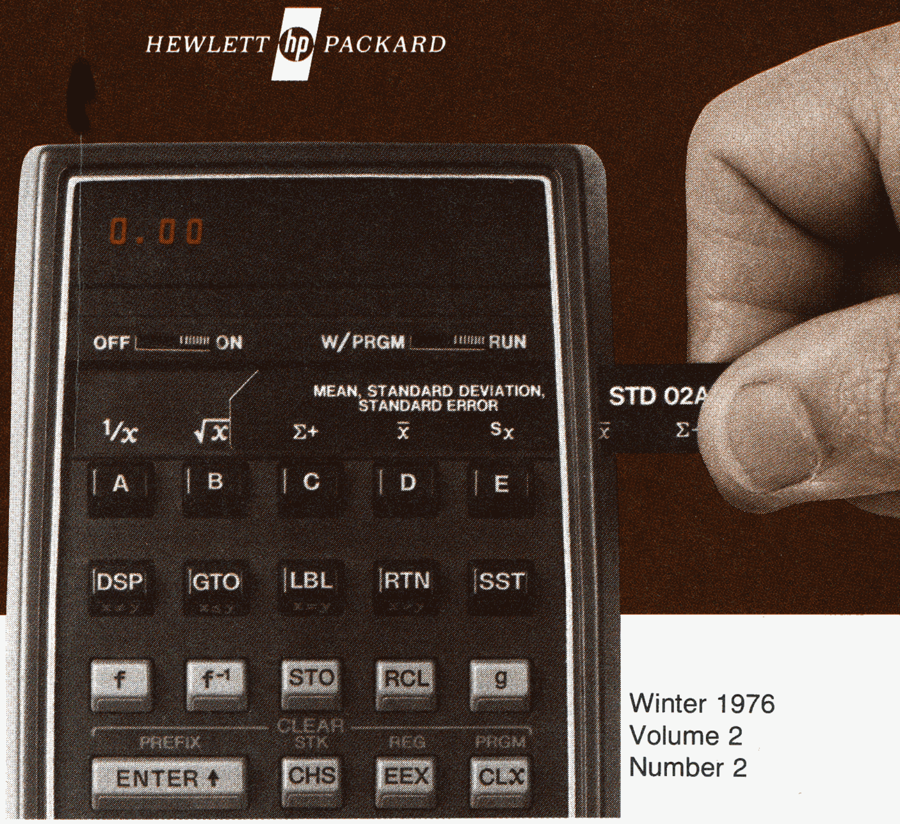
HP-65 KEY NOTE

for HP-65 owners

HEWLETT hp PACKARD



Winter 1976
Volume 2
Number 2

Inside This Issue

Just a Note...

Just before we went to press, we saw the March 1976 issue of *Scientific American*. On page 88 there is an article you don't want to miss. It is titled "The Small Electronic Calculator." If you see the magazine, there's *one* thing you definitely *won't* miss: the cover illustration, which is part of the keyboard of the remarkable HP-65!! We also liked the last line of the article, and so will *all* of you.

"Cogo" Is Ready!

The surveyors among you—and we know there are a substantial number—are in for a real treat. **Carl M. King,** of

Sarasota, Florida, has completed (well, almost) a Herculean task that he calls "The COGO-6500 Series of Coordinate Geometry Calculations for Surveyors."

These COGO programs constitute a comprehensive and self consistent series that solves all the CoOrdinate GeOmetry relationships regularly employed by surveyors and plat designers. The raw data from the surveyors' field notes are reduced to rectangular coordinates for ease of plotting and for detection of errors. From any given *pair* of coordinates, you advance a direction and a distance to a new *point*, identified by a new *pair* of computed coordinates. *All* the programs in this COGO series have certain features in *common*, so that you can switch from one program to another without losing the thread of your calculations. The *bearing traverse* and the *deflection angle* traverse are common to all. The *inverse traverse* is shared by seven of them. When proceeding around a *closed* traverse, along straight legs and circular arc segments, they automatically accumulate the *area*. In addition to the *common* options, each program has a *specialty*. You'll find the first eight programs, COGO-6501 through COGO-6508 listed as Library programs #02825A through #02832A in the new supplement to your Catalog. The two latest programs, recently submitted, are:

COGO-6509 Crandall's Rule Adjustment, Bearing, Deflection Angle, and Inverse Traverse (#04172A) (293 steps)

COGO-6510 Transit Rule Adjustment, Bearing, Deflection Angle, and Inverse Traverse (#04173A) (255 steps)

At the end of each traverse leg, the data is assembled in the "stack" in the same order in each COGO program, so you "roll" it down one step at a time to read out the answers you desire.

About his latest effort, COGO-6509, Mr. King said: "This is definitely my most ambitious program to date. It should serve as an excellent introduction to my entire series of COGO programs. My COGO series teaches a complete and self-contained system of surveying computation that is vital to the plat designer. Boundaries, streets, lots, and legal descriptions can all be calculated most conveniently using my COGO programs."

Mr. King also said about COGO-6509: "I was encouraged to develop this one after having obtained a copy of **Mr. Robert Tatge's** program #03487A. I extend my thanks to Mr. Tatge for demonstrating the practicality of making these calculations with the HP-65. I had been acquainted with the formulae in the reference that he quotes, but I had previously been discouraged from trying it, since it looked so formidable in the text."

As you can see, Mr. King has done an extraordinary job on these programs. It is a pity that lack of space prohibits a detailed explanation of just how complete these programs really are. Perhaps Mr. King's own words best express *our* sentiments:

"As you can readily appreciate, this has been a labor of love. The Crandall's Rule Program is but a recent addition to an evolving sequence. I have gone back and revised several of the earlier programs, and I have totally revised the written instructions for all of them. I am exceedingly pleased with my latest round of revisions, and I wish to call it all to your attention, since I believe it is time that surveyors begin to enjoy the fruits of my work."

Library Corner

As of February 23 there were 4,214 programs logged into the HP-65 Users' Library. But that's not the *really* good news! **You are no longer restricted to the minimum order of $9.00 (three programs). Any program in the Library may be ordered separately;** each has a nominal charge of $3.00 (including those listed below).

NEW INCENTIVE PLAN

Up to now, for each program you submitted to the Library, you received a free program of your choice plus 10 blank magnetic cards. Here's *more* good news! **You now have a choice of two incentives: (1) One free program + 10 blank cards + one plastic card case or (2) Five free programs.** And it is very likely that you'll soon hear more good news from the Library. Keep your eye on this column.

You'll notice a lot of games listed in this column this issue. Why? For some unknown reason, there have been many fine game submittals this new year. Was it the Christmas spirit? Is there more "calculator keying-time" during the long winter evenings? Anyway, we think you'll enjoy this batch of new programs.

If possible, use an order form from the catalog to order these programs. **Use the program number shown here.** Send only checks or money orders, payable to Hewlett-Packard Company. Be sure to include any state or local taxes.

NEW PROGRAMS

Havana—Bingo (#04236A)

This program serves as a nonrepeating random number generator for the games of *Havana* or *Bingo*. For *Havana*, the range of numbers is 1 through 32 plus joker. Players bet on individual numbers or combinations to appear before the joker. *Bingo* uses the numbers 1 through 39. Each player has one or more *Bingo* cards on which he/she attempts to complete a row, column, etc. before any other player does so. (137 steps)

Author: **Randy A. Coverstone**
Cambridge, Massachusetts

Euclid (#04238A)

Euclid is a game played by the user and the HP-65. At the start of the game the calculator selects two random numbers. The user and the HP-65 then alternate turns subtracting some multiple of the smaller number from the larger. The winner is the one who reduces the larger number to zero. The calculator plays flawlessly and is very careful to allow only fair play on the part of its opponent. (100 steps)

Author: **Randy A. Coverstone**
Cambridge, Massachusetts

Section Properties by Coordinates (#04210A)

This program calculates area, x and y coordinates of center of gravity, and moments of inertia about the x and y axes for an area of any shape. The shape is defined by a series of points connected by straight lines, each point having an x and y coordinate, and can be located in any quadrant; i.e., they can have + or − coordinates. There is no limit to the number of points entered. (100 steps)

Author: **John A. Apostolos**
Sacramento, California

Pool on a Rectangular Pool Table (#04211A)

A "game" program that mathematically and accurately simulates a real pool game played with two balls. The game requires plenty of skill and is played by one or more players. Cards 1 and 2 compute everything that program #03841A does. Cards 3, 4, and 5 compute everything that program #03842A does for any rectangular table with six pockets. The person shooting can make the white ball rebound before colliding with the blue ball. Also, this program takes into account table friction and sidewall resiliency. (447 steps)

Author: **Steven B. Cohen**
Rego Park, New York

Three-Pile NIM (#04127A)

The classical three-pile match game. Three piles of matches are chosen. The player alternates with the HP-65, subtracting any number of matches from any single pile. An important feature is that the player who takes the last match *loses*. The three piles are displayed as a single number (e.g. 4.35) after each turn. (99 steps)

Author: **John K. Appledoorn**
Bernardsville, New Jersey

(Note: While similar to Gary Campbell's program #01467A, this has two significant improvements. First is the single-number display. Program #01467A held the number in each pile in the stack, and the stack had to be rolled to see the entire contents, which necessitated use of pencil and paper to keep track. Second, and far more important, the player taking the last match loses, *whereas in Program #01467A he wins. This may seem like a trivial difference, but it involves a reversal in strategy near the end of the game, and this makes the game surprisingly difficult to analyze. The author stated: "Mr. Gary Campbell, the author of #01467A, deserves to be commended. I had given up on NIM until getting his program. Still, I am prouder of my improvement of his program than of any program I have submitted before—including the two you published in KEY NOTE (3-D Sub Hunt and Win, Place, and Show)").*

Roast Goose (#04080A)

This program computes the optimum roasting time for a stuffed goose at 325°F from the weight of the bird. The program accepts the argument in decimal pounds, pounds and ounces, kilograms, or grams, and is intended for geese of 8

pounds (3.63 kg) or greater; goslings smaller than that will be rejected. The program also determines starting or finishing times for cooking your goose. (91 steps)
Author: **Sydney Z. Spiesel**
New Haven, Connecticut

Turing Machine (#04160A)

This program simulates a finite state programmable computing machine similar to a Turing machine. It has nine active states and a halt state. The first card runs the machine. The second converts between binary and decimal for input/output. (183 steps)
Author: **Charles D. Tallman**
Dallas, Texas

This abstract doesn't really do justice to the nature of the program. The Turing machine is an important element in the history of computing theory. Basically, a Turing machine is an exceedingly simple model that has all the logical capabilities of even the largest and most complex computers in existence. It was first suggested by the British mathematician A.M. Turing, in 1935. Turing went on to suggest that any function that is computable can be computed on a Turing machine, and no one has yet disproved this assertion. It is this assertion that lets us say the Turing machine has all the logical capabilities of the computer (or, of course, the HP-65).

The Maze (#03979A)

The Maze has 5 levels of circular passages, each enclosing the previous one. Each passage has 10 possible gate locations for entry into the next level; however, only one position per level is randomly selected. A clue to the general gate location pattern is displayed. Each player, in turn, guesses a gate location, and the maze level status is displayed. The first player to escape the maze wins with a display of flashing zeros. (138 steps)
Author: **Robert A. Plack**
Phoenix, Arizona

Not So Round

An article in the last issue has provoked a few letters. Since they contain much useful information and show the deep interest that HP-65 owners have in their calculators, we reprint them herewith for your pleasure.

Dear Sirs:

I take exception to some comments in your article, "A Well-Rounded Subroutine!", in the Autumn 1975 issue of KEY NOTE. (Not to mention the obvious typographical error that a 10.0 in R_3 will round to the nearest 0.0l.) *(You're right; it should have been 0.1. Ed.)* Necessity is the catalyst that causes people to look for *a* way to do something, usually by brute force, as Mr. Flye says, but not necessarily an easier or better way. In fact, the existence of *a* way may inhibit the search for an easier or better way. The real incentive is *challenge*. I don't mean to disparage Mr. Flye. He had a need and then proceeded to meet that need. Then he went on to do more important things, including communicating his research results.

However, I have no need for a rounding subroutine, so I read the article rather carelessly until I came to, "The subroutine will not accept 0...". That snagged my attention and, when I saw what Mr. Flye had to do to make the subroutine handle 0, I said, "There's gotta be a better way!" I was *challenged*.

Needless to say, there is a better way:

LBL	+	R_3 contains a scaling constant to determine the point of rounding.
B	f	
RCL 3	INT	
×	RCL 3	
f^{-1}	÷	
INT	RTN	
g LST x		

Further, there may be still better ways. I therefore *challenge* the readers of KEY NOTE to send in a better version.

P.S. I'll wager that before the year is over, at least five guys (or gals) will send in better versions.

Yours very truly,

John E. Bigelow
Oak Ridge, Tennessee

Dear Sir:

Mr. Robert Flye's subroutine can be shortened and still accept zeros. Assume n is the number of decimal places desired. Two options are:

Use 10^n in R_3 as the rounding factor:

LBL	23	.	83
C	13	5	05
RCL 3	34 03	+	61
×	71	f	31
0	00	INT	83
g x>y	35 24	RCL 3	34 03
CLX	44	÷	81
1	01	RTN	24
−	51		

17 steps: *saves* 6 steps and a stack register. Or, if RCL 3 is replaced by EEX, n, you have 19 steps, which *saves* 4 steps plus R_3 and a stack register.

Use 10^{9-n} in R_3 as the rounding factor:

LBL	23	g LST x	35 00
C	13	−	51
RCL 3	34 03	g LST x	35 00
+	61	+	61
g LST x	35 00	RTN	24
−	51		

11 steps: *saves* 12 steps and a stack register. Or, if RCL 3 is replaced by EEX, 9^{-n}, you have 12 steps, which *saves* 11 steps plus R_3 and a stack register.

Yours truly,

David B. Westcott
Islington, Ontario, Canada

(Notice the similarity between Mr. Westcott's last routine and, later on, Mr. King's first routine. Ed.)

Gentlemen:

Mr. Flye's requirement for a subroutine that will round either positive or negative numbers and that will not fail when presented with a zero can be met in another way that uses considerably fewer program steps. I am sure that he will be happier with it, since program space is always at a premium.

The method that I am about to suggest utilizes the rounding capability inherent in the calculator. You merely add a large number to the number to be rounded, and the unwanted digits are forced into "overflow." When the same large number is subtracted, the answer has been rounded as desired.

The said large number must have as many digits after the decimal point as you wish to have remain in your rounded numbers. The best choice that I have found is:

$$N = 2 \times 10^{(9-n)}$$

Where N = the large number that you store in a convenient register,
and n = the number of digits that you wish to retain after the decimal point.

The factor 2 is apparently necessary to make the routine work for negative numbers as well as positive numbers. To parallel Mr. Flye's first example, in which instance he wished to round to the nearest 0.01, I would store N as follows: Press [2], [EEX], [7], [STO], [3]. Then, in the program, I would employ the following subroutine:

LBL	23
C	13
RCL 3	34 03
+	61
RCL 3	34 03
−	51
RTN	24

Or, at a cost of only two additional steps, you can save storage register R_3 for other uses, and the subroutine would be as follows:

LBL	23
C	13
2	02
EEX	43
7	07
+	61
g LST x	35 00
−	51
RTN	24

The rounding is either up or down to the nearest indicated digit, but always in the positive direction for the exact ½ digit. In this particular case it requires a deficiency of −0.00501 in order to favor the negative case, which I imagine is the way the rounding is done even in the large calculators.

Sincerely,

Carl M. King
Sarasota, Florida

Dear Sir:

The rounding subroutine contributed by Robert Flye in the latest issue of HP-65 KEY NOTE (Autumn 1975, Volume 2, Number 1) can be shortened to 13 steps by using the integer-fraction command twice.

(continued)

LBL	23	+	61
A	11	f	31
RCL 1	34 01	INT	83
×	71	RCL 1	34 01
ENTER↑	41	÷	81
f^{-1}	32	RTN	24
INT	83		

Register 1 contains the rounding factor; i.e., 10 for the nearest .1, 100 for the nearest .01, etc. This subroutine can handle positive and negative numbers and zero. Also, when the need arises to round to the nearest integer exclusively, this subroutine may be shortened as follows:

LBL	23	+	61
A	11	f	31
ENTER↑	41	INT	83
f^{-1}	32	RTN	24
INT	83		

Here is a handy, although lengthy, program that rounds to the nearest 5:

LBL	23	.	83
E	15	5	05
1	01	+	61
0	00	g x≤y	35 22
÷	81	5	05
f	31	g NOP	35 01
INT	83	5	05
STO 6	33 06	+	61
g LST x	35 00	LBL	23
f^{-1}	32	1	01
INT	83	RCL 6	34 06
.	83	1	01
2	02	0	00
5	05	×	71
g x>y	35 24	+	61
GTO	22	f	31
1	01	INT	83
		RTN	24

This routine can handle positive numbers and zero.
Yours truly,

William A. Chadburn
Casa Grande, Arizona

Gentlemen:
In "A Well-Rounded Subroutine!" in the Autumn 1975, Volume 2, Number 1 issue of KEY NOTE, Robert Flye has a problem accepting zero in his subroutine. A trick used in my program #02242A will overcome his problem with just a few additional steps inserted between LBL C and ENTER. The steps are:

EEX	43
CHS	42
9	09
9	09
+	61

No approximation is involved as the very small number added is eliminated by the round-off function.
Sincerely,

H.C. Mingst
Vallejo, California

Gentlemen:
Your recent HP-65 KEY NOTE publication shows an 18-step reader program that rounds numbers but will not handle zeros. You recommend a 23-step program that does handle zeros. Why not:

LBL	23	.	83
C	13	5	05
0	00	+	61
g x>y	35 24	f	31
RCL 1	34 01	INT	83
+	61	RCL 1	34 01
−	51	×	71
RCL 1	34 01	RTN	24
÷	81		

If rounding is to be in tenths, put 0.1 in STO 1, if it is to be in thousandths, put 0.001 in STO 1, etc. It will handle zeros. Also, it saves six steps.
Sincerely,

James G. Harrington
Helena, Montana

Well, **Mr. Flye,** you started something! And Mr. Bigelow's letter might draw *more* erudite and/or exotic inputs. Also, we want to thank all of you who wrote about this article; your letters came later than these and more or less duplicated the programming shown here.

A Bruised .093

In the middle of the right column on page 4 of the Autumn KEY NOTE (Vol. 2, No. 1), there is a letter with two short programs written by **Bob Edelen** of Aurora, Colorado. If you tried to run the programs, you know by now that we really goofed. There are at least three errors, and we owe it to Mr. Edelen to print a corrected version, which appears below. We've already written a letter of apology to Mr. Edelen, and herewith publicly apologize for our typographical errors.

"In response to the many ways to generate Fibonacci numbers, below is one of the shortest.

HP-65	HP-25
W/PRGM	PRGM
f PRGM	f PRGM
g LST x	01 f LAST x
g x⇌y	02 x⇌y
+	03 +
RTN	04 f PAUSE
RUN	05 GTO 01
	RUN
	f PRGM

Now, to load either machine with the first two terms:

(1) Key in the first term and press [+].
(2) Key in the second term.

For the HP-65, each press of [A] will give the next term. For the HP-25, press [R/S] and the calculator will generate the sequence automatically. For the standard Fibonacci numbers, use 1 for the first and second terms.

The routine is very well adapted to the HP-45, as it embodies only three steps: LST x, x⇌y, and +. These three steps are repeated for successive terms."

(If you don't know what my .093 is, key it into your calculator and turn the display upside down. Ed.)

How to Land on the Moon!

More than 350 HP-65 owners have purchased the *Moon Rocket Landing Simulator* (#00287A) program from the Users' Library. This program simulates landing a rocket on the moon. You are given an initial altitude, velocity, and fuel supply. By a series of rocket motor burns, you must try to land on the moon's surface at zero velocity before you run out of fuel. Now, at least 350 people sympathize with the problems of real astronauts. So let's see if we can use our calculator to figure out how to get down safely. However remember that we are **not** finding the most efficient way to land (i.e., minimizing fuel). We are finding a single burn that is to be keyed in at **every** step of the game and that will allow us to win.

We know that the acceleration experienced by our rocket is the sum of the moon's gravity g and the rocket's burn b (which is directed opposite to g). Thus, we can write:

$$a = (b - g) \qquad (1)$$

Integrating equation (1) with respect to time, we get an equation for the velocity:

$$v(t) = at + v_0 = (b - g)\,t + v_0 \qquad (2)$$

Integrating again, we get an expression for displacement

$$s(t) = \tfrac{1}{2}\,at^2 + v_0t + S_0 = \frac{(b-g)t^2}{2} + v_0t + S_0 \qquad (3)$$

in which v_0 and S_0 are initial velocity and displacement.

Now, in order to land our rocket, we must find some burn such that after a time T both v(T) and S(T) are zero. Setting equations (2) and (3) equal to zero, we have:

$$(b - g)T + v_0 = 0 \qquad (4)$$

$$\frac{b-g}{2}T^2 + v_0T + S_0 = 0 \qquad (5)$$

Solving equation (4) for T and substituting into equation (5),

$$T = -\frac{v_0}{(b-g)} \qquad (6)$$

$$\frac{b-g}{2}\,\frac{v_0^2}{(b-g)^2} - \frac{v_0^2}{(b-g)} + S_0 = 0 \qquad (7)$$

Multiplying through by (b − g), we have

$$\frac{1}{2}v_0^2 - v_0^2 + S_0(b-g) = 0$$

which can be solved for b.

$$b = g + \frac{v_0^2}{2S_0} \qquad (8)$$

Substituting equation (8) back into (6), we can compute the time it will take to land.

$$T = \frac{+v_0}{g - \left(g + \frac{v_0^2}{2S_0}\right)}$$

$$T = -\frac{v_0\, 2S_0}{v_0^2}$$

$$T = \frac{-2S_0}{v_0} \qquad (9)$$

Now, an additional complication must be considered: the HP-65 program is set up only to accept burn values that are integers. If equation (8) is solved for the initial values $g = 5$, $v_0 = -50$, and $S_0 = 500$, a burn of 7.5 results. Let's not burn any fuel at all and see if we wind up some velocity and height for which our formula yields an integral burn. Aha! After 5 seconds with $b = 0$, we have $v_0 = -75$ and $S = 188$, which gives $b = 19.96$ and $T = 5$. We land safely with 5 burns of 20 units each.

Notice that this leaves 20 units of fuel unused. The best descent **we** know of leaves 30 units unused.

Now, wasn't that easy?

Fibonacci Generator with Pause

We don't want to belabor the Fibonacci-number subject until you get sick of it; however, there *are* some very useful applications for these routines. And, below, is an absolutely fascinating program. Try it—you'll like it. Furthermore, it might present all sorts of other possibilities.

Dear Editor:

I enjoyed your last issue of HP-65 KEY NOTE very much—particularly the informative article about the weird displays. To illustrate the useful purpose to which registers A, B, and C can be put, I combined the information in your article with the one following it on Fibonacci series and wrote the program enclosed, which is a "Fibonacci Generator" with a programmed pause. You just press A and, after a brief multiplexing display, the successive terms in the series are shown with a 4-second pause for each term, and you don't have to touch the machine at all until you want to stop the generator.

The two constants in the program position valid results of calculations properly so that when combined with the invalid scratch in R_9, the pause display shows the valid numbers. Also, the 9.0^{-23} constant serves the double function of giving a reasonable pause duration while also blanking out what would otherwise be a distracting decrementing counter, thereby giving the pause display a perfectly stable appearance.

Fibonacci Generator With Pause

Key	Code	Comments
LBL	23	Initialization.
A	11	
9	09	This constant is used to blank
EEX	43	out what would otherwise be a
CHS	42	decrementing counter in the
2	02	pause display.
3	03	
STO 1	33 01	Store constant.
EEX	43	
CHS	42	
3	03	
2	02	
STO 2	33 02	Store constant.
1	01	
CHS	42	
f^{-1}	32	
COS	05	Prepare R_9.
1	01	
STO 3	33 03	Start Fibonacci series.
STO 4	33 04	
LBL	23	
B	12	
RCL 4	34 04	
RCL 3	34 03	
+	61	
STO 3	33 03	Next term in series.
g LST x	35 00	
STO 4	33 04	Last term in series.
RCL 2	34 02	Steps 29-32 position last
x	71	term of series in proper place
RCL 1	34 01	for pause display.
+	61	
RCL	34	
9	09	Recall scratch.
g NOP	35 01	Toggle display on.
÷	81	4-second pause.
B	12	Compute next term.

To run the program, load the card and press A. After a brief multiplexing display, successive terms in the Fibonacci series will be displayed sequentially, with a 4-second pause for each term. You don't have to touch the HP-65 at all after pressing A.

The display will look something like:

88. 80000 0.0.

Terms of the series will appear stationary in the columns of zeros that are underlined. Because only four digits are reserved for the pause function, this program can't be used for numbers larger than 9999. To stop the generator, press any key long enough for a pause function to be completed.

Sincerely,
Dick Hoppe
Casco, Wisconsin

(Well done, Mr. Hoppe. We are impressed! And you'll see another "weird display" if you let your "generator" run for a while, past the 9999 limit. Ed.)

Of Cards And Rounding

Art Leyenberger of Ridgewood, New Jersey, is a member of the *HP-65 Users' Club* that was mentioned in the last KEY NOTE. Although he usually contributes to that club, he felt the following tips would interest *all* HP-65 owners. We also liked the closing on his letter: "Happy Programming (is that what HP stands for?)." We're not sure Messrs. Hewlett and Packard would agree—but we'd bet they'd like that!

Dear Editor:

With the new blank program cards that are now available (the white ones), I use the extreme right area of the card to label any instructions that are used with RTN R/S. For instance, with certain games, I put an "s" there to remind me to enter a seed, then press RTN R/S. With a program I wrote to find Mean, Standard Deviation, and Standard Error of Grouped Data, I put an "init" in that location to remind me to always initialize the program first.

As reported by **Bob Edelen** in a past issue (V2N9) of 65 NOTES (the newsletter of the *HP-65 Users' Club*), the function f →D.MS does not round to the nearest second when converting from decimal degrees/hours. The following subroutine will take a value (in decimal hours) and convert it to Hours Minutes Seconds (using the standard format H.MMSS). It will round down if the decimal is less than 0.5 and round up if greater than or equal to 0.5. The routine takes 29 steps. Usually DSP · 4 is the mode for operation but if DSP · 6 is pressed, you will see a decimal fraction of a second displayed in the fifth and sixth decimal locations. Switch to W/PRGM and press f PRGM.

LBL	23	2	02
A	11	÷	81
f	31	STO	33
INT	83	+	61
STO 2	33 02	2	02
g LST x	35 00	RCL 3	34 03
−	51	f^{-1}	32
CHS	42	INT	83
6	06	.	83
0	00	0	00
×	71	0	00
STO 3	33 03	6	06
f	31	×	71
INT	83	RCL 2	34 02
1	01	+	61
EEX	43	RTN	24

Now, try 32.5 seconds. Switch to RUN, press DSP · 4, key in 32.5, and press ENTER↑. Next, key in 3600 and press ÷ to get decimal hours. Then use the routine—by pressing A. You should get 0.0033; the routine "rounded up."

Now, try 32.4999999 seconds. Key it in as above and use the routine. This time you should get 0.0032; the routine "rounded down."

Happy Programming,
L. Arthur Leyenberger
Ridgewood, New Jersey

Default Tip Defaulted?

Here is an interesting note about a tip in the last issue.

Gentlemen:

Regarding **Fred Goll's** "Default Program Tip" in the Autumn 1975 issue (Vol. 2 No. 1), another method to disable undefined labels is to simply use a R/S as the first line of any program.

Sincerely,
Eric V. Davis
Glendale, California

Are You In Rhythm?

Those of you who are not familiar with the theory of biorhythms, it postulates that every person has a 23-day cycle that correlates with physical vitality, endurance and energy: A 28-day cycle that corresponds to sensitivity, intuition, and cheerfulness; and a 33-day intellectual cycle that relates to mental alertness and judgement. All three cycles start in a positive direction from birth or the beginning of independent life.

Days when accidents are most likely to occur are the days on which the median line (0.00) is crossed. These are called "critical days." The high periods ($0 < x \leq 1$) are the times when you should have the most energy, be the most cheerful, outgoing, and mentally alert. The low periods ($-1 \leq x < 0$) are considered recuperative periods.

There is one restriction to this program. If the birthdate is before February 28, 1900, and result dates will be after February 28, 1900, enter the next day as the birthdate. For example, if you were born January 1, 1900, and wanted to compute biorhythms for today, enter January 2, 1900, as your birthday.

Why are we including an HP-25 biorhythm program in KEY NOTE? First, a large number of people have expressed interest in a biorhythm program. Second, there already is such a program in the Users' Library for the HP-65. Third, since a great many HP-65 owners also have an HP-25, we figured this was the best of two worlds.

HP-25 Biorhythms

Line	Key(s)	Line	Key(s)
01	f FIX 2	26	R/S
02	3	27	RCL 4
03	RCL 1	28	−
04	f x<y	29	STO 6
05	GTO 10	30	1
06	1	31	8
07	+	32	STO 7
08	RCL 3	33	5
09	GTO 16	34	STO + 7
10	1	35	RCL 6
11	3	36	RCL 7
12	+	37	÷
13	RCL 3	38	g FRAC
14	1	39	2
15	−	40	x
16	RCL 0	41	g π
17	x	42	x
18	f INT	43	g RAD
19	x⇌y	44	f SIN
20	RCL 5	45	R/S
21	x	46	GTO 33
22	f INT	47	1
23	+	48	STO + 6
24	RCL 2	49	GTO 30
25	+		

REGISTERS USED: R_0 = 365.25, R_1 = Month, R_2 = Day, R_3 = Year, R_4 = Birth Reference, R_5 = 30.6, R_6 = No. of days since birth, and R_7 = Cycle Time.

Program Instructions

Step	Instructions	Input Data/Units	Keys	Output Data/Units
1	Key in program.			
2	Store values.	365.25 30.6	STO 0 STO 5	
3	Store birthdate.*	M D Y	STO 1 STO 2 STO 3	
4	Compute N (M, D, Y) (birthdate).		f PRGM R/S STO 4	
5	Store result date.	M D Y	STO 1 STO 2 STO 3	
6	Compute N (M, D, Y) (today's date).		f PRGM R/S R/S R/S R/S	Physical Emotional Intellectual
7	To calculate rhythms for the current date + 1 (and subsequent dates).		GTO 4 7 R/S R/S R/S	Physical Emotional Intellectual
8	To calculate rhythms for a new birthdate, go to step 3.			

*If birthdate is before February 28, 1900, and result dates will be after February 28, 1900, enter next day as birthdate.

"Calculator Calculus"

No, this is not a dissertation on higher mathematics. The title of this article just so happens to be the title of a new book that a lot of you might be interested in.

The book, *Calculator Calculus,* by Professor George McCarty, came to our attention immediately because of its unique and colorful cover. The color rendition on the front cover is unmistakably a likeness of the HP-21.

The book sells for $8.95 and will be used immediately at UC Berkeley and Irvine. According to the publisher, Mr. Ficklin, 2,000 copies are being mailed to college faculty people thoughout the country who are teaching calculus courses. Mr. Ficklin expects an initial order for 10,000 copies, which will be sold through college bookstores. (Probably your best place to find one.)

The only direct reference to HP occurs on pages 245 and 246 (in the Bibliography). However, on page 224 the unmistakable HP ENTER key is clearly illustrated.

This spiral-wire-bound book is aimed at students. If you are interested in it, check your local college bookstore. The publisher is:

Page-Ficklin Publications
261 Hamilton Avenue
Palo Alto, California 94306

A Great Honor!

In the Summer 1975 issue (Vol. 1, No. 5), on the back cover, we ran an article about an HP-65 in the *White House*. Its owner, **Gus Weiss,** a Senior Staff Member at the *White House* and an economist on the Council on International Economic Policy, recently received quite an honor. On February 19, the French Ambassador conferred on Mr. Weiss the *Legion of Honor* for "arranging a joint venture between France and the U.S. on aerospace." It was quite an honor, and we know Mr. Weiss is thrilled to have received it. The *Legion of Honor* is a French order of distinction instituted in 1802 by Napoleon, with membership being granted for meritorious civil or military services.

In a telephone conversation with Mr. Weiss, he also said he wanted to thank the HP-65 owners who stopped to see him (and his "famous" HP-65) after his "debut" in KEY NOTE.

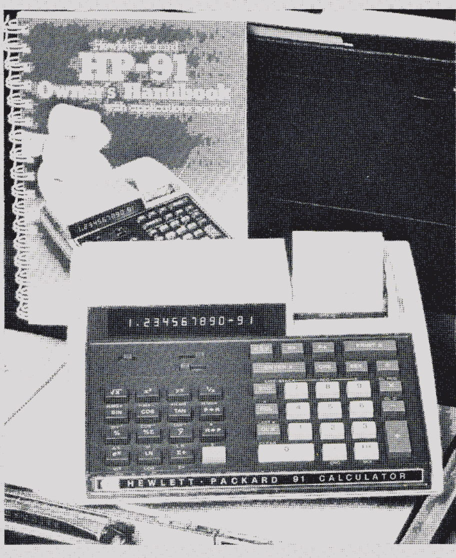
The HP-91 Is Here

On March 1 Hewlett-Packard introduced the first of a new generation of small, portable, printing calculators that operate on rechargeable batteries or line current.

The HP-91, priced at $500, is an advanced scientific calculator with a large, bright display (tilted for easy reading), and a buffered keyboard so data may be keyed in at high speed.

In addition to its scientific functions, the HP-91 includes operations found in other Hewlett-Packard calculators such as averaging and summation.

The 2½-pound calculator is a more powerful version of the widely used HP-45 pocket calculator, with the added capability of creating a printed log of all calculations. It has all of the preprogrammed functions of the HP-45, plus 16 addressable memory registers instead of 9; engineering notation; performs regression and linear estimates; has three percent-functions instead of two; and keyboard buffering seven keystrokes deep.



Four selective clearing operations and three selectable printing modes on the HP-91 contribute to ease of operation. The HP-91 has been designed with separate numeric and function keysets. This layout, coupled with the seven-character buffer, allows the "touch entry" trained operator to use the calculator without reducing normal speed.

An ac adapter/recharger, a carrying case, and a 220-page owner's handbook are included with the calculator. The handbook contains user instructions and a comprehensive application section that gives the most efficient keystroke sequences for solving problems in the fields of mathematics, statistics, finance, navigation, and surveying.

The new HP-91 is available for delivery two weeks after receipt of customer order. It is distributed through Hewlett-Packard and selected retail outlets.

Sun Almanac 1976

From **Tak Y. Lee** of Wellesley, Massachusetts, came this additional input about using the *1974-1975 Sun Almanac* during 1976.

"When the *1974-1975 Sun Almanac* program (#02187A) is used to compute the declination in 1976, the error as pointed out in KEY NOTE (Vol. 1, No. 5) may be approximately represented by a sine function:

48.5″ sin (t + 15°) or 0.1347° sin (t + 15°). Correcting the original formula used in program #02187A simply by combining this term with the cosine term leads to a new cosine term of fundamental frequency as:

$$-23.2659° \cos (t + 10.307°).$$

Since a common factor of −1/1000 has been factored out, the amplitude should be entered as 23265.9 herein. Declinations computed by such a modified program and sampled at 0^h on the first day of every month, with the original declination and higher harmonic terms unchanged, show a discrepancy of no greater than ±7″ throughout 1976—as compared to a published almanac.

On the other hand, the error curve of GHA is rather rich in harmonic contents: there is no simple modification to improve the accuracy for 1976 unless a much longer program is used."

(Mr. Lee is a steadfast and steady contributor to the Library. My personal thanks go to him for sharing this note with other HP-65 owners. Ed.)

65 Program Tip

Dear Sirs:

I have found a variation for the *Field Angle Traverse* program *(Surveying Pac 1, SURV 1-01B)*, which lends itself well to random traverse to set property corners at the same time.

In order to calculate the angle from the last leg of the traverse to the "side tie," the bearing of the last leg has to be known.

By altering the program, the bearing is computed without calculating an inverse. Press **GTO** **E** and single-step through program 1-01B to step #75; that is when the easting is displayed. Then insert the following steps.

RCL 1	34 01	→D.MS	03
f	31	DSP	21
TAN	06	.	83
f^{-1}	32	4	04
TAN	06	R/S	84
f	31		

The bearing is displayed. By comparing coordinates, the quadrant of the bearing is determined.

R.C. Berti
Placerville, California

A Wild Number!

You all know that an HP-65 has an extremely wide dynamic number range. Well, sooner or later, we knew that *someone* would do more than just *think* about the size of that range—or number!! And it sure happened! So, the next time you see that "overflow" indication in your display, you'll surely remember **Bill Wild** of Denver, Colorado.

"I was wondering how large $9.999999999 \times 10^{99}$ is. It's bigger than the volume of the universe in cubic microns!!!

According to the *1972 New York Times Almanac* (page 261), the radius of the observable universe = 10 billion light years.

$$10 \times 10^9 \times 186{,}000 \frac{\text{miles}}{\text{sec}} \times 60 \frac{\text{sec}}{\text{min}} \times 60 \frac{\text{min}}{\text{hr}} \times 24 \frac{\text{hr}}{\text{days}}$$

$$\times 365 \frac{\text{days}}{\text{yr}} \times 5280 \frac{\text{ft}}{\text{miles}} \times 12 \frac{\text{in}}{\text{ft}} \times 25{,}400 \frac{\text{microns}}{\text{in}}$$

$$= 9.44 \times 10^{31} \text{ microns radius}$$

$$\text{Vol} = \frac{4}{3}\pi r^3 = \boxed{3.52 \times 10^{96} \text{ microns}^3\text{ ''}}$$

"Shooting" Zeros!?

HP-65 owners never cease to amaze us. We knew you would enjoy the "display" article in the last issue, but we weren't prepared for such enthusiastic letters about it. Our thanks to all who wrote expressing their delight with that article. We'll try to do even better with the next one. Meanwhile, this will interest you.

Dear Editor:

Immediately after running a program designed to provide an interesting display (see "Why Does My Display....?" in the Autumn 1975 KEY NOTE), another extremely fascinating display occurred. The display flickered while "shooting" a zero every one second!!

I had just read the article, performed the steps listed, and amusingly watched the blinking display. Then, while reading the next article about Fibonacci numbers, I switched from RUN to W/PRGM mode to key in the program steps. At this time, the display, which was still blinking, produced a fast and furious display for a short while and then started "shooting" zeros across the display from right-to-left at precisely one-second intervals!

In addition, by switching to RUN mode, the display blinks at a fast clip with a burst of zeros every 5 seconds, resembling a spring stretching. Sounds incredible but it's true—just like the HP-65 itself.

Incidentally, the original blinking display program may be shortened to just

f	f^{-1}
PRGM	TAN
LBL	RCL 9
A	÷
1	

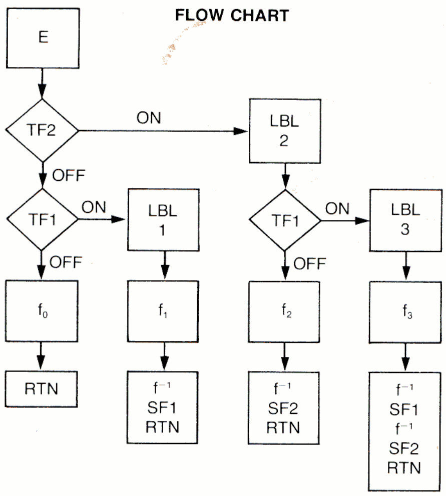
Yours very truly,
Vern R. Bunch
Baltimore, Maryland

Four For One

Although somewhat similar to material we have printed before, the following programming tip is unique enough to pass along to all HP-65 owners. It was submitted quite a while ago by **Thomas J. Maresca** of King of Prussia, Pennsylvania. At the time, we thought it was a duplicate of another published tip but have since determined it is not. We hope you forgive our mistake, Mr. Maresca.

"Here's how to calculate four functions using one user-defined key:

FLOW CHART



CALC	F2	F1
f_0	off	off
f_1	off	on
f_2	on	off
f_3	on	on

PROGRAM:

LBL E f TF 2 GTO 2 f TF 1 GTO 1	LBL 1	LBL 2 f TF 1 GTO 3	LBL 3
f_0	f_1	f_2	f_3
RTN	f^{-1} SF 1 RTN	f^{-1} SF 2 RTN	f^{-1} SF 1 f^{-1} SF 2 RTN
11	5	9	7

The numbers directly above indicate the subroutine overhead steps, they total 32."

Short And Sweet

Here's a suggestion that some of you probably hadn't thought of. It was submitted by **John M. Novak** of San Francisco, California.

"If a program contains more than 100 steps, a few steps can be keyed in manually *after* entering the program. If a R/S is added to the program, the excess steps can be keyed in at the middle or, if preferred, at the end of the program."

For example, after entering a card, press **A** to execute the program. Then, at the end of 100 steps, perhaps you might manually press **f** [LOG], or something to that effect. If the R/S were in the middle of your program, it would stop there, then you could press some keys, then continue the program. As you can see, there are many, many ways to use this idea.

Another Timer?

We thought by now that everyone had heard about the so-called "hidden" timer on the HP-45. But calls still come in about this mysterious trick. Yes, it's true; you can put an HP-45 in a timer mode. We are aware of this capability of the HP-45; however, it is not possible to calibrate or adjust the timer without physically altering the calculator (which we will not do). The HP-65, on the other hand, can easily be operated fairly acurately as a digital timer. There are several timer and stopwatch programs in the Users' Library. (Look in section 99.99.) Accuracy of the HP-45 timer is only ±10%.

Anyway, for those of you not "in the know," here's how you set the HP-45 into a timer mode. First, press **RCL**, then *simultaneously* press **CHS**, [7], and [8]. (This might take a bit of practice.) The "timer" can then be started and stopped by pressing **CHS**. Pressing **EEX** will blank out the "hundreds" digits but will not affect the rest of the display. Press **EEX** again and the "hundreds" digits return in the display. Pressing **CLx** will reset the timer to zero, running or not. Pressing keys 1 through 9 will enable you to record "splits," which can be displayed by stopping the timer and pressing keys 1 through 9. (It is not necessary to press **RCL**.)

HEWLETT-PACKARD COMPANY
Advanced Products Division
19310 Pruneridge Avenue
Cupertino, California 95014

HP-65 KEY NOTE

Winter 1976 Volume 2 · Number 2

Programming and operating tips, answers to questions, and information on new programs and developments. Published quarterly by Hewlett-Packard for owners of HP-65 Pocket Calculators.

Reader comments or contributions are welcomed. Please send them to the above address.

Address Correction Requested
Return Postage Guaranteed

BULK RATE
U.S. POSTAGE
PAID
PERMIT NO. 317
SUNNYVALE, CA

Printed in U.S.A.