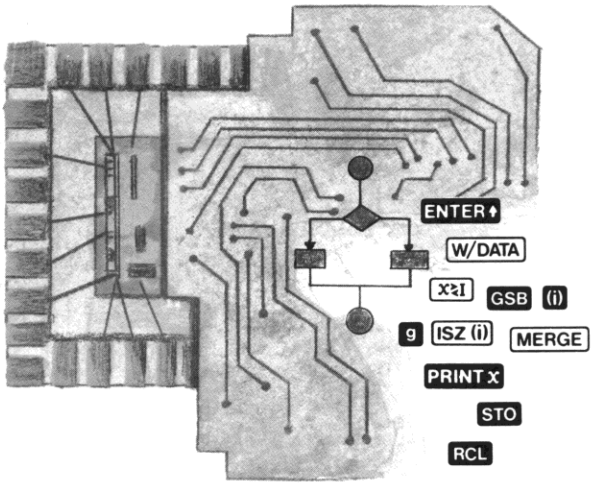


Featuring, this issue:

HEWLETT hp PACKARD

HP Key Notes

May 1978 Vol. 2 No. 2

New Design ... New Prices ... Series E is Here!

On April 17 Hewlett-Packard announced the new Series E calculators, an extensive low-end product line that presently includes five new machines.

Among the new features of Series E are:

- **New level of accuracy:** gives you confidence that your answers are correct.
- **New larger, "tilted" display:** makes it easy to read over a wider viewing angle.
- **New display format:** uses *commas* to aid legibility and number recognition.
- **New Error codes:** tell you *what kind* of mistake you made.
- **New self-test capability:** allows *you* to check the calculator for possible malfunction.

And there is much more, including a new modular handbook system. But you'll have to actually *see* these new calculators before you'll believe what they can do. It would take an entire issue of KEY NOTES to begin to cover the many advances in Series E. Here is a short summary to whet your appetite!

HP-31E SCIENTIFIC $60* A basic preprogrammed calculator with 4 storage registers.

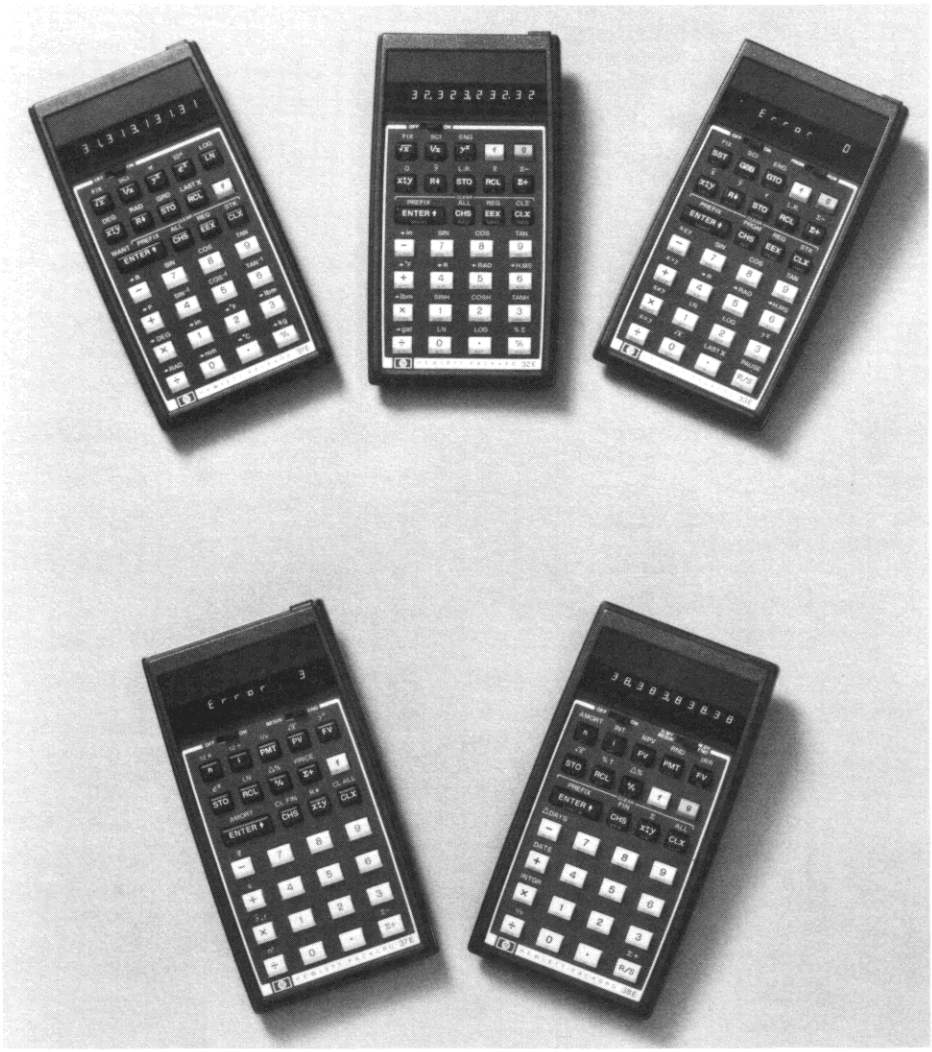
HP-32E ADVANCED SCIENTIFIC WITH STATISTICS $80* An advanced, preprogrammed calculator with a powerful set of statistics functions and 15 storage registers.

HP-33E PROGRAMMABLE SCIENTIFIC $100* A programmable calculator with 49 fully merged program lines and 8 storage registers.

HP-37E BUSINESS MANAGEMENT $75* A basic calculator for business and finance, with financial, retail/percent, and statistical functions and 7 storage registers.

HP-38E ADVANCED FINANCIAL WITH PROGRAMMABILITY $120* Our first programmable financial calculator, with up to 99 fully merged lines of easy programming, plus IRR for 20 groups of cash flows with up to 99 cash flows in each group, a 2,000-year calendar, and up to 20 storage registers.

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



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

Library Corner

As of April 13, there were 2,311 programs in the HP-67/97 Users' Library in Corvallis. It seems they come in bunches, and we are at present facing another large bunch to review, log into the system, and add to the list.

HP-67/97 LIBRARY NEWS

In the last issue we told you that *Library Catalog Addendum 2* would be mailed in April. Let's make that "about May 15." It isn't a pleasure to move the date back, but paper shortages, printing problems, and interstate problems beyond our control have delayed the mailing by one month. On a positive note, the addendum will contain programs from #00918D through #01800D, and we know that you will spend many hours reading abstracts about some really fantastic programs. As you gain ever more experience and expertise, the quality and variety of programming moves from merely "excellent" to "sensational." Keep up the great work!

Another item that will be of interest to you is that subscription renewal letters for HP-67/97 Library members will be mailed starting about the end of May.

Don't forget that there are now *two* ways to buy a Library program from Corvallis: (1) the regular $3* fee for only the program and (2) a $5* fee for the program *and* a prerecorded card.

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

HP-65 LIBRARY NEWS

As you know, the HP-65 Users' Library is closed for submittals and remains at 5,554 programs. However, through an unfortunate oversight, two programs that should have been included were mislaid. This matter is being corrected by publishing the abstracts below. Thus, there are *actually* 5,556 programs from which to choose.

Also, another milestone to remember: The original HP-65 Library Catalog supply has been exhausted and will not be reprinted. However, we still have supplies of Catalog Addenda 1, 2, and 3.

65 Normal/Lognormal, Parameters (#05555A)*

Computes and stores the mean, variance, standard deviation, and standard error for the normal (M,S^2, S,SM) and lognormal (MU,SIG^2, SIG, SIGMU) distributions (given the data XI, I=1 to n). It also computes and stores the median (EXP(MU)), coefficient of deviation (EXP(SIG)), and coefficient of deviation of the mean (EXP(SIGMU)). Negative values and zero are discarded for the lognormal computations. Registers are protected from accidental destruction. (199 steps)

Author: **Frederick A. Olson**
Camillus, New York

**In European areas, order by number 51646A.*

65 Capacity of Curb Opening Inlets (#05556A)*

This program will compute the depth of flow and spread of a known quantity of water flowing in a street, also the capacity of curb opening inlets on a continuous grade, giving the inlet length required to intercept the entire gutter flow. It also will design or analyze known inlet length. The program is used by the Montgomery County Department of Transportation and the Washington Suburban Sanitary Commission (both located in Maryland. (100 steps)

Author: **Philip A. Hendrick**
Wheaton, Maryland

**In European areas, order by number 51647A.*

ORDERING PROGRAMS

Any program you see in HP KEY NOTES can be ordered either from the Users' Library in Corvallis, Oregon, or from the Users' Library in Geneva, Switzerland. (Both addresses are on the back cover.) For most of the world, use the program number listed next to the program's title, then order it from Corvallis. The only exception is if you live in the European areas; in that case use the number listed in italic type below the program abstract, then order it from Geneva. Payment for programs must conform with the instructions from *your* area Library. **Always use order forms if possible,** and be sure to include any state or local taxes.

NEW HP-67/97 PROGRAMS

Here are a few new programs you won't find in the new Addendum 2. Before you order *any* of these programs, be sure you read the paragraph on **Ordering Programs.**

67/97 Lifetime Program (#02025D)*

This program estimates your lifetime in years. Given the answers to various questions in the areas of heredity, health, diet, exercise, education, occupation, and life-style, the calculator determines your expected lifetime. Starting with individuals 20 years old or over, the program first finds the base expected age from your sex and present age, and then a value is added to or subtracted from your base age, depending on the responses to questions. (223 steps)

Author: **Bruce G. Hansen**
Lansing, Michigan

(Fascinating! Juan Ponce de Leon should have had an HP-67/97; it would have discouraged his search. However, for me, the program turned out to be a veritable "fountain of youth!" Try it, you'll like it. And if you hate the answer, well ...you can always cheat a little! Ed.)

**In European areas, order by number 00293D.*

67/97 Three Gravitating Bodies in a Plane (#02161D)*

Given the masses, initial positions, and initial velocities for three bodies (e.g., stars) moving in a plane, this program will integrate their positions forward in time, using a straightforward algorithm. The program also computes the energy of a given configuration and it will adjust particle velocities so that the center of mass appears stationary. There are two cards; one sets up the configuration and the other integrates it. (430 steps)

Author: **Harold T. Coderre**
Princeton, New Jersey

(Very good, Mr. Coderre! The chart and graph help an already fine program. And your second one, below, also is a contribution. Ed.)

**In European areas, order by number 00294D.*

67/97 Fundamental Physical Constants (#02162D)*

This program outputs the values of nine fundamental physical constants in either C.G.S. or M.K.S. units. The user can call up the values of the speed of light, the charge on an electron, Planck's constant, Avagadro's number, the mass of an electon, the mass of a proton, Boltzmann's constant, the universal gravitational constant, or the energy of 1 electron volt. Only one storage register is used and the stack is undisturbed. (205 steps)

Author: **Harold T. Coderre**
Princeton, New Jersey

**In European areas, order by number 00295D.*

"25 Words or Less"

In the last issue, **James S. Hayden** (Edwards, California) suggested that we should add *this* column to HP KEY NOTES. Evidently our readers agree, because we received a lot of encouragement and many inputs for the column. It might not be possible to print all of them, but we'll try our best.

And, remember, don't limit yourself to 25 words if it takes more than that to explain your contribution.

Here are three routines from the first contributor, **Hugh Kenner,** of Baltimore, Maryland.

Y^x FOR LARGE NUMBERS: x ≒ y, LOG, x, FRAC, LSTX, INT, x ≒ y, 10^x, RTN. The result is in the X- and Y-registers in the form x EEX y.

TRIANGLE SOLUTION, ANGLE GIVEN 3 SIDES: Enter sides in stack to start: a ENT↑, b ENT↑, c. Then program: ENT↑, x^2, R↑, x^2, −, R↑, x^2, +, x ≒ y, ÷, x ≒ y, ÷, 2, ÷, arc COS, RTN. The display shows the angle opposite side a.

TRIANGLE SOLUTION, GIVEN 3 SIDES AND INCLUDED ANGLE: Start with side, angle, side in stack: a ENT↑, b ENT↑, c. Then program: x ≒ y, ENT↑, R↑, →R, R↑, −, CHS, →P, R/S (side B in display), CLX, π, R→ D, x ≒ y, R/S (angle A in display), R↑, +, −, RTN (angle B in display).

Now, two examples from **Shellman H. Brown,** Hyde Park, New York.

FLAG FLIP-FLOP: This routine avoids the GTO and LBL of the example shown in "Vector Operations" in the Standard Pac. Program: LBL E (any label), 1 (assume on), F?1 (test flag), CLX (indicate off), SF1 (assume on), x = 0 (test indicator), CF1 (turn off), RTN.

FLAG TEST: Mr. Hayden's simple flag test (Vol. 2 No. 1, page 1) ducks the difficult part: resetting of flags 2 and 3. Best I can manage is 25 steps; maybe someone else can improve it. This program is to be read in after g MERGE. Program: LBLe (load LSTx), ABS, CLx, F?0, GSB4 (display 0 if on), 1, F?1, GSB4 (display 1 if on), 2, F?2, GSB2 (display 2 if on), 3, F?3, GSB3 (display 3 if on), LSTx (restore x), RTN, LBL2, SF2, GTO4, LBL 3, SF3, LBL4, PAUSE, R↓, RTN.

And, for brevity, how about *this* one from **Frank P. Rust,** Salt Lake City, Utah.

PRINT-PAUSE ROUTINE: The following is shorter and saves one label over the print-pause toggles used in the Standard Pac programs. For a 0 or 1 input: LBLE, SF1 (sets flag 1), x = 0? (if input was 0, then clears flag; otherwise, leaves flag set), CF1, RTN.

(However, the above routine requires User inputs; the Standard Pac routine does not. Ed.)

Here is one that was once supplied by our Customer Support to **Richard Scott** (Fairbanks, Alaska) and now is coming back to KEY NOTES readers.

COMPARING THREE VALUES TO FIND THE SMALLEST: Assume A, B, and C in registers 1, 2, and 3. Program: RCL2, RCL1, x ≤ y, GTO1, X ≒ Y, LBL1, RCL3, x ≤ y, RTN, x ≠ y, RTN.

From **Valerie Van** (Sacramento, California) comes this neat routine.

THE LAZY BONES ADDING MACHINE: I gave it that title because it shows one possible use of Data Entry Flag F3 (which is set to "true" when a number is keyed in), coupled with the pause for data entry. The routine works best for the HP-97, because the sound of the printer prompts you to key in your next number. If you fail to do so, you are given extra time in the form of two more PAUSE's. If you *still* don't respond with a number, the routine gives you the current total and stops. The summation is resumed by keying a new number followed by [A]. Register 1 must contain zero initially. For the HP-67, omit the PRINT and SPACE instructions. Labels 1 and 2 aren't really necessary in this format of the routine, but were included to more clearly show the routine's division into subroutines. Program: LBLA, ST+1, PRINTx, CF3, PAUSE, F3?, GTOA, GSB1, RTN, LBL1, PAUSE, PAUSE, F3?, GTOA, GSB2, RTN, LBL2, RCL1, SPACE, PRINTx, RTN.

Around the world from Tel Aviv, Israel, comes this contribution by **Micha J. Schocken.**

INPUTTING/RETRIEVING VARIABLES/RESULTS: Please consider the following sets of subroutines that I have found useful for inputting/retrieving variables/results totalling up to 25 inputs and results, where assigning user-definable keys to various subroutines proved to be impractical. Program: LBLA, STI, R↓, STO(i), RTN, LBLD, STI, LBLe,* RCL(i), R/S, ISZ,* GTOe,* LBLE,* CLx,* STI,* GTOe.* To use this, key in the value to be stored, press [ENTER↑], key in the register to be used (say, 12) and press [A]. Continue until all inputs are keyed in. Then, to recall a register (say, 12) key in the number and press [D]. To see the next register, press R/S. To retrieve *all* registers in sequence, press [E] and then R/S for as many inputs as you made.

If program memory is overloaded, this 16-step set of subroutines can be reduced by omitting the steps marked * at the loss of only the retrieval-in-sequence feature.

This input is more like the "25 Words" part of the column title. Submitted by **Jon Brewster** (Corvallis, Oregon).

SAVE ONE STEP: Sometimes saving one step can save a program. If you are a surveyor, astronomer, navigator, or geometry buff, and you need to generate 180°, try: π, R → D.

The following routine will neatly store 24 values in the order in which they are keyed in. Donated to KEY NOTES by **Richard H. Baker** of Queen Creek, Arizona.

STORE-IN-ORDER: Program: LBLA, STOE, RCLE, STO(i), ISZI, RCLI, 2, 4, x=y?, GTO1, RCLE, RTN, LBL1, 0, STOI, STOE, RTN. To use, key in, say 24 [A], 16 [A], 83 [A], 38 [A], ..., etc. up to 24 values. The values will be stored in the registers in the order they are keyed in. Additional values cause the program to start over with R_0.

How about this entry by **T. R. Bainbridge** of Kingsport, Tennessee?

AUTOMATIC INITIALIZATION OF CONSTANTS: Program: LBLA (start), F0? (CF0 before recording), GTO1, SF0, ...(generate and store one or more constants), LBL1, ...(use constants in remainder of program), RTN. The routine generates and stores constants only the first time the program is run, when F0 is not set.

And **John B. Hart** (Cincinnati, Ohio) didn't make it in "less than 25 words," but he did add a good idea.

NOTHING BUT R/S: A card filled with R/S instructions is very useful when all steps below a given line in a program are to be deleted. Go to the last line to be saved in a program, key MERGE, and load the R/S-filled card. An R/S-filled card is created by running a card through the HP-67/97 just after it is turned on and switched to PRGM mode.

Now we switch to Pittsburgh, Pennsylvania, and this input from **Edwin J. Borrebach.**

SHORT-AND-FLEXIBLE: For inputting a number of constants into memory, such as the beginning of a program, the following subroutine is short and fully adjustable in length. Program: LBLA (initialize), 14 (or any number between 1 and 24), STI, LBL1, RCI, R/S (data input point. Press R/S to enter data), STO(i), LBLB* (data is okay), DSZ, GTO1 (program continuation is automatic after input of last piece of data).

*Having entered a set of data, if only part of the data is to be revised for another run, this permits a skip-over of those data input values that are not to be revised.

What's in an HP-67 Register?

We get a lot of register-checking routines for the HP-67, so we publish one now and then. But rarely are these routines foolproof, and so they elicit lots of mail—which takes many, many hours to answer. The problem is that it takes a terribly long time to fully check *all* the ramifications of a routine or a subroutine. You never can be sure what might precede it, causing all sorts of problems. Also, maybe we will forget to input oddball numbers, and then find to our dismay that a routine won't work with decimal numbers that start with .0000 etc.

Well, *this* routine looks bulletproof, and it has worked flawlessly ...so far! It is the contribution of **Paul Schüle** of Zurich, Switzerland. It displays the contents of all registers from 0 to 24 if they *do not* contain zero. There is no bothersome I-register display at the end. When you press [E], the number of each register is shown for 1 second, then the register contents are displayed for about 5 seconds. If a register contains a zero, the routine bypasses it. For repeated operation, all you need to do is press [E]. It works, and very well.

If you record the routine on a magnetic card, it can be appended by means of g MERGE to any program consisting of not more than 198 steps. Try it; it works wonderfully well. Key in the routine, switch to RUN and load anything but zero in all the registers, and then press [E].

Congratulations, Mr. Schüle. It appears that you have corrected all the faults of previous routines.

```
001   *LBLE        014    X=Y?
002       0        015    GTO7
003    STOI        016    DSP0
004   *LBL9        017     PSE
005    RCLi        018    RCLi
006    X≠0?        019    DSP2
007    GTO8        020    PRTX
008    ISZI        021    ISZI
009    GTO9        022    GTO9
010   *LBL8        023   *LBL7
011       2        024       0
012       5        025    STOI
013    RCLI        026     RTN
```

A New Almanac For Calculator Users

Thanks to **P. Kenneth Seidelmann,** a director of the Nautical Almanac Office (U.S. Naval Observatory), here is news about the *Almanac For Computers*. It was designed for use with small electronic computers and pocket-sized calculators. Instead of the familiar tabulations at short, fixed intervals, this almanac presents astronomical data in the form of efficient polynomial series, valid for prescribed

(Continued)

time spans. The navigational section contains short representations of the Greenwich Hour Angles (GHA) and declinations of the Sun, Moon, and navigational planets and the GHA of Aries. These functions are represented to navigational precision (±0′.1) as power series of fifth degree, which can easily be evaluated with programmable pocket calculators.

An applications section contains many formulas and algorithms of general utility in navigation and astronomy.

In the astronomical section, Chebyshev series represent such data as the apparent right ascensions and declinations of the Sun, Moon, and planets, nutation in longitude and latitude, sidereal time, and the equation of the equinoxes. Since a small computer is required to evaluate the series efficiently, these data are available in machine readable form. An algorithm for evaluating Chebyshev series is provided in the almanac.

Finally, a list of 176 stars is included. In addition to mean places at the beginning of the year, this list provides data for simply calculating apparent places during the year to a precision of ±0′.1.

The 1978 edition of the *Almanac for Computers* can be obtained by sending a check for $3.00 payable to the U.S. Naval Observatory; 34th and Massachusetts Ave., NW; Washington, DC 20390.

Tip for E.E. Pac Owners

If you have our *E.E. Pac 1,* you'll enjoy—and make use of—this neat improvement to one of the programs.

Dear Sirs:
HP-67 Users who have utilized EE1-18A, "Bilateral Design: Gain and Stability Circles, Load and Source Mapping," may be interested in the following helpful tip.

Besides computing constant gain output circles, the program can readily compute input or constant transducer gain circles. Simply interchanging only the S11 and S22 data entry on card EE-17A, while not altering the data for S21 and S12, allows this computation. No alteration of the program is necessary. This modification of data-input entry extends the usefulness of the program to optimization of transducer gain and noise figure.

To use this tip, load card EE1-17A, then enter S11 data; angle, magnitude, 22. Next, enter S22 data; angle, magnitude, 11. S12 and S21 are left unchanged. Now, load card EE1-18A and compute *input* gain circles.

Sincerely,

Alan Victor, Cooper City, Florida

You're Never Too Young...

Do you remember your 8th birthday? If programmable calculators had existed when you were that age, do you think you could have programmed one? Or do you think that 8 years old is too young to start programming? And if you have guessed by now that we are leading up to something, you're right. We are.

Here is a most remarkable account about an 8-year-old youngster in Israel who can program a 223-step game even though he hasn't yet learned to spell all of the words of calculator jargon we so freely toss around. And the program works; we checked it carefully. Also, he did it by himself, which is all the more startling.

Following is an excerpt from a letter his father wrote to our Product Marketing Manager, then the abstract for the program.

Dear Sir:
Our eldest son, Zvi, has become hooked on programming. It's funny, but he took to it like a fish to water and spends almost all of his free time working out programs for all sorts of things and, for his age (8), really seems to do very well. He's been dying to write a program good enough for the Library, and it is hard for me to explain that it would not be taken. I do want to encourage him, so I promised to send this program (all completely his own work) to HP. It would be super if you could take the trouble to just drop him a note from HP

Sincerely yours,

David Schreiber, Jerusalem

67/97 Noughts-and-Crosses (01884D)*

The game of noughts-and-crosses *(also called tic-tac-toe, Ed.)* is played by two players; one writes noughts (zeros), the other writes crosses (x's) in any of the nine squares in the familiar pattern: #. Players 1 and 2 take turns and "write" a nought or a cross by using the [1] through [9] keys on the keyboard. The HP-67/97 will record the move internally but will not show the move in the display. Thus, the other player cannot see which square was chosen. And if a player tries to play twice in a row, the calculator will catch the double move and flash a 0.000 display. If a player tries to put a cross or nought in a square that already contains one, the number (of that square) flashes in the display. When one player gets three noughts (or crosses) in a row, the calculator indicates which player won and which squares were used.

(Congratulations, Zvi, for a sensational job and for proving what all of our Users already know: HP calculators are easy to program. Ed.)

Author: **Zvi Schreiber**
Jerusalem, Israel

**In European areas, order by number 00296D.*

What Is A "Quality" Program?

Why are *some* programs highlighted in KEY NOTES? Why do *some* programs you buy seem (ahem) "less perfect" than others? Well, after seeing over 7,000 programs in the last 4 years, we can make a few observations.

Most really "good" or "excellent" programs get that attribute because the author has taken the time to *carefully* document the problem, the solution, and *how* it was derived. Examples, for example, make all the difference in the world. Also, just because there are only five sheets in a program submittal package does not automatically limit you to only five sheets. Don't attempt to cram a complex matrix algebra program on one sheet—or five sheets—when six or eight are necessary. And keep in mind that some people have HP-97's (or HP-67's) and program accordingly.

Try to realize that someone else will have to interpret your program, perhaps someone who doesn't know the subject matter as well as you do. So, when you find an obscure solution to a tricky problem, reference it! Take the time to show how you got from step A to step B. Don't be afraid of *too much* detail; it can be invaluable to the neophyte and is easily "skimmed over" by the professional.

When necessary, list the definition of symbols used. Show the derivation of equations and algorithms. Flow charts and diagrams are invaluable to *your* compilation of the program; don't short-change the final submittal by leaving them out. Also, a concise and usable sample problem(s) usually will get most people acquainted with the subject so they will be able to *apply* the program. And, last but not least, make the program listing totally useful by carefully listing all register contents and completely describing all labels.

Remember, when *everyone* submits well-documented, high-quality programs, all of *you* are the beneficiaries, not the Library.

Now that we've gotten *that* off our chests, we'll treat you to a real treat. We'll tell you about a whole bunch of "excellent" programs—all by one author. All are neat. All are precise. All are eminently usable. All are totally and fully documented. So we know it can be done. They are the work of **Bruce K. Murdock** of Santa Barbara, California. His LNAP program (01585D) (00252D in Europe) in the last issue was extremely popular. Too bad we don't have enough space to print all of the abstracts. His first nine programs, on filter design (Butterworth, Chebyshev, etc.), appear in your Library Catalog Addendum #1 (Corvallis Library, only), and most are in the *Users Library Solutions* book "Butterworth and Chebyshev Filters."

Here is a list of Mr. Murdock's latest programs, ending with one of the best business programs we've seen in a long time. To order programs in European areas, use the number in blue ink.

NETWORK ANALYSIS

67/97 LNAP; Ladder Network Analysis Program* (21 pages) #01585D (00252D)
67/97 LC-LNAP; LC Ladder Network Analysis Program (18 pages) #02259D (00297D)
67/97 LC-LNAP, Z_{in}; LC Ladder Network Input Impedance (14 pages) #02260D (00298D)
67/97 Lossy Transmission Line Input Impedance (11 pages) #02258D (00299D)
67/97 Voltage Along a Lossy, Loaded Transmission Line (8 pages) #02257D (00300D)
67/97 Second Order Active Network Pole and Zero Polynomial Coefficients (26 pages) #01859D (00301D)

CIRCUIT DESIGN

67/97 Bilateral Transistor Amplifier Design Using S Parameters (10 pages) #02256D (00302D)
67/97 VHF Oscillator Design Using Scattering Parameters (20 pages) #02255D (00303D)
67/97 Transistor Configuration Conversion (9 pages) #02254D (00304D)

MAGNETIC COMPONENT DESIGN

67/97 Inductor Design (Iron Core)—Magnetics* (15 pages) #01516D (00305D)
67/97 Inductor Design—Wire Size Calculation* (8 pages) #01515D (00306D)
67/97 Straight Wire and Loop Wire Inductance Calculation (9 pages) #02169D (00307D)
67/97 Single-Layer Solenoidal Air-Core Inductor Design (10 pages) #02168D (00308D)
67/97 Multilayer Solenoidal Air-Core Inductor Design (17 pages) #02167D (00309D)
67/97 Cylindrical Solenoid Design (19 pages) #02165D (00310D)
67/97 Cylindrical Coil Solenoid Analysis (21 pages) #02170D (00311D)
67/97 Magnetic Reluctance of Tapered Cylindrical Sections (8 pages) #02166D (00312D)

MATHEMATICS

67/97 Elliptic Integrals and Functions* (6 pages) #01616D (00313D)
67/97 Bessel Functions and FM or Phase Modulation Spectra (12 pages) #01850D (00314D)
67/97 Curve Fitting by the Cubic Spline Method* (13 pages) #01391D (00315D)
67/97 Multiple Linear Regression for Two Independent Variables (8 pages) #02173D (00316D)
67/97 2 × 2 Complex Matrix Operations, Part 1 (12 pages) #02171D (00317D)
67/97 2 × 2 Complex Matrix Operations, Part 2 (9 pages) #2172D (00318D)

NAVIGATION

67/97 Cable Catenary Scope and Cable Touch-down Latitude and Longitude (7 pages) #02253D (00319D)
67/97 Cable Cross Track Error and Distance To Go (7 pages) #02252D (00320D)

BUSINESS

67/97 Real Estate Investment Analysis* (9 pages) #00927D (00321D)

**These abstracts will be in Addendum 2. See "Library Corner."*

HP-67/97 "Ersatz" Continuous Memory

Translated from German to English that means "Substitute Continuous Memory." And when you read the following letter from Belgium, you'll know why we used that title!

Dear Editor:

When working in the field with my HP-67, I have found the following program useful to save the stack, the registers, and the program whenever it is necessary to change the battery pack.

First, take three blank magnetic cards and label the first one DUMP PRGM, the second one DUMP STK, and the third one DUMP REG. Next, on side 1 of the DUMP STK card, write the following program.

"DUMP"	STO 2	"LOAD"
LBL A	R↓	LBL B
W/DATA	STO 3	0
CL REG	LAST X	ENTER↑
P→S	STO 4	RCL 4
CL REG	2	+
STO 0	-x-	RCL 3
R↓	W/DATA	RCL 2
STO 1	CLX	RCL 1
R↓	DSP 0	RCL 0
	R/S	R/S

Now suppose the low power LED indicator glows while you are in the middle of a running program, and you *must* change batteries. Don't panic, it's easy!

1. Stop the running program.
2. Switch to W/PRGM.
3. Feed in side 1 of the DUMP PRGM card, then side 2 if needed.
4. Switch to RUN.
5. Feed in side 1 of the DUMP STK card. (The one on which you have written the above program).
6. Press [A].
7. The display shows **Crd.** Now, feed in side 1 of the DUMP REG card, then side 2 if **Crd** is still displayed.
8. The display will show a flashing 2, then **Crd** again. Feed in side 2 of the DUMP STK card.
9. The display will show 0, to tell you that the calculator may be switched off.

Replace the battery pack. Then, to reload:

1. Switch to RUN.
2. Feed in sides 1 and 2 of the DUMP STK card.
3. Press [B].
4. Feed in the DUMP REG card and then the DUMP PRGM card.

The HP-67 has no function that gives access to the current value of the program address, therefore it is necessary to manually introduce this address before resuming program execution. Thus, you will have to know where your program stopped, then use the GTO .nnn function to return to that point. This must **not** be done if the calculator was not running when you initiated the "dumping" procedure.

I always carry these three cards, in the same holder with my cleaning card and my diagnostic card. And, although very elementary, I think this procedure could be of some help to your readers.

Yours sincerely,
Pierre Flament, Brussels.

Now you see why we used that title. And, although the HP-67/97 truly cannot retain a program in memory when shut off, its "smart" card reader gives you an *ersatz* Continuous Memory!

Relieve Congestion in Your I-Register

Dear Editor:

This letter is about HP-67/97 indirect storage, relieving congestion in the I-register, and saving steps in long and/or involved programs. In my struggle with the multiple use of the I-register, such as indirect recall inside a DSZ loop, it frequently occurred to me that it would be nice to have two or more indirect registers. Therefore, with the following routines, any register can be used indirectly for storage and recall. They do use the I-register, but this is not apparent because the I-register is restored to its original value.

Let's suppose it is desired to use register 5 indirectly with instructions like RCL (5)* and STO (5), where RCL (5) means recall the register specified by register 5 and STO (5) means store the contents of x in the register specified by register 5. Routine 1 will implement the RCL (5) instruction.

	RCL (5)	Final Stack Contents
1.	LBL 1	T = (5)
2.	RCL 5	Z = y
3.	x ≒ I	Y = x
4.	RCL (i)	
5.	x ≒ y	X = RCL (5)
6.	x ≒ I	
7.	R↓	x = original
8.	RTN	contents of X, etc.

Note that T does not contain z but the contents of register 5. All other registers are

(Continued)

undisturbed. Steps 5 and 6 restore I to its original value.

Similarly, routine 2 implements STO (5).

	STO (5)	Final Stack Contents
1.	LBL 2	T = (5)
2.	RCL 5	Z = z
3.	x ≒ I	Y = y
4.	R↓	X = x
5.	STO (i)	
6.	R↑	
7.	x ≒ I	
8.	R↓	
9.	RTN	

Of course, any primary register can be used for this purpose, not just register 5. But things become interesting when the RCL 5 instruction (step 2 of routines 1 and 2) is replaced with RCL (i). This causes routine 1 to RCL ((i)), that is, recall indirectly from the register specified by the I-register. This is a powerful instruction, but alas, the I-register is burdened with the task of continually pointing to the register desired for indirect recall. The same problem exists for routine 2 when it becomes STO ((i)).

One recourse for this predicament is to write a routine to RCL ((x)) instead of RCL ((i)). Surprisingly, routines 3 and 4 are only 2 steps longer than 1 and 2, respectively. Routine 3 recalls indirectly from the register specified by X.

	RCL ((x))	Final Stack Contents
1.	LBL 3	T = (x)
2.	x ≒ I	Z = z
3.	RCL (i)	Y = y
4.	x ≒ I	X = ((x))
5.	R↓	
6.	RCL (i)	
7.	x ≒ y	
8.	x ≒ I	
9.	R↓	
10.	RTN	

Note that the original value of X (the pointer) is replaced by the recalled number, and that the T-register is modified.

Routine 4 stores the contents of Y indirectly into the register specified by X.

	STO ((x))	Final Stack Contents
1.	LBL 4	T = (x)
2.	x ≒ I	Z = x
3.	RCL (i)	Y = z
4.	x ≒ I	X =y
5.	R↓	
6.	{↓	
7.	STO (i)	
8.	R↑	
9.	x ≒ I	
10.	R↓	
11.	RTN	

The original contents of Z and Y are moved down for subsequent calculations, displacing the pointer (which ends up in Z). Again, T is modified.

As an example of routine 4, put 6 in register A, 1 in T, 2 in Z, 576 in Y, and 20 in X. Execute GSB 4, storing 576 indirectly in register 20 (register A). Since A contains 6, 576 is stored in register 6. Now check the stack and then RCL 6. To illustrate routine 3: Modify register 6 with STO+6, key in 20 and GSB 3 to recall 1152 (the number in register 6). Register 20 has been used as if it were another indirect register.

Keying in 20 before each subroutine call can be expensive in a program, so use the smaller numbered registers. This expense in steps can be traded for one register (like E, for example) by inserting RCL E as the first step following the label in routines 3 and 4. This gives RCL((e)) and STO((e)) routines, where E points to the register to be used indirectly.

In summary, routines 1 and 2 allow a particular primary register to act as a register for indirect storage and recall. Routines 3 and 4 allow *any* register (specified by x) to act as such. By adding one step to routines 3 and 4, the job of specifying this indirect register is given to a different (primary) register, instead of X. In all six routines the I-register is restored to its original value and the LSTx register remains intact throughout, just like the normal STO and RCL instructions.

I hope others find these routines helpful in structuring the use of the I-register, as I have. Perhaps with the "new instruction set," programs with still greater capability are possible.

Sincerely,

Emerson J. Perkins
Huntington Beach, California

**(Very good, Mr. Perkins! We'll add but one note: Some folks might not know that computer symbology you use. For example, R is a register; (R) is the contents of register R; and ((R)) is the contents of register (R). Ed.)*

The Replete Traveler

Although we have not gotten complaints about the type of problem outlined in the following letter, it is entirely possible that such a thing could happen. So if you travel a lot, perhaps you should heed the writer's warning.*

Dear Sir:

Since my job entails considerable travel, my HP-65 calculator and I pass through many X-ray machines at various airports. While the X-rays may not affect the magnetic tapes during a single pass, I found that after several trips the quality of the recordings begins to deteriorate. For some unknown reason this was more prevalent with those I recorded myself, rather than the pre-recorded ones.

My solution is to always carry my calculator and tapes inside one of the bags used to protect camera film. The bag I use is called *Filmshield* and is manufactured by Sima Products of Lincolnwood, Illinois. However, most any camera store carries it. This bag is lead-and-barium-lined vinyl that easily holds my calculator. Since using this bag I have had no further problems.

I believe this information may be of interest to other HP calculator users who travel throughout the country. The use of a bag of this type is considerably easier than having to re-record the tapes after several trips.

Sincerely,

George R. Fisette, Hackensack, NJ.

**(It is probable that the tapes are being altered by a magnetic device,* not *by X-rays. However, the letter was printed here to warn you that magnetic devices used by airports can, and will, alter the magnetic particles on a card and thus destroy your recorded program. Ed.)*

A Notation From Poland

Occasionally, someone will forget to press one of the prefix keys and find, to their amazement, that the correct key code or function appeared in the display. For example, check page 232 in your *HP-67 Owner's Handbook* (page 207 for the HP-97), and you will find that the (i) and RCL (i) keystrokes are described as performing the same function. Try it. (However, STO (i) remains a two-key code on both calculators.)

When we designed the keyboards for the HP-67/97 calculators, a few decisions had to be made. The keyboards, for example, were quite different, so the keycodes necessarily had to be different. However, we did not want the two machines to be imcompatible. Thus, a few "oddities" exist. Many of you have noticed these "shortened" keycodes and have written to us about them. The first letter was from Konstantly Boufal, of Warsaw, Poland, who no doubt had some "expert" help when he found the keystroke-saving codes, as Mr. Boufal also sent a photo to show us his "assistant," probably the world's youngest HP fan and User. In his words: "My lovely daughter, Magdalen, 8 months old, has perfected one operation on my HP-67 ...CLx!"

In w/PRGM mode on the HP-67 and HP-97, you can save one keystroke for GSBa commands. On the HP-67, instead of g GSB f a for 32 22 11, *you can always press* f a and still get 32 22 11. And so on through GSB b, c, d, and e. On the HP-97, instead of GSB f a for 23 16 11, *you can always press* f a and still get 23 16 11.

This trick also works in W/PRGM mode for GSB A through E. On the HP-67, instead of

f GSB a for 31 22 11, *you can always press* A and still get 31 22 11, **except for the first step of program memory.** It will not work then because the default functions are automatically reset when you turn on the calculator. On the HP-97, instead of GSB A for 23 11, *you can always press* A and still get 23 11.

These are handy tricks when editing a program because they save keystrokes. But don't use them if they confuse you. Remember, the factory-authorized keystrokes were devised to prevent errors and to save having to memorize a lot of rules.

There are a few other keystroke-saving codes, but since they involve being useful *only* at the first step of memory, they tend to be impractical and cause more errors than benefits.

For "Mark-Elangelos"

As part of a continuing effort to try to please everyone, here are some more products found by readers to be good card-marking devices. We cannot guarantee their success, but we do know that they worked for the following people.

Gentlemen:

I have found a better marking pen than the SHARPIE, which has a blunt tip and produces relatively coarse markings. A much better result can be obtained from the Schwan STABILO Pen 96P (fine). I found this pen advertised for the purpose of permanently marking magnetic cards. It can be obtained in many colors. I have the following source information:

Schwan Pencil Co., Inc.
221 Park Avenue, South
New York, NY. 10003

Distributed by:

Federal Sales Service, Inc.
1008 North Randolph Street
Suite 104
Arlington, VA. 22201
Phone (703) 525-5215

The set of four colors (red, blue, green, and black) is called No. 9607P *Fine Tip Pens.* It sold some time ago for about $3.

Regards,
H.P. Stratemeyer, Hollis, NH.

Gentlemen:

I believe I've stumbled onto another way of temporarily marking magnetic cards. Although other brands probably are available, I've found the 0.5 mm Pentel PF335 lead for "film" to be superior to graphite for marking cards on a temporary basis.

I understand that this lead is designed to write on mylar stock used in drafting work, and consequently it smears much less than ordinary pencils. I've found markings made with this lead to resist smearing by greasy fingers. (I went so far as to rub one marked card with butter and alcohol.) However, the markings come off very easily with a conventional pencil eraser, but I prefer the "plastic" type of eraser.

Sincerely,
Larry Browning, W Lafayette, IN.

We also have run across another finer-tipped permanent marker. It is marked: *PILOT SC-UF Ultra Fine Point Permanent,* and it can be obtained in most office supply or artist/drafting supply stores. It might even come in colors.

Of course, not everyone has trouble marking cards. For instance, here's a letter we received *over two years ago!**

I have been reading with some surprise about the problems of marking program cards. In my naiveté, I didn't realize that a problem existed. Since the only fine-tipped pen I owned is a *CASTELL TGH* No. 0, which was filled with *CASTELL TG BLACK MAGIC* ink, that is the one I used.

No surface preparation was used prior to writing and no coating was applied afterward. One card, recorded on both edges, has been run between 300 and 400 times and is still perfectly legible.

Sometimes ignorance is bliss.

Sincerely yours,
J.C. Hanselman, San Diego, CA.

**(Actually, 2 years and 8 months ago. Right, Mr. Hanselman? And I'll bet you thought I had ignored your letter or forgotten it! And what do I use? Okay, I'll confess. Since I get samples either from you or from the stores, I use* all *of them. Nothing like being a nonpartisan, right? Ed.)*

Flags Revisited

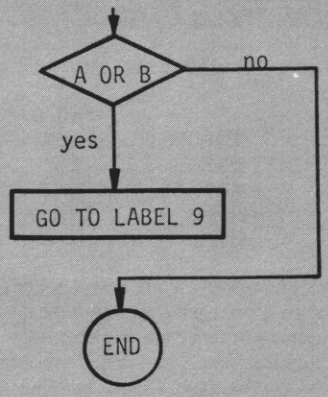
In the October 1977 issue (Vol. 1 No. 3) we printed an article, "On Understanding Flags," by **William M. Kolb** of Annapolis, Maryland. Some people did not understand the table accompanying that article, so here is an explanation graciously furnished by Mr. Kolb.

The table lists various keystroke sequences which are the equivalent of the logic equation at the top of the column. The equation $\bar{A}$ (not A), for example, can be programmed at least three different ways:

F3?	F0?	CFO
F3?	F3?	F1?
	F0?	F0?
	F3?	

Which sequence you use will depend on how many steps you are trying to save and what flags are available in your program. The question marks used in HP's documentation after all tests were left out to save space in the table. Two logic equations appear at the top of each column. One is the opposite of the other and is listed for convenience. If you prefer to think about the conditions under which you must skip a step rather than the conditions under which the next step is executed, use the second equation to find the appropriate keystroke sequence. The letter in parentheses after some keystrokes tells you which flag represents which variable in the logic equation.

These explanations may be easier to follow with an example. The letters A and B will represent two conditions that are to be tested by some undetermined keystroke sequence. In this example, we will determine if either condition A or condition B is true at the end of our main program. If it is, we will branch to LABEL 9; if neither condition is true, we'll halt and end the program. The flowchart for this logic would look like this:



The program corresponding to this flowchart has some unknown steps combined with the two desired results:

```
*
*
*
?
?
GTO 9
RTN
```

The table is designed to help you fill in the missing code required. In this example, we want the non-skip case whenever A or B is true. Reading across the top line of the table for the non-skip equations, we find "A OR B" listed and choose any of the sequences shown; e.g.,

```
F0 (A)
SF1
F1 (B)
```

The program is now completed by inserting these keystrokes:

```
*
*
*
F0?
SF1
F1?
GTO 9
RTN
```

We must remember to include whatever code is necessary to set flag 0 when condition A is true and flag 1 when condition B is true. In this particular example, we could just as well have selected an alternative keystroke sequence and used flags 0 and 3. The code would be:

```
*
*
*
F3?
F3?
F0?
GTO 9
RTN
```

Suppose condition A had the following context: Set when new data is entered, clear otherwise. Condition B has this context: Set when R8 is non-zero, clear otherwise. At some point in the program, it will be necessary to test R8 and set flag 0 if it is not zero. If R8 is in the X-register at the time of the test, however, we can avoid using

(Continued)

flag 0 by substituting x≠0 for flag 0 (condition B). Both methods are illustrated:

*	*
*	*
RCL 8	*
x≠0	*
SFO	*
*	*
*	RCL 8
F3?	F3?
F3?	F3?
F0?	x≠0
GTO 9	GTO 9
RTN	RTN

Note that we didn't worry about setting flag 3 for condition A because it is automatically set when new data is entered via the keyboard.

In general, flag 2 and flag 3 are interchangeable within the table. Flag 0 and flag 1 are also interchangeable and may often be replaced by a relational test as in this example.

Where-To-Find-It List

Because it is frustrating to be unable to acquire for your calculator an accessory you need (or want) in a reasonable time, we are starting a new service that will alleviate this problem. As of the first week in June, a list of all dealer outlets in the U.S. that participate in our Accessory Promotion Plan will be available to our WATS operators (phone number 800-648-4711, or in Nevada, 800-992-5710). The list also will be available to all of our Sales Offices, and to our Customer Support group and the Service Center at the factory in Corvallis, Oregon. The telephone number for the factory is 503-757-2000, **and this is not a WATS line.**

The dealers on this list will have committed to continually keeping in stock those batteries, rechargers, thermal paper, software, and blank cards needed to adequately support our calculators. And to keep this service as good as possible, we would appreciate feedback on how well this service works.

HP KEY NOTES

May 1978 Vol. 2 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
Editor, HP KEY NOTES
1000 N.E. Circle Boulevard
Corvallis, Oregon 97330 USA

Hewlett-Packard S.A.
Users' Library
P.O. Box 349
CH-1217 Meyrin 1/Geneva
Switzerland

Double Stick Revisited

Dear Sirs:

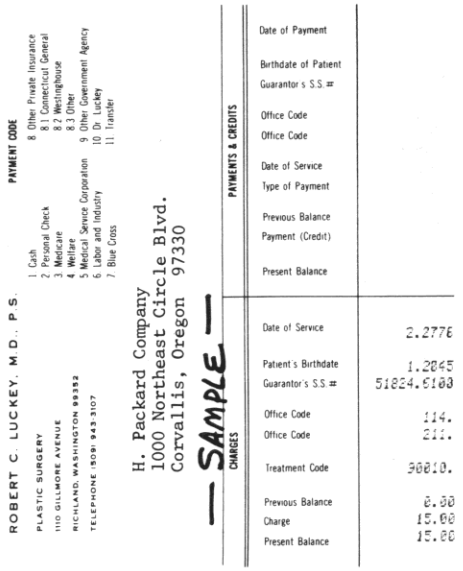
The February 1978 issue (page 8) referred to the use of *Scotch*® Double Stick tape for mounting HP-97 program tapes. It did not mention use of the *Scotch*® C-12 double thick tape applier. This makes perfect application of the tape a reality. We use the HP-97 for all statement receipts, ledger cards, and ledger book entries.

Sincerely,

Robert C. Luckey, M.D.,
Richland, Washington

Thanks for the information, Dr. Luckey, and thanks for the sample ledger card, which is reproduced here so that other readers can see how handy the HP-97 tapes can be for business purposes.

®*Scotch is a registered trademark of the 3M Company.*



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 CE PAC

Program CE1-02A2, "Section Properties," has been found to contain an error. If I_x equals I_y you will not get a correct answer. To correct your book, add the following steps. On page L02-04, after step 178 (RTN), add:

*LBL8	X⇄Y	*LBL9
X⇄Y	X=0?	0
9	GTO9	RTN
0	X⇄Y	
CHS	RTN	

On page L02-03, after step 66 (x≠0?) add:

x = 0? and GTO 8

To receive a revised card (CE1-02B2), **you must mail your old card to: HP Service Department, P.O. Box 999, Corvallis, Oregon 97330.** This new card will be available on May 29.

Program CE1-04A, "Stress on an Element," also contains an error. You will not get a correct answer when S_x equals S_y. To correct your book, add the following steps. On page L04-02, after step 164, add:

*LBL8	X=0?	÷
X⇄Y	GTO9	GTO7
9	x	*LBL9
0	LSTX	0
X⇄Y	ABS	GTO7

After step 125 (TAN^{-1}) add: LBL 7. Replace step 123 (x≠0?) with: x=0? and GTO 8.

To receive a revised card (CE1-04B), **you must mail your old card to: HP Service Department, P.O. Box 999, Corvallis, OR 97330.** This new card will be available on May 29.

Address Correction Requested
Return Postage Guaranteed

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

Printed in U.S.A.