (West) for a while, so let's try this suggestion from **Henning Legell**, who lives in Eutin.

(67) Another method for "Too Many Inputs/Outputs" (Vol. 3 No. 2) is to program a software prefix. This may have advantages in some cases. The best is to explain it by an example.

001	*LBLA	013	1
002	1	014	1
003	GTO9	015	GTO9
004	*LBLB	016	*LBLB
005	2	017	1
006	GTO9	018	2
007	.	019	GTO9
008	.	020	.
009	.	021	.
010	*LBLC	022	.
011	RTN	023	*LBL9
012	*LBLA	024	RTN

Line 010 is the software prefix. (1) Every routine has to end with GTO9 instead of RTN. (2) Two equal labels are separated by the software prefix. (3) The last routine is LBL9, RTN.

Thus, you can build up to 10 non-numerical labels (or even more, by programming another software prefix). In my example, the numbers are short "programs." So, key in the routine and

Press A	You see 1
Press E A	You see 11
Press E B	You see 12
Press B	You see 2

and so on.

Now, here is an interesting letter from a new HP-41C owner, **James C. Van Pelt**, of Branford, Connecticut. As you will see, he almost couldn't wait to write!

(41C) Most programs have an initializing subroutine in which registers are cleared, flags and constants are set, etc. With the HP-41C, letters can be displayed during this subroutine so they appear over the top row of keys and in effect "label" them at the beginning of each program run.

It is very important to be sure to include a title in each program; e.g., LBL TITLE, or whatever. If the labels in a program are all numbers and single letters from A through J, you can easily lose access to a program and not be able to recover it. (If this is incorrect, please let me know!)

This "title label" can also serve as the label for the initializing routine. For quick access to a program, insert LBL TITLE ("TITLE" is any title) at the point in the program where you want to begin, and then assign XEQ TITLE to your initialization key—usually the "a" key. After that you will be able to run the program immediately just by pressing that key in USER mode.

An easy way to get your program to run as soon as you turn on the calculator: At the end of each major subroutine, substitute GTO 14 for RTN. Right before the initialization subroutine, enter:

```
LBL 14
FS? 11
RTN
SF 11
LBL TITLE (beginning of program)
```

As a result, the program will automatically start executing at the beginning of the program whenever the HP-41C is turned on. Since Flag 11 will be cleared by turning on the calculator, execution will skip over the RTN, set Flag 11, and then go into the initializing subroutine. After this, until the HP-41C is turned off, execution will stop on RTN after LBL 14 because Flag 11 will always remain set.

I use my HP-41C mostly for financial applications. Using this trick, whenever I turn on my HP-41C the letters "N, I, PMT, PV, FV" appear in the display, in effect "labeling" the top five keys used in the standard finance program; this program is then all ready to run without further initializing. If I run another program in the machine, I can "reset" the HP-41C for the finance program simply by pressing the initializing key.

To add another wrinkle, insert LBL 13, BEEP just before the LBL 14 in the above. Then any major subroutine ending in GTO 13 will cause the HP-41C to alert you with a "beep" that the answer is ready. (Of course, LBL 13 and LBL 14 could be reassigned any other number of letters, but the numbers from 00 to 14 use the least memory.)

To add some fun when you introduce the HP-41C to the children, enter the following in PROGRAM mode: LBL TONE0, TONE 0, RTN. Then do the same for digits from 1 through 9. Next, assign the keys 0 through 9 in USER mode as follows: key 0 to TONE0, key 1 to TONE1, etc. (There's probably a more elegant way to do this, but I just got my HP-41C yesterday!) Now whenever the keys 0 through 9 are pressed in USER mode, the corresponding tone will sound: you

have a little electronic music-maker! The notes are placed along the minor scale and don't allow much melody-making, but you can play a passable "Sixteen Tons" in no time!

I hope you can use all or part(s) of this. With the HP-41C, we are standing on the brink of a whole new world in programmable calculating; these are just some first glimpses.

**(You are incorrect, here, Mr. Van Pelt. Refer to the article, "HP-41C Tips and Techniques," on page 3. Ed.)*

Contributions can originate from Houghton, Michigan too! Here is one from **W. A. Hockings**, who is a Research Fellow at Michigan Technological University.

(67/97) I would like to submit the following chi-squares evaluation routine for your "25 Words" column. It uses only 14 lines and no storage registers, as compared to 25 lines and 3 registers in the routine by **Michael Tarnowski** in the Vol. 3 No. 3 KEY NOTES.

To enter the routine press **A**, then key in the values of E_i and O_i, following each key-in with an **R/S**. The cumulative value of χ^2 is displayed after each pair of numbers is entered. To leave the routine press **B**. Wrong values can be deleted by keying in the wrong values with their signs reversed.

001	*LBLA	008	X ²
002	CLX	009	X ² Y
003	*LBL0	010	=
004	R/S	011	+
005	ENT↑	012	GTO0
006	R/S	013	*LBLB
007	-	014	RTN

What about a routine that will work on any HP calculator? Here is one from **Rodney E. Wood** of Chatsworth, California.

(All) Once in a while a user has a need to compute a percentage of total (%T) on an HP calculator. Most would divide the numbers and multiply the total by 100, but this routine would require at least four lines and would eliminate both entries. The contribution below will accomplish the same answer in only three lines (plus LBL A and RTN) and have the added benefit of retaining the TOTAL in the Y-register while X becomes the %T. The reverse function is, of course %.

The routine is: LBL A, 1/x, %, 1/x, RTN. The TOTAL is entered into the Y-register, while the number to be converted goes in x. The number x is replaced by %T after the operation, but the rest of the stack remains unaffected. It should be noted that any HP calculator currently produced, programmable or not, should be able to use this routine.

In Marathon Shores, Florida, it is only natural that calculators are used for navigation. But here is a rather clever application, thanks to **Charles H. Bowles**.

(67/97) Want to set your HP-67, take a celestial navigation sight or any other useful chore, press **B** when the task is accomplished and see the exact time of your act? Then key in a watch time as hours, minutes, minutes, seconds, sec-

onds. When your watch says exactly that, press **E**. When you reach your "mark," press **B** (twice, if needed). The exhibit will be the exact time of your "mark" in sexagesimal format (h.mmss).

To test the program, run as above, but at mark, note your watch and key in that time after pressing **B**. Then press **C**: you will see the real watch time, then a % error, then elapsed time, then the actual error in seconds (h.mmss). Press **A/S** and see the % error in digit form.

001	*LBLA	024	HMS+
002	.	025	STO4
003	0	026	→HMS
004	0	027	RCL1
005	0	028	RCL5
006	1	029	CHS
007	HMS+	030	HMS+
008	STOA	031	STO3
009	*LBL0	032	→HMS
010	JX	033	=
011	LSTX	034	RCL1
012	SPC	035	X↔Y
013	SPC	036	RCL3
014	GTOA	037	RCL4
015	*LBLB	038	PRST
016	RCLA	039	R/S
017	R/S	040	R↓
	*LBL5	041	R↓
019	STOE	042	EEX
020	GTOA	043	2
021	*LBLC	044	x
022	STO1	045	R/S
023	CHS		

Comment: LBL0 fine-tunes the program for accuracy. It was arrived at by much experimentation, but I am sure your readers can improve on it. For up to 5 minutes it is nearly perfect. After that an error of 1 or 2 seconds may appear. That is why LBL0 needs more work on it. I invite improvement!

("Timer" routines are not accurate, Mr. Bowles. There are too many variables. All you can do is recognize how accurate your routine is over certain intervals and live with that. No timer routine can ever begin to approach the accuracy of a quartz-crystal or chronometer movement. Ed.)

Are you ever in doubt about what display mode you are in? If so, then **Emerson J. Perkins** of Huntington Beach, California, has the answer to your problem.

(67/97) Here is an unusual routine that literally calculates what display mode the HP-67/97 is in. It can be used in a program to detect how many digits accuracy the user has selected.

001	*LBLA	007	-
002	FIX	008	1/X
003	3	009	LOG
004	1/X	010	INT
005	ENT↑	011	RTN
006	RND		

Example: (1) Key-in routine. (2) In RUN mode, press **DSP** 5. (3) Press **A**. (4) See "5" in the display.

Notice that the routine sets the display format to FIX. If the calculator is already in FIX mode, then this is not necessary in the routine. SCI and ENG modes produce results offset by 1 (one) and "Error" for DSP 9.

An unusual application of this routine is to recall a digit 0 thru 9 that was stored in the display mode using DSP (I) to store it. In this way, the contents of the I-register may be saved and retrieved without using an extra register! Again, only the 10 integers 0 thru 9 may be retrieved. An "Error" display will occur if the number is outside the range $-10 < R_i < 10$ when using DSP (I). (See page 224 of the owner's handbook.)

This scheme can also be used for a software flag with 10 states. For example, DSP 0 could clear it and DSP 1 could set it (to use only two states). After using the display as a flag, the display format may be changed as desired for suitable output of results.

Here is another "routine" very nearly the same as the one above. Notice how **John P. Beyer** of Woodland Hills, California, has done very nearly the same thing as Mr. Perkins, but in a different way.

(67/97) Here is a small program fragment that might make an entry for your "25 Words" feature. I've put label and return statements around it to make it look like a subroutine, but calling it a subroutine would be like calling a carburetor a Fuel-Air System.

The purpose of the routine is to determine the setting of the display control within the program so that it can then be manipulated numerically. The routine returns an integer that is equal to the setting of the display control. For example, if the calculator is set (manually) to DSP 6, the routine will produce the answer "6." I first used it as part of a larger program in which I used the display setting to provide error control, but then found that I needed to control the control, so to speak. Here is the routine:

001	*LBLA	007	ABS
	3	008	1/X
003	1/X	009	LOG
004	RND	010	INT
005	LSTX	011	RTN
006	-	012	R/S

Notice that in the second line the constant "3" could as well have been a "6."

This routine refers way back to an early issue of 1979 (Vol. 3 No. 1) but we think you'll like it. This was submitted by **Gilbert M. Halpern, M.D.**, of Honolulu, Hawaii.

(67/97) **Peter Baldwin's** routine for storing in all registers in KEY NOTES, February 1979, page 6, is fine, but I was unable to perform arithmetic after recalling the contents of the registers because "N" used to recall the contents of any particular register is interposed in the stack at Y and the contents are in X and Z: I added $x \rightleftharpoons y$ and R_i at step 022 and now the recalled contents of the registers are in X and Y of the stack.

001	*LBLA	014	GTOA
002	STO1	015	RTN
003	RCL1	016	*LBLC
004	ISZI	017	ENG
005	RCL1	018	X↔I
006	R/S	019	RCL1
007	RTN	020	X=0?
008	*LBLB	021	RCL1
009	STO1	022	DSP4
010	RCL1	023	X↔Y
011	0	024	R↓
012	0	025	STOE
013	ISZI	026	RTN

Again we have a reference to a former routine. This one is the contribution of **Vic Schmidt**, who is in the Geology Department of the University of Pittsburgh in Pennsylvania.

(67/97) **Coy Morris's** routine for conversion between spherical and cartesian coordinates (Vol. 3 No. 2, page 9) is good for keyboard entry of the data, but less useful as a routine to be imbedded in a program. I have found the following four routines to be very efficient.

Spherical to Cartesian. Begin with the stack for X:r, Y:5u, Z:φ, T:A, where A is any number and use R_{\leftarrow} , R_i , R_{\leftarrow} to get X:x, Z:A, T:z.

Cartesian to Spherical. The exact inverse is obtained with $\rightarrow P$, R_i , $\rightarrow P$ (remember to start with z in T).

Latitude, Longitude to Direction Cosines. Begin with X:1, Y:north latitude, Z: east longitude, T:A; then use R_{\leftarrow} , $x \rightleftharpoons y$, R_i , R_{\leftarrow} to get X:c₁, Y:c₂, Z:A, T:c₃.

Direction Cosines to Latitude, Longitude. The inverse is obtained with $\rightarrow P$, R_i , $x \rightleftharpoons y$, $\rightarrow P$.

If you use the fifth program in the *ME Pac*, here is a new idea for you from **Dennis M. Ryan**, who is a Staff Mechanical Engineer at Ampex Corporation in Redwood City, California.

(67/97) Users of ME1-05A, who deal with predominantly rectangular or circular cross-section beams, may find it useful to re-record the program with the addition of one of the following routines in the space available:

RECTANGULAR CROSS-SECTION

b†h

211	R/S
212	*LBL5
213	STO9
214	X ²
215	x
216	6
217	-
218	STOD
219	ST×9
220	2
221	ST÷9
222	RCL9
223	RTN

(Continued)

CIRCULAR CROSS-SECTION

```

211  R/S
212  *LBL E
213  STO9
214  3
215  YX
216  P↓
217  X
218  4
219  ÷
220  ST00
221  ST×9
222  RCL9
223  RTN
    
```

With the input "b ↑ h" or "r" respectively, this routine calculates section modulus,

$$Z = \frac{bh^2}{6} \quad \text{or} \quad Z = \frac{\pi r^3}{4}$$

places it in register D for subsequent stress calculations, then calculates, stores (R_9), and displays moment of inertia,

$$I = \frac{Zh}{2} \quad \text{or} \quad I = Zr$$

ready for input into LBL b. Register R_9 may later be replaced with sums of shear (V) as you (HP) have suggested, or used to retain I for the future calculations.

Obviously, these same routines also can be incorporated into ME1-06A, using the unoccupied register B to store Z.

We hadn't heard from Canada for a while, but Terry Mickelson of Duncan, B.C., solved that problem with this neat, short routine and example.

(67) The $A/B = C/D$ ratio equation can be solved in the 21 steps of this program. Key in the values for A, B, C, and D, with the unknown quantity input as zero. The program will output the answer for the unknown value after [A] is pressed. Example: (A) 13, ENTER; (B) 22, ENTER; (C) (unknown) 0, ENTER; (D) 44, [A]. After [A] is pressed, the answer "26" will appear in the display. This might be an ideal subroutine since no registers or flags are used.

		EXAMPLES:
001	*LBL A	
002	X=0?	13.00 ENT↑
003	GT01	22.00 ENT↑
004	R↓	0.00 ENT↑
005	X=0?	44.00 GSBA
006	GT02	26.00 ***
007	R↓	
008	X=0?	0.00 ENT↑
009	GT01	22.00 ENT↑
010	R↓	26.00 ENT↑
011	*LBL 2	44.00 GSBA
012	R↓	13.00 ***
013	÷	
014	X	
015	RTN	
016	*LBL 1	
017	R↓	
018	X	
019	X≠Y	
020	÷	
021	RTN	

HP-41C Owner's Handbook and Programming Guide Addendum

This addendum contains updating information for the *HP-41C Owner's Handbook and Programming Guide*, part number 00041-90001 Rev. B, printed in August, 1979.

The following two paragraphs should be included as the first items under Service on page 242:

Using state-of-the-art technology, the HP-41C Continuous Memory circuits operate continuously—even while the calculator is turned off. Because these circuits are always drawing very low power from the batteries, they are susceptible to disruption at all times. Disruption can be caused by inserting or removing plug-in modules or peripherals while the power is turned on; electrostatic discharge to the unit; strong magnetic fields; plugging devices into the HP-41C that are not supported by Hewlett-Packard for use with the HP-41C; or other conditions that can traumatize the calculator.

Of course, all causes of disruption should be avoided, but should disruption occur, the most common symptom is a loss of keyboard control of the calculator. The HP-41C has been designed to allow recovery from these conditions. The procedure for resetting the calculator is to simply remove the battery pack and replace it again immediately. This will reset the HP-41C without causing a **MEMORY LOST** condition (unless the trauma itself was great enough to cause a **MEMORY LOST** condition). After several attempts, if this procedure fails to reset the calculator, work through the service procedure in the *HP-41C Owner's Handbook and Programming Guide*, page 242.

HP KEY NOTES

November 1979 Vol. 3 No. 4

Programming and operating tips, answers to questions, and information about new programs and developments. Published periodically for owners of Hewlett-Packard fully programmable personal calculators. *Reader comments or contributions are welcomed. Please send them to one of the following addresses.*

Hewlett-Packard Company
Users' Library
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330 USA

Hewlett-Packard SA
USERS' CLUB EUROPE
7, Rue du Bois-du Lan
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

Address Correction Requested

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 814
PORTLAND, OR