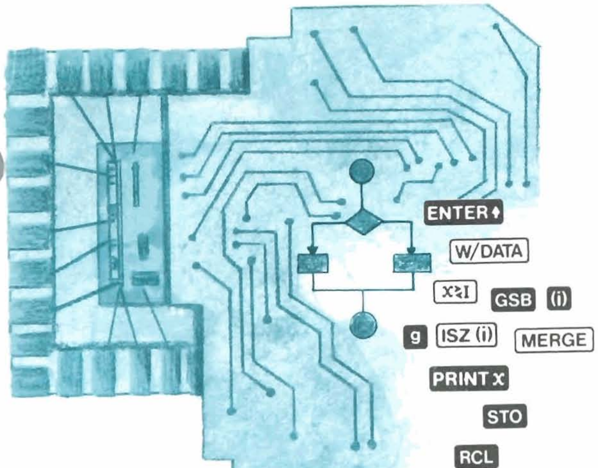


Featuring, this issue:

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.

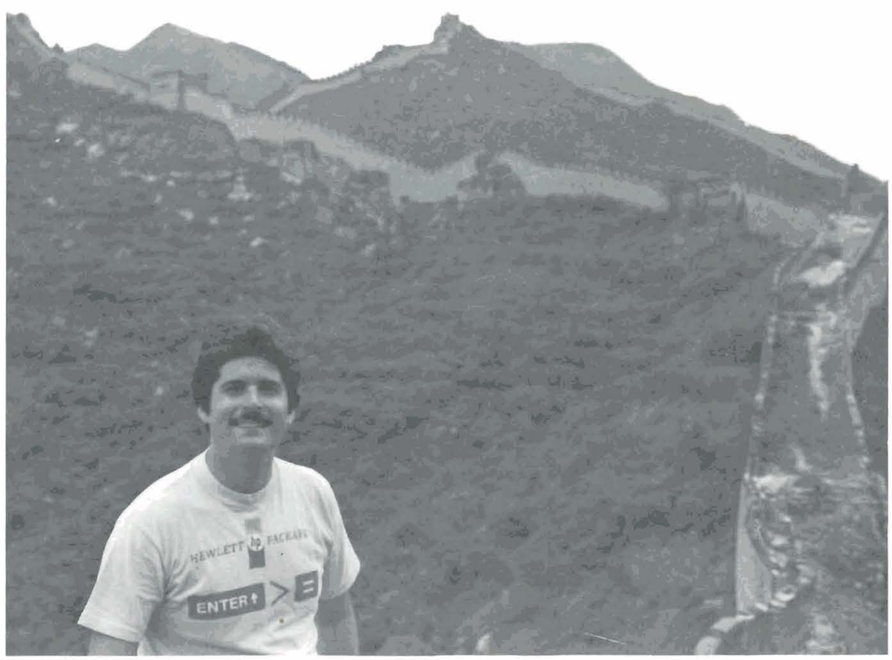
The Chinese interpreter for out group had some knowledge of engineering and math. Because he spoke English, I was able to demonstrate some of the programmable features of my HP-67 by programming some simple arithmetic functions and some branching routines. He was especially impressed by the "Polynomial 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.)*



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!"

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.)**

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

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

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	GSB8	026	*LBL6
005	GSB9	016	GSB8	027	X⇄I
006	GSB9	017	GSB8	028	DSP0
007	GSB9	018	GSB8	029	PSE
008	GSB9	019	*LBL8	030	X⇄I
009	X≠0?	020	RCLi	031	DSP9
010	GSB6	021	X≠0?	032	PRTX
011	STOI	022	GSB6	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 B. This resets the display and turns off the main program pause option (no longer needed). If you now key .21 B 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 B → "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

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

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     Yˣ     049      ×
017   ENT↑     050    RTN
018   ENT↑     051  *LBLB
019   RCLi     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   RCLi     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      ×
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    ***

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

EXAMPLES

```
                 FIX
                  Pi
 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      ×    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

HP-67 and HP-97 Accessories

Description	Model Number	Price
DC Recharger/Adapter	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.	82058A	$ 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)	00097-13143★	$ 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
HP-97 Standard Pac	00097-13101
Business Decisions Pac	00097-13144
Civil Engineering Pac	00097-13195
Clinical Lab & Nuclear Medicine Pac	00097-13165
EE Pac I	00097-13131
Games Pac	00097-13185
Math Pac I	00097-13121
ME Pac I	00097-13155
Navigation Pac	00097-13205
Stat Pac I	00097-13111
Survey Pac I	00097-13175

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

SEE SPECIAL "KEY NOTES" OFFER ON NEXT PAGE!

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

✱No longer available.

★Also usable on the HP-65.

HEWLETT-PACKARD ORDER FORM **HP CODE NO. 300ZB**

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

Please enter my one-year subscription to the HP-67/97 Users' Library

☐ For US, Puerto Rico or Virgin Islands	$15.00	
☐ For countries outside US, PR, V.I., and Europe	$20.00	

SPECIAL OFFER TO "KEY NOTES" READERS!

For every two **Users' Library Solutions** books you order from the list on page 7, we will send you one free. Just list your choice of the free book (or books) below the two you order and write "free" in the "Price Each" column. This offer expires March 31, 1979.

SHIPPING CHARGE★ ____________

STATE AND LOCAL TAX ____________

TOTAL PRICE★★ ____________

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

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.

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

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	*LBLD	012	*LBLE
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	*LBLa	006	*LBLA
002	R↓	007	Σ+
003	LSTX	008	R/S
004	Σ-	009	GTOA
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	*LBLA	(Start with y_2, ENTER, y_1)
002	-	
003	LSTY	
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	*LBL1
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	*LBLE	007	STO7
002	P⇄S	008	STO8
003	0	009	STO9
004	STO4	010	P⇄S
005	STO5	011	RTN
006	STO6	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 [x̄]! 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 Before	After
1. ISZ		
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 Before	After
1. DSZ		
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, Calif.

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 $\sqrt{x^2 y^3 - z}$ 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

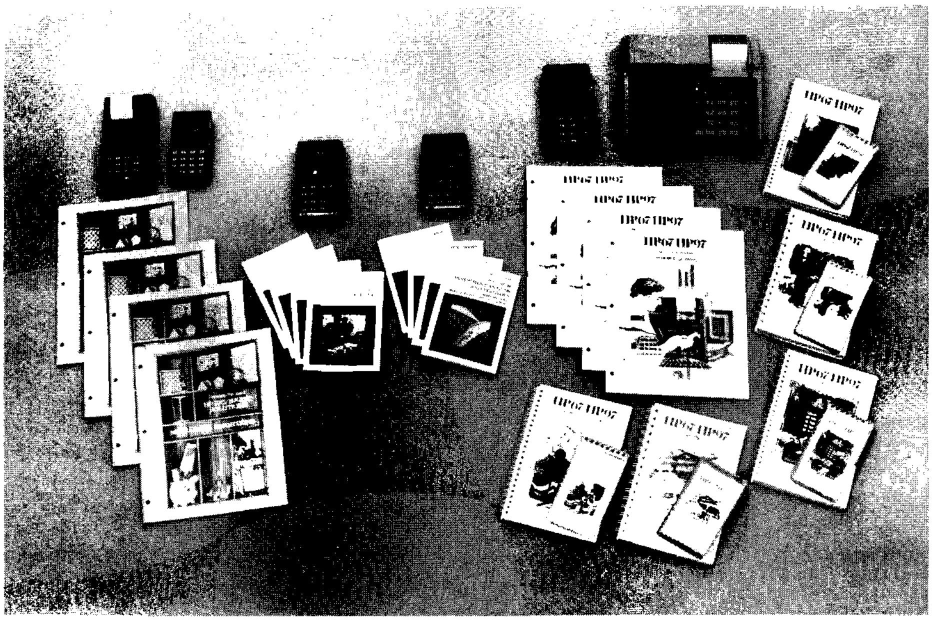
Pearl Robbins
Prentice-Hall, Inc.
Mail Order Sales Dept.
Old Tappan, NJ 07675

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 ad vanced 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

Printed in U.S.A.