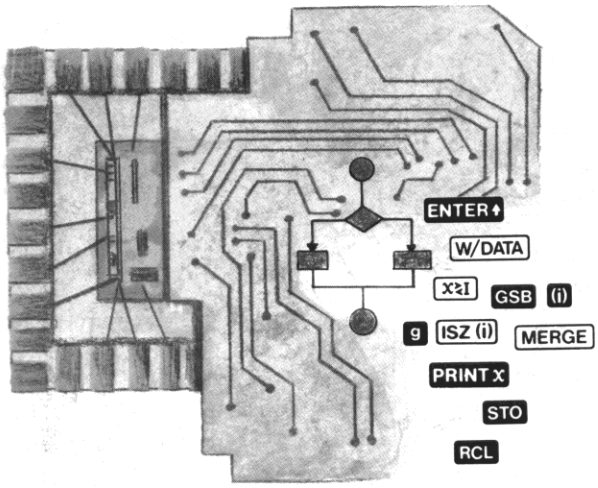


Featuring, this issue:
Watzin the Registers? 3
The "Other" Software? 3
Anti-Subroutines **and** Ersatz Memory? . . 4
HP-67/97 Looped Program Merge 4
"25 Words" (More or Less!) 5, 6, 9
KEY NOTES Order Form 7, 8
The Iddi Wizard of Id! 10
Let's Hear it Again for RPN 11
Free Software Offer 12

February 1979 Vol. 3 No. 1

HEWLETT hp PACKARD

HP Key Notes

And The Walls Came Tumbling Down ...

Yes, that *is* the Great Wall of China in the accompanying photograph! And the person in the photo is **Gary M. Tenzer,** of Pacific Palisades, California. Mr. Tenzer recently visited China on his vacation and took with him his HP-67. Here's a letter from him about this unusual experience.

Dear Henry:

With all of the attention given to the Peoples Republic of China in the past few months, I thought that it might be interesting to share with your readers some of my experiences on a recent trip to China with my trusty traveling companion, my HP-67.

My tour of China was limited to 16 days and three cities: Canton (Kwangchow), Shanghai, and Peking (Beijing). The Chinese people are very solicitous and warm but the language barrier forclosed much verbal communication.

As you know, China's technological progress has been minimal over the last 30 years. It is indeed odd to walk into a department store in Canton (yes, they do have department stores!) and see the electronic equipment for sale. The TV sets and radios are all the vacuum tube variety, the type sold in this country 25 years ago. It is of little wonder that, when the Chinese people interact with Americans, a cultural shock takes place.

It is impossible for an American to walk the streets of a Chinese city or through a store without being mobbed by literally hundreds of people curious about visitors from the West and their sophisticated American gadgets. Often, I was approached by people anxious to examine my camera equipment and my digital watch. They were particularly excited when I showed them my HP-67 in order to share an example of superior American technology. They were awed! I would then do some arithmetic calculations, communicating in the universal language of mathematics. Several of the young people around were familiar with algebra, as it is required in the secondary schools. They were most excited about the machine's potential for use in their studies.

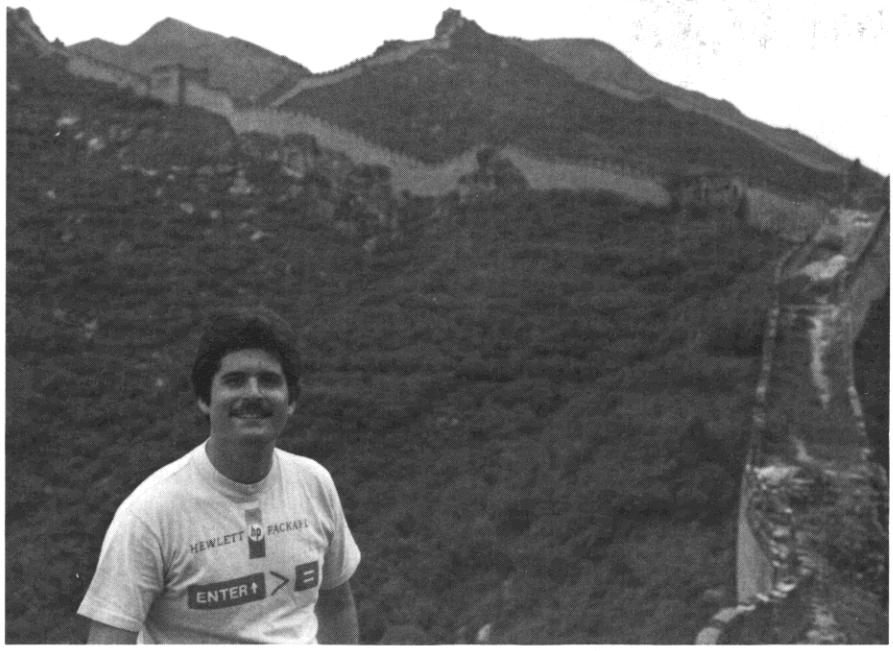
The Chinese interpreter for out group had some knowledge of engineering and math. Because he spoke English, I was able to demonstrate some of the programmable features of my HP-67 by programming some simple arithmetic functions and some branching routines. He was especially impressed by the "Polynomial Evaluation" and the "Matrix Operations" programs as well as some mathematical games such as "Bagels."

The experience of visiting a country on the verge of a technological revolution was like a trip back in time. The Chinese people are fascinated with American technology and are eager to grasp its wonders. I look forward to visiting China again to view the effects of further Westernization and technological progress.

I have enclosed a photograph that was taken of me standing on the "Great Wall of China," proudly wearing my Hewlett-Packard, *ENTER GREATER THAN EQUALS T-shirt.*

Kindest regards,
Gary M. Tenzer

(Mr. Tenzer holds an AB degree in Economics from the University of California (Berkeley) and two Masters degrees (MBA and MSBA) in Finance and Real Estate Finance from the University of Southern California. He also holds a California Real Estate Broker's license. Currently he is entering a career in the Real Estate Development industry. Gary spends much of his spare time putting his HP-67 and HP-97 to work for him in investment analysis as well as in consulting with businesses to help them get the most beneficial use from their HP programmables. He is an active member of PPC, the calculator user's club, as well as a regular contributor to the PPC Journal, *the club's newsletter. Ed.)*



All prices in this newsletter are suggested retail prices excluding applicable state and local taxes–Continental U.S.A., Alaska and Hawaii

Library Corner

Every paid subscriber to the Users' Library should have received *Catalog Addendum #3* by now. It brings the total number of programs in the Catalog to 2,750. If you haven't received your copy, you may need to renew your subscription.

If you have not already done so, check out appendixes A and B of *Addendum #3*. Appendix A lists all those popular programs from HP Application Pacs and *Users' Library Solutions* books. Appendix B lists collections of Library programs at reduced prices, plus supplies and accessories for your HP-65, HP-67, or HP-97.

ORDERING PROGRAMS

None of the individual programs in this issue are available in Europe at this time. They will probably be added to the European Library Catalog at a later date. And, don't forget, if you live in the U.S.A., you can order Library programs you see here in KEY NOTES by calling the toll-free number on the Order Blank.

Library programs are available in two forms: A set of the program listings and instructions (software) is $3*, and the fee is $5* for a set of software *and* a recorded magnetic card.

NEW HP-67/97 PROGRAMS

Here is a "set" of programs that deals directly with vapor pressure or vapor-liquid equilibrium in mixtures or solutions of various liquids. All of the programs in this set are the work of **Ora L. Flaningam,** who is a chemist with Dow-Corning Corporation in Midland, Michigan. Much of his work is in physical chemistry and thermodynamics of solutions (vapor pressure, vapor-liquid equilibrium, and distillation).

ORA L. FLANINGAM

Many of his programs started out being written in BASIC for a time-share computer. With these now rewritten for his HP-67, he does much less walking to the computer terminal. And in his words, "Being able to write them on my own calculator teaches me much more about solution thermodynamics than simply using 'canned' programs written by someone else." Some programs were fairly straightforward and some required several weeks of steady work.

Mr. Flaningam used to collect BASIC routines, but has now switched to HP-67/97 routines. His new collection already fills more than five looseleaf notebooks.

And does he *like* his HP-67? He answers: "I really do love my HP-67. It, like its predecessor HP-35, has saved me untold hours of calculation time. It is always at hand, on my belt, ready for whatever I need. And my wife and daughter enjoy playing games with the calculator, especially *Mastermind* and *Arithmetic Teacher*. My daughter, Lian (age 11), has also started working her way through the *HP-67 Owner's Handbook*. She has gotten as far as section 6!" 17 apr 79

Below are the programs in this set. You can order individual programs by their Library number or the entire set as: **Vapor Pressure Set #67000-99988,** for $35.50* (Not available in Europe. See **Ordering Programs**.)

01025D Antoine Equation for Vapor Pressure Correlation (8 pages, 299 steps)
01628D Composition Conversion Program (5 pages, 144 steps)
02197D Binary Vapor-Liquid Equilibrium, Part 1 (6 pages, 224 steps)
01939D Binary Vapor-Liquid Equilibrium, Part 2 (6 pages, 189 steps)
02062D Bubble and Dew Points for Non-Ideal Binary Solutions (7 pages, 333 steps)
02348D Distillation of Ideal Binary Solutions (5 pages, 106 steps)
02353D Generalized Vapor Pressure Equation for Non-Polar Fluids (7 pages, 137 steps)
02488D Smoker Equation for Binary Distillation (5 pages, 181 steps)
03240D Ternary Vapor-Liquid Equilibrium (5 pages, 106 steps)
03238D Binary Vapor-Liquid Equilibrium, Part 3 (8 pages, 371 steps)

During a recent long illness, **William A. Griswold** of Nashville, Tennessee, found time to write and beautifully document a fine selection of programs that will be useful to anyone involved in Mechanical Engineering Stress Analysis.

We are pleased to report that Mr. Griswold has recovered and has gone back to work at AVCO, Aerostructures Division. And we think that you will find Mr. Griswold's programs of value individually, or better yet, as an entire set.

Below are the programs in this set. (The third one has six pages, the rest have five.) You can order individual programs by their Library number or the entire set as: **Analysis for ME's #67000-99989,** for $36.50.* (Not available in Europe. See **Ordering Programs.**)

03271D Mechanical Properties (Rectangular Sections) and MC/I Stresses (112 steps)
03272D Mechanical Properties (Rectangles, Triangles, Circles, Sectors, Fillets) (224 steps)
03273D Mechanical Properties (Oblique Rectangles, Known Sections, Rotation) (224 steps)
03274D Mechanical Properties (Principal Axes) (78 steps)
03275D Mechanical Fastener Analysis (164 steps)
03276D Fastener Reaction—Eccentric Load (210 steps)
03277D Inter-Rivet Buckling (195 steps)
03278D Reinforced Hole Analysis (189 steps)
03279D Lug Analysis (171 steps)
03280D Column Strength (184 steps)
03281D Effective Width of a Stiffened Web in Compression (35 steps)
03282D Beam Column (222 steps)
03283D Compression Buckling (75 steps)
03284D Shear Buckling Stress (56 steps)

(A superb creation, Mr. Griswold; one of the neatest packages I've ever seen. My congratulations to you. Ed.)

Now, here is a truly monumental accomplishment in HP-67/97 programming. The following program, written by **Ronald M. Eades** of Hampton, Victoria, Australia, is 63 pages long, uses 9 cards, and totals 1,697 steps. It is carefully and beautifully documented, with plenty of explanations, charts, **actual listings from programs,** etc.

The nine cards include calculations for testing, tabulating, listing, printout, etc. The program is designed to achieve the following types of analysis with a minimum of effort.

1. By organizing data into statistical tables of varying types in which the grouping helps to show the characteristics of the data in a way that is not otherwise possible.
2. By graphing the data in various forms to give further information in a pictorial manner.
3. By analyzing the data mathematically to yield a variety of statistically acceptable measurements.

The nine cards are titled:

1. Testing for Optimum Scales and Intervals (137 steps)
2. Tabulating (223 steps)
3. Listing of Frequencies and Percentages (155 steps)
4. Histogram Bars and Polygon Points (193 steps)
5. Percentage and Frequency Ogives (206 steps)
6. Fitting Expected Frequencies Against Observed Frequencies (218 steps)
7. Testing for Skewness and Kurtosis (223 steps)
8. Goodness of Fit Test (1) (198 steps)
9. Goodness of Fit Test (2) (144 steps)

Each of the 23 separate functions performed is optional and independent. Only one input of data is needed, and all functions operate with a single keystroke.

Continued

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

17 Apr 79

This program, **Analysis of Grouped Frequency Tabulations #67000-99986,** is $28.50*. (Not available in Europe. See **Ordering Programs.)**

(An elegant, precise, and spectacular piece of programming, Mr. Eades. I offer my heartiest congratulations, plus a tip to other readers: Mr. Eades has written a great many other programs. Check your Catalog! Ed.)

Watzin The Registers?

As you all know by now, there are many ways to attack and conquer programming problems. Some people care more about execution speed than finesse; with others, there is a passion to get to the very fewest steps. Here's an example of how one person enhanced a routine we printed in the last issue. The clever title (above) was his, not ours!

Dear Editor:

I am writing in reference to **David D. Loeffler's** register-checking routine No. 1, published in Vol. 2 No. 4 KEY NOTES (center column, page 3). His very useful routine searches the 26 registers, 0 through I, and reports all registers with non-zero contents.

I enclose an enhanced routine for the same purpose. Like his, this routine leaves the 26 registers and last-X unaltered but, in addition, this routine saves and restores the X- and Y-registers. Therefore it is useful during procedures when X and Y contain intermediate results.

This routine is also much faster. When all registers contain zero, its running time on my calculator is 9 seconds, versus 18 seconds for Mr. Loeffler's. Although non-zero registers lengthen both running times, the 9-second difference remains.

The increased speed is primarily due to the method used for looping. I have found this looping method to be very useful, so it is worth noting the mechanism. Through repetition of GSB steps in the main routine and at the first subroutine level, the second subroutine level is executed 25 times, as desired. All backward branching is accomplished by RTN steps, which take less time than GTO steps to a previous label, because the latter require circular memory searches. Although this routine contains 9 GSB steps dedicated to loop control, its total length is 33 steps, versus 34 steps for Mr. Loeffler's routine. Apparently these 9 "extra" steps paid for themselves by simplifying the stack manipulations which were otherwise involved in testing for the last-loop.

Sincerely,
Robert W. Harris, Crofton, Maryland.

001	*LBLE	012	R↓	023	ISZI
002	0	013	RTN	024	R↓
003	X≠I	014	*LBL9	025	RTN
004	GSB9	015	GSB9	026	*LBL8
005	GSB9	016	GSB9	027	X≠I
006	GSB9	017	GSB9	028	DSP0
007	GSB9	018	GSB9	029	PSE
008	GSB5	019	*LBL9	030	X≠I
009	X≠0?	020	RCLi	031	DSP9
010	GSB8	021	X≠0?	032	PRTX
011	STOI	022	GSB8	033	RTN

Editorial

In the August 1978 issue (Vol. 2 No. 3), in the lower left corner of page 9, I used a shortcut in the routine submitted by **Arnold M. Miller,** and that slight slip of the hand caused quite a few letters about my "special" HP-97 printer. The routine called for the entry of some unspecified numbers, which are usually represented by the lowercase letter n. Since that isn't a printable character, I used N!, eliminated the ! symbol on the printer tape, and *voila,* there was my "n," although admittedly in uppercase. So, the answer to all the many letters is: No, I don't have a "special" HP-97. Mine's just like yours.

Received a letter last month from **James Neely** of Carmel, California. He writes a lot of astrology programs and receives a lot of mail about them. However, not everyone includes an SASE when writing to him, and he, like many other Library program authors, asked us to remind everyone to include an SASE when asking for information. And in case you don't know what SASE means, it stands for "Self-Addressed and Stamped Envelope."

Mr. Neely also mentioned problems arising from a person not checking the **Set Status** block on the Program Submittal form. Perhaps it would help to add a note on the *User Instructions* form on *all* Library programs, so the user could not possibly miss any **Set Status** instructions.

In the *Math Pac Handbook* for the HP-67/97, an error has been found by **Murray L. Lesser** of Yorktown Heights, New York. If you own this pac, turn to page 18-03 and the paragraph immediately above the Remarks heading. The last line of the paragraph states "and a safer specification would be DSP6." Change that to read "DSP 3." Example 4 on page 18-05, however, is still correct, and there is no change to the prerecorded card.

If you purchased the booklet, *Airplane Stability Calculations With a Card Programmable Calculator,* that was mentioned in Vol. 2 No. 3, drop me a note and I will send to you some corrections that you probably did not get with your booklet.

Henry Horn, Editor
Hewlett-Packard Co.
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330

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!

The "Other" Software?

Do you know about the engineer who owned an HP-67, bought a *Games* application pac and two *Users' Library Solutions* books—"Games" and "Games of Chance"—and went to Las Vegas and subsequently retired a millionaire at age 38? We don't either ... the person is a fictitious character. But, if the person did exist, he/she is probably lying as low as D.B. Cooper of hijacking fame. And most of us in HP—with out puritan gyroscopes working in our consciences—would rather know about some real HP-67/97 owner who became a millionaire, thanks, instead, to our *Real Estate Investments* "solutions" book ... or who became equally rich in the Stock Exchange with our books on *Portfolio Management/Bonds & Notes* and *Options/Technical Stock Analysis* ... or who got money **back** on their Income Tax with our superbly popular *Taxes* solutions book!

And there are a few more of these books that can help the owner of an HP-67 or HP-97 to realize substantial savings or to manage his or her investments: *Small Business, Marketing/Sales, Energy Conservation, Home Management,* and *Home Construction Estimating.*

We cannot guarantee you riches, but we surely can promise you a lot of fun, not only with our "games" pacs and books, but also with those on *Avigation, Navigation, Aircraft Operation,* or *Darkroom Photography*—if you are an aficionado of any of the above.

And if you care to know on what day of the week you were born, or in what month and year of the Mayan or Chinese calendars it was, you can find it in the *Calendars* solutions book, along with such other things as "Biorhythms" and "Moon Phases."

Hewlett-Packard software traditionally has been considered one of the main assets of the brand. There are plenty of good-quality applications and solutions books not only for the HP-67/97 but also for the HP-33/38, the HP-19C/29C, and the nonprogrammable HP-92. *Users's Library Solutions* books stand out as a novelty. These were another HP innovation in the recent history of personal calculator software, and other calculator brands promptly followed the idea.

The charm of the *Users' Library Solutions* books is that much in them is in the original format, as contributed by the nearly 900 user-contributors. Their originality is in the fact that most of them cover subjects that cut across most "vertical" fields and specialties. To be sure, there are some rather specialized books such as *Beams & Columns, COGO/Surveying, Thermal & Transport Sciences,* and so on. But the most successful of all, in terms of orders, have been those that we mentioned in the first paragraphs of this article. That is why we exemplified an "engineer" (in no matter what type of engineering) making it on a (shame on us) *Games Pac* and a *Games of Chance* solutions book.

Continued

Most often, a software pac or solutions book is purchased at the same time the programmable calculator is purchased—or shortly afterwards. This usually means that the software is intended to expand the performance of the calculator as a professional tool, for faster and more efficient routine calculations *of the type that a user typically encounters at work.* It is when the user takes the calculator home that this gains a new dimension as either a *fun-producing* or a *money-saving,* truly *personal* "appliance"—or both. This is where "horizontal," general-interest, and hobby-type "solutions" books come in, and it is what explains their tremendous success over the more esoteric books.

Present owners of HP-67/97's keep coming back to us and their nearest dealers for more of this fun. That is why we are running a special promotion of *Users' Library Solutions* books during the months of February and March, the details of which you will find on the order blank in this issue.

Anti-Subroutines *And* Ersatz Memory?!

If you have been following the saga of our "Ersatz Continuous Memory" routine, started by **Pierre Flament** (Brussels, Belgium) in Vol. 2 No. 2, you'll remember it had a problem, then some partial corrections. Now, from Surrey, England, comes this letter from **James P.H. Hirst.** Has he finally solved the problem?

Dear Editor,

Readers **Joseph V. Saverino** and **Murray L. Lesser** (KEY NOTES Vol. 2 No. 3), not to mention **Pierre Flament,** can stop worrying about Pierre's "Ersatz Memory" being sunk by a subroutine.

Here is a simple anti-subroutine defence.

(a) Execute Pierre's steps 1, 2, 3 to dump the program.
(b) Record the program step number (psn).
(c) Key h RTN.
(d) Switch to RUN.
(e) Press SST.
(f) Switch to W/PRGM.
(g) If the psn now displayed is not equal to (2 + last recorded psn) go back to step (b); otherwise continue with Pierre's sequence to dump the data, change batteries, and reload.

To resume the calculation proceed as follows:

(h) Use GTO.nnn to return to the first recorded psn and press R/S.
(j) When the program stops, use GTO.nnn to return to the next recorded psn and again press R/S.
(k) Repeat step (j) until you have used up all your recorded psn's. The calculation is now back on course.

Use of this procedure is really only justified when you have to stop in the middle of a long-running calculation, though addicts of computling (sic) might not agree. At any rate, you don't want to take risks with a long-running calculation, so it would be as well to practise on short ones.

A suitable program to play with is "Calculus and Roots of f(x)," SD-11A, in the *Standard Pac.*

Execute the keystrokes of example 2 given on page 11-07 of the handbook and press R/S during one of the pauses. Of course, this will stop the program at step 107, which is not in a subroutine. You can then practise Pierre Flament's normal procedure.

To practise stopping in a subroutine,you can insert a pause after label 2. Let us assume you have successfully completed the solution of example 2 (with or without interruption).
Proceed: GTO 2; switch W/PRGM; h PAUSE; switch to RUN; DSP 9 f E. This resets the display and turns off the main program pause option (no longer needed). If you now key .21 E to start the calculation there will be seven pauses, during any of which you can press R/S. The first two of these occur when subroutine 2 is "nested" within subroutine B. So try pressing R/S during the second pause.

.21 E → 0.2099895" → 0.2100105

Now switch to W/PRGM and feed in sides 1 and 2 of your DUMP PRGM card. Record the psn (118). Key h RTN and switch ro RUN. Key SST and switch to W/PRGM. Record the psn (042). Key h RTN and switch to RUN. Key SST and switch to W/PRGM. Record the psn (086). Key h RTN and switch to RUN. Key SST and switch to W/PRGM. The psn (088) is equal to (2−86) so switch back to RUN and feed in side 1 of the DUMP STK card. Press A. The display shows **Crd.** Now feed in side 1 of the DUMP REG card. After a flashing "2" the display shows **Crd** again. Feed in side 2 of the DUMP STK card. Switch to OFF.

Switch to ON and feed in sides 1 and 2 of the DUMP STK card. Press B. Feed in the DUMP PRGM card.

Now for the moment of truth!

GTO . 118 (your first recorded psn) R/S
→ -0.001838979
GTO . 042 (the second recorded psn) R/S
→ 0.045428571
GTO . 086 (3rd & last recorded psn) R/S
→ "0.250491161"
etc
0.244345974

Hooray!

Perhaps somebody will find a program that this procedure doesn't work on. I hope not.

(Nice work, Mr. Hirst! And you have a good sense of humor, too! But does the Royal Navy know about your anti-subroutine defence? Ed.)

Last Word (Routine?) On Factorials

And we hope this one is! The response to this subject has been startling, to say the least. It seems there are as many opinions as to how to find the factorials of large numbers as there are ways to do it and, together, they must equal a number larger than the total number of stones in the Great Wall of China. *(Right, Gary?)*

Anyway, here is one more good routine, from **James E. Coxon** of Christchurch, New Zealand, followed by a short note from **David E. Rushing** of Salt Lake City. And we thank all of you who wrote routines or letters about this subject.

In Vol. 2 No. 3 I noticed an article on page 9 under "25 Words or Less." The small routine generates factorials of numbers above 69. I noticed that the factorial of 521 took 8.5 minutes to run.

Here is a routine that is shorter and will find the factorial of a number up to and including 10^{98}, with an accuracy of at least seven decimal places and a running time of *less* than 5 seconds for any number!

001	*LBLA	019	2
002	STO0	020	x
003	1	021	1/X
004	e^x	022	1
005	÷	023	+
006	LOG	024	LOG
007	RCL0	025	+
008	x	026	STO1
009	Pi	027	9
010	2	028	8
011	x	029	X≷Y
012	RCL0	030	X>Y?
013	x	031	FRC
014	√X	032	10^x
015	LOG	033	R/S
016	+	034	RCL1
017	RCL0	035	RTN
018	1		

To use the routine, key in the number for which the factorial is required, then press A. The display will show x! Then press R/S and the display will show log x! In other words, try 521! Press A and see 9.18, then R/S and see 1190.96. This output is actually

$$\frac{x!}{10^{\text{INT LOG x}}}$$

If you want to include values less than or equal to 69, insert the following steps after the LBLA in the above routine. 69, x≷y, x>y?, GTO1, N!, RTN, LBL 1.

And, now, last but not least, Mr. Rushing's contribution, for those who want to battle it out to the infinite end!

Re: N!, page 9, KEY NOTES, Vol. 2 No. 4: Forsyth's formula is

$$N! \cong \sqrt{2\pi}\left\{\frac{\sqrt{N^2+N+1/6}}{e}\right\}^{N+\frac{1}{2}}$$

and is a better approximation to N! than is Stirling's formula.

(Quot homines, tot sententiae! Ed.)

HP-67/97 Looped Program Merge

Here is a contribution from an HP fan in London, England. We think you'll like it.

Dear Sirs,

One of the most useful attributes of the HP-67/97 is its ability to accept programs longer than 224 steps by means of the merge function. Using a simple "MERGE/PAUSE" instruction sequence is risky, however, because if you miss the 1-second pause you have lost your chance to merge the next program card. Setting up the

loop is the answer, but if you want program execution to continue automatically when the new card has been read in the usual method—using flags—one of the merge instructions's most valuable resources is wasted, the ability to transfer flag status from one part of a program to the next.

Here is a programming technique that allows you to write very long programs in which all program card-reading operations are carried out automatically via looped program merges, without interruption of program execution. No use is made of flags or other tests; instead, a simple and easily remembered group of instructions is repeated at the start of each card, and the looping is controlled by a subroutine that forms part of this group.

The following example illustrates the method for a four-card program:

CARD 1	CARD 2	CARD 3	CARD 4
*LBL8	RTN	RTN	RTN
MRG	RTN	RTN	*LBL7
PSE	RTN	*LBL7	GSB8
RTN	*LBL7	GSB8	GTO7
RTN	GSB8	GTO7	*LBLA
RTN	GTO7	*LBLA	4
RTN	*LBLA	3	PRTX
*LBL7	2	PRTX	RTN
GSB8	PRTX	GTO7	
GTO7	GTO7		
*LBLA			
1			
PRTX			
GTO7			

If this program is run on the HP-97, the numbers '1', '2', '3', and '4' are printed in succession, as each successive card is read. When a number has been printed, the display flashes to tell you that the calculator is ready to receive the next card.

There are three parts to the instructions that control the looped program merge. The first, which is recorded on Card 1 only, is the three-step group LBL8, MRG, PSE. This sequence goes into the first three steps of program memory when Card 1 is read, and remains there throughout the succeeding card-reads, all of which are merged from step four onwards.

The second part is a group consisting of a number of RTN instructions followed by the sequence LBL7, GSB8, GTO7. It is this part of the program that decides whether to loop or, after a new card has been read in, to continue program execution.

The final part is the main program, which in this case starts at the instruction LBLA. The main program can occupy all remaining program steps. If a further looped program merge is required, it is initiated by the instruction GTO7 in the main program. The only restrictions on the main program are that it must not contain LBL7 or LBL8, nor any references to them (apart from GTO7 referred to above).

The decision whether to loop or to continue execution depends on the program step to which the first RTN instruction encountered after LBL8 returns control. If no program card has been read during the loop, control is returned to the GTO7 instruction after GSB8. If a card has been read, control is returned to the same location in program memory, but now that location contains the first step of the new main program, so execution of the new program continues automatically.

It can be seen that, for the technique to work, all that is required is a number of dummy instructions between the RTN of subroutine 8 and LBL7. I use RTN steps as dummy instructions as an aid to programming, because the number of RTN instructions required at this point for card 'a' of an 'n'-card program is 'n−a+1'. The steps LBL7, GSB8, GTO7 on the final card are also, in effect, dummy instructions, but it helps to leave them in this form for ease of identification.

I hope your readers will find this technique helpful.

Yours faithfully,
A.G. Burns, London, England

Didactic Programming

We have run across a new publication called *Didactic Programming*. It is a journal of calculator-demonstrated math instruction for math majors and is currently free to high-level math instructors who request it on their school's letterhead. It does NOT cover elementary school arithmetic. Publication costs are being met by Educational Calculator Devices, Inc., the company that manufactures the EduCALC, a teacher's or lecturer's calculator/display device built on the order of a lectern.

If you live in the U.S. and would like to have a copy of the first issue, published in autumn 1978, send one dollar to cover postage and handling to:

Didactic Programming
Post Office Box 974
Laguna Beach, CA 92652

If you live outside of the U.S., send $2.25 to cover air mail and handling to the above address.

If you are a high-school or college-level person involved in math instruction with a calculator, we think you will find this new journal very worthwhile, regardless of the type of calculator you presently use.

"25 Words" (More or Less!)

The variety of mail we receive about this column is truly staggering. It is quite evident that a lot of you spend a great many hours figuring out better and trickier ways to solve problems on your calculators. And since you obviously like what you see here, we will continue to print as many routines as space allows.

The following 66-step routine for the HP-67/97 provides control of up to 825 flags. It should help in investigating random selection problems without replacement—for example, Bingo, Blackjack from a limited deck, attendance checks, etc. To use the program, key in the flag number between 0 and 824, followed by:

A—Displays 1 if flag is set, 0 if it is clear.
B—Sets the flag
C—Clears the flag

The routine starts assigning flags 0 thru 32 to R_0, 33 thru 65 to R_1, and so on up to R_E. If less than 825 flags are required, the unused registers are available for other routines or programs.

001	*LBLA	034	−
002	3	035	GTO1
003	3	036	*LBL0
004	÷	037	LSTX
005	STOI	038	÷
006	LSTX	039	*LBL1
007	X⇄Y	040	INT
008	FRC	041	1
009	×	042	0
010	.	043	÷
011	5	044	FRC
012	+	045	5
013	INT	046	×
014	2	047	FRC
015	X⇄Y	048	2
016	Y^x	049	×
017	ENT↑	050	RTN
018	ENT↑	051	*LBLB
019	RCL i	052	GSBA
020	X⇄Y	053	X≠0?
021	÷	054	RTN
022	LSTX	055	R↓
023	X⇄Y	056	R↓
024	INT	057	ST+ i
025	X⇄Y	058	RTN
026	×	059	*LBLC
027	RCL i	060	GSBA
028	X≤Y?	061	X=0?
029	X=Y?	062	RTN
030	GTO0	063	R↓
031	LSTX	064	R↓
032	÷	065	ST− i
033	1	066	RTN

How about a "warning" for a change? This is a note from **Jim Britton** of Houston, Texas.

I felt prompted to comment on the first point of **Tom Cadwallader's** piece in the "25 Words" column of Vol. 2 No. 4. He mentions the preservation of flag status from one card to the next on multi-card programs. The owner's handbooks for the HP-67/97 mentioned this problem only in passing.

The MERGE function not only saves program steps from a previous card but also preserves the flag, display, and trig mode status of the previous card. Beginning every successive card with LBL 9, MERGE, PAUSE and then calling for LBL 9 at the end of that card will solve the problem.

Continued

For a change of pace let's go to County Down in North Ireland, for a neat subroutine from **William P. Brown.**

If one proposes to carry out serious numerical calculations, one should be aware of the real dangers that arise through working with a machine of limited precision. Using 10-figure floating-point precision, one may well fail to realize how near danger lies. The subroutine enclosed allows one to work in 'n'-figure floating-point precision, where 'n' is the 'display' number (N=0, ... 9).

In the form presented, the subroutine uses no storage registers and saves the entry in the Y-register of the stack. It may be shortened by using a storage register and, additionally, the subroutine may then save entries in the Y- and Z-registers.

```
001   *LBLA    013   *LBL0
002    X=0?    014     INT
003     RTN    015    10^x
004    X<0?    016      ÷
005     SF2    017   LSTX
006     ABS    018    X≠Y
007    ENT↑    019    RND
008     LOG    020      x
009    X<0?    021     F2?
010    GTO0    022    CHS
011       1    023    RTN
012       +    024    R/S
```

To express π in four-figure floating-point precision, key in π, DSP4, GSBA. The subroutine handles positive and negative numbers, and zero. It is designed to run in FIX mode. The reader may care to amend it for SCI mode.

EXAMPLES

```
          Pi
        DSP4
3.1416   ***
        GSBA
3.1420   ***
         SCI
3.1420+00 ***
```

EXAMPLES

```
          FIX
          Pi
           x
100.0000  ***
314.1593  ***
        GSBA
314.2000  ***
```

```
          Pi
          ÷
1000.0000
        GSBA
   0.0031 ***
         SCI
3.1420-03 ***
```

This next routine appears to be similar to Mr. Brown's but it isn't. However, if *one* interests you, surely both will. This one is the work of **Ernest R. Reuther** of Miami, Florida.

In a program where the fixed decimal point (FDP) display number (DSP#) is frequently or variably changed by the user, it may be required to *return* to whatever the FDP DSP# was at some particular point in the program. It is not obvious how to accomplish this without user input, so I worked out this interesting routine. LBL D will "store" the DSP# currently in use. LBL E will "recall" that FDP DSP# back to the program. Lines 2, 20, 26, and 27 could be deleted at the expense of not saving the previous x and I register values.

```
001   *LBLD    015    ST+9
002    STO8    016    LSTX
003     CLX    017     FRC
004    STO9    018    X≠0?
005       1    019    GTO1
006    ENT↑    020    RCL8
007       9    021     RTN
008       ÷    022   *LBLE
009   *LBL1    023    RCL9
010     RND    024    X≠I
011       1    025    DSPi
012       0    026    X≠I
013       x    027     R↓
014     INT    028     RTN
```

Example:

```
          Pi
3.14      ***
        DSP9
3.141592654 ***
        DSP4
3.1416    ***
        GSBD
3.1416    ***
        DSP7
3.1415927 ***
        GSBE
3.1416    ***
```

Now, here's a short routine that works well. It was donated some time ago by **Peter Baldwin** of Vernon, Connecticut, and finally made it to this column.

Label A enables one to store into all registers beginning with 0. Label B enables one to store into all registers beginning with 0 and every other one after that. Label C enables one to recall the contents of a specified register.

```
001   *LBLA    013    ISZI
002    STOi    014    GTOA
003    RCLi    015     RTN
004    ISZI    016   *LBLC
005    RCLI    017     ENG
006     R/S    018    X≠I
007     RTN    019    RCLi
008   *LBLB    020    X=0?
009    STOi    021    RCLi
010    RCLi    022    DSP4
011       0    023     R/S
012       0    024     RTN
```

Labels A and B may be used in conjunction with each other to produce odd-even register storage.

To try the above routine, key in a few random numbers, pressing [A] each time to load the register. Then try some with [B]. Now, when you press, say, key [0] and key [C], you will see the contents of register 0.

Let's move now to Porto, Portugal, and a contribution from **Dr. Ing. Henrique E. Adler.**

SAVE STEPS: Problem: Repeat subroutine 2 n=10 times and jump then to LBL 8.

SOLUTION (A)

```
001    GSB2
002    GSB2
003    GSB2
004    GSB2
005    GSB2
006    GSB2
007    GSB2
008    GSB2
009    GSB2
010    GSB2
011    GTO8
012   *LBL2
013     R/S
```

SOLUTION (B)

```
001    GSB2
002    GSB2
003    GSB2
004    GSB2
005    GSB2
006    GSB8
007   *LBL2
008    GSB2
009   *LBL2
```

Solution (B) saves three steps. The trick can be used for any n greater than 4 and not prime.

Let's try a "power play," now, and see how the calculator can handle *that!* It's from **Charles J. Robinove,** who works for the U.S. Geological Survey in Reston, Virginia.

Here is a short HP-67 routine that makes complete use of the capabilities of the indirect register. Sometimes it is necessary to use several powers of a number in a single calculation. This routine calculates n^1 through n^9 and stores the values in registers 1 through 9 for subsequent calculations. The tricky part is that is uses the indirect register both to calculate n to the power in the indirect register and also stores the resulting value in the register addressed by the number in the indirect register. The power is the same as the register number, making it easy to remember.

The routine is:

```
001   *LBLA    008    RCLI
002    STO1    009     Y^x
003       1    010    STOi
004       0    011    DSZI
005    STOI    012    GTO1
006   *LBL1    013     RTN
007    RCL1    014     R/S
```

The program can be checked by recalling each register, 1 to 9, or keying [h] (or [f]) [REG]. Powers of n up to 24 can be stored by replacing the 10 in the program by 24, but this, of course, leaves no registers available for later calculations. Hope this will be of help.

(Watch steps 007 and 008! They are RCL 1 and RCL I. Ed.)

Now, back overseas to Hamburg, Germany, for an input from **Dipl.-Ing. Herbert Gudehus.**

As all HP-67/97 owners know, side 2 of the magnetic cards is assigned for program steps 113 to 224. It is less known, and not mentioned in the handbook, that smaller programs can be recorded otherwise. Write into program memory one or more programs with, together, not more than 112 steps, then record them onto side 1 of

Continued on page 9

a card. After that, clear the program memory and key into the calculator some other program(s), beginning with step 001 and the last step 112, or less. Insert side 2 of the card and these steps (program) are recorded on side 2, but with step numbers 001 to 112, or less.

In this way, both sides of the card can be used for storing steps 001 to 112 of different programs. They can be read in alternately by inserting either side 1 or 2 in RUN mode. This use of the card offers some advantages:

1. Subsequent program changes on one do not change the step numbers on the other side.
2. By using the same labels for both sides, you dispose of 40 labels on one card, and you can more frequently use the labels A to E instead of a to e, which require two keystrokes each.
3. For the programs on both sides, there are left at least 112 free steps for the addition of supplementary routines (by hand or with MERGE from other cards) or data from a serial data card as proposed by Mr. Botkin (KEY NOTES Vol. 1 No. 3).

(This technique was used in program SD-12 in the HP-67/97 Standard Pac. *Ed.)*

And, back again, is **Ernest R. Reuther** of Miami, Florida, with something for navigators.

Perhaps some other navigation-buffs might find some use for my routines for converting degrees, minutes, seconds to degrees, minutes, and tenths of minutes, and back.

D.MMSS TO D.MMM/10		D.MMM/10 TO D.MMSS	
001	*LBL D	012	*LBL E
002	DSP3	013	DSP4
003	HMS→	014	INT
004	INT	015	LSTX
005	LSTX	016	FRC
006	FRC	017	.
007	.	018	6
008	6	019	÷
009	×	020	+
010	+	021	→HMS
011	RTN	022	RTN

Back to Europe again, and this time to Stabroek, Belgium, for a short routine from **John van Thielen.**

In most of the programs about statistics, you must input a set of data and, if you make a mistake with the input, it is useful that you can correct it. I think this nine-step subroutine can help some people.

001	*LBL a	006	*LBL A
002	R↓	007	Σ+
003	LSTX	008	R/S
004	Σ-	009	GTO A
005	R/S	010	R/S

To input the data-pairs, key in y, ENTER, x, and then press [A] for the first pair and [R/S] for the other. Now, if you make an error during input, just press [f] [a] and try again with the right pair.

Now, let's try a new slant on the interpolation routine printed in the last issue. This routine is from **John P. Gould,** Professor of Economics at the University of Chicago's Graduate School of Business.

I noticed **Thomas Hirata's** 33-line interpolation routine in Vol. 2 No. 4, page 5, and your 16-line alternative using four registers. The following routine uses only 15 lines and no registers. In addition, it does not require reentering the routine or entering data in registers each time interpolation bounds are needed. The notation is indicated in the table:

x	y
x_1	y_1
x_2	y_2

with $x_1 \leq x \leq x_2$

The program:		USER ACTION
001	*LBL A	(Start with y_2, ENTER, y_1)
002	-	
003	LSTX	
004	X≷Y	
005	R/S	(x_2, ENTER, x_1, R/S)
006	-	
007	LSTX	
008	R↓	
009	÷	
010	R/S	(x, R/S)
011	R↑	
012	-	
013	×	
014	+	
015	RTN	y value displayed

Of course, if the user wished to repeatedly interpolate between the *same* values, your 16-step routine would save time in data reentry.

Next, a small routine from a 10th-grade student, **John Diamant,** of Cincinnati, Ohio.

This short subroutine, when placed at the beginning of a program card, can connect that card to another one, so that a long program can run continuously through two or more cards. After feeding in the first card and beginning execution of the program, place the second card in the card reader. It will automatically read the card at the proper time and continue execution without manual instructions. (If the card is not in the slot at the proper time, the calculator will stop. To continue; insert second card, press [R/S].)

001	*LBL 1
002	MRG
003	PSE
004	R/S

1. Place this subroutine in the first four steps of program memory (to give the second card the maximum amount of merge space).
2. Insert "GTO 1" after the last program step on the first card (at the point where the second card is to take over execution).
3. Remember: (a) The second card may be filled only to step 221. (b) For each new case, the first card must be fed in again.

Back to Europe again; this time for some "ersatz" flags *that work!* Here's an interesting tip from **Ralf Kern** of Karlsruhe, West Germany.

If you need more than the usual four built-in flags, use the formatting (FIX, SCI) or trigonometric (DEG, RAD, GRD) options (if they are not otherwise fixed in the program). Test these "flags" by the following two routines. First, program 1, SIN^{-1}, LOG, INT, 1, −, which will display −1 for RAD, 0 for DEG, and 1 for GRD (a three-valued "flag"!), whereas programming 11, ENTER, DSP0, RND, −, displays 0 for FIX, and 1 for SCI and ENG.

Finally, does anyone know an algorithm that distinguishes between SCI and ENG?

Next is an idea based on an entry in this column in the last issue. It's from **A.M. Platt** of Richland, Washington, and it's about *Nested Loops* (in the left column of page 6).

Reference is made to the comments on nested loops in Vol. 2 No. 4. If the control registers from innermost loop to outer loops are designated as I, 0, 1, 2, ..., substantial programming steps will be saved. This is particulary true if the innermost loop does not involve a premature exit.

Now, from **John Craig** of Anaconda, Montana, here's a little off-shoot from the HP 9825A Desktop Computer.

Here is a short sequence of steps I use on my HP-67 to mimic the "mod" function on my HP 9825A at work. It might prove useful enough to use in your "25 Words" column.

Y Mod X: Assume x and y in proper registers; then: LBLA, ÷, LSTx, X≷Y, FRAC, ×, RTN. Notice that the stack drops correctly and that the Z- and T-register data is not lost. The LSTx register, however, won't have the "last" x.

(However, don't forget that this works only if both arguments have the same sign. Ed.)

And every once in a while, someone gets a change to pull our leg. Here is one such case, contributed by **Guy Dresser** of Lawrence, Kansas.

The following routine is trivial, but doesn't it clear a small group of storage registers in several fewer steps than the one you printed in the Vol. 2 No. 3 "25 Words or Less" column?

001	*LBL E	007	STO 7
002	P≷S	008	STO 8
003	0	009	STO 9
004	STO 4	010	P≷S
005	STO 5	011	RTN
006	STO 6	012	R/S

It has the following additional advantages over the original 21-step routine:

a. It doesn't affect the I-register.
b. It doesn't force you to clear a group of registers in descending order—any register or registers may be cleared.
c. It takes a couple of seconds less to run.

If you would rather work from the stack than the memory, simply enter 0 and store it in whatever registers you want to clear. Then you don't need to use any memory at all.

For a change of pace, how about some neat *ideas?* Here are two from **Roland K. Kolter** of Dearborn, Michigan.

Your program cards can be labeled magnetically with any decimal number (if you haven't used all 224 steps). No extra key is necessary. Simply begin each program with its number, say 5.23, by programming 5 • 2 3 RTN. After loading from the magnetic card, just press R/S and the number will appear in the display. Don't forget to write the same number on the card!

You can extend the PAUSE function indefinitely if you hold down any key. You can then leisurely choose the next (correct) digit, push it down, and let go of the other key. However, you must let it go and press it once again to get it into the display.

Let's now head north to New Hope, Minnesota, for a contribution from **Neal Neuburger.**

I came across **John S. Prigge, Jr.'s** routine for the greatest common divisor (gcd) of two positive integers (Vol. 2 No. 4, page 4). This routine seemed vaguely familiar—it turned out to be the Euclidean Algorithm—and I was amazed at its simplicity. However, there is an addition that can be made that will find the least common multiple (lcm). The complete routine, incorporating Mr. Prigge's routine as steps 007 to 020, is as follows:

```
001  *LBLA      014    R↑
002    X≷Y      015    ×
003   STO1      016    -
004    X≷Y      017   X≠0?
005   ST×1      018   GTO1
006  *LBL1      019    +
007   ENT↑      020   RTN
008   ENT↑      021  *LBLB
009    CLX      022   RCL1
010      +      023    X≷Y
011     R↓      024      ÷
012      ÷      025   RTN
013    INT      026   R/S
```

The sample given below shows not only the routine but how it can be used to obtain the lcm for a triple (or, with repetition, an n-tuple) of positive integers.

EXAMPLE: 18, 24, 52

Pressing A:	*Pressing B:*
gcd (18,24)=6	lcm=72
gcd (72,52)=4	lcm=936

Try it. Key 18, ENTER, 24, A and you'll see 6 (gcd) displayed. Press B and see 72 (lcm) displayed. Press ENTER, A and see 4 (gcd) displayed. Then press B and see 936 (lcm).

Now, here is a "Root Finder" from the pen of **Dr. Helmut Weiss** of Newport Beach, California.

When some major program involves the iterative solution of equations f(x)=0, storage space may become quite crowded. For such situations, I wrote an HP-97 routine that gets by with only three storage registers.

Since you do not have a column "3 Registers or Less," it is fortunate that the program also qualifies for "25 Words or Less."

The program presumes display in the FIX mode and computes the root to the accuracy determined by the number of DSP digits. Speed compares favorably with that of the root finder in SD-11B.

The user-defined function f(x) is presumed to be recorded as a separate subroutine (labeled e), which begins by recalling x from R_1 and ends with RTN.

```
001  *LBLA      017   STO3
002   STO1      018     -
003      1      019   ST÷2
004      %      020   RCL3
005   STO2      021   ST×2
006   GSBe      022   GTOc
007   STO3      023  *LBL0
008  *LBLc      024   RCL1
009   RCL2      025    RTN
010   ST+1      026  *LBLe
011    RND      027   RCL1
012   X=0?      028      .
013   GTO0      029      .
014   GSBe      030      .
015   RCL3      031    RTN
016    X≷Y      032    R/S
```

To use the program, key the estimated root and press A.

For another change of pace, let's try a trick submitted by **Fabio Lusiani** of Carrara San Giorgio, Italy.

Here is a way to save steps in a program. Instead of recalling two registers and then carrying out a division, be shrewd and put the dividend in R_{S4} and the divisor in R_{S9}. You can then obtain the same result (division) by pressing ÷! I frequently used this method when I owned the HP-25.

(Very true, Mr. Lusiani. However, make sure you don't need what's in the Y-register! For those unfamiliar with the "mean" function, study again page 100 in the HP-97 handbook or page 111 in the HP-67 handbook. Ed.)

"Talented Tabulator"

The Wednesday, November 15, 1978, edition of the *Livingston County Press* (Howell, Michigan) carried the above title below a photograph of **Doug K. Parrish** and his faithful HP-67. There was also a long article about the HP-67, its capabilities and background, and Mr. Parrish's use of it for various and sundry things. Most notably, the article stressed the fact that the reporter, **Debbie Pore,** unwittingly challenged Mr. Parrish's HP-67 to a game and here's what ensued:

"Losing a game of Tic-Tac-Toe to a calculator is not my idea of a good time, but that is exactly what it turned out to be. Little did I know when I gave my all against Mr. Parrish's prized possession, an HP-67, that his little 10-ounce calculator was programmed to never lose. Not only is his calculator quite stubborn about winning, it also has a mind of its own."

No doubt many more of you have had similar experiences, but this is the first time we've seen such an extensive write-up in a newspaper.

Doug Parrish is a German and English instructor at Howell High School. And none of his students are ever in the dark about where they stand academically, because Mr. Parrish has programmed his calculator so that they can find out their grade point instantly.

Thanks for the newspaper clipping, Mr. Parrish.

The IDDI Wizard of ID!

And *what,* you'll probably ask, is *that?* Well, we have to admit it's a "catchy" title, but it *does* describe the subject of the following letter.

Dear Henry:

Your appreciation of my indirect storage and recall routines in the May '78 HP KEY NOTES is largely responsible for the following contribution. I call it the "IDDI Wizard of ID" for reasons that will become clear.

The following routine resulted from experimentation with ISZ and DSZ instructions on the HP-67. This one does a lot for four steps.

"IDDI"	**I-Register**	
1. ISZ	**Before**	**After**
2. DSZ	0	1
3. DSZ	1	0
4. ISZ		

If the I-register is zero, this routine will change it to a 1 and vice-versa. This suggests a complementing function for the I-register used as a flag. But a flag is useless unless it can be tested. Fortunately there is an easy way to test this flag.

"ID"	**I-Register Test**
1. ISZ	If I=1, do step 3
2. DSZ	If I=0, skip 3
3.	(I-register unchanged)

An added bonus is that this software flag can also be test-cleared with this test: DSZ, ISZ. Note that both tests maintain the "do if true" convention. After being test-cleared the flag may be set with ISZ of course.

What is really nice about all of this is that the stack is not bothered by these operations. Further investigation of this type of routine revealed a unique modulo 3 counter (not everything is binary).

"Mod 3"	**I-Register**	
1. DSZ	**Before**	**After**
2. DSZ	0	1
3. ISZ	1	2
4. ISZ	2	0
5. DSZ		
6. ISZ		

Each time this routine is executed, the I-register counts. It counts from 0 to 1, 1 to 2, and 2 back to 0 to repeat the sequence. Note that "IDDI" may also be considered a modulo 2 counter.

While the ISZ and DSZ instructions were not

intended for such routines, it is interesting that they provide useful functions. I use the "IDDI" routine to alternately call one of two subroutines (using LBL 0 and LBL 1) with the GSB(i) instruction, or to keep track of odd and even. I have no immediate use for the "mod 3" routine, but I am sure somebody has. Of course, all of these routines can be implemented by other means, but not as elegantly as with the ISZ and DSZ instructions.

Here is another way to alter program flow, depending on the contents of the I-register:

	I-Register Test
1. DSZ	If I=0, then GSB A
2. GSB A	If 0<I<1, then do nothing
3. ISZ	If 1≤ <2, then GSB B
4. GSB B	If I≥2, then GSB A and GSB B
	(I-register unchanged)

Steps 2 and 4 are subroutine calls by way of example only. Also, although this is an I-register test, it can easily test the X-register by using x⇄I.

A couple of notes are called for here:

(1) It so happens that the instruction following the last ISZ in both "IDDI" and "mode 3" will never be skipped under the conditions defined in this letter. This means no precautions are necessary to use them as subroutines.

(2) By replacing ISZ and DSZ with ISZ(i) and DSZ(i) all of the routines described here can be made to operate on any primary or secondary register.

Incidentally, the sole purpose of the name "The IDDI Wizard of ID" is to remember the complement and test routines with acronyms.

Sincerely,
Emerson J. Perkins, Huntington Beach, Calf.

Let's Hear It Again For RPN

Next time someone tries to make you justify the use of RPN logic in your HP calculator, read to them this letter from **John Robert Kennedy II,** a math instructor at Santa Monica College in California.

Few readers of KEY NOTES need to be convinced that RPN logic provides the simplest, most efficient, and most consistent method to perform mathematical computations. I recently discovered another argument to support promoting this new and more versatile approach to problem solving. Thinking in terms of RPN can help clarify certain mathematical concepts.

While recently teaching a math class, I was trying to get across the idea of a "basic operation." Very simply, every mathematical expression, no matter how complicated, can be reduced to a single *last* operation. The last operation you perform when you numerically evaluate a mathematical expression is the *basic* operation for the expression. In determining a basic operation we must apply all the standard rules of operator hierarchy by first performing all powers, then multiplications and divisons, and lastly, additions and subtractions, all in left-to-right order within groups of parentheses.

Thus, $(a+b)(c+d)$ is a basic product, $\frac{a+bc}{d-e}$ is a basic quotient, $\frac{xy}{z}+w^2$ is a basic sum, and $\sqrt{x^2y^3-z}$ is a basic square root.

RPN helps clarify the basic operation concept which in turn helps clarify cancellation in fractions. One of the most common mistakes made by students in arithmetic and algebra is improper cancellation in simplifying fractions.

$$\frac{8+7}{21}=\frac{8+\cancel{7}1}{\cancel{21}3}=\frac{8+1}{3}=\frac{9}{3}=3$$

$$\frac{\cancel{w}}{(xy+z)\cancel{-w}}=\frac{0}{(xy+z)}=3$$

Students are told it is wrong to cancel an addition or subtraction when simplifying a fraction. Beginning students then question why the following example of cancellation is correct when the previous two examples are incorrect.

$$\frac{(xy+z)\cancel{(a+b)}}{\cancel{(a+b)}(pq+r)}=\frac{xy+z}{pq+r}$$

A basic sum and a basic difference occur in the numerator and denominator in the first two cancellation examples. In the last example, both numerator and denominator consist of a basic product. Legal cancellation occurs only when both numerator and denominator are basic products. Of course, the real operation involved is division, so the canceled terms should be replaced by the number 1.

The real case for RPN came up in my first semester calculus class. We had just finished a chapter containing rules for differentiating functions. The standard rules consist of the product rule, quotient rule, power rule, and chain rule. None of these rules by themselves are difficult to understand, but when they occur in combination, especially with the chain rule in the composition of several functions, then there is room for misunderstanding.

The problem to differentiate $f(x) = \sqrt{\frac{(2x^3+5)(x^2-7)}{(x+3)}}$

came up in the chapter review section. I was asked if the correct solution was to begin by using the power rule or quotient rule. I explained that a correct solution could be obtained by beginning with either rule, but the expression would have to be rewritten in the form

$$f(x)=\frac{\sqrt{(2x^3+5)(x^2-7)}}{\sqrt{(x+3)}}$$

if one were to begin differentiating by using the quotient rule. I further explained that in either case, both rules would have to be involved at some point no matter which rule was applied initially.

Dead silence followed, so I proceeded to explain that as the function was written originally it was a basic power, since a square root is a ½ power. But rewriting the expression as a quotient of two square roots turns the function into a basic quotient.

I said that the last operation you perform when you numerically evaluate an expression determines the basic operation. "Well, you know, if you key in a number for x in your calculator and compute f(x) the last operation..." I stopped in mid-sentence. What was I thinking? "Well that is, if you have an RPN logic calculator, then the last button you push is the *basic* operation. If you have an algebraic calculator the last operation button you press isn't necessarily the basic operation." There were a few frowns in the class as I continued. "The last computation you would do on a slide rule is the basic operation. So maybe some of you should trade in your algebraic calculators for slide rules!"

The few students in the class who had RPN machines immediately understood the point I was trying to get across. The rest of the class also caught on. After all, beginning calculus students don't need slide rules to understand the concept of a basic operation. But the point is that "thinking in RPN" returns real rewards, not only in making numerical calculations simpler, but also in understanding other mathematical concepts. Viva RPN!

"How To" Book Finally Here

On the back cover of Vol. 2 No. 3 we told you about a new book, *How to Program Your Programmable Calculator,* by Stephen L. Snover and Mark A. Spikell. It was orignially scheduled for a November or December production date but was held up by production problems and also to add information about the new HP-33E and HP-38E.

The book has over 160 carefully sequenced examples, exercises, and problems that are solvable on any programmable calculator. It also gives you information on how to design programs to solve problems.

The book will be available in March 1979 and will cost $7.95* for the paperback edition and $16.95* for the hardcover edition. In the Continental U.S.A., Alaska, and Hawaii, send a check or money order to

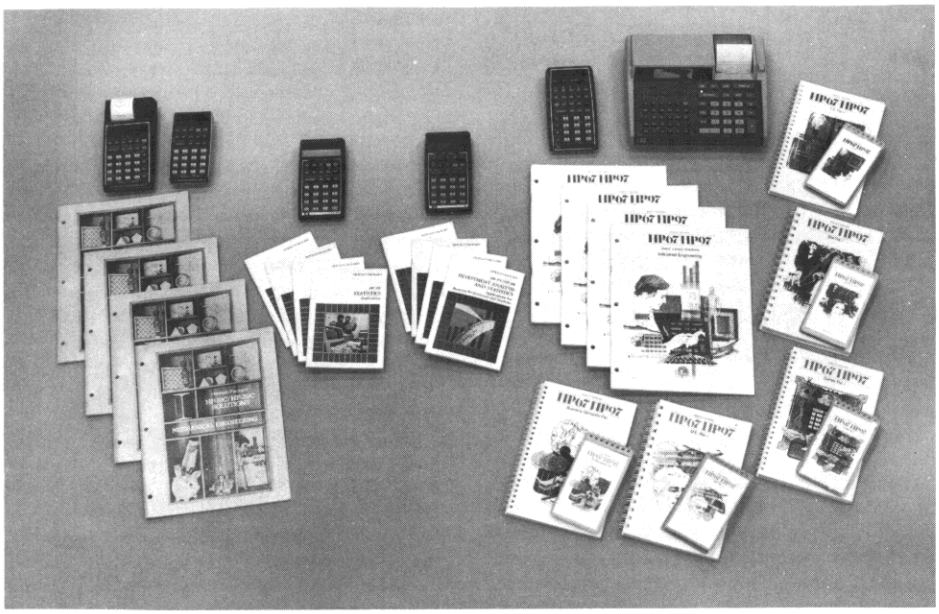
Pearl Robbins
Prentice-Hall, Inc.
Mail Order Sales Dept.
Old Tappan, NJ 07675

and your book will be shipped to you postpaid. If you live outside of these areas, write to

Prentice-Hall International, Inc.
66 Wood Lane End
Hemel, Hempstead
England HP2 4RG

for information on how you may order the book, postage, cost, etc.

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



Free Software Offer*... Up To $85 Value!

If you have been delaying the decision to move up from the HP-65 to the HP-67/97, *now is the time to make a positive move*. Or, maybe you've been toying with the idea of a second programmable for yourself, or maybe even a programmable for a college student or employee. Well, just purchase one of our advanced programmable calculators between March 1 and April 30, 1979, and you'll receive up to $85** worth of free software! The offer is:

HP-67/97—Buy a fully programmable HP-67 or HP-97 and you can choose any five *Users' Library Solutions* books—a $50 value—plus any one Application Pac—a $35 value *absolutely free!*

HP-19C/29C—Purchase a keystroke programmable with Continuous Memory and you'll have a choice of any four of our *HP-19C/29C Solutions* books—a $30 value *absolutely free!*

HP-33E/38E—Or purchase one of the new generation programmable calculators and choose any two of the four HP-33E Applications Books or any two of the three HP-38E Applications Books—a $10 value *absolutely free!*

Stop in at your local HP dealer after March 1 and check out this offer. You'll be able to personally examine all the advanced programmable calculators and see for yourself the range of free software available for this offer.

** This offer is available only in the Continental U.S., Alaska, and Hawaii.*

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

HP-97 Saves *Us* Money!

Every once in a while we hear about an application of one of our calculators that strikes a different chord. This time, it was in a letter from a firm that does some work for Hewlett-Packard.

Gentlemen:

The Boise office of CH_2M Hill is currently involved in the civil and structural building design modifications for two new buildings at the local Hewlett-Packard facilities site. The Boise site architects are Kolbo, Bowman, Smallwood and Associates of Boise.

During the course of the structual design modifications, several new programs for our HP-97 were developed. These are being prepared for submittal to your Users' Library soon.

These HP-97 programs, in addition to some in the *Civil Engineering* pac and the *Beams and Columns Users' Library Solutions* book, were used for steel base-plate design, reinforced concrete foots, pedestals and beams, steel columns, and moment distribution. This resulted in savings in engineering time and in materials which, of course, were passed to our client, Hewlett-Packard.

For example, we modified the standard building designed for UBC Zone 4 seismic forces in order to use the design in Zone 2. Thanks to the capability to check many possible combinations of base-plate sizes and thicknesses quickly, we were able to save 2.8 tons of steel in the base plates alone. Similar savings in time and/or materials were accomplished throughout the project.

To paraphrase a TV commercial, "Thanks, HP!"

Sincerely,
Joseph B. Worcester, Boise, Idaho

Thanks, Mr. Worcester, for the nice letter. Here's one HP-97 that really paid off for both the manufacturer and the customer! Ed.

HP KEY NOTES

February 1979 Vol. 3 No. 1

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

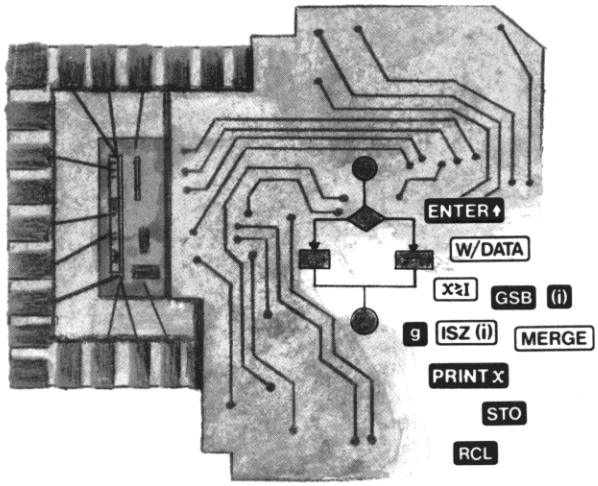
Hewlett-Packard SA
USERS' CLUB EUROPE
7, Rue du Bois-du Lan
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

Address Correction Requested
Return Postage Guaranteed

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

0001298918
M R DEZEARN
1310 G EVERETTE
ABERDEEN PR GRD MD 21005

Printed in U.S.A.



Featuring this issue:

825 Flags . . . Continued 3
Too Many Inputs/Outputs? 4
The Ultimate in Merging Proficiency 5
Spring is "Seed" Time! 6
"25 Words" (More or Less!) 6, 9, 10, 11
KEY NOTES Order Form 7, 8
Use the Right Paper! 11
"SST-ing" Thru Subroutines 12
Base Conversion Routine 12

May 1979 Vol. 3 No. 2

HEWLETT hp PACKARD

HP Key Notes

2½ Years Later

"Where is Corvallis? What is it like?"
"What does the factory look like?"
"How large is it? How many people?"

Those are questions we see over and over in your letters. Therefore, since we haven't said much about the "home" of your calculators since Volume 1 Number 1, we decided to answer those questions for all of you.

Corvallis is at latitude 44°34′ north and longitude 123°17′ west. Or, if that doesn't help, we are located 35 miles (56 km) south of Salem (state capital), 11 miles (18 km) west of Albany, 40 miles (64 km) north of Eugene, and 53 miles (85 km) east of Newport, which is on the Pacific Coast. Portland is just 81 miles (130 km) north of us.

Corvallis lies in the beautiful Willamette Valley, which is bordered on the east by the Cascade Range mountains and on the west by the Coast Range mountains. The Willamette River flows along the east border of the city and then north to Portland, where it empties into the Columbia River.

Oregon State University (OSU) is located here, and it averages about 15 to 17 thousand students. Total population of Corvallis is about 40 thousand.

It rains a lot, but it's green and clean here, and we can easily see Mt. Hood (well, most of the time!), which is about 100 miles (161 km) northeast. All in all, it is a nice place in which to live and to make calculators.

As you can see in the photo, there are two main buildings at the factory, which is on a 140-acre site just northeast of the city. Each building totals 152,000 square feet of area, and there is a 60,000 square foot storage area in the basement of the building in the foreground. Total square footage at the site is 416,000.

The building in the back houses the calculator assembly lines, service department, shops, personnel, marketing department, accounting, and all other administrative functions. The other building houses a highly sophisticated and up-to-date integrated-circuit (IC) facility plus the research and development functions. Approximately 1200 people work at the factory.

At right, center, there is a 5-acre natural lake. Clean waste water used in the IC processing area is recycled by pumping it to the lake and then onto the landscaped grounds. It then returns, in part, through the earth's water table to the lake. And, of course, the lake serves as a source of water for fire and other emergencies.

Like all Hewlett-Packard sites, it is very well-kept and is a beautiful place in which to work. We are as proud of our factory as we are of our calculators!

A Pleasant Surprise!

Are you tired of seeing prices going up, up, up? Well, here's some good news for a change. Effective May 21, prices of the HP-31E, HP-32E, and HP-33E were reduced as follows:

HP-31E Scientific Calculator. Was $60, now $50.* This is a basic scientific model in the tradition of the original HP-35 and the popular HP-21.

HP-32E Advanced Scientific Calculator. Was $80, now $70.* This calculator also includes statistical functions and 15 addressable storage registers.

HP-33E Programmable Scientific Calculator. Was $100, now $90.* This is a keystroke programmable model with 49 lines of program memory and many more features.

Stop in to see these bargains at your local HP dealer's store. It might seem early to buy a back-to-school present, but how often—these days—do you see prices going *down*?

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

Library Corner

As you can imagine, there are quite a few HP-65, HP-67, and HP-97 calculators being used all over the world by people like yourselves. So it takes a lot of paper to print enough catalog addendums to satisfy this elite group. Unfortunately, the northwest has recently suffered some problems in the paper industry and so we've had to postpone a scheduled addendum until we were positive we could get enough paper to print it.

However, we are happy to report that *Catalog Addendum #4* will be distributed during the month of June. It will contain 500 new programs and will include a greatly expanded Application Category Table to help you find programs in fields that interest *you*. The table was revised to reflect the broader range of fields and subjects extant in this day and age. Thus, we can now highlight such things as solar energy, computer science, home computing, dentistry, nutrition, education, etc. Programmable calculators are helping to change our world, so we must change with it. We think you'll find the new categories a great asset.

Now, before you jump the gun, remember that *Addendum #4* will be mailed via third-class bulk mail, so allow at least until the end of July before you check with us. Also, make sure your address is up to date; **the addendum will not be forwarded if you have moved.**

ORDERING PROGRAMS

None of the individual programs in this issue are available in Europe at this time. They will probably be added to the European Library Catalog at a later date. And, don't forget, if you live in the U.S.A., you can order Library programs you see here in KEY NOTES by calling the toll-free number on the Order Blank.

Library programs are available in two forms: A set of the program listings and instructions (software) is $3*, and the fee is $5* for a set of software *and* a recorded magnetic card.

NEW HP-67/97 PROGRAMS

Here is another "special" program. It is *complete* . . . right down to program flow diagram, nomenclature list, and terrific documentation. There are 448 steps, 2 cards, and 30 pages. The name is: **Compressible Flow #67000-99987,** and the price is $13.00.* It is the exacting work of **Gerard A. A. Westen** of Newark, Delaware (whose other "special," *Orifices for ME's,* appeared on page 2 of Vol. 2 No. 3). Here is the abstract:

This program solves adiabatic and isothermal compressible fluid flow problems in conduits, and will solve pressure drop under many conditions. It is applicable for the subsonic, sonic, and supersonic flow region and includes two program cards, one for English units and one for SI units. A calculator printout tape can be attached to the compressible fluid calculation sheet for permanent record purposes. This program will free engineers from laborious complex calculations and charts.

(Congratulations–once again–Mr. Westen. I know engineers will love this one! A very fine job! Ed.)

The next offering for this issue is, again, a "special" program. It consists of 1291 steps, 7 cards, and 30 pages. The program is designed to fit electrolytic conductance/concentration data to the Fuoss-Onsager-Skinner equation. Inputs are the viscosity and dielectric constant of the solvent, temperature, and the concentration-conductance data. Both associated and non-associated electrolytes are treated.

The curve-fitting is done by the Gauss-Newton regression method. While the equations require a starting approximation, this is not a problem, because convergence will occur for almost any approximation. Details of the Fuoss-Onsager-Skinner equation are given in an attached supplement, along with additional references.

Outputs are: limiting conductance, the L parameter, ionic radius, and, for associated electrolytes, the ion association constant.

The name of this program is: **Curve Fit for Electrolytic Conductance #67000-99977,** and the price is $15.50.* This Herculean effort is the work of **Bruce W. Clare** of Coolbellup, Western Australia, who has done a remarkable job of documenting his program.

For a change of pace—and direction—here is a recent program that makes use of some former programs so that accountants, and others, can have a so-called "single source" for this type of computation. The author said it was actually a joint venture with **Harold Scheinhaus** (Wheaton, Maryland) because the program incorporates some "perpetual calendar" data contributed earlier by Mr. Scheinhaus.

67/97 Interest Calculator Between Any Two Precise Dates (#03608D)

This program calculates interest between two precise dates (day/month/year), using a 365/360 interest rate factor and then cumulates interest if there are multiple transactions between borrower and lender. It uses a modified version of *Perpetual Calendar* (02198D). It is somewhat similar to program #00227D but uses a different method of days calculation, which works for any two dates since the adoption of our calendar. (138 steps, 5 pages)

Author: **Steven Guralnick**
Daly City, California

Now, believe it or not, a *third* "special" program, and this one is from **Bernard L. Golding,** who is a professional engineer and Chief, Water Resources (South), for Howard, Needles, Tammen & Bergendoff of Orlando, Florida. His program is titled, **Dissolved Oxygen Concentration in Rivers #67000-99978,** and the price is $9.50.*

Mr. Golding has written and documented an impressive program. It is 410 steps long, is contained on 2 cards, and includes 21 pages of typed, easy-to-understand documentation. His abstract does a good job of describing his program, so here it is:

The program consists of a two-part mathematical model that simulates in rivers and canals the steady-state dissolved oxygen concentration resulting from both carbonaceous and nitrogenous biological oxygen demands (CBOD and NBOD). In the first part, BLEND, mass balances are performed to determine inputs to a river reach. In the second part, DOREM, the dissolved oxygen remaining at the end of the river reach or at one mile intervals along the river reach are computed.

Seven pages of the documentation are devoted to a small textbook describing the program, presenting the equations, and defining the variables. The three-page sample problem shows all keystrokes and outputs in an easy-to-follow format, and detailed, clear User Instructions are given in seven pages. The final four pages give the program listings, with extensive comments for the two cards.

(An excellent example of proper documentation, Mr. Golding; we congratulate you for carefully and diligently recording your work. Ed.)

We Get Letters

It is nothing new to readers of KEY NOTES that our calculators are sometimes used for some very unusual tasks, or in some very unusual places. Therefore, to make sure we keep you well informed about the many things you can do with your calculator, we present the following letter.

Gentlemen:

This is to show that an HP-67 calculator may have other uses besides mathematical calculations.

Convince yourself and your friends that the human eye has a blind spot: Turn the HP-67 on and press DSP 0. Press h REG and look with your *right* eye at the zeroes blinking at the left side of the display as the register contents are displayed. If you slowly move the calculator back and forth around a distance of approximately 8 inches from your eye, you will find a distance at which the register numbers on the right will not be visible! You just located the blind spot on your right eye. (Your left eye should be closed or covered.)

Conversely, looking with your *left* eye (covering the right) at the register numbers blinking on the right side of the display, you will find a blind spot there, too, when you cannot see the zeroes displayed on the left side.

Sincerely,
Carlos R. Schwantes, Monroeville, Pennsylvania

(Thanks for the letter, Mr. Schwantes. I'll have to admit I was skeptical, but it does work. Ed.)

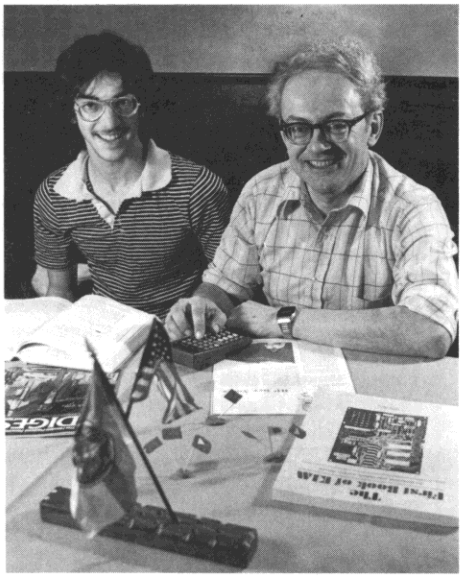
**U.S. dollars. See note at bottom edge of cover.*

825 Flags . . . Continued

Did you notice the blank space at the bottom of the center column on page 5 of the last issue? Unfortunately, it was not left there on purpose. There should have been a paragraph, in blue ink, as follows:

"Starting the column this month is a real breakthrough in programming. We've seen programs that can provide up to 750 software flags, but this one, from **Cass R. Lewart** of Holmdel, New Jersey, can provide up to 825 flags! The flag program was jointly developed by Mr. Lewart and his 16-year-old son, Dan."

Through some unexplained error during the printing of the newsletter, the paragraph disappeared, so we asked Mr. Lewart to forgive us and to take us up on an offer to print his picture along with the missing paragraph. So here it is. And please accept our apologies, Mr. Lewart.



Dan and Cass Lewart. Dan has won several state-wide competitions in mathematics, and both father and son are active members of PPC, the calculator user's club. (A neighbor, **Dankwart Koehler,** took the photo, processed the film, and made this print.)

The 825-flag routine by the Lewart's elicited a few questions, so here's a continuation of the subject.

825 Flags—A Base b Encoding/ Decoding Programming Technique

In our last issue of KEY NOTES, we had a 66-step routine for the HP-67/97 that provides control of up to 825 flags. The routine used a programming technique that may be described as, "Base b Encoding/Decoding." Using a base of 2, the routine stored 33 flags in one HP-67/97 data register. This article describes the techniques used in the routine and shows how other combinations of numbers may be packed into an HP-67/97 register using other bases.

To understand Base b encoding/decoding, we must understand what is the base of a number system. In our day-to-day dealings with numbers, we use the numbers—or digits—0 thru 9. So the base of our number system is 10. When we count, we start with 0 and proceed to 9. When we get to one less than the base, i.e., 9, we carry one count to the next place and repeat the count from zero. Ten follows 9, 100 follows 99, etc. We observe that 123 is a number that may be described as 1 in the hundreds place plus 2 in the tens place plus 3 in the units place. This may be illustrated:

$$\underline{1}\ 100\text{'s} + \underline{2}\ 10\text{'s} + \underline{3}\ 1\text{'s}$$

This may also be expressed using the place value and the base of the number system being used.

$$\underline{1} \times 10^2 + \underline{2} \times 10^1 + \underline{3} \times 10^0$$

Table 1 further illustrates the idea.

Table 1. Place Numbers and Values for 123

Place	Third	Second	First
Place Value	100	10	1
Place Count	2	1	0
Number	0	0	0
	0	0	1
	0	0	2
	0	0	3
	.	.	.
	.	.	.
	.	.	.
	1	2	3

Now, let's apply these ideas to the Lewart's flag routine in the February 1979 issue of KEY NOTES. As you know, a flag has two states, set (1), and clear (0). So a base of 2 is suitable for this application. Table 2 shows base 2 numbers in the same format as table 1.

Table 2. Binary Values for Base 10—0 thru 16

Place		5th	4th	3rd	2nd	1st
Base 10 Place Value		16	8	4	2	1
Place Count (flag #)		4	3	2	1	0
Base 10 Number:	0	0	0	0	0	0
	1	0	0	0	0	1
	2	0	0	0	1	0
	3	0	0	0	1	1
	4	0	0	1	0	0
	5	0	0	1	0	1
	6	0	0	1	1	0
	7	0	0	1	1	1
	8	0	1	0	0	0
	9	0	1	0	0	1
	10	0	1	0	1	0
	11	0	1	0	1	1
	12	0	1	1	0	0
	13	0	1	1	0	1
	14	0	1	1	1	0
	15	0	1	1	1	1
	16	1	0	0	0	0

In base 2 a number place is called a BIT for *BI*nary digi*T*. Five bits are shown. The number 7 in base 2 (binary) is 111, and 3 bits are required to represent it. The base 10 value for binary 111 may be obtained by adding the base 10 place values as shown below.

$$\underline{1} \times 2^2 + \underline{1} \times 2^1 + \underline{1} \times 2^0 = 4 + 2 + 1 = 7$$

Storing 33 flags in one register is accomplished by representing the flags as a binary number stored in base 10 form.

If each place can represent three flags, then ten places can represent 33 flags. Each place cannot represent four flags (bits) because not all combinations of set and clear, 1's and 0's, can be represented by the highest digit 9, even though it is a four-bit number (1001). The upper limit of flags is better seen by considering that $2^{33} - 1$ is 8,589,934,591, which is the highest all-flags-set, base-10-value less than 11 digits. The next full set of flags is $2^{33} - 1 =$ 17,179,869,183, which is too large.

Study the simplified routines below to follow the Test, Set, and Clear process. Labels A-Test, B-Set, and C-Clear function the same as in the Lewart's routine. The error-correcting and housekeeping steps that direct each group of 33 flags to successively higher registers have been omitted for illustration and study. Routine A tests a given flag by dividing the register contents by 2 raised to the flag number power, then taking the integer part and testing for odd or even by dividing by 2. An even result displays a zero, for clear, by the last step, FRAC. A remainder (0.5) indicates the flag is set. (Key DSP, 0 to show "1.") The ABS step in routine A is used to store 2^n in LAST X (n is the flag number) and it is used by routines C and B if required.

Test Flag n	Clear Flag n	Set Flag n
*LBL A	*LBL B	*LBL C
2	GSB A	GSB A
X≷Y	X=0?	X≠0?
Y^x	RTN	RTN
ABS	X≷Y	X≷Y
RCL 0	ST-0	ST+0
LSTX	RTN	RTN
÷		
INT		
2		
÷		
FRC		
RTN		

Routine B uses routine A to test the flag. If it is clear (0), no further action is taken and program execution stops at the first RTN. If the x=0? test shows the remainder (0.5) because the flag was set, the value of 2^n remaining in the stack is subtracted from the register after the x≷y step.

To better visualize the process, refer to table 2 and follow the process using the more familiar base 10 values. For example, suppose flags

0, 1, 2, and 3 are set. (Table 2 only shows all combinations of four flags.) All the other flags are clear. Under these conditions, R_0 would contain 15. Now suppose we wish to clear flag 2. In base 2, the first four flags set condition of 1111 would become 1011 when flag 2 is cleared. This corresponds to a base 10 value of 11. When subroutine B determines that the flag is set, it subtracts 2^n, which is 2^2 in this example. Considering only flag 2, the binary value is 0100, which is 4 in base 10. Subtracting from 15 gives the desired result:

Base 2	Base 10
1111	15
−0100	−4
1011	11

Routine C is identical to routine B except it does nothing if the fraction part of subroutine A is not zero, but adds the value of 2^n if the flag is clear. It is just the reverse of the example above.

The simplified routines above are fast and short, but have a limit of 30 flags per register. Because the routine uses the fractional part in testing a flag, the large numbers result in round-off error when flag numbers larger than 29 are used. The Lewart's additional 39 steps avoid the round-off error and provide for automatic addressing of the registers. If the full storage capability of the registers is used, it is possible to store up to 41 flags in each register. (Two more by using the signs of the characteristic and mantissa, and six more by using the characteristic to gain the additional eight flags.)

The principles described above can be generalized to increase register number capacity. In the more general sense, only two routines are required, one to store and one to recall the number. However, an additional input is required: the location in the register must also be entered.

In the Flags examples above, the register was divided into parts identified as flags 0-29. To identify the location within a register, the positions will also be numbered 0, 1, 2, 3 etc., up to the limit of the 10-digit register capacity. The number to be stored is an integer, i, to be stored into position p. Almost *any* base may be used. The base is stored in register B. If you use these routines in a program, you can free register B by having the program contain the base in each step that recalls register B. The following instructions apply.

1. Store base b in register B.
2. To store number i in position p, key p, ENTER, i, and press E.
3. To recall number i from position p, key p and press A.

Recall Position p	Store i in Position p
*LBLA	*LBLE
RCLB	STOE
X≷Y	X≷Y
Y^x	GSBA
ABS	×
RCL0	ST−0
LSTX	X≷Y
÷	RCLE
INT	×
LSTX	ST+0
RCLB	RTN
÷	
INT	
RCLB	
×	
−	
RTN	

Table 3 gives the limits of a few of the more common bases used in programs that require data packing to more fully utilize the storage capacity of the HP-67/97. Other methods have been used; for example, decimal point encoding, and logarithmic encoding. Of all the methods used, however, the most amazing is Base b encoding/decoding using base 2 to represent flags.

Table 3. Base b Encoding/Decoding Limits

Number Range, i	Positions	Base
0-1	0-29 (30)	2
0-2	0-18 (19)	3
0-4	0-12 (13)	5
0-9	0-9 (10)	10
0-20	0-6 (7)	21
0-99	0-4 (5)	100
0-99999	0-1 (2)	10^5

For Cryptology Fans ...

CRYPTOLOGIA is a quarterly journal (100 pages per issue, 8½ by 11-inch format) with January, April, July, and October issues. It is the only journal dedicated to a broad, yet in-depth, coverage of cryptology and related fields. Among the subjects discussed are: Computer Security, Military Science, History, Mathematics, Literature, and Ancient Languages. Regular features include: Crypt analyst's Corner, Devices and Machines, Course in Cryptology, Book Reviews, and a column with bits about sources and happenings. If you are interested in this sort of thing, write for more information to: *CRYPTOLOGIA; Albion College; Albion, Michigan 49224. A year's subscription costs $20.** Also, back issues and bound volumes of past years are available.

(Thank you, Dr. Winkel, for sending us information about this service. Ed.)

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

Too Many Inputs/ Outputs?

We recently received a very interesting letter from **Cass R. Lewart** (Holmdel, New Jersey) and, since we have had questions about this same function, we decided to share it with you.

What to Do When Your Program Has Over 10 Inputs/Outputs.

A good practice that produces an easy-to-use program is to assign only the ten non-numerical labels A-E and a-e to individual program inputs or outputs. However, it frequently happens that a program requires more than 10 inputs or outputs. Assigning numerical labels 0-9 necessitates pressing three keys (f GSB n) before entering a value or obtaining a result and should be avoided. Another straightforward method, namely keying GTO .nnn followed by R/S is even more cumbersome. What follows are several alternate methods of providing additional inputs or outputs, which you may find quite convenient when you run out of non-numerical labels. Each method has some advantages and disadvantages; the choice will depend on your particular program and personal preference.

1. Key in 0 through 9 followed by a non-numerical label. Good for up to 10 additional inputs or outputs.
2. Key in a non-numerical label by itself or press *any* numerical key followed by a non-numerical label. This method is frequently used for sharing the same label for input or output. Make sure that F3 is clear when you record on the magnetic card.
3. Repeated keying of the same non-numerical label. **This and the following methods have the disadvantage that inputs or outputs must follow in the same sequence.** Use of a prompting routine is recommended.
4. Alternate use of the same non-numerical label.
5. Use of a non-numerical label followed by repeated R/S for inputs, PAUSE or -x- for outputs. Several outputs can then be displayed in succession. This method is particularly suited for programs with many inputs or outputs.
6. Use of a non-numerical label followed by SST's, followed by R/S. This method saves on steps but does not allow for prompting; therefore it should be used as a last resort.

The following examples illustrate the six methods. An asterisk (*) shows suggested locations for a call to a prompting routine. The main program should reset the counter of the prompting routine R_i to 0.

1.	*2.*	*3.*	*4.*	*5.*	*6.*
LBLA	LBLA	LBLA	LBLA	LBLA	LBLA
STOI	F3?	*	F2?	*	R/S
GSBi	GTO1	.	GTO1	.	STO/RCL1
RTN	.	.	*	.	STO/RCL2
LBL1	.	RTN	.	R/S or	.
.	RTN	LBLA	.	PAUSE	.
.	LBL1	*	SF2	*	RTN
RTN	.	.	RTN	.	*Prompting*
LBL2	.	.	LBL1	.	*Routine:*
.	RTN	RTN	*	R/S	LBL9
.			.	*	ISZ I
RTN			.	.	DSPO
			RTN	RTN	RCLI
					PAUSE
					DSP5
					RTN

HP-97 For Basketball?

Well, maybe not for basketball *per se,* but definitely for basketball *coaches!*

Dr. Aaron N. Moen, presently a professor and research scientist at Cornell University, has written a most remarkable book for coaches who are serious about evaluating team statistics, and for college professors in athletic theory who want to introduce students to a most up-to-date approach to evaluating team statistics. The book, *Basketball Performance Profiles,* describes ways to profile games *in progress,* and to evaluate seasonal trends from game to game. The first part of this book describes ways to do this with a clipboard and a handheld calculator. The techniques described in the second part of the book are based on fast, programmed electronic computing. Efficiency ratios are calculated during the game and as game summaries, providing insight into the floor functions that were strong, and those that were weak. Over 200 evaluations are completed during a game (26 evaluations per profile), and game summaries provide 78 additional evaluations. All of these are completed within 5 minutes after the game is over, using the instructions, program steps, and equipment described in this 175-page book.

If you live in the U.S. and want to obtain this book, send a check or money order for $10.00* to Corner Brook Press; Box 106; Lansing, New York 14882. If you live outside the U.S. or just want more information, write to the above address.

And if your local high school, college, or company basketball team is losing too many games, perhaps you should tell the coach about this newfangled science. And, just think: All along you thought the HP-97 was just another superlative scientific calculator for serious number-crunching in such fields as mathematics, medicine, statistics, and so on. As it turns out, it also happens to be the most powerful coaching aid ever available at courtside.

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

The Ultimate in Merging Proficiency

That title may seem a bit strong, but after you read the following letter from **James H. LeMay** of Houston, Texas, perhaps you'll agree. At any rate, we think you will enjoy his contribution to KEY NOTES.

Dear HP KEY NOTES:

I agree with **A. G. Burns** (in Vol. 3 No. 1) that one of the most useful attributes of the HP-67/97 is the ability to accept programs longer than 224 steps, but I think I have found a simpler technique to accomplish unlimited merging and yet maintain all the advantages of his method for programming as well as data storage.

The technique for programming takes advantage of two aspects of the HP hardware. One is the execution of step 001 immediately following step 224. The other is that, when a second card is merged, the calculator accepts from the merging card only the number of steps that would fill up 224 steps of programming in the calculator.

For the first card in a long series to be merged, the program would be:

```
001    PSE   220     2
002   GTOi   221   STOI
003  *LBLA   222   PRTX
              223  *LBL2
               .   224   MRG
               .
               .
```

Now the second card, card 2, can be programmed to be:

```
001   GTOi   219     3
002  *LBL2   220   STOI
               .   221   PRTX
               .   222  *LBL3
               .   223   MRG
                   224   R/S
```

Notice that card 2 can have a maximum of only 223 steps.

Starting with step 220 on card 1, the "2" command serves three purposes. First, for the HP-97 only, the number of the next card to be used can be printed on tape to keep track of the card in use. Secondly, as the calculator begins the merge loop, the number of the next card to be merged is displayed during the pause as a visual reminder of the next card to be used. (Not all programs use cards in consecutive order.) Thirdly, as the calculator executes the GTO i statement, it returns back to the merge loop at LBL2 and sits in a holding pattern as it flashes the number of the next card each time the pause function is executed.

As card 2 is called and input into the calculator, the programming will now look like:

```
001    PSE   220     3
002   GTOi   221   STOI
003  *LBL2   222   PRTX
               .   223  *LBL3
               .   224   MRG
               .
```

The only command left from card 1 is PSE, in step 001. Steps 001 to 223 on card 2 are now steps 002 to 224 in the calculator. Immediately after merging card 2 from the PSE command at step 001, the GTOi searches for LBL 2 because "2" was stored in the I-register back on card 1. But, because the program memory was rewritten with card 2, LBL 2 is now immediately after the GTOi statement in step 003, and it automatically continues the program execution. As the body of the program has finished execution, the new programming added from card 2 allows card 3 to be merged the same way. This time, 3 is displayed and stored in I and LBL 3 is the address of the new merge loop. The next cards will look like:

CARD 3		CARD 4		LAST CARD
001	GTOi	001	GTOi	001 *LBLA
002	*LBL3	002	*LBL4	.
	.		.	.
	.		.	.
	.		.	RTN
219	4	219	5	
220	STOI	220	STOI	
221	PRTX	221	PRTX	
222	*LBL4	222	*LBL5	
223	MRG	223	MRG	

The last card needs only a program body.

Merging cards (programs) in this manner can continue *ad infinitum.* It is convenient to number the merge loops in increasing order from card to card. For the program above, the merge loop was LBL 2 on card 1, LBL 3 on card 2, LBL4 on card 3, etc., up to LBLe and back to LBL 0 for card 20, but it does not have to be in order. The address can simply alternate between any two numbers, say 2 and 3, as more cards are merged. Card 3 (above) could have used LBL 2 for the merge loop, but it requires the command "4" after STOI to indicate the number of the next card to be used. For convenience and shorter programming, it is easier to leave it in order.

In the way of suggestions . . . if there is data stored in the I-register from the body of the program and it is necessary for it to be kept for future execution on other cards, the value can be easily recalled to the stack and juggled while a new card is being merged and then it can be replaced.

Also, if the total number of steps in the program does not add up to 223 and it is desired to merge future cards, the remainder can be filled with superfluous R/S commands any place convenient in the body of the program.

Finally, the PRT x command on the HP-67 is not necessary, but be sure the program retains the proper number of steps.

The advantages of this technique are:

1. Looped automatic merging (no keys to press).
2. Unlimited number of cards that can be merged.
3. Maximum programming space.
4. Total preservation of Set Status (no flags are used).
5. Visual indication of card to be merged.
6. Unlimited number of coffee (tea?) breaks between calculations.

What more could a programmer ask?

The technique for merging data cards automatically is less involved but more versatile. For example, in the course of programming, storage registers 0 to 13 need data from the corresponding registers on the data card, card 4. The program would look like:

```
               .     4     F3?
               .   PRTX    F3?
               .   CF3     GTO0
               1  *LBL0      .
               3   MRG       .
            STOI   PSE       .
```

As the program stores the number of the last register to be merged, the card number is printed on the HP-97 (a step not necessary on the HP-67) and continues to be displayed in the merge loop as long as no data is entered. Once the data is entered from both sides of the card, flag 3 is set and the GTO 0 is skipped, so execution of the body of the program can continue automatically.

This also allows two more options. The merge loop can be defeated by simply pressing any integer key on the keyboard while the display is flashing. Then the merging of data will be completely skipped and the data will be untouched (except that the stack was raised).

For the second option, the programmer can store just secondary registers. For example, suppose it was necessary to merge only secondary registers 0 to 3. The above program is identical except, to warn the programmer, the card indicator "4" could be changed to "4.2" meaning card 4, side 2. Then, when the calculator displays "4.2," side 2 of the data card should be entered, and only storage registers 10, 11, 12, and 13 have been added to memory. The calculator now displays **Crd.** Press any key to erase the display and the calculator will automatically continue execution.

Again, the advantages are:

1. Looped automatic merging.
2. Variable data entry.
3. Visual indication of the card and side to be merged.

I hope this has shown the power of the HP-67/97 merge capability and simplicity to its fullest.

(Indeed it has, Mr. LeMay. Many thanks for a fine job and for sharing your contribution with our readers. Ed.)

Spring is "Seed" Time!

John Craig, of Anaconda, Montana, contributed the following ideas on seeds. If you use seeds at all, this should be of interest to you.

Problem:

1. Seed given by HP for random number is hard to remember.
2. Some programs—like, games—need different starting sequences every once in a while.
3. It's hard to test other seeds to make sure they are as good as the one given.

Solution:

I experimented on our HP 9825A Desktop Computer and discovered several facts:

1. Seeds do recycle at exactly 500,000 as mentioned in the manual.
2. The 10 seeds formed from .n123123 (where n is any one of the 10 digits) form 10 starting seeds spaced exactly 50,000 seeds apart in the same tested sequence.
3. The 100 seeds formed from .ab23123 (where "ab" is 00 thru 99) form 100 starting seeds, each spaced exactly 5,000 apart, again in the tested sequence suggested by HP.

As an example of use, after loading a game, perhaps the first instructions would be to key in a two-digit integer between 0 and 100. The starting seed could then be formed by the program by tacking-on the "23123" and shifting the decimal point. This would guarantee a good sequence and would provide new seed sequences.

"25 Words" (More or Less!)

It is apparent from the mail we receive that most of you enjoy this column and read and/or use it. So you will notice that the column has been growing not only in size but also in scope. And it will continue to do so, as long as space, time, and inputs permit. We are only sorry that we cannot print all of the inputs we get. Anyway, here are some more interesting tidbits we know you will enjoy.

First entry this issue is a trick from **Frank A. D'Amico** (Pascagoula, Mississippi). It makes one of the *HP-67/97 Standard Pac* programs easier to use.

The following procedure will permit the use of both sides of the *HP-67/97 Standard Pac* program SD-12B ("English-SI Conversions") without having to reprogram each time you switch sides of the card.

Merge both sides of the card onto a blank card (169 steps) then insert "LBL 1" before step 001 and "LBL 2" after what is now step 081. CAUTION: To make sure you're in the proper place, GTO1 **must** be keyed *each time* before using LBLA-E or a-e on side 1, and GTO2 **must** be keyed *each time* before using LBLA-E or a-e on side 2.

Now, what's new in Wahiawa, Hawaii? That's easy; it's a neat and very simple plotting routine from **Dale P. Weber.**

I have come up with a Nonfunction Plotting Routine that can be used on both the HP-67 and HP-97. It uses one card for both the program and storage of the "print constant."

001	*LBLB	017	RCLA
002	STOA	018	-
003	-	019	RCLB
004	9	020	÷
005	÷	021	RND
006	STOB	022	10^x
007	RCLA	023	STOC
008	RCLB	024	RCLD
009	2	025	X≷Y
010	÷	026	÷
011	-	027	FRC
012	STOA	028	RCLC
013	RCLB	029	x
014	R/S	030	PRTX
015	*LBLA	031	R/S
016	DSP0		

The number (constant), 987654321.0, is stored in register D. To store it on the card, clear all registers except D, and put any instruction (say, CHS) in program line 113. (This line will not be used or recorded. Now load the program on side 1, then **Crd** will be displayed. Press CL x and then W/DATA. Now load side 2. Side 1 will contain the program and side 2 the constant. To use the plotting routine, enter the maximum (say, 250) and press ENTER↑. Then enter the minimum and press B. Now, just key in your values and press A after each one (with printer set for MAN). HP-97 users will obtain a graph and HP-67 users will be able to construct one by looking at the left-most number.

(It works well, Mr. Weber, but it really is more convenient on the HP-97, with an already-plotted output. Also, the narrower the range the better the plot. Ed.)

This next routine was the result of a son's request to check his long division problems from school. The answers required the remainder to be shown, so **David L. Smith** (Burlington, Massachusetts) wrote, for his son:

Check Long division and Show Remainder.

001	*LBLA	010	x
002	÷	011	EEX
003	LSTX	012	4
004	X≷Y	013	÷
005	ENT↑	014	R↑
006	FRC	015	+
007	X≷Y	016	DSP4
008	INT	017	RTN
009	R↓	018	R/S

To use the routine, key in the dividend and press ENTER↑, then key in the divisor and press A. For example:

4359 ENTER↑ , 57 A --- 76.0027
Quotient = 76, Remainder = 27

(Now none of you has an excuse for not checking your son's or daughter's homework! Ed.)

For a change of pace—and geography—here is a 32-step routine from **Win Acton** of Vancouver, Washington, that is guaranteed to make your life easier if you insist on expressing length measurements in feet and inches with binary fractions, instead of succumbing to metrication.

Input format is ffii.nndd, where ff is integral feet, ii is integral inches, nn is the fraction's numerator, and dd is its denominator. For example, 6 feet $3^5/_8$ inches is keyed 603.0508. The entire fraction may be zero but a zero denominator following a non-zero numerator won't work. No storage registers are used. The previous value in X is saved in Y (and Z and T), so chain calculations are facilitated.

001	*LBLC	017	*LBL0
002	INT	018	+
003	LSTX	019	EEX
004	FRC	020	2
005	X=0?	021	÷
006	GTO0	022	INT
007	EEX	023	LSTX
008	2	024	FRC
009	x	025	EEX
010	INT	026	2
011	LSTX	027	x
012	FRC	028	1
013	÷	029	2
014	EEX	030	÷
015	2	031	+
016	÷	032	RTN

(Continued)

HP-67 and HP-97 Accessories

Description	Model Number	Price
DC Recharger/Adapter (HP-67)	82054A★	$ 35.00
Reserve Power Pack (HP-67)	82004A★	$ 20.00
Reserve Power Pack (HP-97)	82037A	$ 35.00
Security Cradle (HP-67)	82015A★	$ 30.00
Security Cable (HP-97)	82044A	$ 10.00
Hard Leather Case (HP-67)	82016A★	$ 35.00
AC Recharger/Adapter (HP-67) 110/220 Vac., Switchable	82002A★	$ 20.00
(HP-97) 110 Vac.	82059A	$ 12.50
(HP-97) Euro. 220 Vac.	82066A	$ 16.00
Battery Pack (HP-67)	82001A★	$ 10.00
Battery Pack (HP-97)	82033A	$ 18.00
Soft Case, Black Leather (HP-67)	82017A★	$ 10.00
Soft Case, Synthetic (HP-67)	82053A★	$ 7.00
Soft Case, Synthetic (HP-97)	82035A	$ 10.00
Thermal Printing Paper, 6 rolls (HP-97)	82045A	$ 6.00
Program Pad (HP-67 & HP-97)	00097-13154★	$ 4.00
3 Program Card Holders (HP-67/97)	00097-13142★	$ 10.00
Blank Program Cards (HP-67/97) 40-Card Pack with Holder	00097-13141★	$ 20.00
120-Card Pack with Holders	00097-13143★	$ 45.00
1000-Card Pac	00097-13206★	$195.00

HP-67/97 Application Pacs Each $35.00

HP-67 Standard Pac	00067-13101	Games Pac	00097-13185
HP-97 Standard Pac	00097-13101	Math Pac I	00097-13121
Business Decisions Pac	00097-13144	ME Pac I	00097-13155
Civil Engineering Pac	00097-13195	Navigation Pac	00097-13205
Clinical Lab & Nuclear Medicine Pac	00097-13165	Stat Pac I	00097-13111
EE Pac I	00097-13131	Survey Pac I	00097-13175

HP-67/97 Users' Library Solutions Books Each $10.00

BUSINESS:		**MEDICAL:**	
Options/Technical Stock Analysis	00097-14009	Medical Practitioner	00097-14005
Portfolio Management/Bonds & Notes	00097-14010	~~Anesthesia~~	~~00097-14019~~✱
Real Estate Investments	00097-14012	Cardiac	00097-14018
Taxes **(UPDATE FOR 1978)**	00097-14004	Pulmonary	00097-14037
Home Construction Estimating	00097-14033	**PHYSICAL/LIFE SCIENCES:**	
Marketing/Sales	00097-14032	Chemistry	00097-14006
Home Management	00097-14031	Optics	00097-14016
Small Business	00097-14039	Physics	00097-14015
ENGINEERING:		~~Earth Sciences~~	~~00097-14017★~~✱
Antennas	00097-14021	Energy Conservation	00097-14029
Butterworth & Chebyshev Filters	00097-14003	Space Science	00097-14028
Thermal & Transport Sciences	00097-14023	Forestry	00097-14034
EE (Lab)	00097-14025	Biology	00097-14040
Industrial Engineering	00097-14035	**OTHER:**	
~~Aeronautical Engineering~~	~~00097-14036★~~✱	Games	00097-14013
Beams & Columns	00097-14027	Games of Chance	00097-14038
Control Systems	00097-14026	Aircraft Operation	00097-14001
COMPUTATION:		Avigation	00097-14002
High-Level Math	00097-14011	Calendars	00097-14024
Test Statistics	00097-14008	Photo Dark Room	00097-14022
Geometry	00097-14007	COGO/Surveying	00097-14020
Reliability/Quality Assurance	00097-14030	~~Astrology~~	~~00097-14014★~~✱

★ Also usable on the HP-65.

✱ *No longer available.*

HEWLETT-PACKARD ORDER FORM

HP CODE NO. 300ZC

Model Number or Program Number	Description	Qty.	Price Each	Total Price

Please enter my one-year subscription	☐ For US, Puerto Rico or Virgin Islands	**$15.00**	
to the HP-67/97 Users' Library	☐ For countries outside US, PR, V.I., and Europe	**$20.00**	

SHIPPING CHARGE★ _____

STATE AND LOCAL TAX _____

TOTAL PRICE★★ _____

You may order any of the HP-67 and HP-97 accessories and software shown by calling our toll-free number, 800-648-4711 (in Nevada 800-992-5710), and asking for the nearest Hewlett-Packard Accessory Dealer. (Toll-free number not available in Alaska or Hawaii.)
Or you may use the convenient order form on the reverse side and mail to: Hewlett-Packard Co., 1000 NE Circle Blvd., Corvallis, Oregon 97330.

NOTE: All Users' Library Programs mentioned in Key Notes can be ordered on this order form. Simply note the appropriate Program Number and fill in the description and price. All programs with pre-recorded program cards are available at $5.00 each.

Hewlett-Packard reserves the right to make changes in materials, specifications, and prices without notice.

Orders cannot be shipped to any European countries.

★ For all orders shipped outside US please add $5.00 shipping charge.
★★ Paid in US dollars drawn on a US bank.

PAYMENT OPTION (Please check one)

☐ **CASH PAYMENT.** Check or money order enclosed in US dollars drawn on US bank. (Please be sure to include your state and local sales taxes.)

☐ **BILL MY CREDIT CARD.** To use our seven-day 24-hour charge-by-phone service, call us Toll Free at (800) 648-4711, Ext. 1000; in Nevada call (800) 992-5710. Toll-free number not available in Alaska or Hawaii. You can use the deferred payment plan available through your Master Charge or Visa/BankAmericard account simply by filling in the information below.

CHARGE TO: ☐ American Express ☐ VISA/BankAmericard ☐ Master Charge—Include 4-digit number appearing on card just above your name:

Your credit card account will be billed when your order is shipped.

MY COMPLETE CARD NO. IS: EXPIRATION DATA: _____

If using a credit card and the billing address does not correspond to that shown for shipping, please indicate correct billing address.

_____
NAME APPEARING ON CREDIT CARD

_____
STREET ADDRESS city, state, zip

_____
SIGNATURE TITLE BUSINESS PHONE NO. (Include area code)

SHIP TO: _____
NAME

_____
ADDRESS

_____
CITY STATE ZIP

To find the volume of a box 3 feet 0-47/64 inch wide, 4 feet 11-9/16 inches long, and 9-¾ inches high:

Key	Press	Display
	DSP3	0.000
300.4764	C	3.061
411.0916	C_yx	15.194
9.034	C_yx	12.345

Now, how about a "Slow Register Review" routine specifically for all our HP-67 users, so they can write down the contents of the registers? Here it is, thanks to **Curtis J. Caldwell** of Jacksonville, Florida.

Occasionally it is desirable to perform the HP-67 "h REG" function slowly to allow the copying of register contents, or to interrupt such a review by halting execution while retaining the ability to resume the review from that point, retaining the sequence compatability with the "h REG" function for rapid register review. The I register is used to step through memory, and thus does not assume the importance which it might when the "h REG" function is used. Three variations are:

STEP	1	2	3	Remarks
001	*LBLA	*LBLB	*LBLC	
002	CLX	CLX	CLX	Zero the I register.
003	STOI	STOI	STOI	
004	*LBL1	*LBL2	*LBL3	
005	1	1	1	Skip secondary registers.
006	0	0	0	
007	RCLI	RCLI	RCLI	
008	X=Y?	X=Y?	X=Y?	
009	+	+	+	
010	STOI	STOI	STOI	
011	2	PSE	PSE	Begin display routine, testing for the I register as the indicator to quit (#25), displaying the register number about to be reviewed, and then displaying the contents of the register of interest.
012	5	RCLi	RCLi	
013	X≷Y	PRTX	X=Y?	
014	X=Y?	ISZI	R/S	
015	RTN	GTO2	PRTX	
016	PSE		ISZI	
017	RCLi		GTO3	
018	PRTX			
019	ISZI			
020	GTO1			

Routine #1 is useful when a return is required from the output routine to continue some other task. It is the most suitable of the three for programs created for use by non-programmers and casual users in that no special user instructions are required and no "unnerving" messages are displayed.

Routine #2 is useful when data output is the last program task, when program space is critical, and the user is not distrubed by halting with "Error."

Routine #3 is a substitute for routine #2 when the user is uncomfortable with an "Error" message. The user is burdened with pushing "R/S" whenever the register number equals its contents.

I would be interested in other routines that perform this function; those that do not use storage registers, those that require fewer program steps, or those that use not more than one extra step yet have a shorter execution time.

(Okay, Mr. Caldwell, we'll be the first to respond. Try this routine. Maybe you'll see more in the next issue. Ed.)

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

Let's now try a little different subject and a new approach to an old problem. It's from **Coy R. Morris** of Leesburg, Virginia.

Enclosed is a very simple technique for conversion from Spherical to Rectangular coordinate systems and vice versa. The simplicity of the technique lies in its use of the calculator's built-in conversion from Polar to Rectangular and Rectangular to Polar Functions. Anyone familiar with the coordinate transformation equations (shown below) for both systems will realize the savings in keystrokes and memory. The technique is particularly useful for quick conversions from the key pad and excellent for gaming and scientific programs.

Spherical to Rectangular

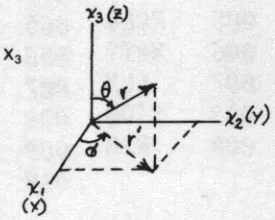
$x_1 = r \sin\theta \cos\phi$
$x_2 = r \sin\theta \sin\phi$
$x_3 = r \cos\theta$

75.00 ENT↑ → θ
15.00 → R → r
3.88 *** → x_3
CLX
90.00 x ⇄ y → ϕ
14.49 *** → r′
→ R
0.00 *** → x_1
x ⇄ y
14.49 *** → x_2

Rectangular to Spherical

$r = \sqrt{x^2 + y^2 + z^2}$
$\phi = \arctan y/x$
$\theta = \arctan \sqrt{x^2 + y^2}/z$

2.00 ENT↑ → x_2
12.00 → P → x_1
12.17 *** → r′
x ⇄ y
9.46 *** → ϕ
CLX
15.00 → P → x_3
19.31 *** → r
x ⇄ y
39.04 *** → θ



for r = 15
θ = 75
ϕ = 90
the rectangular values are:
x_1 = 0.00
x_2 = 14.49
x_3 = 3.88
(rounded to two places).

Mr. Caldwell hit the jackpot with two routines for this column, this issue. Here is his second contribution:

In navigation problems it is desirable to have solution angles bounded between 000 and 360. It is not uncommon for angles to be bounded by −540 to 720. The following routine performs the desired conversion where the original angle is conditioned only by an upper bound of 720. Programming considerations were: (1) data storage space is critical, and (2) run time is not as important as saving program storage space.

STEP	1	Remarks
001	*LBLA	
002	3	Establish bounding constant.
003	6	
004	0	
005	−	Ensure solution angle is below upper bound of 360.
006	*LBL5	See if solution angle is above lower bound of 000.
007	X>0?	
008	RTN	LST X contains 360.
009	LSTX	
010	+	
011	GTO5	

(Continued)

Now, somewhat in the same vein, here is a nice trick from **Ward W. Miller** of Lagrangeville, New York.

While calculating some celestial navigation problems recently (with DSP 4), I found that 7.3499 hours were converted via g → H.MS to 7.2060 (7h 20m 60s) when I wanted 7.2100 (7h 21m 00s) displayed. Well, the following sequence produces the expected rounded display of hours (or degrees), minutes, and seconds, assuming DSP4 and that hours (or degrees) are originally in decimal forms.

KEY ENTRY	DISPLAY	MEANING
	7.3499	7.3499h
g → H.MS	7.2060	7h 20m 60s
f RND	7.2060	7h 20m 60s
f H←	7.3500	7.3500h
g → H.MS	7.2100	7h 21m 00s

(Thanks, Mr. Miller, especially for the sentence I didn't *print; those were very kind words about KEY NOTES. Ed.)*

Next, let's see what **Dr. Hans Stocklmair** is up to over in Klagenfurt, Australia.

As a contribution to the "25 Words or Less" column, I would like to suggest the following solution:

In many business programs, it proves useful to get results displayed/printed optionally

(a) In currency units;
(b) In thousands of currency units (divided by 10^3); and
(c) In millions of currency units (divided by 10^6).

This can easily be achieved by inserting the three steps RCL 0, 10^x, ÷ at the end of the calculation, with the exponents 0, 3, and 6 alternately set in R_0. All you need, then, is a short and comfortable routine to change the contents of R_0, perpetually, just by pressing a single program key (similar to the usual way of setting/clearing the print flag).

There are many ways to achieve this result; the shortest routines I have found are given below. Version I changes the contents of R_0 in the seqquence 0 → 3 → 6→ 0 every time key "C" is pressed, while version II works in the sequence 0 → 6 → 3 → 0 (which may be of greater comfort in case the 10^6 transformation is more frequently used).

	VERSION I		VERSION II
001	*LBLc	001	*LBLc
002	3	002	6
003	ST+0	003	ST+0
004	9	004	RCL0
005	RCL0	005	X≠Y?
006	X=Y?	006	RTN
007	CLX	007	9
008	STO0	008	ST-0
009	RTN	009	RCL0
		010	RTN

Our next contributor has not only two inputs for this column but also a short article elsewhere! Congratulations to **Harden Schaeffer** of Goldthwaite, Texas.

This routine is for plotting, with 13-increment resolution, a curve on the HP-97 printer. It is called by label 0 and uses three labels. It also uses the I-register and may destroy the contents of the stack.

001	*LBL0	011	DSPi
002	FIX	012	PRTX
003	1	013	RTN
004	0	014	*LBL1
005	X≠Y?	015	CLX
006	GTO1	016	3
007	X≷Y	017	-
008	*LBL2	018	SCI
009	STOI	019	GTO2
010	0	020	R/S

These two routines, used together, permit storing and listing 10 x,y data pairs per card, with each card being numbered automatically. The first x-value is stored in R_9, the first y-value in R_{59} and so on, down to the 10th x-value in R_0 and the 10th y-value in R_{50}. The number of each card in the series is stored in R_A. Registers B thru E are available to store the data. The y-value of each pair is listed first, then the x-value. (To reverse this, move the first P ⇌ S instruction in the listing routine from step 12 to right after step 17.)

TO STORE DATA: Load the storing routine and press [A]. Key in the y-value and press [ENTER↑], then key in the x-value and press [R/S]. Keep entering the data pairs and pressing [R/S] until the display indicates **Crd,** then "write" a data card. Enter more data pairs and keep writing cards until you run out of data pairs . . . or blank cards.

001	*LBLA	013	R/S
002	2	014	STOi
003	0	015	P≷S
004	STOI	016	X≷Y
005	ISZi	017	STOi
006	1	018	P≷S
007	0	019	F0?
008	STOI	020	GTO0
009	*LBL0	021	WDTA
010	CF0	022	GTOA
011	DSZI	023	R/S
012	SF0		

TO LIST DATA: Load the listing routine, read a data card, then press [A]. The calculator will print the number of the card, then 10 data pairs, then pause with a flashing display until you read in another data card, at which time it will repeat the process. Do not press a keyboard key during pauses; this will cause the calculator to list the last card again.

001	*LBLA	011	SF0	021	SPC
002	RCLA	012	P≷S	022	SPC
003	PRTX	013	RCLi	023	*LBL1
004	SPC	014	PRTX	024	CF3
005	1	015	P≷S	025	PSE
006	0	016	RCLi	026	F3?
007	STOI	017	PRTX	027	GTOA
008	*LBL0	018	SPC	028	GTO1
009	CF0	019	F0?	029	R/S
010	DSZI	020	GTO0		

Because the stock market is always big news, here is a neat conversion trick from **Fortson H. Williams** of Cincinnati, Ohio.

For stock market calculations, the following routine (label A) allows entry of the stock prices in eighths, which are then converted to decimals. This can save many keystrokes if there is much of this data to be entered.

001	*LBLA	006	8
002	STO1	007	÷
003	FRC	008	RCL1
004	ST-1	009	+
005	.	010	RTN

Label B allows entry of two stock price quotations in eighths, such as the high and the low, converts them both to decimals, and averages them. This routine is a combined eighths-to-decimals conversion which is much shorter than if each of the two prices had been converted separately by means of the routine of label A.

001	*LBLB	009	.
002	STO1	010	8
003	FRC	011	÷
004	X≷Y	012	RCL1
005	ST+1	013	+
006	FRC	014	2
007	+	015	÷
008	ST-1	016	RTN

Do you remember **Nicomachus**? If so, you must have studied mathematics pretty well. If not, see the note following this letter from **Peter Luschny** of München (Munich), Germany.

Nicomachus vs Euclid. There is a beautiful, simple algorithm—thanks to Nicomachus—for calculating the greatest common divisor (gcd). He stated that the general method is to "continually take the less from the greater until reaching a unit or two equal numbers." This algorithm is not as fast as Euclid's, but is is considerably simpler (no division, no multiplication).

The following eight-step routine uses only one label and no storage registers. The two values for which the gcd is to be found must be in the X and Y stack registers.

001	*LBLA	005	ABS
002	-	006	X≠Y?
003	LSTX	007	GTOA
004	X≷Y	008	R/S

It is easy to extend the routine so that it will also find the least common multiple (lcm). Again, no storage registers are used.

001	*LBLA	010	LSTX
002	ENT↑	011	X≷Y
003	ENT↑	012	ABS
004	R↑	013	X≠Y?
005	x	014	GTO0
006	LSTX	015	R↓
007	R↑	016	÷
008	*LBL0	017	R↑
009	-	018	R/S

The gcd and lcm are in the X and Y stack registers, respectively.

Sometimes it might be useful not to destroy the input. In such cases, use the following routine:

```
001  *LBLA  014    R↓
002   ENT↑  015    R↓
003   ENT↑  016   ENT↑
004    R↑   017    R↓
005   ENT↑  018   X≠Y
006    R↓   019    ×
007  *LBLB  020   LSTX
008    -    021    R↓
009   LSTX  022   X≠Y
010   X≠Y   023    ÷
011   ABS   024   LSTX
012  X≠Y?   025   R/S
013  GTOB
```

After running this routine the stack registers contain:

T = a
Z = b
Y = lcm (a, b)
X = gcd (a, b)

and here is a sample:

```
   40902.00   T
   24140.00   Z
29040420.00   Y
      34.00   X
```

(Nicomachus of Gerasa was a Neo-Pythagorean philosopher and mathematician who wrote Arithmetike eisagoge, *which was translated in 1926 to* Introduction to Arithmetic, *and it was the first work to treat arithmetic as a discipline independent of geometry. Considered a standard authority for 1000 years, it sets out the elementary theory and properties of numbers and contains the earliest known Greek multiplication table. In his book, numbers are no longer denoted by lines, as in Euclid, but are written in the Arabic numerals. General principles are stated with particular numbers taken as illustrations. A Latin translation by Apuleius of Madaurus–c. AD 125–is lost, but Boethius' version survived and was used as a school book up to the Rennaissance. Ed.)*

Let's move now to Mattstetten, Switzerland, for an input from **Hans Marbet.**

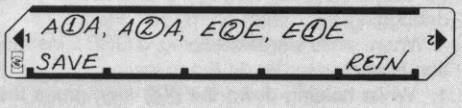
I have noted with interest the "Ersatz Memory" articles and follow-on comments in past issues of KEY NOTES. And I agree with the comments that the method is only applicable *when any program in execution is terminated.* However, I hardly can believe that the battery run-out time allows the user to consult the program description. Thus, I should like to present an alternative and easier-to-handle program which, furthermore, does the same job with less writing/reading of cards.

The 62-step program saves all register and stack contents on *a single magnetic card.*

Save data contents as follows:
1. Load the program.
2. Press [A], enter side 1 of data card, press [A].
3. Press [A], enter side 2 of data card, press [A].
4. Turn off the calculator.

Return the data as follows:
1. Turn on the calculator.
2. Load the program.
3. Press [E], enter side 2 of data card, press [E].
4. Press [E], enter side 1 of data card, press [E].

Caution: Ignore any **Crd** displays that might occur. Just follow the straightforward instructions above, pressing the [A] or [E] keys only when any current section of the program is terminated. The [A] or [E] keystrokes before and after each card-entering can easily be remembered by labeling the program card as follows:




```
001  *LBLA  022   RCL4  043   P≠S
002   WDTA  023    +    044   STO0
003   R/S   024   RCL5  045   P≠S
004  *LBLA  025    +    046   R/S
005   STOA  026   RCL6  047  *LBLE
006    R↓   027    +    048   P≠S
007   STOB  028   RCL7  049   RCL0
008    R↓   029    +    050   ST-0
009   STOC  030   RCL8  051   X=0?
010    R↓   031    +    052   R/S
011   STOD  032   RCL9  053  *LBLE
012   LSTX  033    +    054    0
013   STOE  034   X=0?  055   RCLE
014   P≠S   035   R/S   056    +
015   RCL0  036  *LBLA  057   RCLD
016   RCL1  037   WDTA  058   RCLC
017    +    038   R/S   059   RCLB
018   RCL2  039  *LBLA  060   RCLA
019    +    040   R/S   061   R/S
020   RCL3  041  *LBLE  062  *LBLE
021    +    042    2    063   R/S
```

(Since KEY NOTES is distributed in Europe at least one month later than in the U.S., Mr. Marbet did not see James P. H. Hirst's solution to "Ersatz Memory" in Vol. 3 No. 1 before he wrote the above program. Nonetheless, I thought it worthwhile to add Mr. Marbet's contribution to the "Ersatz Memory" puzzle. Ed.)

Use the Right Paper!

Lately, some of the HP-97 (and HP-91 and HP-92) calculators that have been sent back to our Service Department for repair have had the wrong thermal paper in them. Please remember that **HP thermal paper is the only approved paper for your calculator.** (Refer to page 285 in the *HP-97 Owner's Handbook.*) **Other types of thermal paper might work in your HP calculator, but those papers will cause premature printhead failure, and possibly cause repairs at your expense.**

Editorial

That *ENTER greater than EQUALS* T-shirt on **Gary M. Tenzer** on the cover of the last issue caused quite a few inquiries. So I tried to find the original source. The shirt was originated by **Jim Virgin** of the Stanford University Bookstore and was made by Shoreline Sportswear of Santa Barbara, California. However, I am sorry to report that they are no longer available.

I received a letter last month from **Robert Schild, Jr.** of Dover AFB, Delaware, wherein he suggested we add the KEY NOTES issue numbers at the bottom edge of pages so you can more easily locate specific material when paging through bound copies of the newsletter. So you can thank him for persuading me to do it.

Mr. Schild also pointed out a small error that not many people saw in the last issue. Refer to **Neal Neuburger's** routine near the bottom of the first column on page 10 of Vol. 3 No. 1. When running the example, "52" should be added to the third line, after ENTER.

Mr. Schild and many others have asked why we no longer punch holes in KEY NOTES so that it can be inserted in ring binders. Well, as the newsletter has grown and grown many times, it has become necessary to print it by more economical means. That is why you do not see staples in it and three holes drilled in the binding edge. We presently run the newsletter on a very large web press that does the entire process in one pass, and to drill holes as a separate, manual, process would cause a heavy additional expense. Instead of doing that, we put the money into extra pages for you, and we feel sure you'd rather have more information than three small holes in each copy.

If you've written to me and haven't received an answer, don't give up hope! Mail has been very heavy lately, and the situations that delayed this issue have put an unusual strain on your editor's available time. 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

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, as a final word, don't forget to notify us of address changes. You won't be able to read your KEY NOTES if we don't know where to send it!

Wood You Help a Forester?

You all know that coal and oil are not renewable energy resources. But wood is. So we decided to print the following letter and see what our far-flung audience can do to help —maybe even from other areas of the world.

Gentlemen:

I am a forester with an HP-97 and am looking for forestry programs; that is, in addition to those listed in the HP-67/97 Users' Library. It seems, also, that most of your past contributors are westerners, and I hope to find some southern "67/97" foresters.

Please publish this letter and my address in the next KEY NOTES. This may lead to a greatly enriched forestry section of the Library.

Sincerely,

Colin Bagwell, R.F./A.C.F.

1601 Sun Valley Road, Huntsville, AL 35801

(Glad to help, Mr. Bagwell. Let us know what response you get from our readers. Ed.)

When $2.00 Equals $20.00!

In appendix B of *Catalog Addendum 3*, dated November 1978, we had a significant typographical error. The President wants us to keep prices down, but *this* price was ridiculous!

Thanks to **Robert Schild, Jr.** (Dover AFB, Delaware), who notified us of the mistake, we can inform you about it. On the page of "Supplies & Accessories for Your Programming Needs," the 00097-13141 40-Card Pac with Holder should be $20.00* instead of $2.00

On that same page, mark these other corrections:

1. On the first line, mark "(HP-67)" after "DC Recharger/Adapter."
2. The correct model number for the HP-97 110 Vac Recharger/Adapter should be 82059A (not 82058A).
3. On the fourth to last line, the correct number for "3 Program Card Holders" is 00097-13142 (not -13143).

Also, although it is not stated on the page, the asterisk after the model number indicates the supply or accessory can be used for the HP-65.

** U.S. dollars. See not at bottom edge of cover.*

"SST-ing" Thru Subroutines

Here is a rather clever tip you might not have thought of just because it never occurred to you. The following letter is from **Harden Schaeffer** of Goldthwaite, Texas.

This small item might be of interest to readers of KEY NOTES. It is a technique I discovered for single-stepping through a subroutine while debugging a program on my HP-97.

When, while single-stepping, a GSB x instruction is encountered, do the following:

1. While holding down the SST key, press the GTO key (or any other key in the same row).
2. While holding down the GTO key, release the SST key.
3. Release the GTO key.
4. Press GTO x.
5. The calculator will now single-step through the subroutine and will return to the proper place when it encounters a RTN instruction.

I don't know for sure if this will work with the HP-67, since its keyboard is electrically very different from the wiring of the HP-97's keyboard. And while the above procedure appears rather complicated "on paper," it's really quite simple to do in actual practice.

This procedure was developed during the course of some very helpful and enjoyable correspondence with one of the people in your Service Department.

Better Late Than Never!

If your KEY NOTES arrived late, we apologize. Because of recent strikes and some other setbacks, there was a delay in both printing and distribution. However, we know you enjoy getting the newsletter regularly, so we want to assure you that the August issue *will* be on schedule again.

Base Conversion Routine

Although a Library program will do the same thing the following routine will do, we think this contribution from **Cass R. Lewart** (Holmdel, New Jersey) is well worth printing in KEY NOTES.

Here is my program that converts an integer argument from *any* base to *any* base in only 49 steps. I saw a description of a similar program in the Users' Library (00409D) but it required 135 steps. I think that many KEY NOTES readers will be interested in this one. For example:

$(013\ 015\ 199)_{215} = (?)_{16}$

Argument ENTER↑ Old base ENTER↑ New base A.

13015199 ENTER↑ 215 ENTER↑ 16 A.

Answer: 90308113 = 938BD hex.

001	*LBLA	026	ENT↑
002	STO0	027	INT
003	R↓	028	STO3
004	ENT↑	029	−
005	GSB3	030	RCL1
006	GSB0	031	×
007	RCL0	032	RCL5
008	GSB3	033	×
009	RCL0	034	ST+4
010	*LBL0	035	RCL2
011	STO1	036	ST×5
012	R↓	037	GTO1
013	STO2	038	*LBL2
014	R↓	039	RCL4
015	STO3	040	RTN
016	0	041	*LBL3
017	STO4	042	1
018	1	043	−
019	STO5	044	LOG
020	*LBL1	045	INT
021	RCL3	046	1
022	X=0?	047	+
023	GTO2	048	10ˣ
024	RCL1	049	RTN
025	÷		

HP KEY NOTES

May 1979 Vol. 3 No. 2

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

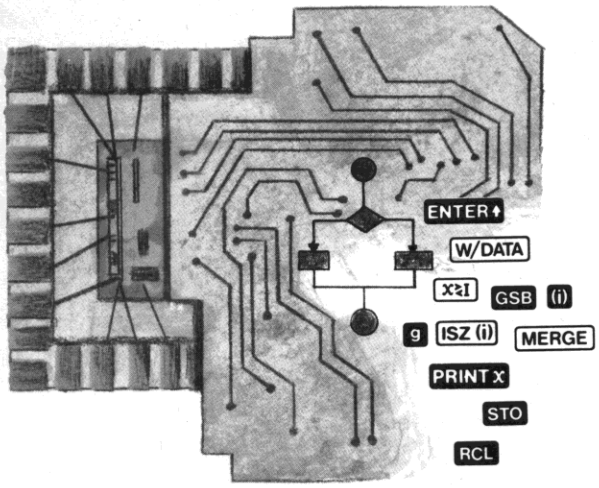
Hewlett-Packard SA
USERS' CLUB EUROPE
7, Rue du Bois-du Lan
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

Address Correction Requested
Return Postage Guaranteed

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

0001298918
M R DEZEARN
1310 G EVERETTE
ABERDEEN PR GRD MD 21005

Printed in U.S.A.



Featuring, this issue:
The HP-41C Defined 2
A User's View of the HP-41C 3
The Designer's View 5
HP-67/97 Compatibility 6
Software for the HP-41C 6
Library Corner 7
Another Excellent "Calculator" Book 8
Random Numbers, Means, Regressions etc. 9
About Batteries 10
More About Merging 10
"25 Words" (More or Less!) 11

August 1979 Vol. 3 No. 3

HEWLETT hp PACKARD

HP Key Notes

HP Creates A Whole New Standard . . .

On July 16, 1979, Hewlett-Packard unveiled *a whole new standard* in handheld personal calculation: The HP-41C Alphanumeric Programmable Calculator. Not *just a calculator,* however, it is *a complete system*! And here, and on following pages, we are proud to present this wonderful, powerful personal programmable and its plug-in peripherals.

As you have already guessed—or already know—there are quite a few "firsts" on this HP calculator. It has a liquid-crystal display (LCD). It has full alphanumeric capability. It has a completely redefinable keyboard. It has four "ports" at the top end for plug-in peripherals, Application Modules, and Memory Modules. And it is the first HP fully programmable calculator with Continuous Memory.

Yes, the HP-41C is *fully* programmable, with incredible power that can expand to over 2000 lines of program memory. Or 319 registers for data storage. Or any mix of data storage and program memory *you* choose. But with all this incredible sophistication, the HP-41C is also remarkably easy to use. It communicates in friendly English, so operation is simple even for a novice. There are even status annunciators to remind you of operating modes and battery life. And so you always know what's going on, you can *name* a program, then call it up *by name* for execution. How's *that* for convenience!

In addition, with only the lightest touch of a switch, the HP-41C can switch from a predetermined calculator to a user-customized instrument, complete with "specialized" keyboard, thus allowing you to create a highly personalized calculator, precisely matched to *your* own special applications.

Plug in the "extra smart" Card Reader and you can easily store your own personal program library on magnetic cards that can be read again anytime you wish—even in any other HP-41C anywhere in the world. Yet, you can also create "private" programs, *for only your use*! And here's real music to your ears: *HP-67/97 recorded cards are compatible!*

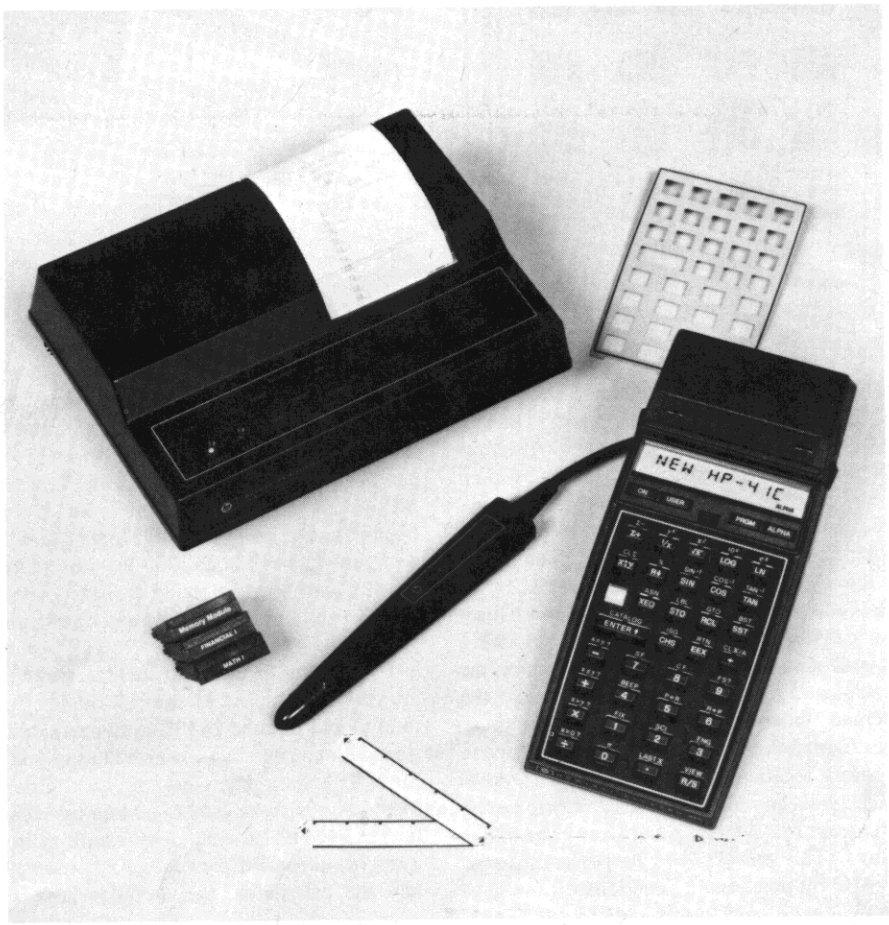
Then, plug in the Printer, which is a complete alphanumeric, plotting printer with three operational modes for documenting your calculations and program listings and for generating hard-copy and graphic outputs. Or, plug in both the Printer *and* the Card Reader and use them in support of each other!

And, lastly, there's a plug-in Wand that lets you conveniently load long programs by reading bar codes printed on paper. (But, alas, you won't be able to order *that* innovation until early 1980.)

But we saved the best part for the end: The HP-41C uses the straightforward keystroke programming and time-proven RPN logic system that lets you slice with ease through the most difficult problems.

Now, is it any wonder that we call this magnificient calculator—this powerful system—

A Whole New Standard?



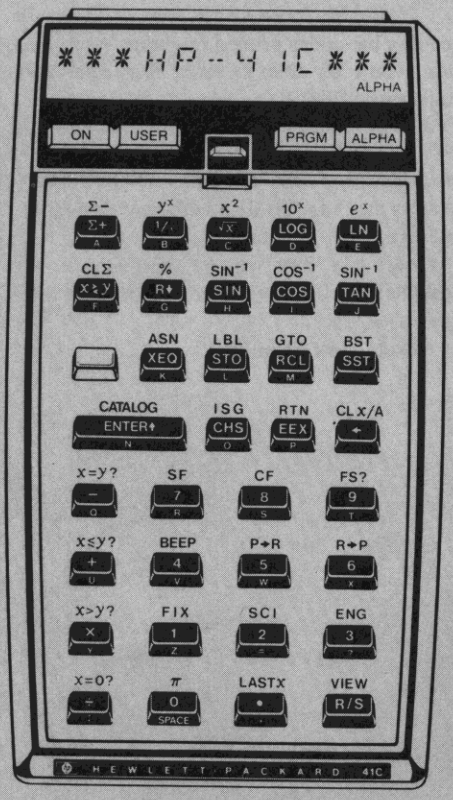
All prices in this newsletter are suggested retail prices excluding applicable state and local taxes–Continental U.S.A., Alaska and Hawaii

The HP-41C Defined

As we told you on the cover, the HP-41C is a system. But an extremely powerful, yet friendly system. It will do a whole lot more than is obvious at first glance. However, it would take a book to describe *all* of its functions and capabilities, so we'll give you an overview of each part of this magnificent new system.

THE CALCULATOR

By itself, the HP-41C can accomplish an astonishing array of calculations. The 39 keys on the face of the calculator control an amazing number of functions: in all, 130. Yet, there is only *one* shift key! The message, here, is very clear: keep it simple and friendly and easy to use.



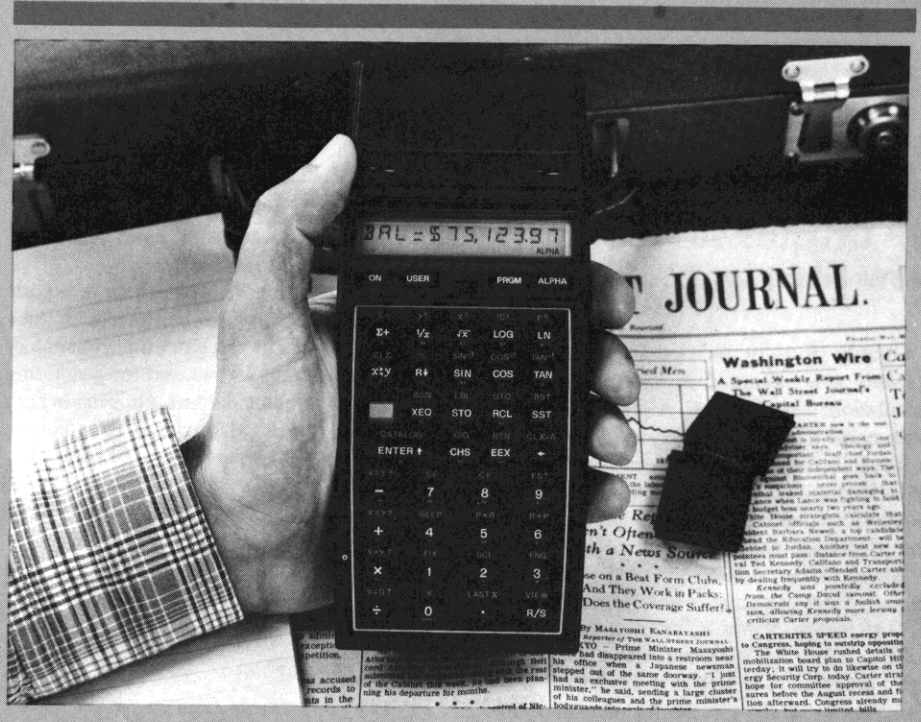
By utilizing its full alphanumeric capability, the HP-41C communicates in understandable messages to make it easy to use. It prompts you for inputs when required. It gives clear direction when calculating errors are encountered. And it keeps you alerted to which operating modes are being used. It even has an audible "beeper" to signal key events.

Because we know that not everyone uses their calculator the same way, we included a versatile new feature. Nearly any function the calculator can perform—those built into the machine, those programmed into it, or those from Application Modules—can be assigned to most key locations on the keyboard. In other words, you can redefine the EEX key to be the LBL key, or nearly whatever your heart desires. This enables you to personalize the HP-41C by positioning functions on the keyboard where they are most convenient. And, in addition, two keyboard overlays and a set of labels enable you to change the keyboard markings to keep track of your "custom" setup.

With its liquid-crystal display using very tiny amounts of power, the batteries in the HP-41C should normally last about 9 to 12 months, depending on how much the calculator is used.* Also, the four size N alkaline batteries used in the HP-41C are commercially obtainable. (They are *not* rechargeable.)

The HP-41C package includes:
- An HP-41C Alphanumeric Programmable Calculator.
- An HP 82111A Soft Case.
- An HP-41C *Owner's Handbook and Programming Guide*.
- An HP-41C *Standard Applications Handbook*.
- An HP 82152A Overlay Kit.
- An HP 82151A Module Holder.
- Four Size "N" Alkaline Batteries.

and the price of this package is $295.*



PLUG-IN MODULES

By itself, the HP-41C has a basic (built-in) program memory of approximately 400 lines or 63 data storage registers; it all depends on how you want to allocate it. But Memory Modules are a convenient way to dramatically expand the calculator's memory capacity. Each module contains 64 data registers or up to approximately 400 lines of program instructions, for any combination of data storage and program memory selected by the user. (Each Memory Module costs $45.*) If all four of the HP-41C's ports are filled with Memory Modules, the total available memory is 319 data registers or about 2000 lines of memory. As you can see, we made it highly versatile so you can define just about any configuration you desire.

If you plug in the Card Reader, three ports are still available for Memory Modules, so the total would then be 255 data registers or 1600 lines of memory. Add the Printer, and you still have 191 data registers or 1200 lines of memory, even with only two Memory Modules in the HP-41C. Any way you look at it, it's a mighty powerful package. And, remember, like the calculator, the modules have Continuous Memory. As long as it's plugged into the HP-41C, a module maintains data and program information *even when the calculator is turned off.*

The HP-41C's large memory capacity creates a two-fold benefit: You are now able to maintain *more* and *longer* programs than previously possible in an HP handheld calculator. But an even better asset is the fact that *each program is autonomous;* it can be called up by name, run, edited, recorded, or erased *without affecting other programs in the calculator.*

And that's not the only software capability of this little giant. Sixteen Application Modules have been or are being developed for use with the HP-41C and more are planned for later introduction. Each Application Module contains approximately 4000 or 8000 program lines and plugs into one of the HP-41C's ports, and has its own handbook and keyboard overlays. Each Application Pac costs $45.* These solid-state software modules are *permanently recorded.*

Also available is a series of 25 *Solutions Books* that contain step-by-step keystroke listings for manually entering programs. Each book costs $12.50.*

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

CARD READER

The HP-41C Card Reader is similar to the reader in the HP-67 except that, instead of being only smart, it is **extra smart!** In fact, it can read and translate HP-67/97 cards (program or data) for execution in the HP-41C. (More about that in a following article.)

In addition, cards can be read in any order because the HP-41C notifies you—in the display—which track of the card is still required. And, because so many of you have asked for it, the new Card Reader provides "program security." Through user commands, the Card Reader can be instructed to record a card so that the program on that card can only be executed and not viewed or altered. How's that for being "extra smart"!

The Card Reader is powered by the batteries in the HP-41C. Although it uses very little power—and the HP-41C even less—it does lessen the battery life for the calculator, depending on how often the Card Reader is used. However, even with fairly heavy use, battery life will still be counted in months, not in hours, as it has been in the past.*

The HP-41C Card Reader package includes:
- An HP 82104A Card Reader.
- An *Owner's Handbook*.
- A Head Cleaning Card.
- 20 Blank Magnetic Cards.
- A Card Holder for Magnetic Cards.

and the price of this package is $195.*

THE PRINTER

The HP-41C Printer is a portable, quiet, thermal printer. It prints numbers, upper- and lower-case letters, special characters, and incorporates easy-to-use plotting routines. Like the HP-97, its three modes are *Normal, Manual,* and the popular *Trace*. It will even print double-wide characters, for highlighting or what-have-you. And again because you asked for it, it has a five-position print intensity switch to make your printouts as light or as dark as *you* want them to be. It runs on rechargeable batteries (same pack as the HP-97) or ordinary house current, and will print about 4000 to 5000 lines on one charge. Also, it uses the same paper used in the HP-97.

There are a total of 127 standard characters that can be printed on the HP-41C Printer. You will not quickly run out of letters, symbols, signs, Greek letters, and so on with such a choice. But, if you want your own symbols of any kind, all you have to do is build your own, from the 7×7 matrix of dots that makes up each character-space across the 24-character-wide field available. This means you can create 2^{49} *different patterns* with the HP-41C Printer. And if your calculator isn't handy, that's over 560 trillion patterns! As you can see, with the HP-41C Printer, your imagination, not the machine, is the limiting factor.

The HP-41C Printer package includes:
- An HP 82143A Printer.
- An AC Adapter/Recharger.
- A Rechargeable Battery Pack.
- An *Owner's Handbook*.
- Two Rolls of Thermal Paper.

and the price of this package is $350.*

THE WAND

The HP-41C Wand, which will be available in early 1980, makes the HP-41C the first calculator to accept programs and data from printed bar codes. You simply pass the Wand over a printed bar code and in seconds it translates these codes into information that is loaded into the calculator.

After the Wand is available, the *HP-41C Solutions Books* and Users' Library programs will have printed bar codes to facilitate easy program entry.

THE SYSTEM

As you can see, this remarkable machine and its amazing peripherals form a system that is almost indescribably powerful and versatile. Yet it is friendly and easy to use. This system can *adapt* to meet changing requirements, and it can *grow* as your needs increase.

A WHOLE NEW STANDARD

No matter what can be said about the HP-41C, above all else, it has the programming power to solve the most complex problems. *Even programs written for 8-kilobyte computers have been translated into the HP-41C!*

We could go on for pages and pages about this new standard of the calculator world. You simply have to see it and experience it to realize what a tremendous contribution it is. It's been a long time coming, but we know you will agree that the HP-41C was worth the wait. So go see your nearest HP dealer and see the HP-41C; we'll bet that you won't be able to resist it!

And to give you some *other* viewpoints, the following HP-41C articles include one by a user, one by one of our engineers, and one about HP-67/97 compatibility.

**See article "About Batteries."*

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

A User's View of The HP-41C

As many of you know, **Richard Nelson** was an early HP-65 User. He founded PPC* in June 1974 (then called the HP-65 Users Club) and is Editor of the *PPC Journal,* a monthly publication of PPC. When not devoting most of his free time to PPC he works as a free-lance writer/consultant and Applications Engineer for Statek Corporation in Santa Ana, California. He has written numerous calculator articles for such magazines as *Byte* and *Electronics* and has published eight papers on calculator applications.

Some time before announcement of the new HP-41C, we furnished Mr. Nelson with a prototype machine so he could become familiar with it before its introduction. We also thought that it would be a good idea for you to read the assessment of a skilled, long-time calculator user *before* you had a chance to see the HP-41C. In the following article are Mr. Nelson's thoughts and comments about the HP-41C. We hope you enjoy this "preview."

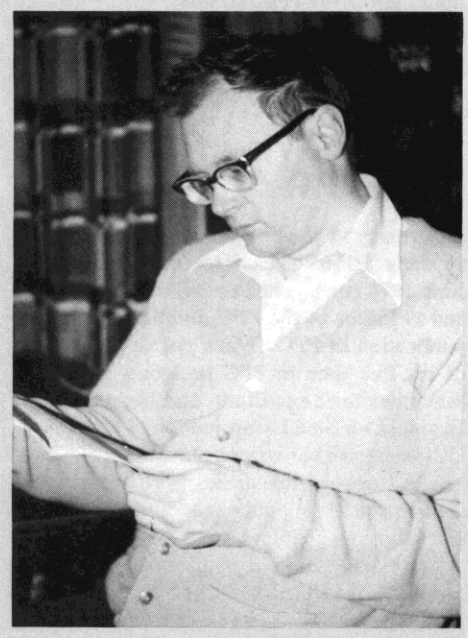
The recently introduced HP-41C calculator is a machine of great depth. The capabilities that exist in this programmable handheld calculator are so numerous that it would take reams of paper to describe *all* of them! Yet, the most significant characteristics are the friendliness and beautiful simplicity that this second-generation machine offers to the user—any user, from the person who doesn't program to the person who enjoys developing his own software.

Compared to the HP-65 and HP-67, the HP-41C is smaller, lighter, faster, and has a

simpler, less-cluttered keyboard. The machine has a Series E look. All keys are black with white lettering on the top except the single, unmarked, gold prefix key. Above each marked key is a function printed in gold. A blue letter or symbol is on the front sloping surface of all but four keys.

Instead of the two slide switches used on the HP-67, there are four "mode" keys below the display. These are easier to use and provide more functions.

Even with so many new features, the user with previous HP calculator experience will find the HP-41C easy to use. Major new features are the alphanumeric liquid-crystal display and the four plug-in ports at the display end of the machine. These vividly point out that the HP-41C is the heart of a calculator *system,* so there is no built-in card reader on the basic HP-41C. The system includes ROM (read-only memory) modules for plug-in HP software, RAM (random-access memory) modules to add program and data memory, and peripherals that presently include a card reader, a printer, and a wand that reads bar codes. And with all of that, the HP-41C also has Continuous Memory that retains all programs, data, and key assignments—similar to other HP programmables such as the HP-29C.



Richard J. Nelson

The most exciting feature that makes the HP-41C a second-generation machine is its alpha capability. With this capability, there are no key codes to remember, you may label your programs with words and symbols in any language, program prompts and instructions become part of the program, and a computed answer may be displayed or printed on the same line as the description of the answer. The HP-41C processes alpha data much like it does numeric data. Words may be stored, printed, and even compared.

The alpha capability opens a new vista of applications that has the potential to make the HP-41C into a personal information center. Names, telephone numbers, price lists, bank balances, and even a simple set of books are possible. Memory size is limited because the HP-41C is not a data-processing handheld computer, but it *is* possible to store 32 names and telephone numbers in one Memory Module. Word games are practical, and the famous game Hangman is part of the *Standard Applications* handbook. However, alphabetical sorting is not practical on the HP-41C unless the required time to compare each data, character by character, is tolerable.

The computing power of the HP-41C is many times greater than the HP-67 or HP-65. HP-67 users often dream of more than one "I" register for indirect addressing. ALL registers of the HP-41C can be designated as "I" registers, and that includes the stack and last-X registers. Indirectly addressable instructions include just about every one it would make sense to address indirectly, including: store, recall, register arithmetic, display setting, tone, flags, position of statistical registers, catalogs, x-exchange any register, and many more.

The ISZ and DSZ counters of the HP-67 are greatly improved in the HP-41C. They are now called ISG for "increment and skip if greater" and DSE for "decrement and skip if equal." The control number in the designated register has three parts. The counter starting value, the branching value, and the incrementing value. Any register may be an ISG or DSE register, and the ISG/DSE function may be indirectly addressed. Full indirect addressing, full counter control, and six levels of subroutines provide the HP-41C programmer with unparalleled computing capability. The familiar scientific functions are included, along with a few new ones: modulo, LN1+x, sign, four flag tests plus set and clear operations, and an astonishing set of flags, just to mention a few.

The memory capacity of the HP-41C may be increased with up to four, 64-register, Memory Modules. The HP-67 has 26 data registers plus 32 memory registers of 7 steps each, for a total of 58 registers. The HP-41C has 63 registers and four bytes. (A byte is 1/7 of a register.) All HP-67 instructions are one byte long, but HP-41C instructions are one byte, two bytes, three bytes or even longer.

Program steps relate to discrete operations. On the HP-67 the number 1234567890 requires 10 steps (10 bytes). On the HP-41C, 1234567890 also takes 10 bytes, *but only one program line.* An HP-41C program line may hold from one to 16 bytes. Because of this, it is difficult to compare program lengths on the basis of steps and lines.

One of the features that makes the HP-41C simple to use is the USER mode. User programs, ROM programs, peripheral functions, and HP-41C functions may be assigned to any keyboard key except the gold key and the "correction" (⊟) key. The press of a single toggled USER key converts the keyboard into a set of custom functions that are labeled with supplied keyboard overlays. Thus, the HP-41C may be made into a custom machine that looks familiar to the user because it is identified with his terms in the display and on the keyboard. Non-technical users will appreciate the HP-41C as a friend and not some scientific or highly technical gadget.

A Catalog function displays user program names, ROM function names, and peripheral function names. CAT 1 displays user-written program names, CAT 2 displays peripheral and ROM program names, and CAT 3 the 130 machine function names. R/S stops the rapid display of catalog content, and SST or BST allows you to step to a desired function. *(Pressing and holding any key other than R/S slows the listing so you can follow it easier. Ed.)* If the function is one of your programs, you can execute or edit it because the program pointer is at the beginning of the program when its name is displayed by CAT 1.

CAT 3

+	ENTER↑	RCL
-	E↑X	RDN
*	E↑X-1	RND
/	FACT	RTN
1/X	FC?	SDEV
10↑X	FC?C	SCI
ABS	FIX	SF
ACOS	FRC	Σ+
ADV	FS?	Σ-
AOFF	FS?C	ΣREG
AON	GRAD	SIN
ARCL	GTO	SIGN
ASHF	HMS	SIZE
ASIN	HMS+	SQRT
ASN	HMS-	SST
ASTO	HR	ST+
ATAN	INT	ST-
AVIEW	ISG	ST*
BEEP	LASTX	ST/
BST	LBL	STO
CAT	LN	STOP
CF	LN1+X	TAN
CHS	LOG	TONE
CLA	MEAN	VIEW
CLD	MOD	X=0?
CLP	OCT	X≠0?
CLRG	OFF	X<0?
CLΣ	ON	X<=0?
CLST	P-R	X>0?
CLX	PACK	X=Y?
COPY	%	X≠Y?
COS	%CH	X<Y?
D-R	PI	X<=Y?
DEC	PROMPT	X>Y?
DEG	PSE	X<>
DEL	R↑	X<>Y
DSE	R-D	XEQ
END	R-P	X↑2
ENG	RAD	Y↑X

The card reader that attaches to the HP-41C is quieter, and "smarter" than the HP-67 card reader. The HP-41C card reader contains its own functions and has the ability to read and translate HP-67/97 programs to HP-41C programs—and translated programs run faster. A verify function can assist the user in sorting out magnetic cards by reading the card and displaying its type—HP-67 data or program, HP-41C data, program, status, or write-all. A "private" record feature writes a program and key assignment card that may be used but not viewed, altered, or copied.

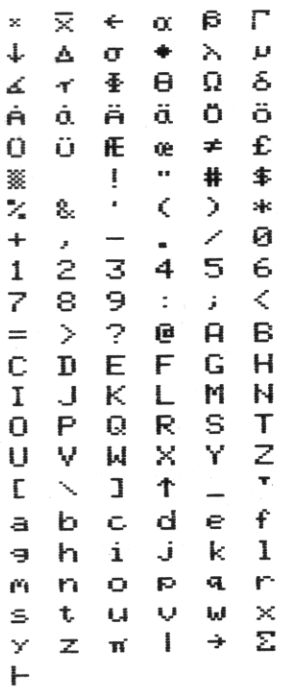
(Continued)

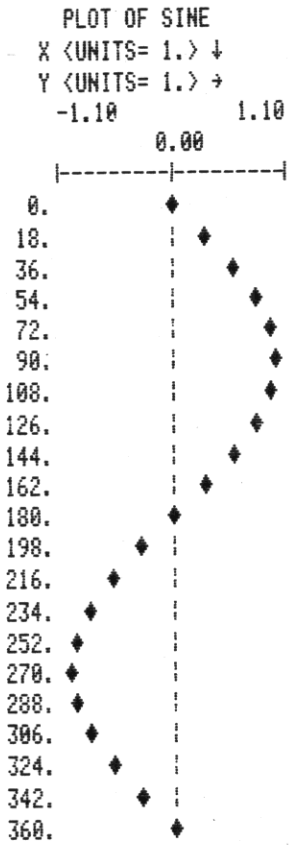
Other features make the HP-41C a pleasure to own and use. The primary (non-rechargeable) batteries give a new freedom of use. Six to nine months of usage without a battery replacement makes life much simpler and, considering battery life and recharger costs, quite cost-effective.* With your programs and data *always* in the machine, the HP-41C will always be ready when you are.

Many other features make the HP-41C user-oriented. Ten tones can be programmed and a four-tone combo is called BEEP. Using the extensive control flags of the HP-41C allows: programs that automatically start when the machine is turned on, recording on a clipped-corner card if desired, ignoring range or system errors, disabling the tone function to speed up program execution if desired, printing lower-case or double-wide on the printer, detecting alpha or numeric data entry, and even adjusting the digit separators to periods if desired. And there are 56 flags on the HP-41C; the first five are indicated as being set in the display, along with six other annunciators.

The HP-41C printer is the most impressive of the three peripherals. It uses the same battery and paper as the HP-97 and has a similar moving print-head, but that is where the similarity ends. The HP-41C printer is belt-driven to be whisper quiet. Printing a program listing takes less than one second per line, and the print character set has 127 upper- and lower-case letters, numbers, and special symbols. Four direction arrows, reverse slash, ampersand, brackets, and even a dotted square are only a few examples of the special characters. The graphics capability of the printer even allows the user to make up his own character set, compose figures, and even a company logo. The 24-character line is composed of a 7 × 7 dot matrix for each character. This gives a resolution of 112 "dots" per inch across the paper.

STANDARD CHARACTER SET:





Plotting is a function that is also part of the HP-41C printer. The PRPLOT function plots in an 18-character field, with a resolution of 124 positions. It also plots an axis, upper and lower limit marks, and proper scales. An annotated program listing of PRPLOT given in the HP-41C printer handbook allows the user to modify the program to tailor it to his own needs. Multi-strip plotting is another approach that increases the plotting capability of the printer.

The printer also provides all the normal "housekeeping" functions of program listing, flag status listing, user assigned keys, and catalog listing. The single flag control of double-wide, upper- or lower-case, and the combination of the two makes the printer easy to use and a valuable addition to the basic HP-41C.

The bar code reading wand will make low-cost software in printed form easy to load into the machine. The number of instructions per page is fewer for bar codes than for printer listings. From a user's viewpoint, the wand also provides the possibility of an isolated data input port from another device.

The HP-41C is an extremely sophisticated and yet simple machine to use. Calculator users are accustomed to single-button execution of functions and programs. The ability to address and execute thousands of programs on a hand-held, 39-key keyboard must have presented a challenge to the HP-41C designers. In larger computer systems the overall control of the computer is managed by the operating system. For the first time, calculator users will sense a mini-operating system in their programmable calculator that takes care of a lot of details. This frees the user to concentrate on his solutions and not the problems. The interfacing of the HP-41C to its peripherals, the carefully thought out methods of logical keystroke sequences, and the commitment of system support made by Hewlett-Packard will maintain the HP-41C as the best personal calculator system for years to come.

*Personal Programmers Club. For more information about this Club, send a self-addressed large (9″×12″) envelope with first-class postage for 2 ounces to:
PPC JOURNAL (Dept. KN)
2541 W. Camden Place
Santa Ana, CA 92704 U.S.A.
and you will receive a sample issue of the PPC JOURNAL and other information. PPC is not sponsored, nor in any way sanctioned, by Hewlett-Packard.
*See article "About Batteries."

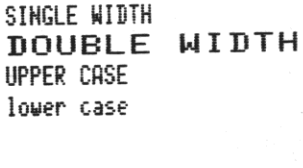
The Designer's View

To present yet another facet of the new calculator, we asked one of our HP-41C design engineers in Research and Development to write an article on *his view* of this new standard in calculators.

MEET THE HP-41C

Since most of you are familiar with the HP-65, HP-67 and/or HP-97, the thrust of this article will focus on the major points of distinction between the products.

One of the first things you will notice about the HP-41C is the comparatively uncluttered keyboard. There is a single shift (gold) key and there are only two functions per key. The front slope of each key contains an alphabetic or special character for use in ALPHA mode. Yet the HP-41C has *more* functions than the 67/97. We just don't show all of them on the keyboard. Of the 130 or so functions in the HP-41C, only 58 are actually printed on the keyboard. The others are "buried" in the machine. But we have provided a way for you to "resurrect" any of these functions and assign it to any key or shifted-key location of your choosing. Then when you switch the machine to USER mode, the calculator no longer behaves like we've programmed it, but like *you've* programmed it.



In fact, you can extend the concept of USER mode to include not only HP-41C functions, but also functions added by the plug-in Application Modules and peripherals as well as "functions" you program yourself in user memory. As you add a peripheral such as the HP 82143A Printer, the complete catalog of printer functions is available to you for execution or key reassignment. Or, you can write your own program, give it a name, and assign it to a key.

So that you don't forget how your keyboard is configured, we provide removable overlays which you can use to relabel the keys. Also, if you press a key and hold it down, the alphanumeric liquid-crystal display (LCD) will show the assigned function mnemonic (yours or ours, whichever one will execute) in the display. Releasing the key will initiate the function execution. And, since the HP-41C has Continuous Memory, it remembers everything, key-

assignments included, when you turn it off. In fact, because of Continuous Memory and the LCD, it is really not necessary to turn the unit OFF to save power! (If you let it sit for approximately 10 minutes without exercising a function, it will shut itself off.)

There are several uses of the alphanumeric display in addition to the function-prompting capability. First there is the ability to display function mnemonics rather than numeric key codes when in PROGRAM mode, to aid you in program development and debugging. Also, the "alpha register" concept enables you to construct alpha strings that can be combined with the contents of numeric registers to provide alpha or alphanumeric messages for input prompting ("KEY IN SPEED"), or output labeling ("RATE = 23.72%").

The ability to name programs or subroutines with alpha strings and have that name appear in the display is an aid not only for key reassignments as mentioned above but also for program development: XEQ (execute) "ROOT", GTO "COST", etc.

Error messages (and others) are now more decipherable. We now have a useful meaning to a conditional executed from the keyboard. Press FS?23 (is flag 23 set?) from the keyboard and the HP-41C display will respond YES or NO.

Another exciting new capability that the HP-41C provides is the ability to configure your system with the addition of plug-in modules and peripherals. Additional user memory, plug-in applications software, a Printer, and an "extra-smart" Card Reader are all available for the HP-41C, with more devices and capabilities on the way. And when you plug in the optional Card Reader the HP-41C will behave like an HP-67 if you feed it an HP-67 card, or like an HP-97 if you have the optional Printer plugged in as well.

These are only a few of the features of the HP-41C. We could go on and on for pages about this new and very powerful system. So, for the complete story, visit your HP dealer and experience the HP-41C for yourself.

HP-67/97 Compatibility

By now you are wondering how you will find the time to translate your library of HP-67/97 programs for your new HP-41C. That's easy! Just feed your HP-67/97 cards into your HP-41C Card Reader, wait a few seconds, and presto, you have an equivalent HP-41C program, ready to run. Data cards will work too!

To use HP-67/97 programs or data recorded on magnetic cards, you need:

1. An HP-41C.
2. An HP 82104A Card Reader.
3. One HP 82106A Memory Module.

With this HP-41C system you will be able to use about 97% of all HP-67/97 program cards and all data cards.

The following account for most incompatibilities:

1. Programs using unsupported HP-67/97 capabilities usually will not operate (e.g., pseudo-alpha displays generated by use of non-normalized numbers).
2. The "Rapid Reverse Branch" (GTOi or GSBi when the content of I is negative) of the HP-67/97 is not supported.
3. Multi-step digit entry strings immediately following conditional tests do not execute properly. For example, the HP-67/97 steps:

   x=y
   1
   2

   will not be translated properly because the HP-41C will construct a single program line with the two digits and skip (if false) the entire digit string. The HP-67 and HP-97 would skip (if false) the individual digit 1 and place a 2 in the X-register.
4. The HP-67/97 "Merge" function is not supported.

These, and other minor incompatibilities, are described in detail in the *HP 82104A Card Reader Owner's Handbook*.

The First HP-41C Program!

Before you take us to task on that title, let us clarify it. We are referring to the first HP-41C program *contributed by a customer and accepted into the Users' Library*.

The author of this program, **Charles I. Dinsmore, P.E.,** has been an active contributor to the HP-67/97 Users' Library. His contributions include excellent programs for structural steel and concrete analysis. In fact, his HP-67 submittals were so well done that we decided to ask for his help for the *Structural Analysis Pac* for the HP-41C. His contributions in frame analysis and concrete design helped make the pac a powerful tool for the structural engineer.

Mr. Dinsmore credits a good deal of his professional achievement to his first programmable calculator, an HP-25. It got him started in programming and motivated him to investigate topics he might otherwise have passed over. However, he soon found the need for more capacity, so he purchased an HP-67. And, now, his new HP-41C should help him to continue to grow within his profession.

Mr. Dinsmore currently lives in Seattle, Washington, with his wife, Rita, and their son, Michael. He is employed by Victor O. Gray & Company, Consulting Engineers.

Mr. Dinsmore's HP-41C program encompasses 26 pages of first-class documentation and 6 magnetic cards. The title is **Column Solver #00270C,** and the cost is $6.* This program does an extremely complete analysis of steel W-shaped, I-shaped, or H-shaped columns, according to the 1978 A.I.S.C. specifications for structural steel buildings. Because of the structure and nature of this program, three Memory Modules must be used in the HP-41C to use this program.

Software For The HP-41C

One of the assets of owning a card-programmable calculator is being able to conveniently record programs already written, checked, and ready to run. The HP-41C has an abundance of programs, books, and Application Pacs, and more are on the way. Here are lists of the various categories of software.

APPLICATION PACS ($45 each)*

Aviation *(Available late 1979)*
Clinical Lab and Nuclear Medicine
Circuit Analysis
Financial Decisions
Mathematics
Securities
Statistics
Stress Analysis
Structural Analysis *(Available late 1979)*
Surveying
The following pacs are under development:
Games
Home Management
Machine Design
Navigation
Real Estate
Thermal and Transport Science

HP-41C SOLUTIONS BOOKS ($12.50 each)*

Business
- Business Statistics/Marketing/Sales
- Home Construction Estimating
- Lending, Saving, and Leasing
- Real Estate
- Small Business

Computation
- Geometry
- High-Level Math
- Test Statistics

Engineering
- Antennas
- Chemical Engineering
- Civil Engineering
- Control Systems
- Electrical Engineering
- Fluid Dynamics and Hydraulics
- Heating, Ventilating, and Air Conditioning
- Mechanical Engineering
- Solar Engineering

Other
- Calendars
- Cardiac/Pulmonary
- Chemistry
- Games
- Optometry I (General)
- Optometry II (Contact Lens)
- Surveying
- Physics

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

Library Corner

Most of you who were current subscribers as of last month should have received your copy of *Catalog Addendum #4*. If you do not receive a copy by late August, please contact the Users' Library. With truck strikes and so forth, bulk mail is not getting a priority these days. Also, *if your address is not kept current,* you are going to miss your particular copy.

Addendum #4 contains 520 new program abstracts, and this makes 3,270 that are now available to you. In this addendum we hope you find the new Application Category Table an improvement over the former table. It is impossible to accommodate every subject, but the new table should make your *Catalog* easier to use and the program you need, easier to find.

The Library is already getting started on the next addendum. Forward any new programs, corrections, or errors you find, or suggestions you might have, to the Users' Library (address on back cover).

By now you know that this issue is dedicated to telling you about the incomparable HP-41C. And that also means big news for Library subscribers! Why? Because most of the HP-67/97 programs in the Library will run on the HP-41C. (But HP-41C programs will not run on the HP-67 or HP-97.) That means there are *already* about 3,270 programs available for the HP-41C. In addition, the Library has about 300 new programs that were written just for the HP-41C. So, if you have not taken advantage of the many benefits the Library offers subscribers, you might consider it now. Contact the Library directly.

For the present, there will not be a separate HP-41C Library. Thus, the Users' Library will be even bigger and better than ever, because the HP-41C programs will appear together with the HP-67/97 programs. However, HP-41C program numbers will end with a "C." The abstracts for the new calculator will appear in the *Catalog* with a "41-Title," as they have in the past with "67" and "97" denoting the programs written on those calculators. (The only differences in 67 and 97 programs are the key codes.) However, we must emphasize again that the examples in all of the contributed programs are run by our applications engineers, and we will reject programs that do not run on both the HP-67 and HP-97 calculators. However, HP-67/97 programs do not have to run on an HP-41C.

ORDERING PROGRAMS

None of the programs in this issue are available in Europe at this time. However, if you live in the U.S., you can order Library programs you see here in KEY NOTES by using the Order Form in a previous issue, or you can order by calling toll-free at (800)648-4711, ext. 1000 (in Nevada 800-992-5710).

Until November 1, 1979 (see next paragraph), Library programs are available in two forms: a set of the program listings and instructions (software) is $3*, and the fee is $5* for a set of software *and* a recorded magnetic card(s).

PRICE CHANGES

November 1, 1979, some changes go into effect in the Users' Library. After that date, all Library programs (67, 97, 41C) will be $6* each. Each program will include documentation *and a prerecorded magnetic card!* Magnetic cards by themselves will NOT be available. There will be no more $3 and $5 programs. These changes are the result of added paper and postage costs that we have so far absorbed and can no longer continue to do.

Also on November 1, 1979, all Library subscriptions will be $20* a year for the U.S. and Canada. All other foreign subscriptions will be $30 a year.

Prices for the *HP-65 Library Catalog* and programs will remain at $8* and $3* each, respectively.

NEW PROGRAMS (HP-67/97)

Although *Addendum 4* is probably now in your hands, there are some recent program submittals we think you might enjoy. But before you order any of these, be sure you read the preceding paragraphs.

67/97 Receiver Parameters (#03719D)

This program interrelates several parameters important for radio receivers and similar devices. Sensitivity may be found in voltages, power, and dBm as a function of bandwidth and noise figure. Intermodulation distortion may be related to power level, intercept point, and dynamic range. Conversions may be made between noise figure and temperature, and between voltage and power ratios and dB. (224 steps, 7 pages)

Author: **William J. Riley, Jr.**
Pittsford, New York

(Well done, Mr. Riley! This program calculates an awful lot, and very easily. Should interest amateur radio operators and radio engineers. Ed.)

Now, here are a *pair* of programs that are very similar in their capabilities and operation. The data storage is such that, while computing a backwater curve using one program, the other program card may be entered if the channel cross-section changes! For example: A circular storm sewer emptying into a trapezoidal channel. The backwater calculations may be continued uninterrupted and *without restoring any data.*

67/97 Trapezoidal Channel Water Surface by Standard Step Method (#03680D)

This program calculates water surface profiles for trapezoidal, rectangular, and triangular channels. Required input is Manning's "n," bottom width, channel side slopes, flowrate, and channel slope. After being given the beginning station and flowline elevation, and known or assumed depth at the beginning station, the program outputs station, elevation of energy grade, hydraulic grade, and flowline, followed by depth, velocity, velocity head, and friction slope. Subsequent stations are then input and the above information is again printed out (or read out). Equations in stationing, flowline drops, and channel data changes are allowed. Normal and critical depth calculations may be made without interrupting backwater calculations. Manning's equation and English units are used, but a metric conversion routine is included. Data storage is compatible with Circular Channel Program #03681D. (224 steps, 11 pages)

Author: **Christopher R. Stevens**
Phoenix, Arizona

67/97 Circular Channel Water Surface by Standard Step Method (#03681D)

This program description is virtually identical to the one above except that the calculations are for circular channels (pipes). Also, data storage is compatible with #03680D. (222 steps, 10 pages)

Author: **Christopher R. Stevens**
Phoenix, Arizona

(Although the subject of Mr. Stevens' programs might not interest everyone, surely his documentation would! Excellent *is the word that comes to mind when reading the pages of comments, diagrams, etc. Kudos to you Mr. Stevens! Ed.)*

Next, we have some "special" programs for you. These two, below, were written in Japan by **Kenichiro Akita,** and are a *tour de force* in programming and documentation. Mr. Akita is Deputy Manager of the Textile Project Group for Nissho-Iwai Company, Ltd., in Osaka, Japan. And, although these programs can be used on an HP-67, they were designed primarily to use tape readouts from the HP-97.

The first "special" is: **Constant Payment to Principal Loan, Exact Amortization #67000-99974,** and the price is $10.00.* Here is the abstract:

Once fundamental data inputs on an annual basis have been entered, exact amortization schedules of Constant Payment to Principal Loan may be generated in different forms: all outputs on a single tape or on separate tapes of the same length, the Payment Numbers, Dates, Total Payments, Interests, CPMT to Principal, and Balances. By being of the same length, the tapes can be joined together to produce lists. There is one Input Card for fundamental data, seven cards for individual amortization, and seven cards for yearly amortization. Basic data is retained for repeated use by different cards. Options are: Rounding Mode, Accrued Interest for 365 or 360 Day Basis, and Totals-To-Date. And you can freely set the "time frames" to fit not only the calendar year but also fiscal years, taxation years etc. (2,781 steps, 15 cards, 60 pages)

The second of Mr. Akita's "specials" is:

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

Mortgage Loan, Exact Amortization #67000-99975, and the price is $10.00.* The abstract is essentially the same as for the preceding program except that the schedules of mortgage loans are generated. And this program also allows setting of "time frames" to match your requirements. (2,338 steps, 15 cards, 60 pages)

Author: **Kenichiro Akita**
Amagasaki, Japan

And, finally, here is a set of programs that is being offered as a "special." These seven programs in the field of x-ray crystallography should prove useful to others working in this and such related disciplines as physical chemistry and conformational analysis. In the words of the author, **John P. McAlister:** "These programs perform calculations normally available only on large multi-purpose computers to those actively pursuing research in x-ray crystallography. Such programs are less commonly available to others whose principle interest is in another area but whose work is augmented by the results of crystallographic analysis. As a crystallographer, I find these programs useful because it is frequently more convenient to perform a few calculations at my desk than to endure the interruptions and general hassle required to log on and use the general laboratory computer. While these programs were developed with the analysis of crystallographic data in mind, those performing coordinate manipulations or calculations based on coordinate geometry might well find applications in other areas. All the programs function in the context of a general oblique coordinate system and therefore will accept the more usual Cartesian system as a special case when properly initialized."

The title of this set is **X-Ray Crystallography #67000-99976** and the price is $30.50.* The set includes seven separate, but coordinate, programs, and the last six depend on the data card generated by the first program. The program titles are:

#03739D Crystallographic Data Card Initialization.
#03740D Coordinate Assignment in Crystallographic Systems.
#03741D Crystallographic Preliminaries.
#03742D Crystallographic Coordinate Transformations I.
#03743D Bond Distances and Angles From Crystallographic Data.
#03744D Torsion Angles From Crystallographic Data.
#03745D Least Squares Plane From Crystallographic Data.

Author: **John P. McAlister**
Madison, Wisconsin

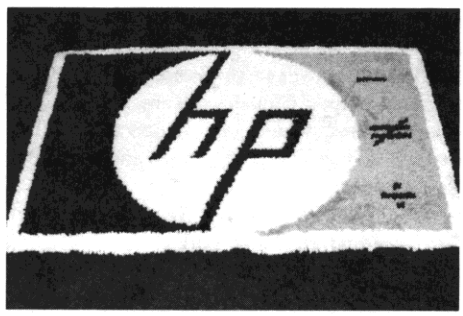
(Quite a feat, Mr. McAlister! In all, 62 pages of documentation, 1,855 steps, and 9 magnetic cards. But I should warn readers that although each program is available separately for the standard fee, you need #03739D to run any of the others, and the programs are definitely a better buy as a "package." Beautifully documented and typed; excellent work. Just one question, Mr. McAlister: Who did all *that typing? Wow! Ed.)*

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

"Hooked" on HP!

Not long ago we received a letter from a calculator owner who has owned an astonishing array of calculators. Most of his first ones operated with algebraic logic and were made by several different manufacturers. But going from one to the other was a confusing mess. So 16 months ago he bought an HP-97, got rid of the others, and life has changed somewhat. Here's an excerpt from his letter.

Since the purchase of the HP-97, I have also added the HP-67, HP-22, HP-19C, and HP-21 to my collection. The fact that all of these calculators use RPN and function basically the same, makes going from one to the other very simple. Also, I have bought a second HP-21 for my wife to use, and she is very well satisfied with it. In fact, I am so well pleased with the HP line of calculators that I even designed and hooked a rug to let everyone know that I feel that Hewlett-Packard is unequalled by anyone else. Enclosed you will find some photos of the above rug and calculators.



Hope you enjoy the photos, It's just too bad that I can't send the rug to you, but I think as much of it as I do of my calculators.

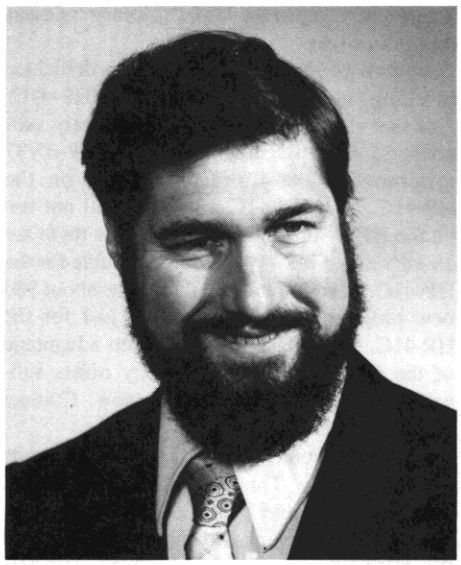
Sincerely yours,
David D. Walton, Cincinnati, Ohio

(Thanks for the color photos, Mr. Walton. Too bad we can't print in full color. That rug is a remarkable work of art, right down to the "HP blue" color! Congratulations for a fine job. Ed.)

Another Excellent "Calculator" Book

Since all of you are reading this newsletter because you have a deep interest in learning how to apply your calculators to your specific needs or problems, and since we know that a lot of you are in one of the many fields (!) of the electronics industry, it is a pleasure to tell you about an extremely good book that will soon be in the bookstores. And, because the author of this book has contributed many excellent programs to the Users' Library and has been featured in KEY NOTES before (Vol. 2 No. 2, page 4), it gives us even more pleasure to present this new book, *Handbook of Electronic Design and Analysis Procedures Using Programmable Calculators.*

Besides being an avid HP calculator fan and a contributor of excellent programs, the author



Bruce K. Murdock, is a staff consultant to the Electronic Design Engineering section at Delco Electronics in Santa Barbara, California. There, his responsibilities include the design and analysis of such electronic equipment as underwater acoustic and sonar systems, radio and cable telemetry systems, servo systems, highpower amplifiers, and power conditioning equipment. Mr. Murdock has also lectured at the University of California at Santa Barbara on the subject of network analysis and synthesis.

This book should be of real use to electrical engineers and technicians who have access to programmable calculators or desktop computers. The book collects a vast amount of material and is clearly the result of a great deal of work. It contains 39 program descriptions concerning problems within the electrical engineering field; however, this range is very wide, from magnetic component design (transformers and solenoids) to microwaves, and from circuit and signal analysis to circuit and filter synthesis. Also, this range can be seen from the titles of the five sections in the book: Network Analysis, Filter Design (active and passive), Electromagnetic Component Design, High Frequency Circuit Design, and

Engineering Mathematics. The usefulness of the book is further enhanced because many of the programs are really sets of solutions to closely related problems (the interchangeable solution concept).

Although each program description ends up with a superbly annotated and flowcharted HP-67/97 program listing (and some algebraic program listings), the main strength of the book is in the program descriptions. After a clear statement defining the use of the program and some background on the subject, there is a concise step-by-step development of the equations that are to be used, along with references (58 in all) for those readers wishing more background on the particular subject. Every equation is numbered, and the book is fully cross referenced by equation number. The reader is literally "guided by the hand" through the development of the equations and resulting algorithms. The author never says "it *can* be shown," he *shows* it in detail. This material alone would be a valuable addition to any engineer's technical library, and represents over half of the book's 544 pages.

After the program algorithm development are step-by-step operating instructions for the associated programmable calculator program. Following these operating instructions are several program examples, complete with HP-97 printout showing all program input and output. These examples enable the user to fully verify his use of the programs. Many answers given in the examples are compared with answers obtained by other means (such as from published tables) to demonstrate the correctness of each program solution.

The HP-67/97 (and algebraic) program listings are completely and fully documented. The program comments and equations on the right-hand side of the program listing read like a flowchart. The flowchart nature of the programs is further enhanced with lines and arrows on the left-hand side of the program listing; i.e., a line connects "GTO 5" with "LBL 5," or "GSB A" with "LBL A," so the reader can easily follow the program flow. The contents of all registers are also completely listed.

The many elegant routines within the programs that both shorten the program length and maximize program execution speed reflect both the author's cleverness and experience in his field.

The book has received excellent reviews from both **Sidney Darlington** (the father of modern filter design and for whom the Darlington transistor connection is named), and **Philip R. Geffe,** noted filter designer and consultant.

The price of the book is $26.50,* which buys over 540 pages (8½-by-11-inch format) of excellent data, 39 very complete and very-well-documented programs, and a sturdy hardbound cover with dust jacket. The book will be available in October 1979.

To order your copy of the book, send your check or money order (including any local sales tax) to:

Van Nostrand Reinhold Company
Mail Order Service
7625 Empire Drive
Florence, Kentucky 41042

The publisher will ship your book postpaid, on a 15-day money-back guarantee basis. This offer is available only in the Continental U.S., Alaska, and Hawaii. For shipments outside these areas, write to the following address for information on price and how to order.

Litton Educational Publishing
International
135 West 50th
New York, N.Y. 10020 U.S.A.

(Congratulations, Mr. Murdock! I cannot imagine an engineer, in fields related to this book, not wanting it. This is an excellent book. And, remember, it's very likely you can deduct the cost of it from your income tax. In many more ways than that, it is truly a bargain. Furthermore, none of Mr. Murdock's programs from the Library are in the book in their original form. What few Library programs are in the book are significantly changed. Even the program listings have been revised to be faster, shorter, more accurate, and to provide more output. Finally, there is some possibility of preprinted prerecorded magnetic cards being made available if enough interest is shown. If you wish to correspond with the author regarding magnetic cards, write to the address below. Ed.)

Bruce K. Murdock
P.O. Box 2325
Goleta, CA 93018 U.S.A

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

Random Numbers, Means, Regression, And The Programmable Calculator

The title of this article is also the title of a new book. It was written by **Thomas W. Beers,** Professor of Forestry at Purdue University's School of Agriculture in West Lafayette, Indiana.

The five chapters comprising this monograph were originally written as individual documents to be used in the teaching of undergraduate students in a Forestry Biometry-Forest Inventory course. The goal was to make use of the programmable handheld calculator as a tool to remove the tedium of the calculations and to provide some "excitement" to a potentially boring subject. The HP calculator family is used because of the convenience of the RPN logic system employed in the HP line. The programs described throughout this monograph are presented in considerable detail, and with examples, to facilitate their use by the novice. The chapters are:

1. Random Numbers for 3-P Sampling by Handheld Calculator.
2. The Meaning of Means and the HP-67.
3. Simple Linear Regression and the HP-67.
4. Multiple Regression, Simultaneous Equations and the HP-67.
5. Multiple Regression: Significance, Estimation, Deletion, and the HP-67.

The book is softbound in 8½-by-11-inch format, contains over 138 pages, and should interest not only Foresters but also professionals in education. For the U.S., Canada, and Mexico you can purchase the book for $8* per copy, postpaid (fourth class mail). If you order from other countries the price is $9* postpaid. Mail all orders to:

T & C Enterprises
Box 2196
West Lafayette, IN 47906
U.S.A.

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

Want To Help Find More Oil?

Do you want to possibly help relieve the energy crisis? Well, here's your chance.

Edwin J. Ballantyne, Jr., special editor for an upcoming Society of Exploration Geophysicists (SEG) manual of handheld calculator programs for geophysics applications, is requesting contributions of programs. Specifically requested are programs for data reduction and interpretation in both algebraic and reverse-Polish notation. Separate manuals for calculators of each type are projected. Areas of special interest are: gravity, magnetics, radio-metrics, refraction seismics, reflection seismics, resistivity, induced polarization, electromagnetics, well logging, and physical properties. Other topics of interest to geophysicists are encouraged.

Each contribution should include: (1) a brief program description, including equations; (2) user instructions; (3) a program listing; and (4) test examples.

Send all contributions to:
Jerry W. Henry
Society of Exploration Geophysicists
P.O. Box 3098
Tulsa, Oklahoma 74101

Deadline for receipt is December 31, 1979. Target date for publication of the manual is mid-1980.

Any questions or comments should be directed to:

Edwin J. Ballantyne, Jr.
Amoco Minerals Co. – Mail Code 5407
200 E. Randolph Drive
Chicago, Illinois 60601
Telephone (312) 856-2503

(Here's your chance to maybe help these folks find some new sources of oil. Anyone knows by now that we sure do need it! Ed.)

Editorial

The HP-41C has been on the market two weeks as of this writing. Already it is a great success. The most-asked question I hear about it is "Will it push the HP-65, HP-67, and HP-97 into oblivion?" The answer is no. KEY NOTES has no intention of abandoning these predecessors to our newest marvel. And if you don't see HP-65 articles or news, it's because there hasn't been any or I haven't received any from you. Also, don't forget that the HP-65 Library is still there, still functioning, still available, with over 5,500 programs in the U.S. Library.

Something is missing from this issue for the first time. The address block on the back cover no longer carries the statement "Address Correction Requested, Return Postage Guaranteed." You will find another statement in its place. The reason for this is easy to explain. Americans move around a great deal, and postage costs are escalating with inflation, so our costs for keeping track of where you are have become too high to continue this way. In the future, if you want to continue getting KEY NOTES, be sure you inform me of your address changes. I'll forward your address change to the Library if you're a member.

It is not unknown to us that some of you examine our products with a *very* critical eye. Some of you are looking for those "hidden" functions that you just *know* are lurking in there—somewhere. Then too, some of you are just curious as to what makes these things work. But the scrutiny we are referring to transcends *all* of that. For example:

Several weeks ago the phone rang, I innocently picked it up—as usual—and there was **Craig Pearce** (Berwyn, Illinois), who is the *ne plus ultra* of calculator enthusiasts. You'd have had to have been here to have really appreciated the conversation: "Hello, Craig; how are you?" "Fine, thanks. Say, Henry, I just got one of those remarkable Swiss Army knives—this one even has a nice magnifying glass in it—and I've been busy playing with it and looking at all sorts of things." "That's nice, Craig, but what does that have to do with calculators?" "Henry, did you know that the small decimal point in the HP-19C display has a tiny HP logotype on it?"

Your editor's life is never dull ... .

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!

Base Conversion, Slightly Off-Base ...

On the back cover of Vol. 3 No. 2 we had an article, "Base Conversion Routine," contributed by **Cass Lewart.** Just before we went to press, a piece of crooked typesetting from our phototypesetting machine was replaced where the "answer" appears, just above the routine's listing. Unfortunately, the machine injected a double error. The last line of text *should* read:

Answer: 903081113 = 938 BD hex.

The infernal machine dropped a "1" and jammed-up some letters.

Also, we received a letter from **Bob Edelen** of Denver, Colorado, informing us that Mr. Lewart's routine has a slight malfunction in it. And that is true. To correct the original routine, insert an "INT" step between original steps 033 and 034.

Mr. Edelen found the omission of "INT" when he tried to run 3486784400 decimal to base 9. He got 8888888889, and he should have gotten 8888888888.

Thanks, Mr. Edelen, for sharing this "fix" with all of our readers.

About Batteries...

Because the new HP-41C uses alkaline batteries instead of rechargeable batteries, there are some things you should know about them. As with *all* batteries, battery life is very dependent on the age of the battery (shelf life) and the amount and type of use.

With no plug-ins and with normal use, meaning day-to-day calculations and running several programs, an HP-41C will have a battery life of 9 to 12 months.

An HP-41C that is continuously executing (nonstop) a program will run from 50 to 100 hours. As you can see, this is an *extreme* case. Intermittent use will extend battery life enormously.

Most HP-41C users will find that, with normal use of the card reader and the plug-in modules, the batteries will last from several months to six months or even more.

With *extremely heavy* card reader usage, battery life will average somewhere between a week and two weeks. However, even though the batteries will no longer operate the card reader at that point, they will continue to operate the calculator for quite some time.

Early in 1980, an accessory AC adapter will be available to run the HP-41C on ordinary house current.

Don't Use Solvents!

Because it is possible for batteries to leak under certain circumstances, some people are tempted to spray the battery terminals and battery compartment in their calculator with, typically, a "silicone spray" from a pressurized can to protect that part of their calculator. These aerosols usually contain a solvent of one kind or another, and most common solvents have a detrimental effect on most common plastics. Therefore, we do not recommend the use of these sprays on the batteries or in the battery compartment.

Proper care of the batteries in your calculator will preclude the need for spray coatings. Do not leave discharged batteries in your calculator, and remove the batteries if you do not expect to use the calculator for extended periods.

More About Merging

For the last several issues, we have been publishing a lot of material about merging so that you can find a comfortable or convenient way to write multiple-card programs and thus extend the power of your HP-67/97. We don't want to bore you with the subject, but here is another excellent letter on merging. The author of this letter is also the author of a new book that is reviewed elsewhere in this issue. We think you'll like his ideas.

My reason for writing has to do with the article on merging. (Vol. 3 No. 2, page 5.) While the methods in the article work as advertised, chaining of long programs can be done more simply and still achieve the same results. If a PAUSE instruction is encountered during the execution of a program, and a new program is loaded during this time, the new program will replace the old one. When the pause time-out ends, the calculator will still be running a program, *but it will be the new program.* The new program starts execution at step 001.

I have used this technique in my book to chain programs requiring more than one card. I generally place the PAUSE instruction in a loop: LBL, PSE, GTO. Whatever was in the X-register prior to entering the loop will be displayed during the pause. This display can be the identifying number of the magnetic card to be read. The user need not wait for the pause. The next magnetic card to be read can be inserted into the card reader as soon as the program has been started. When the pause is encountered, the card will be read.

This instruction sequence does not have to be placed at the end of the current program, nor at steps 222 through 224; it can be placed anywhere it is convenient. The next card in the sequence does not have to have anything special at the beginning (steps 001 through 003). The first step of the new program is the first step the programmer wants executed. By not using the MERGE instruction, even greater program efficiency is achieved.

Remember, the MERGE instruction is used to modify a section of current program. Generally, when programs are chained, no part of the

previous program needs to be saved; hence, the MERGE instruction is not appropriate in this instance.

I hope this dissertation has shed some more light on program chaining.

Very truly yours,

Bruce K. Murdock, Goleta, California

Pac Corrections

If you own some of our application pacs, check the following corrections and mark them in your copy. If the correction includes a revised card, **you must mail in your old card to get a new one.** Be sure to include your name and address. If your pac copy is correct, you have a later, revised issue.

HP-67/97 SURVEYING PAC

Program SU1-19A1, "Geographic to Alaska 2-9," has been found to contain some errors that can cause inaccurate results in some calculations. On pages L19-01 and L19-02, mark the following corrections in the listing.

1. At bottom of L19-02, add "USED" in the box for flag 2.
2. Delete step 185.
3. Delete step 182 and add "x^2."
4. Delete step 178.
5. Between steps 175 and 176 add "F2?, CHS."
6. Delete step 133 and add "x^2."
7. Delete step 130.
8. Delete step 088.
9. Delete steps 075 and 074 and add "RCLB."
10. Between steps 054 and 055 add "X<0?, SF2, STO B."

To receive a revised card, **you must mail your old card to: HP Service Department; P.O. Box 999; Corvallis, Oregon 97330.** This new card will be available on October 15, 1979.

Program SU1-19A3, "Alaska 2-9 to Geographic," has been found to contain some errors that can cause inaccurate results in some calculations. On pages L19-05 and L19-06, mark the following corrections in the listing.

1. At the bottom of L19-06, add "USED" in the box for flag 2.
2. Between steps 198 and 199 add "F2?, CHS."
3. Delete step 179.
4. Between steps 149 and 150 add "x^2."
5. Between steps 087 and 088 add "x<0?, SF2."
6. Between steps 001 and 002 add "STO C."

To receive a revised card, **you must mail your old card to: HP Service Department; P.O. Box 999; Corvallis, Oregon 97330.** This new card will be available on October 15, 1979.

"25 Words" (More or Less!)

The column is somewhat truncated this issue because of the space devoted to the HP-41C, but if you have contributed something in the past weeks or months, don't give up. We'll expand the column in November to try to "catch up."

Our first input is from **Justus A. Villa** who lives in St. Thomas in the Virgin Islands.

Here is my contribution to your "25 Words" feature. It is a flexible increment/decrement routine. To set it up: ±n, STO9, P≷S (n can be an integer or not). Thereafter: increment by Σ+. and decrement by Σ−. Notice that the other statistical registers are not *necessarily* disqualified as accumulators.

Here is another tip for statisticians. It's from **Michael Tarnowski** of Wiesbaden, Germany.

I have found a routine that I think would be appropriate for your "25 Words" column. The routine calculates the value of χ^2 statistics for the goodness of fit test.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

To test the routine, press [A], key in O_i, [ENTER↑], E_i, and press [B]. To delete wrong values, key in O_i, [ENTER↑], E_i, and press [C]. After the last input has been keyed in, press [RCL]1 for χ^2. That's it! The routine can be used by itself or as part of a larger statistical program.

```
001  *LBLA      014   ST-1
002   CLRG      015     1
003    CLX      016   ST-0
004    RTN      017   RCL0
005  *LBLB      018    RTN
006   GSB0      019  *LBL0
007   ST+1      020   STO2
008      1      021     -
009   ST+0      022    X²
010   RCL0      023   RCL2
011    RTN      024     ÷
012  *LBLC      025    RTN
013   GSB0      026    R/S
```

What do they do with HP-67/97's in Maydon Wharf, South Africa? That's easy to answer. Just read what **Richard Helps** has to say:

Here are two short routines that I have not seen published in KEY NOTES. I hope they may be of some use to your readers.

Sometimes in a programme it is necessary to know what trigonometric mode the calculator is in (e.g., using trig mode as an extra three-way flag). The routine: EEX, 2, COS will return −0.17 for DEG, +0.86 for RAD, and 0.00 for GRD, which can be tested using X<0?, X>0?, and X=0? for branching, etc.

If you wish to test only for DEG and RAD, then use: 2, COS, X>0?, SF2 which will set flag 2 for DEG mode and leave it clear for RAD mode. Of course, any flag or branch instruction can be used at the last step (SF2).

Now, from **Rex H. Schudde** of Carmel, California, comes this alternate solution to the angle conversion routine in the last issue.

On page 9 of Vol. 3 No. 2 you printed an 11-step program for 000° to 360° angle conversion by **Mr. Caldwell.** Attached is a listing of a nine-step program, sent to a mutual friend by **Mr. Ian McKinnon** of Toronto Islands, which accomplishes the same result in a very elegant manner. Further, this routine is not limited to angles less than or equal to 720°; Mr. Caldwell's routine does have this limitation. Thought that you and Mr. Caldwell would be interested.

```
001  *LBLA
002     3
003     6
004     0
005    →R
006    →P
007   X≷Y
008   X<0?
009     +
010   R/S
```

Here is a letter about an article in the last issue (page 6). It is from **Richard G. Cosway** of Phoenix, Arizona. His routine is pretty neat!

With regards to **Mr. Craig's** article "Spring is 'Seed' Time!" in the May 1979 issue, I have what I feel is a much simpler and shorter random number generator.

Assuming the seed is in register E and is between zero and one (though no starting seed is necessary):

```
001  *LBLE
002   RCLE
003     Pi
004     +
005     eˣ
006    FRC
007   STOE
008    RTN
009    R/S
```

This routine has the advantage of requiring only one register since no multiplicand needs to be stored. Also, it produces numbers which, from what I've been able to determine, are random.

But how's this one for brevity? It was contributed to KEY NOTES by **Charles Close,** who lives in Alexandria, Virginia.

I believe the following statement almost meets the criteria for the "25 Words or Less" column.

"If the total number of identical repeated steps in a program exceeds eight (8), the program will be shorter if those repeated steps are placed in a separate subroutine, to be called when needed."

(Continued)

Not quite 15 words, but almost! Keep up the good work in KEY NOTES.

(Thanks for the compliment, Mr. Close. And, by the way, take another look at the column title. You were closer (!) than you thought! Ed.)

Perhaps a lot of our readers have the same problem mentioned in this letter from **Guy Dresser** of Lawrence, Kansas. There are undoubtedly other solutions than his, but it *does* work.

I use the following routine, which compares decimal numbers to fractions, in an employee time card totalling program, and thought it might qualify for "25 Words or Less."

```
001  *LBLE      017   X≤Y?
002   ENT↑      018   GTO1
003   ENT↑      019    R↓
004   RCLA      020   RCLA
005      2      021   RCLB
006      ÷      022      +
007      0      023   GTO0
008  *LBL0      024  *LBL1
009   STOB      025    R↑
010    R↑       026   PSE
011    FRC      027    R↑
012   X≷Y       028   INT
013      -      029   RCLB
014   ENT↑      030      +
015    R↓       031   RTN
016    ABS      032   R/S
```

The routine selects the fraction closest to the decimal portion of a number, displays the difference between the fraction and the decimal number, then ends with the integer and "fraction" added together in the display.

Since we pay on the basis of quarter-hours but the time clock is calibrated in hundredths of an hour, the routine will establish how many hours and quarter hours to pay for, and also the number of hundredths over or under to carry forward to the next week. It will work, though, for any fraction, so it might also be useful in teaching children the relationships between fractions and decimals.

To try the routine, put ¼ (.25) in register A, key in a value such as 40.68 and press [E]. The calculator will pause, displaying –0.07 (difference between fraction selected and decimal), and then it will display 40.75 (40¾). Then put 1/6 (6, 1/X, STO A) in register A, key in 40.68, and the results will be 0.01 and 40.67 ($40^4/_6$). If you are unsure of the fraction you end up with, (how many sixths is .67?) use FRAC, RCL A, ÷, which could be appended to LBL 1 in the program, after a pause.

Do you find yourself watching the clock at the end of the day? If so, here is a neat routine that will interest you, or perhaps you can apply it to other projects. It is from **John L. Gafford** of Jackson, Michigan.

Here's one for your "25 Words or Less." I call it "Clockwatcher." Load the program, look at your digital watch, key in the time, press A and presto—hours and minutes (and seconds if you DSP4) until quitting time! It will accommodate any time between 6 A.M. and 6 P.M. Negative numbers mean you are working overtime! The test in step 005 determines whether the time keyed in is before or after noon. The appropriate routine under LBL0 or LBL1 then makes the calculation. Quitting time is placed in steps 010 through 013 in 12-hour format, and in 018 through 022 in 24-hour format.

```
001  *LBLA
002   ENT↑
003      6
004   X≷Y
005   X≤Y?
006   GTO0
007   GTO1
008  *LBL0
009    CHS
010      4
011      .
012      4
013      5
014   HMS+
015    RTN
016  *LBL1
017    CHS
018      1
019      6
020      .
021      4
022      5
023   HMS+
024    RTN
```

Our quitting time happens to be 4:45 P.M., so the user should correct the entries to correspond to his quitting time.

On a more serious note, the program could be adapted to quickly calculate the time between the present instant and some future event.

Just Before We Went To Press...

As we were putting this issue "to bed," Hewlett-Packard announced that, effective August 1, 1979, the price of the HP-67 would be reduced to $375.* The price of the HP-97 remains the same, at $750.*

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

HP KEY NOTES

August 1979 Vol. 3 No. 3

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

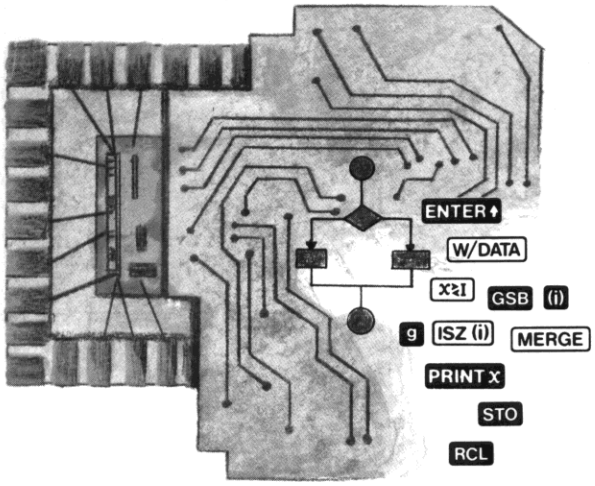
Hewlett-Packard SA
USERS' CLUB EUROPE
7, Rue du Bois-du Lan
P.O. Box, CH-1217 Meyrin 2
Geneva-Switzerland

Address Correction Requested

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

0001298918
M R DEZEARN
1310 G EVERETTE
ABERDEEN PR GRD MD 21005

Printed in U.S.A.



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

HEWLETT PACKARD

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)!

All prices in this newsletter are suggested retail prices excluding applicable state and local taxes–Continental U.S.A., Alaska and Hawaii

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 submittals: 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 54 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-

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 program SD-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,** deserves 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 Σ+ ? 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 , • , × . 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. Lables 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 Σ+ PRGM. Now press ■ VIEW • X 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 Σ+ (01). Repeat with 1.005 XEQ Σ+. 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 Σ+. 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 Σ+ sequence. Now press X≷Y 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, STOP	Stops for input. Prints alpha display.
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, STOP	Stops for input. Prints alpha message.
CLA, ARCLX, AVIEW	Prints input (left justified).

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 Σ+ PRGM, then SST to the TONE 8 instruction. Delete with ← and key XEQ ALPHA TONE ALPHA ■ • X PRGM. Now key 5 and press A. This time the tone corresponding to the "count" number is heard. Try 1.005 XEQ Σ+. The count started at 1 and went to 5. Try 4.008 XEQ Σ+.

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 d 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   STO I      022   R/S
004  *LBL C      023  STO i
005      4       024  ISZ I
006   RCL I      025  GTO C
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  STO I
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 safety 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 "SZE"	Assigned to: 1/x key
44 Bytes	Exits USER mode upon execution? Yes.
20 Lines	
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 AVIEW
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 1/x key.)
.1 To find the SIZE that has been allocated, press USER then 1/x. 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: Σ+ key.
17 Bytes	Exits USER mode upon execution? Yes.
07 Lines	
03 Registers	System Requirements: Std. 41C

```
01♦LBL "RCLΣ"     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 "SZE".

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

```
01♦LBL "VIEWSZE"   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 √x key.)
2. To find the SIZE the HP-41C is adjusted to, press USER and then √x. 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 execution? Yes.
33 Lines	
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⇄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 ×	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 ½ 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 fire 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 displeasure 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 (55 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 ⇄ y
R↓
```

The index number, without zeros, is less likely to be confused with data, hence the DSP0. The x ⇄ 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:		**Input:**	
	LBLA		1309.23
	÷		ENTER
	INT		360
	×		ENTER
	−		1309.23
	RTN		ENTER
			360

After keying in the program and the inputs, press [A] 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	LST*x*		013	RTN	
007	X ⇄ 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   RCLi        018   X=Y?
008    X≠Y        019    RTN
009    X≠I        020    R↓
010    X≠Y        021   GTO5
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   RCLI        017   RCLB
005   X>Y?        018   LSTX
006    RTN        019     %
007   DSP0        020   ST+i
008    PSE        021   ST+9
009   DSP2        022   RCLi
010   RCLA        023   PRTX
011   RCLi        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 repeitions 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 ARCL X
10 RCL 06         22 AVIEW
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 **Ramón 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   X≠Y
002     0       010     2
003   STOI      011     5
004  *LBL1      012   X=Y?
005   RCLI      013   RTN
006    PSE      014  ISZI
007  RCL i      015  GTO1
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: Mod(x,y)= (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

```
001  *LBLA
002     3
003     6
004     0
005   ENT↑
006   ENT↑
007    R↑
008   X≠Y
009     ÷
010   FRC
011     x
012   X<0?
013     +
014   RTN
```

PRGM II

```
001  *LBLA
002     3
003     6
004     0
005   ENT↑
006   ENT↑
007    R↑
008   X≠Y
009     ÷
010   FRC
011     x
012   RTN
```

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.

PRGM III

```
001  *LBLA
002   RCL0
003     ÷
004   FRC
005   RCL0
006     x
007   X<0?
008   GTO0
009   RTN
010  *LBL0
011   RCL0
012     +
013   RTN
```

PRGM IV

```
001  *LBLA
002   RCL0
003     ÷
004   FRC
005   RCL0
006     x
007   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   X≠Y
008   X<0?
009     +
010   RTN
```

Examples: 3603° = 3°
−896° = 184°

$-180 < \theta \leq 180$

```
001  *LBLA
002     1
003    →R
004    →P
005   X≠Y
006   RTN
```

Example: 182 [A] = −179°

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° (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   X≤Y?      018   CHS
008   GTO0      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   X≤Y?      017  *LBL0
008   GTO0      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:

The equivalent operation is also possible with the HP-67.

```
001  *LBLA
002     0
003    R↓
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  *LBLE        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    X²
002    CLX        009   X≷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 [R/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   RCLE
006     1       029   CHS
007   HMS+      030   HMS+
008   STOA      031   STO3
009  *LBL0      032   →HMS
010    √x       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↓
018  *LBLE      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 outisde 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
002      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 $x \rightleftarrows y$ and R↓ at step 022 and now the recalled contents of the registers are in X and Y of the stack.

```
001  *LBLA      014   GTOA
002    STOi      015    RTN
003    RCLi      016  *LBLC
004    ISZI      017    ENG
005    RCLI      018   X≷I
006     R/S      019   RCLi
007     RTN      020   X=0?
008  *LBLB       021   RCLi
009    STOi      022   DSP4
010    RCLi      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' 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←, R↓, R← to get X:x, Z:A, T:z.

Cartesian to Spherical. The exact inverse is obtained with →P, R↑, →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←, $x \rightleftarrows y$, R↓, R← to get X:c_1, Y:c_2, Z:A, T:c_3.

Direction Cosines to Latitude, Longitude. The inverse is obtained with →P, R↑, $x \rightleftarrows y$, →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  *LBLE
213    STO9
214     X²
215      x
216      6
217      ÷
218    STOD
219    ST×9
220      2
221    ST÷9
222    RCL9
223     RTN
```

(Continued)

CIRCULAR CROSS-SECTION

r

```
211    R/S
212  *LBLE
213   STO9
214      3
215     Y^x
216     Pi
217      x
218      4
219      ÷
220   STOD
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.

```
001  *LBLA
002   X=0?
003   GTO1
004     R↓
005   X=0?
006   GTO2
007     R↓
008   X=0?
009   GTO1
010     R↓
011  *LBL2
012     R↓
013      ÷
014      x
015    RTN
016  *LBL1
017     R↓
018      x
019    X≠Y
020      ÷
021    RTN
```

EXAMPLES:

```
13.00 ENT↑
22.00 ENT↑
 0.00 ENT↑
44.00 GSBA
26.00  ***

 0.00 ENT↑
22.00 ENT↑
26.00 ENT↑
44.00 GSBA
13.00  ***
```

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