



» [HP 10bII+ Financial Calculator](#)

Gene Wright

Learn how HP has upgraded the HP 10bII with a host of new and useful functions. Now business and financial students and professionals have a low cost machine suitable for all of their calculating needs.



Your articles



» [Tweaking the HP42S](#)

Jake Schwartz

Jake has spent a great deal of time analyzing HP calculators and his unique perspective on keyboard clutter and layout is applied to one of HP's more interesting machines, the HP42S.



» [Problem Solving Part II](#)

Richard J. Nelson

This second part explores the classic problem of Triangle solving and what is involved. Why hasn't there ever been a good, efficient Triangle Solver on a calculator?



» [What Is FOCAL?](#)

Richard J. Nelson

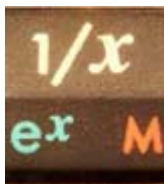
What is FOCAL and how did the HP User Community learn of this "HP announcement"? HP Solve asks you to provide any historical information that will help remove FOCAL from its urban legend status.



» [Upgrading the HP-41](#)

Monte J. Dalrymple

Read how Monte was able to redesign the Nut processor to utilize more modern technology for greater speed and greater memory with a project he calls NEWT.



» [Fundamentals of Applied Math Series #6](#)

Issue 23 April 2011

Welcome to the twenty-third edition of the HP Solve newsletter. Learn calculation concepts, get advice to help you succeed in the office or the classroom, and be the first to find out about new HP calculating solutions and special offers.

» [Download the PDF version](#) of newsletter articles.

» [Contact the editor](#)

From the Editor



Learn more about current articles and feedback from the latest Solve newsletter including RPN tips, One Minute Marvels and Math problem challenges.

[Learn more »](#)

Customer Corner

» [Meet an HP Calculator user](#)

Meet Monte Dalrymple, an IC designer based out of California. Read all about his past and present experiences as an HP calculator user.

Richard J. Nelson

Does e stand for earnings, exceptional, or exacting? The sixth installment follows up on the previous article in this series which discussed Logs and pointed out that most commonly used log bases are 2, e, & 10.

» [Update Profile](#)

» [Change Email](#)

» [Unsubscribe](#)

» [HP Home](#)

» [Support & Drivers](#)

If you received this e-mail from an associate and/or would like to receive email of this type directly from HP, please [click here](#).

HP respects your privacy. If you'd like to discontinue receiving these type of e-mails from HP, please [click here](#).

For more information regarding HP's privacy policy or to obtain contact information please visit our [privacy statement](#) or write to us at: Privacy Mailbox, 11445 Compaq Center Drive W., Houston, Texas 77070, ATTN: HP Privacy Mailbox.

Hewlett-Packard [website](#)

[Sign up](#) for Driver and Support Alerts.

The HP 10bII+ Financial Calculator

HP Solve #23 page 3

Article – [Next →](#)

The HP 10bII+ Financial Calculator

Gene Wright

What is it?

The HP 10bII+ is Hewlett Packard's revised entry-level financial calculator. It updates the functionality of the HP 10bII for the first time since the introduction of the original HP 10B back in January of 1989. These changes really increase the functionality provided on this entry-level financial calculator – so much so that calling it entry-level just seems wrong! The images below show the current HP 10bII on the left and the updated HP 10bII+ on the right. Packing all these new features into the HP 10bII is incredible!



Fig. 1 – HP 10bII (current)

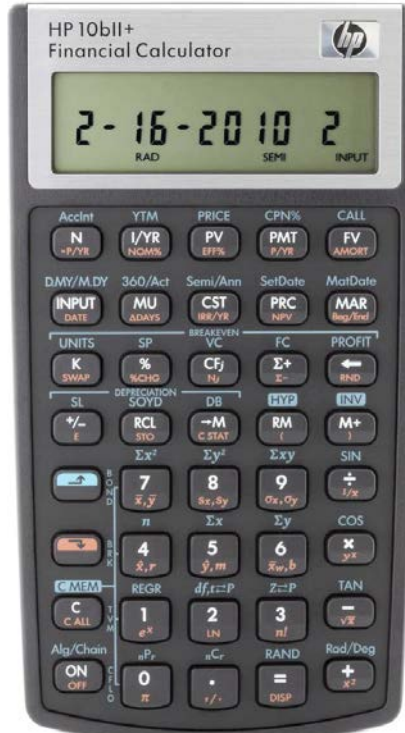


Fig. 2 – HP 10bII+ (new release)

The four images below show previous versions of the HP 10b and HP 10bII family including a never released version on the far right.

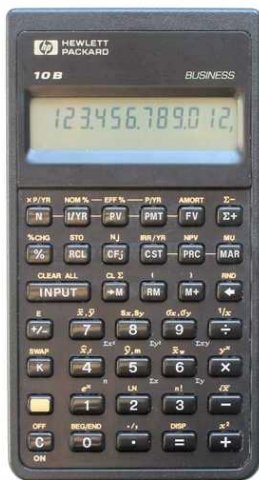


Fig. 3 – Early version



Fig. 4 – Ugly version?



Fig. 5 – Early 10bII

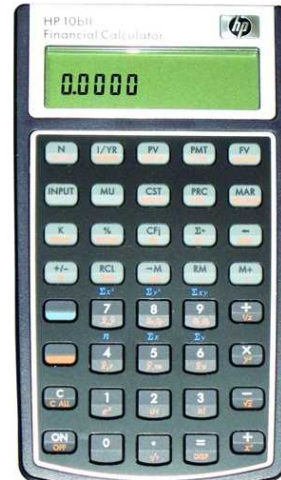






Fig. 6 – Unreleased version

Functional consistency

Despite all the added features, the HP 10bII+ still operates just like the HP 10bII. Care was taken to ensure compatibility with existing valid keystroke sequences. If pressing a series of keys is a valid key sequence and solves a problem on the HP 10bII, it will solve the problem on the HP 10bII+. Any existing books or classroom materials will work as written with no changes needed.

Physical differences

Two shift keys are now present: The familiar yellow for functions written on the front of the keys and a new light blue for functions written above the keys. The shift keys even show an arrow that actually indicates the “up” or “down” nature of the shifted functions,  . The silver band around the key area is gone. More display annunciators are present: two shift indicators, an indicator for the inverse key, indicators for the 360-day calendar mode and semiannual coupons, and an indicator for radians mode. The STATS indicators now indicate whether an X or Y value is being displayed. These X and Y indicators are also used in other circumstances, such as when a function returns a primary value to the display and a secondary value in the “SWAP” position. For example, computing the Net Present Value (NPV) also computes the Net Future Value (NFV) and stores it in the “SWAP” position. The X indicator is turned on when the NPV is being displayed. Pressing   will display the NFV and turn on the Y indicator. The layout of the screen is shown below.

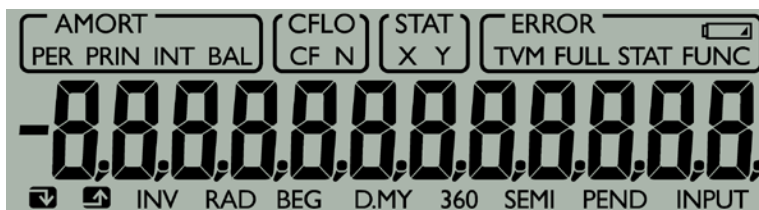


































Fig. 7 – HP 10bII+ annunciators

Chain and algebraic mode

The HP 10bII+ can be placed into either chain mode or an algebraic mode with hierarchy by the mode toggle . When pressed, the display flashes a message to indicate which mode is active. In chain mode,       returns a result of 9, while in algebraic mode, it evaluates to 7. This will reduce the need for parentheses use in many problems. Sadly, RPN is not present on this model. The operator stack has 13 locations available for a combination of operators and parentheses.

Trig and hyperbolic functions

The HP 10bII+ includes trigonometric functions and their inverse functions as well as the corresponding hyperbolic functions with their inverses. The angle mode can be set to degrees or radians using the mode toggle . The INV key is intelligent enough to know that if you just pressed  , you intend to compute an inverse sine if you follow those two keypresses with the  key, since the sine function is above it. This means that you do not have to press     , just    . Similarly, the HYP key can be pressed without additional shift key presses. To calculate the hyperbolic sine of 2, press    . To compute the inverse hyperbolic sine of the result, press     or    .

Bond calculations

The HP 10bII+ receives a nice upgrade by providing functions to calculate bond price and yield. A bond with a call provision is also handled nicely by changing the default value of 100 stored in the CALL register to the appropriate call value.

Date calculations

The HP 10bII+ now does date calculations in actual or 360-day calendar modes. To compute the days between two dates, key the first date, press $\boxed{\text{INPUT}}$, key the second date and press $\boxed{\text{DATE}}$. For a date given a date and a number of days, key the date, press $\boxed{\text{INPUT}}$, key the number of days and press $\boxed{\text{MUDAYS}}$. Alternatively, these two functions can also be inline operators where the first value is keyed, the function accessed, then the second value keyed followed by $\boxed{=}$. This approach works for all two-value functions other than arithmetic: they can be done as inline or “postfix” operators. Note that a date given a date and a number of days is always computed in actual days mode. Dates may be entered in M.DY or D.MY formats, depending on the $\boxed{\text{D.MY/M.DY}}$ mode setting. The screen image below shows a date in the M.DY format. The last digit on the right indicates the day of the week.



Fig. 8 – Date as shown in the M.DY format.

Breakeven calculations

The breakeven relationship including a target profit is calculated using the third row of blue-shifted functions as shown in the formula below. FC is the fixed cost, SP is the sales price per unit and VC is the variable cost per unit. This is very handy for the student.

$$\text{Breakeven in units} = \frac{(\text{FC} + \text{Profit})}{(\text{SP} - \text{VC})}$$

Depreciation calculations

The HP 10bII+ handles depreciation calculations in the same way that the HP 12c models do. The original cost is stored in $\boxed{\text{PV}}$, the salvage value is stored in $\boxed{\text{FV}}$ while $\boxed{\text{I/YR}}$ is used to hold the declining balance factor and $\boxed{\text{N}}$ holds the expected life of the asset. The user enters the year for which the depreciation is to be calculated and presses $\boxed{\text{DEPR}}$ followed by the $\boxed{\text{SL}}$, $\boxed{\text{SOYD}}$ or $\boxed{\text{DB}}$ function to compute straight line, sum of the year's digits, or declining balance method. The depreciation is returned to the display and the X and TVM indicators are turned on. Pressing $\boxed{\text{KSWAP}}$ will display the remaining depreciable value and turn on the Y indicator. The TVM indicator is on to indicate a TVM calculation is in progress and that TVM registers are being used.



Fig. 9 – Depreciation calculation in progress with TVM annunciator.

Enhanced statistics

The HP 10bII+ now has a random number generator, permutations, and combinations built-in. The random number seed can be set by keying a value and pressing $\boxed{\text{RCL}}$ $\boxed{\text{STO}}$ $\boxed{\text{RAND}}$. Having two key positions dedicated to directly compute permutations and combinations may not seem like a big deal, since the calculation using factorials isn't all that complicated. However, if a student is computing a hypergeometric probability calculation by hand (and yes, I used to make my students do this), such as

shown in the formula below, having these built-in rather than computing them using the factorial key will be a big time saver, since it is essentially three combination calculations in sequence.

$$P(X | N, X_t, n) = \frac{\frac{(N - X_t)!}{(n - X_t)!(N - X_t - (n - X_t))!} \times \frac{X_t!}{X!(X_t - X)!}}{\frac{N!}{n!(N - n)!}}$$

Fig. 10 - Hypergeometric probability distribution formula.

In addition, the factorial function will compute the gamma function if given a non-integer argument. The HP 10bII+ also has two probability distributions built-in: the normal and student's t-distributions. For the normal distribution, enter a standardized z-value and press $\left[\frac{Z \rightarrow P}{3} \right]$. To find the inverse normal distribution, enter the cumulative probability and press $\left[\frac{INV}{M+} \frac{Z \rightarrow P}{3} \right]$ to compute the z-value. To find a probability using the student's t distribution, enter the degrees of freedom, press $\left[\frac{df \rightarrow P}{2} \right]$, enter a t-value and press $\left[\frac{=}{=}$ $\right]$. To compute an inverse student's t distribution calculation, enter the degrees of freedom, press $\left[\frac{INV}{M+} \frac{df \rightarrow P}{2} \right]$, enter the probability, and press $\left[\frac{=}{=}$ $\right]$. Both of these calculations can be done using $\left[\frac{INPUT}{=}$ $\right]$ to separate the arguments as well by keying the first argument, pressing $\left[\frac{INPUT}{=}$ $\right]$, then keying the second argument followed by $\left[\frac{=}{=}$ $\right]$. The student's t calculation can also be set up as a constant calculation using $\left[\frac{K}{=}$ $\right]$, the constant key. This is very useful for students, as tables won't need to be referenced in the back of a textbook any longer. You can almost hear the cheers of students going up across classrooms even now.

Six regression models and best fit

In addition to linear regression, the HP 10bII+ can fit logarithm, exponential, power, exponent and inverse models to data by opening the regression environment by pressing $\left[\frac{REGR}{=}$ $\right]$.

Number and Mode	Description
0-Best Fit	Automatically selects fit
1-Linear	$m \cdot x + b$
2-Logarithm	$m \cdot \ln(x) + b$
3-Exponential	$b \cdot e^{(m \cdot x)}$
4-Power	$b \cdot x^m$
5-Exponent	$b \cdot m^x$
6-Inverse	$m / x + b$

The presently selected model is displayed and $\left[\frac{+}{=}$ $\right]$ and $\left[\frac{-}{=}$ $\right]$ move up and down through the choices. When the display indicates the model desired, pressing $\left[\frac{INPUT}{=}$ $\right]$ will select that model. The actual fit occurs when a regression function is executed. Pressing $\left[\frac{4}{X \uparrow} \right]$ will estimate an X value given a Y value and return the correlation in the SWAP position. Pressing $\left[\frac{5}{Y \downarrow} \right]$ will estimate a Y value given an X value and return the slope (the m coefficient for other models) in the SWAP position. Pressing $\left[\frac{6}{S_{w,b}} \right]$ will compute the weighted average (X weighted by Y) but also returns the slope (the b coefficient for other models) in the SWAP position. When best fit is selected, the HP 10bII+ quickly selects the model with the best correlation and then displays "Best fit" for a moment followed a quick display of the model chosen.

New data editors

The HP 10bII+ also includes the ability to review and edit statistics and cash flow data. Pressing $\left[\frac{RCL}{=}$ $\left[\frac{CF}{=}$ $\right]$ or $\left[\frac{RCL}{=}$ $\left[\frac{\Sigma}{=}$ $\right]$ opens up a data review editor where pressing $\left[\frac{+}{=}$ $\right]$ or $\left[\frac{-}{=}$ $\right]$ moves you up or down through entered X, Y data values or cash flow values and cash flow frequencies. As you move through them, the X and Y annunciators come on and off to indicate whether you are viewing an X or Y value in the statistics editor and the CF and N annunciators do the same as you view cash flow data.



Fig. 11 – Examples of annunciators during the new data editing mode.

Values can be changed by keying a new value and pressing $\boxed{\text{INPUT}}$. Pressing the $\boxed{\text{X}}$ inserts a blank entry into the data area and the $\boxed{\text{+/-}}$ key removes the presently displayed pair of values. Pressing $\boxed{\text{←}}$ erases the presently displayed value and replaces it with a zero, which can be kept or overwritten by keying another value and pressing $\boxed{\text{INPUT}}$. These abilities are a welcome addition to the HP 10bII's functionality.

There is now dedicated space for 15 cash flows and 15 statistics data points. In addition, there is a shared data space for any combination of up to 30 additional cash flows or statistics data points. So it is possible to have 15 cash flows and 45 statistics data points, 30 cash flows and 30 statistics data points, 45 cash flows and 15 statistics data points, or any other valid combination.

This is another big change from the existing HP 10bII, which only used register accumulations for statistics. The HP 10bII+ actually stores the data values entered unless you try entering more than the maximum of 45 allowed pairs. At that point, the FULL indicator is turned on to alert the user that they have entered the maximum number of pairs for which the data values can be retained. If another data value is entered, the existing individual data values are discarded and the HP 10bII+ reverts to register-based statistics only. This does change how some entries would behave between the two models. On the HP 10bII, it was possible to enter 1, 2, and 3 into the statistics data but then to remove 4 and 5 from the data even though they had never been entered. The HP 10bII+ will not allow this while in list-based statistics mode, since the 4 and 5 would not be present in the list to remove.















The HP 10bII+ also allows the direct computation of values such as the coefficient of variation, which is the standard deviation divided by the mean. Just compute the standard deviation, press $\boxed{\text{+/-}}$, compute the mean and press the $\boxed{\text{=}}$ key. On the earlier HP 10bII model, the mean must be computed first and stored into a memory. Why did it work that way? I have no idea, but it is much better now on the HP 10bII+.









What are some other goodies included?





The HP 10bII+ gets a previous answer function whereby the value previously computed when $\boxed{\text{=}}$ was pressed can be recalled at any point by pressing $\boxed{\text{RCL}} \boxed{\text{=}}$. This is very useful.

The HP 10bII+ also includes 10 extra memories, 10-19, which are accessed using the old method of pressing the decimal point before the memory digit. Memory 1 is accessed by $\boxed{\text{RCL}} \boxed{1}$, while memory 11 is accessed by $\boxed{\text{RCL}} \boxed{.} \boxed{1}$ for example. Given the "M" register, the HP 10bII+ now has 21 addressable data memories. Of course, the memories for the five breakeven variables can be used as data memories if needed as well. Simply press $\boxed{\text{RCL}} \boxed{\text{STO}}$ $\boxed{\text{→}}$ $\boxed{\text{UNITS}}$ $\boxed{\text{X}}$ to store a value into the UNITS memory location and $\boxed{\text{RCL}} \boxed{\text{←}}$ $\boxed{\text{UNITS}}$ $\boxed{\text{X}}$ to recall it. Each of the breakeven locations is usable in that way. That's an additional 5

memory locations. Similar approaches can be used with the CST, PRC, MAR and MU locations as well as bond and even TVM locations, if not used for other purposes.

A new clearing function has been added as the  of the  key. It is labeled as  and is one of three keys in “reverse video” where the other two are the  and  keys. The “reverse video” means that these keys require an additional key press to execute a function.  and  expect a subsequent , , or one of the trig functions to complete the desired operation, as stated earlier. The  function is connected by a thin line to the , , , and  keys, which all have a small vertical abbreviation next to them.

- Pressing  and , for example, clears the cash flow data only, leaving the rest of the calculator’s memory intact.
- Pressing   clears the TVM data,
- Pressing   clears the breakeven data, and
- Pressing   clears the bond data.

Since the existing HP 10bII and the earlier HP 10b allowed only the clearing of a digit with the  key, clearing a calculation with the  key, clearing the stats with  or clearing the entire calculator with , these new clearing functions are very useful given the expanded data space on the calculator.

Compared to the competition

The basic competitor for the HP 10bII+ calculator is probably the Texas Instruments BAII+ calculator. The TI engineers are probably not happy with this new HP entry, since each of the functions shown below are present on the HP 10bII+ and are missing from the TI BAII+ calculator. Perhaps TI should rename their model the TI BAII– instead?

Functions on the HP 10bII+ not found on the TI BAII Plus

Entry of a number in scientific notation	Student’s t distribution	Gamma function
Entry of a number with 12 digits	Inverse Student’s t distribution	Markup calculations
11 extra storage memories	Best fit regression choice	Initial CF ₀ frequency other than one
PI	Exponent regression model	No CPT (compute) key needed to solve
Normal distribution	Inverse (1/x) regression model	
Inverse Normal distribution	Weighted average	

That is quite a list!

Conclusion

HP has upgraded the HP 10bII with a host of useful functions. If a primary market for the HP 10bII were university students, they can now purchase an HP 10bII+ which should have all the functions for their business and science/mathematics classes. At the suggested MSRP, this is an incredible price / performance point. Now, HP, we’re all still hoping (and waiting) for a new low-end scientific model ...

About the Author



Gene Wright is the author of textbook "Quantitative Analysis for Business", a business math textbook using the HP-10BII and HP-12C, available from Amazon.com. He is also a video lecturer for a CFA exam review course. A former teacher at Lipscomb University in Nashville, Tennessee, he now works for a consumer electronics company. Gene has written many articles on HP calculators and serves on the annual HHC committee.

From The Editor

HP Solve #23 page 10

[← Previous](#) **Article** – [Next →](#)

From The Editor – Issue 23

Winter is in full force with many areas getting lots of snow and ice. *HP Solve* readers have spent more time indoors.

It seems that the RPN Tips Column has run its course. Unless a reader provides an input the 22 previously published RPN tips will have to be it.

I am also saddened to report that I will discontinue the Math Problem Challenge series. The reason for this is the Internet. One of the most important skills anyone needs these days is the ability to find what you want to know on the Internet. All you have to do is enter the “problem” into Google or other search engine, and you get the answer. That is no challenge.

Here is the content of this issue.

S01 – HP 10bII+ Financial Calculator – Gene Wright This new version of the HP 10bII is essentially a brand new machine. While the foundation is the 10bII the feature and function set expansion is truly astounding. Not since the HP-27s has an entry level machine had such a powerful mix of financial, scientific, and statistical functions. Now the business/financial student has a low cost machine that is suitable for all of his courses. Gene has done a superb job in covering this latest HP accomplishment.

S02 – Regular Columns

This is a collection of repeating/regular columns.

- ◆ **From the editor.** This column provides feedback and commentary from the editor.
- ◆ **One Minute Marvels.** This OMM reverses explores the so called Ulam’s Conjecture. If you have any curiosity about numbers and how they are related you will find the two routines listed are fun to play with. There are some interesting relationships that you may discover examining “Ulam’s Conjecture.”
- ◆ **Math Problem Challenge # 3 - Solution.** Here is an equivalent resistance problem that is really very easy to solve, IF, you use the right approach. You must think “out of the box” on this one. All aspects of the Cube Edge Resistor problem are explored in the solution to this last *HP Solve* Math Problem Challenge.

S03 – Customer Corner – Monte Dalrymple Monte is a California microprocessor designer who is also an HP calculator enthusiast. His story is an interesting one and his biggest HP calculator project is also described in this issue.

S04 – Tweaking the HP42S by Jake Schwartz Jake has spent a great deal of time analyzing HP calculators and his unique perspective on keyboard clutter and keyboard layout is applied to one of HP’s more interesting machines, the HP-42S. The HP-42S was a follow-on to the famous HP-41C series and it was designed to be compatible with it. The HP-42S Pioneer case, used for 11 different models, is a thin convenient design that is always popular with HP calculator users. Its soft key menus are analyzed by Jake with extensive images. He even examines the user interfaces of other machines such as the HP 35s, HP 10bII, and the HP 17bII+ in order to suggest a more effective/efficient way of laying out the machine.

S05 – Better Problem Solving Part II The basic idea of a calculator is that it is low cost, simple to use, and fast to solve problems. Calculators are efficient problem solvers that are unique when compared to versions found on computers and cell phones. Better Problem Solving Part II explores the classic problem of Triangle solving, what is involved, and asks the question: Why hasn’t there ever been a *good efficient* Triangle Solver on a calculator? A suggested user interface is described for this most important calculator problem.

S06 – What Is FOCAL? Is this HP-41 Programming Language an Urban Legend? What is FOCAL and how did the HP User Community learn of this “HP announcement”? *HP Solve* readers are asked to provide any historical information that will remove FOCAL from the urban legend category.

S07 – Upgrading the HP-41 by Monte J. Dalrymple The HP-41 was an astounding machine. It was part of a system that included bar code reading of programs, interfacing to all aspects of the instrument and computer

world, and using disc drives, printers, and a long list of other peripherals. Well over a hundred plug-in modules provided an extensive array of applications programs that extend from complex math to medicine.

Monte has re-designed the Nut processor to utilize more modern technology for greater speed (50x) and greater memory (all 41 modules) with a project he calls NEWT – Nut, Extended, With Turbo. After many years of part time work he is not able to offer an HP-41 replacement board for your treasured HP-41 to enhance its

performance beyond your dreams. The most popular programming machine ever in terms of being programmed is described in this article. Check Monte's website at <http://systemyde.com/hp41/>

S08 – #6 in the Fundamentals of Applied Math Series – e Does e stand for earnings, exceptional, or exacting?

This #6 installment of the Applied Math series is a #5 (Logs) follow on. e is often elusive and this review should help eliminate some of its easily enabled episodes.

That is it for this issue. I hope you enjoy it. If not, tell me!

Also tell me what you liked, and what you would like to read about.

X < > Y, Richard Email me at: hpsolve@hp.com

HP 48 One Minute Marvel – No. 10 – Ulam's Conjecture *Joseph Horn*

One Minute Marvels, OMMs, are short, efficient, unusual, and fun HP 48 programs that may be entered into your machine in a minute or less. These programs were developed on the HP 48, but they will usually run on the HP 49 and HP 50 as well. Note the HP48 byte count is for the program only.

Professor Ulam says this is not his idea, but many math students have seen this conjecture. It states that given any integer if you repeatedly apply one of two operations on the number, and the result, you will eventually reach one. The operations are:

1. If odd, multiply by three and add one.
2. If even, divide by two.

Take the first interesting integer, 3. The resultant values are: $3 \Rightarrow 10, 5, 16, 8, 4, 2, 1$. The "Ulam" process was applied seven times or completed seven operations or cycles. Note that when a power of two is reached the sequence directly divides by two, repeating operation 2, to reach one. The two programs below provide the basic Ulam values.

'ULAM' applies the rules and returns the next value in the series. 'ULM' repeatedly applies 'ULAM' until one is reached keeping count of how many times 'ULAM' is applied.

'ULAM' << IF DUP 2 MOD THEN 3 * 1 + ELSE 2 / END >>

13 commands, 52.5 Bytes, # 2F47h. Timing: 27 \Rightarrow 82 in 12.7_ms.

'ULM' << 0 OVER DO ULAM SWAP 1 + SWAP UNTIL DUP 1 SAME END DROP
SWAP \rightarrow TAG >>

16 commands, 55.0 Bytes, # FBE2h. Timing: 27 \Rightarrow 111 in 2.78_sec.

Questions

- (1) 12 & 13 require 9 cycles. 14 & 15 require 17 cycles. What are the first three consecutive numbers that require the same number of cycles?
- (2) Three requires 7 cycles. The ratio of cycles to the number is 2.333. What is the next number that exceeds this ratio?

- (3) What is the lowest number in which the number of cycles exceeds 100?
- (4) Five requires 5 cycles. What is the next number that is the same as the number of cycles?

HP Solve Math Problem Challenge #3 - Solution

This problem/solution is the third in a series of real world practical or teaching problems offered as a challenge to **HP Solve** readers.

This problem is an equivalent resistance problem. The math involved is not very complex, but if you have any basic electrical experience you will understand Ohms law and equivalent resistance.

Here is the problem.

Twelve equal resistors are connected together as if they were the edges of a cube. The equivalent resistance to be calculated is across the internal corners of the cube, i.e. from a to g in Fig. 1. The value of each resistor is 2,982 ohms.

The best solution idea is one that gives a clear step-by-step explanation of the solution. The best solution is based on the guidelines listed in Issue #22.

When I proposed this problem I did so remembering when I first encountered it long before the Internet. If you Goggle "Resistor cube problem" you will get 222,000 hits. Obviously the problem isn't a very difficult one to even think about under these conditions. Here is my solution prepared before I checked the Internet. See my additional challenge at the end.

Solving for the equivalent resistance across the internal diagonal corners of the equal valued resistors connected on the edges of a cube, see Fig. 2, is a messy series parallel equivalent resistance problem.

This problem doesn't have much practical value except to stimulate your basic understanding of resistors, current flow, and voltage drops. Most importantly this problem requires that you understand Kirchhoff's laws⁽¹⁾.

The two important characteristics of this circuit you should observe is that all the resistors have the same value and that the circuit has complete symmetry.

The value of studying and solving this problem is to illustrate and understand an analytical approach. The exact value of the resistors is not important if you imagine the following.

First let's make each resistor one ohm. Next we will connect an unknown voltage across points A and B such that one ampere of current flows. (Let's assume that the resistors have an adequate power rating) Assume that the current flows into point A. It "sees" three identical paths. See Fig. 3.

Path 1 is R1, R4 & R5, and then R10 & R11.

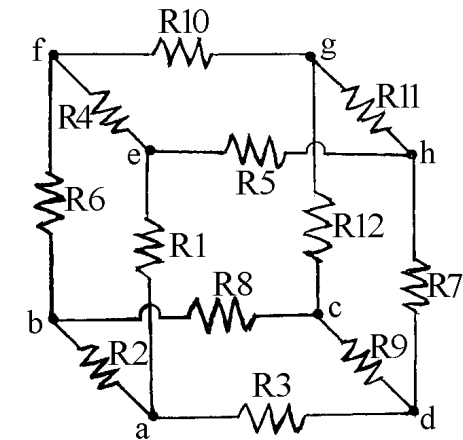


Fig. 1 – Resistors along the edges of a cube. Find the equivalent resistance from a to g across the internal diagonal of the cube.

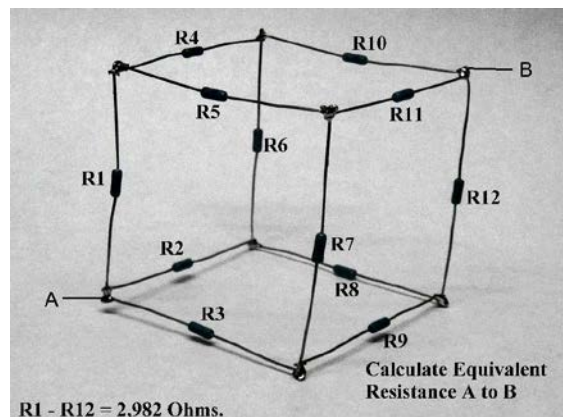


Fig. 2 – Resistors connected on edges of cube.

Path 2 is R2, R6 & R8, and then R10 & R12.

Path 3 is R3, R7 & R9, and then R11 & R12.

We don't know what voltage is being applied to A & B, but if we could calculate it the resistance could then be calculated ($I = 1$) and the problem is half solved.

Using A as a reference (ground) V_1 , V_2 , and V_3 (as shown in Fig. 4) is obvious by inspection. V_1 is the same as the voltage across R3. If $I = 1/3$ ampere and R3 = one ohm V_1 is $1/3^{\text{rd}}$ volt.

V_2 is the sum of the voltage across R3 and R7. R7 has $1/6^{\text{th}}$ volts across it. $V_2 = 1/3 + 1/6$ volts.

V_3 is the sum of the voltage across R3, R7 and R11.
 $V_3 = 1/3 + 1/6 + 1/3 = 2/6 + 1/3 + 2/6 = 5/6$ volts.

The equivalent resistance = $5/6$ volts divided by one ampere = 0.8333333333 ohms.

The problem specified that each resistor is 2,982 ohms.
 Scaling: $2,982 \times 0.8333333333 = 2,484$ ohms.

Because of the symmetry there are only three different values you may measure across the eight resistor connection points of the cube.

1. The internal diagonal is the problem as shown above. There are two possibilities. Both are the same value, 2,484 ohms.

2. Across each resistor. All 12 values are the same no matter what resistor you measure.

3. The diagonal of each cube face, 12 total. These are all the same.

Further investigation: Can you now calculate the two values, 2 & 3 listed above? See measured answers below. How could you arrive at these answers?

ANSWER: 2. 1,730 ohms. 3. 2,230 ohms. (as measured).

Cube Resistance Problem Solution - Note

(1) There are two basic laws.

i. Around any closed circuit, the algebraic sum of the voltage sources and IR drops must total zero.

ii. At any junction of two or more conductors in a circuit, the algebraic sum of the currents into a junction must equal the current out of the junction.

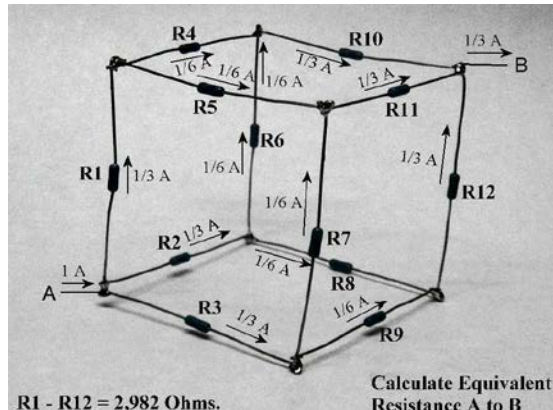


Fig. 3 – Current division from A to B.

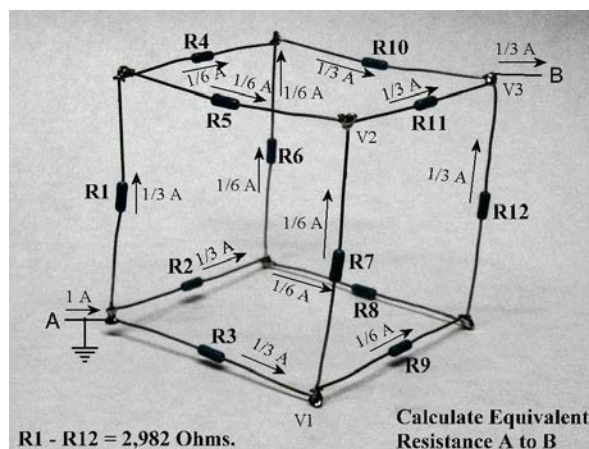


Fig. 4 – Summing the voltages.

Customer Corner: Meet Monte Dalrymple

HP Solve #23 page 15

[← Previous](#) **Article** – [Next →](#)

Meet Monte Dalrymple

Editor's note. Customer Corner has appeared in past issues of *HP Solve* where we interviewed the worldwide users of HP's calculators. Past interviews have been of users who live and work in the US, UK, Canada and Germany. We now go to Livermore, California for our next interview.



1. ***HP Solve: What is your background?***

Monte: I was born in a small town in Idaho, but have lived in California since I was six. I was always building things, and discovered electronics in Junior High.

2. ***HP Solve: What did you study at school?***

Monte: I majored in Electrical Engineering at UC Berkeley. Most of my coursework was in analog design, and I didn't really do anything digital until my senior year. My graduate work was about half analog and half digital. When I started working I had to choose one way or the other, and digital won.

3. ***HP Solve: What is your occupation?***

Monte: I have been an IC designer all of my career. My first design was a serial communications controller, and it's still in production today (although in a much different technology). I migrated to CPUs, and have done more of those than I care to count. The NEWT project was a refreshing change, because the bit-serial operation was fundamentally different from anything I had done before with a CPU. And it was quite fun figuring out how to expand the capabilities of the machine in a backwards-compatible fashion.

4. ***HP Solve: Do you do much traveling?***

Monte: Not really. In fact, for the past fifteen years I have telecommuted, so I don't even have to deal with a daily commute. Most of my interaction with customers is via email or teleconference these days, although I still have to occasionally visit the customer site. When projects involve people in half a dozen time zones teleconference is the only thing that makes sense.

5. ***HP Solve: When did you first see or use an HP calculator?***

Monte: I was in the library reading Scientific American when I saw the advertisement for the HP-35. I was so impressed that I bought a copy of the magazine. I don't think I still have it though. When Macy's started carrying the HP-35, I used to go to the store just to play with the display calculator. This was my senior year in high school and I barely knew what trigonometric functions were, but I found it incredible that a machine that fit in my hand could do that.

6. ***HP Solve: So when did you buy your first HP?***

Monte: I started college with a slide rule, because the \$395 price of an HP-35 was more than tuition for one quarter at Berkeley. But by the time of my first finals it was obvious that to be a Berkeley engineer you had to have an HP calculator. At the time my uncle worked for HP, and employees were able to buy one calculator per year at cost. The first year he bought one for himself, but the second year I didn't ask him in time to take advantage of the deal. By then the HP-45 was out, so that was what I had to have. Given the size of the investment I carried it in the hard belt case. Final exams were three hours, so you also had to carry a fully-charged spare battery pack for a final. And heaven forbid you had back-to-back finals.

7. ***HP Solve: What machine did you buy after your first?***

Monte: My HP-45 was all I ever needed in college and when I first started my working career. But when I found myself doing repetitive calculations figuring transistor sizes for my first chip, I got my HP-41CV. Keystroke programming made the programs as easy as entering the number of fanouts, a guess for their size, and a guess at the wire length.

8. **HP Solve:** *What HP calculators have you used since?*

Monte: When it was first announced I had to have a 95LX, and I still use it as a personal organizer. I have one of most of the RPL models, but never really use them. The 4-level stack of the HP-45 is hard-wired into my brain, and I find myself cursing whenever I try to use the unlimited stack of an RPL machine. I use my HP-16C quite frequently for hexadecimal calculations. My desk always has my original 41CV and my 16C at hand. I have an HP-33S and HP-35S, but don't really use them regularly. I really like the styling of the HP-35S, but the handling of hexadecimal numbers is just too cumbersome.

9. **HP Solve:** *What have you used your calculators for?*

Monte: Anything and everything. I have copies of most of the HP modules for the 41CV, and enjoy just exploring what they can do.

11. **HP Solve:** *Have you noticed anything interesting about calculator usage during your travels?*

Monte: It's interesting that you can tell how old an engineer is by the calculator that he or she carries. I still carry my 41CV, which really dates me in meetings, but I still see a lot of 32SIIs. Many young engineers seem to view a calculator as almost disposable, which is unfortunate. Most programmers that I interact with are doing very low-level programming and they usually have a 16C that is worth its weight in gold to them.

12. **HP Solve:** *Does your calculator usage involve more than your profession?*

Monte: Yes. I already mentioned the 95LX as a personal organizer, and I also play chess on it. TVM calculations are a snap on that machine too. I use my 41CV whenever I am helping my kids with homework. They had to have TI machines for school, but for the life of me I can't use a machine with an equals sign.

13. **HP Solve:** *What was the largest problem you have solved with an HP calculator?*

Monte: Well, it wasn't one big calculation, but I had to calculate the appropriate transistor size for each of the seventeen thousand transistors on my first chip. Fortunately the computer does that work automatically these days.

14. **HP Solve:** *What are you currently using HP calculators for?*

Monte: I use the 16C for hexadecimal calculations almost daily. Cost/benefit calculations related to chip size and yield, balancing the checkbook, and so on for the 41CV.

15. **HP Solve:** *What appeals to you about HP calculators?*

Monte: One word: quality. My HP-45 still works and I pull it out occasionally. My 41CV is older than any of my children and I still use it daily.

16. **HP Solve:** *Do you have a website?*

Monte: Yes, <http://www.systemyde.com> is where I advertise, keep my resume, and so on. I have the rights to most of the designs that I have done recently, so I use the website to give potential customers the information they need about these designs.

17. **HP Solve:** *What are the important aspects of your website?*

Monte: If you go to <http://www.systemyde.com/hp41> you'll see my work on the 41CL project in all its glory.

18. **HP Solve:** *Do you write or post calculator articles on your website?*

Monte: Not really, but I did keep a running status update for the 41CL project, and the manuals for the NEWT processor and the 41CL are available there too. I did do an article about the 41CL design for Circuit Cellar magazine. It was in the October 2010 issue.

19. **HP Solve:** *Can you explain in more detail?*

Monte: The 41CL is a project that replaces the processor in my 41CV with an upgraded version that I designed myself. It runs up to fifty times faster, and contains images of over a hundred ROMs on board.

It's fully compatible with all of the 41 peripherals. With this upgrade I expect that my 41CV (now CL) will outlive me.

20. *HP Solve: Have you had contact with HHC?*

Monte: I first presented the idea of a replacement processor for the 41C series at HHC2004. I am pretty sure that everyone there thought "yeah, right". Since my day job got in the way, it has taken a long time to come to fruition, but I actually do have a 41CL now, and have sent some out for testing by people I met at HHC.

21. *HP Solve: How many HHCs have you attended?*

Monte: Only two, mainly because I don't travel much. Since 2004 I have tried to contribute a paper though.

22. *HP Solve: What have you gained from the HHCs?*

Monte: Mostly an appreciation that there are still people who are passionate about the technology in HP calculators. Lots of people from HHC have answered dumb questions or provided feedback that helped me get the 41CL project actually working.

23. *HP Solve: Do you have a calculator collection? How extensive is your collection?*

Monte: I wouldn't really call it a collection. I am more a user than a collector. I would love to have an HP-35 and an HP-65. The HP-65 was announced just a few weeks after I got my HP-45, and I was crestfallen at the time. But I couldn't have afforded it anyway. I would like to get an HP-15C at some point.

24. *HP Solve: Do you have any "special" HP calculators?*

Monte: Only to me personally. My HP-45 will always hold a special place for me, as it got me all the way through grad school at Berkeley. Clearly my 41CV holds a special place, given the amount of time I've put into upgrading it to the 41CL.

25. *HP Solve: What does your family think of your interest in calculators?*

Monte: I think they call it "calculator derangement syndrome" behind my back. But three of my kids are engineers, and the fourth has started on that path, so they sort of understand the fascination with technology.

26. *HP Solve: What kind of HP calculator would you like in the future?*

Monte: Well, the 41CL is basically all that I ever wanted. The HP-35S was so close to being ideal for a practicing engineer. But the hexadecimal handling killed it for me. Even though I know its quality, I cringe to think of the day when my 16C might die. I would buy a small pile of a redo of that machine.

27. *HP Solve: What other HP calculator related projects have you worked on?*

Monte: I think the 41CL project was big enough. Although now that it's done I wouldn't mind reverse-engineering the processor used the 71B and original RPL machines, just for fun. But I doubt I'll have time for that anytime soon. Actually, since the Voyager series used the same CPU as the 41C, it would be possible to clone the 15C or 16C using the NEWT design, but the electronics is only a small part of such a project, so I'll probably leave that to someone else. Having the CPU on a daughter-board, so that I could reuse the case, display and keyboard is the only reason that the 41CL project was possible.

28. *HP Solve: Do you have any additional comments that may be of interest to HP Solve readers?*

Monte: Go to an HHC, and meet people who know HP calculators inside and out. Many of the early machines still have secrets to reveal, and it's great seeing the ingenuity and engineering that went into their design.

29. *HP Solve: Thank you for spending your time with us.*

Tweaking the HP42S

HP Solve #23 page 19

[← Previous](#) **Article** – [Next →](#)

Tweaking the HP42S

Jake Schwartz

Introduction

In 1972, the HP35 calculator appeared on the scene and its keyboard was a straightforward affair with basically one function assigned to each key (with the exception of an “arc” key which served as a shift for the SIN, COS and TAN keys in order to provide ASIN, ACOS and ATAN functions). In those early years to follow, more and more functions were made available via increasing the number of keyboard shift keys and shifted-function “key planes”.

As a result, an increasingly cluttered keyboard was unavoidable and the HP-67 and HP-34C handhelds from 1976 and 1979 represented the peak with 3 shifts each, permitting up to four functions per key. (Let us define keyboard *clutter* as the average number of functions per keyboard key. Based on this definition, the clutter value for the 35-key HP-67 would be 3.26 and for the 30-key HP-34C 3.40. It could also be argued that the HP48/49/50 series machines are equally complex with their left, right and alpha shifts, although the 50g clutter has a slightly lower value of 3.00.) By having all functions visible on the keyboard however, no command was more than two keystrokes away, and the overall moral to the story of this article is that fewer keystrokes per function are always better.

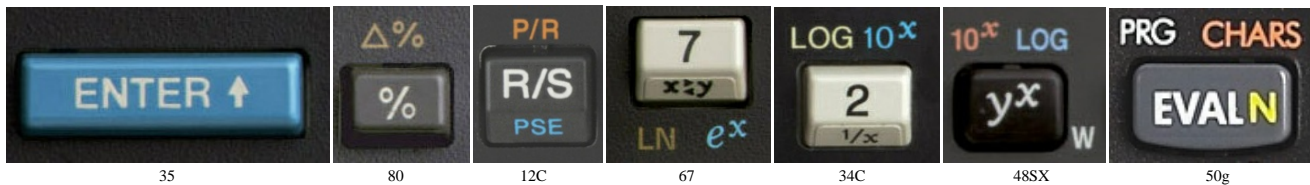


Fig. 1- HP calculators and the progression of keyboard clutter. Left-to-right: HP35 unshifted key, HP80 key with a single shifted function, HP12C key with two shifts, and the HP67, HP34C, HP48SX and HP50g, each with three shifts.

The HP41 Changes the Game

In 1979, Hewlett-Packard released the HP41C, which (among many other things) set a new standard in RPN-calculator functionality and programming power. As it appeared to the designers that adding still more shift keys and shifted functions beyond three would not be a good idea, they changed the user-interface philosophy with the HP41, whereby the majority of its function set was hidden from the user’s view. This machine, with a single gold-colored shift and shifted-function key plane, only barely began to reveal its built-in capabilities. In order to access functions which were not shown on the keyboard, one would have to spell them out alphabetically in the single-line segmented LCD from the new ALPHA key plane, with its letters placed on each key front. Counting both the gold- and ALPHA-shifted markings, the clutter value for the HP41 was 2.64 functions per key. Additionally as a spelling and reference aid, the HP41 provided a CATALOG command which listed the name of every function one-by-one in alphabetical order. Spelling out function names unfortunately increased many times over the number of keystrokes required for the average command. For example, the keyboard “BEEP” shifted function only required two keystrokes, but the non-keyboard “ABS” function needed $\boxed{\text{XEQ}} \boxed{\text{ALPHA}} \text{ABS} \boxed{\text{ALPHA}}$, or six keystrokes.

Clamshells, Soft Keys and Menu Trees

In June of 1986 after the final iteration of the HP41 (the 41CX) was a couple of years old, HP went in another direction by bringing out the first of a new line of “clamshell-case” machines with multi-line dot-

matrix LCDs – the HP18C Business Consultant. With only a single shift key and corresponding sparsely-filled right-hand shifted key plane, it continued to leave the majority of built-in functions hidden, but provided access via six-at-a-time soft-key menus using the top row of keys which were re-definable and labeled in the bottom row of the LCD. (The clutter value for the HP 18C is only 1.43 functions per key and it would be even lower, were it not for the left-hand keyboard’s numerous shifted alphanumeric characters.) Multiple soft-key menus were reached starting with the “MAIN” menu and traversing the menu “tree” with keys leading to submenus named “FIN”, “BUS”, “SUM”, “TIME”, “SOLVE” and “MATH”. Furthermore, many of those submenus contained menus at further lower levels as well. Since a maximum of six functions could be displayed at once, menus requiring additional functions had their sixth (and rightmost) key position labeled “MORE” or “OTHER” which when pressed revealed another

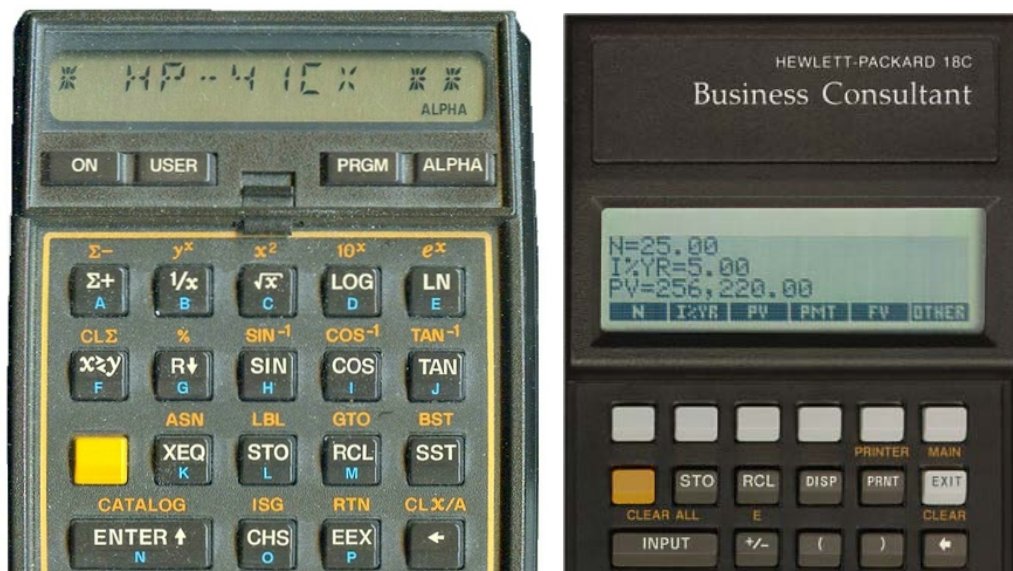


Fig. 2 - The single-line segmented HP41 LCD (left) permitted spelling out function names while the multi-line dot-matrix HP18C screen (right) on the right side of its clamshell case provided a method for labeling a menu of up to six “soft” keys in the top row.

row of functions. Also, despite the fact that the physical shape of the keys allowed for an additional function to be labeled on each key front, no machines in the clamshell series (18C/19B/19BII financials or 28C/28S scientific) ever took advantage of them.

The Pioneer Series

A year and a half later in January of 1988, a new line of vertical-format machines – dubbed the “Pioneer” series - debuted with both scientific (HP-27S) and business (HP-17B) models containing 2-line dot-matrix LCDs and continuing with a row of soft keys at the top being defined by the bottom row of the display.

However, a new rounded key shape would only accommodate a primary function to be marked on each key, effectively removing the possibility of any key-front shifted key plane with a corresponding second shift key. When later Pioneer HP-20S, 21S and 32SII models with 1-line screens required the addition of a second shift key and key plane, the function labels had to be crammed side-by-side above each key, contributing significant clutter to the look of the keyboard. However, these units achieved reduced keystroke counts as a result. (Figure 3 compares the 27S and 32SII.)

Enter the HP42S

Toward the end of 1988, HP introduced the non-expandable HP42S Pioneer model as a successor to the HP41 series. (Both appear in Figure 4.) Both the functionality and programming features were significantly enhanced while still maintaining backward compatibility with HP41 programs. However, one handicap which the 42S shared with the 41 series was the concealment of the vast majority of its functions “under the hood”. This time around, the user had the choice of either spelling them out in ALPHA mode like before or seeking the appropriate soft-key menus containing the desired functions. In both cases, however, the number of keystrokes required remained quite high as compared to a machine which was able to have all its functions visible out on the keyboard. Also, the ALPHA mode scheme on the 42S was very different, with most of the characters embedded in a tree of soft-key menus rather than being out on the keyboard. With its single shift key and no ALPHA characters marked on the keyboard, the 42S’ clutter value was only 1.95 functions per key versus 2.64 for the HP41. Despite access to many



Figure 3. Two Pioneer-series models: the HP27S (left) from 1988, with its clutter value of 1.73 functions per key and HP32SII (right) from 1991, with its second shift key and higher clutter value of 2.73. With their rounded-surface keys, the provision for a function on each key front had been removed.

more available ALPHA characters than in the HP41, getting to the letters A through Z required two keystrokes each, versus one on the HP41. Figure 5 shows examples of the HP42S ALPHA navigation scheme.

The HP41 had 37 functions requiring no parameters (such as “HMS+” or “ABS”) which were not on the keyboard and therefore required spelling via ALPHA, taking a total of 260 keystrokes. The HP42’s 25 no-parameter functions (such as “COSH” or “STR?”), found neither on the keyboard nor in soft-key menus required a total of 253 keystrokes via ALPHA. Comparing these, the HP41 averaged 7.03 keystrokes per function whereas the HP42 required a significantly-higher average of 10.12 keystrokes per

function. For ALPHA-spelled functions requiring parameters, the HP41 included “DSE”, “FC?”, “FS?C”, “FC?C”, “ΣREG”, “X<>” and “TONE” and their direct and indirect stack, register-number and flag-number parameters. This totals 1,223 function combinations⁽¹⁾ requiring 13,646 keystrokes or an average of 9.52 keystrokes per function. On the HP42S, the only non-menu ALPHA-spelled functions with parameters were those corresponding to the “X<>” function – a total of 210 function combinations requiring 2,730 keystrokes or an astronomical average of 13.00 keystrokes per function. In both situations, the HP42S scheme appears to have been inferior to that of the HP41 from the standpoint of minimizing keystroke counts. (It should be mentioned that an additional 13 HP42S no-parameter functions not appearing in soft-key menus could be accessed via the comprehensive 42-page CATALOG FCN menu using fewer keystrokes than their ALPHA-spelling counterparts. Nevertheless, these 13 still required 84 keystrokes for an average of 6.46 keystrokes per function.)

Despite the shortcomings of the HP42S ALPHA-function system, the remaining aspects of the machine allowed its overall 5,954 function combinations to require 32,104 keystrokes, or an average of 5.39 keystrokes per function. On the other hand, the HP41’s 4,873 functions and 27,042 keystrokes represented a slightly-higher average of 5.55 keystrokes per function. Nevertheless, had the 42S employed a second shift key and additional keyboard function plane, the function keystroke counts would have been further reduced and machine ease of use further improved.

Back To the Future

Flashing forward to 2007-2008, we saw the return of HP’s usage of keys with slanted fronts, permitting shifted key functions to be printed on the keys themselves. Some examples of these are the HP35s (which supported menus) and the HP10bII (with no menus), each with two shift keys, along with the single-shift, menu-plentiful HP17bII+ as shown in figure 6. Not only did this reduce the clutter around the keys, but in the case of the HP35s, keystroke counts were kept in check by filling all available shifted key positions with functions not otherwise occupied by menu-access commands (such as “LOGIC”, “CLEAR” and “BASE”). Another way that the HP35s managed clutter was to place its alphabetic-key markings also on the key fronts just to the right of its blue-shifted function names. Compared to the HP32SII as shown in figure 3, with its alphabetic markings in the space to the lower right of each key, the 35s appears less intimidating to this user.

Now, returning to the present with the migration to a new, much faster CPU as the basis for the HP20b, HP30b and the latest incarnation of the HP12C, it can only be imagined what a speed boost like this would do for the scientific machines. Based on this possibility, let us consider a revived HP42S with the speed of a modern CPU plus some additional “tweaks”:

(1) *By function combinations, we mean all the ways a function may be combined with its parameters. For example, the STO function may directly address the 100 numeric registers 00 through 99 or the 5 stack registers L, X, Y, Z or T; plus it may indirectly address values in those same 100 numeric or 5 stack registers. This totals 210 possible parameters. In addition, the ALPHA variables may also be directly or indirectly addressed, however the number of possible combinations of these are too numerous to count.*

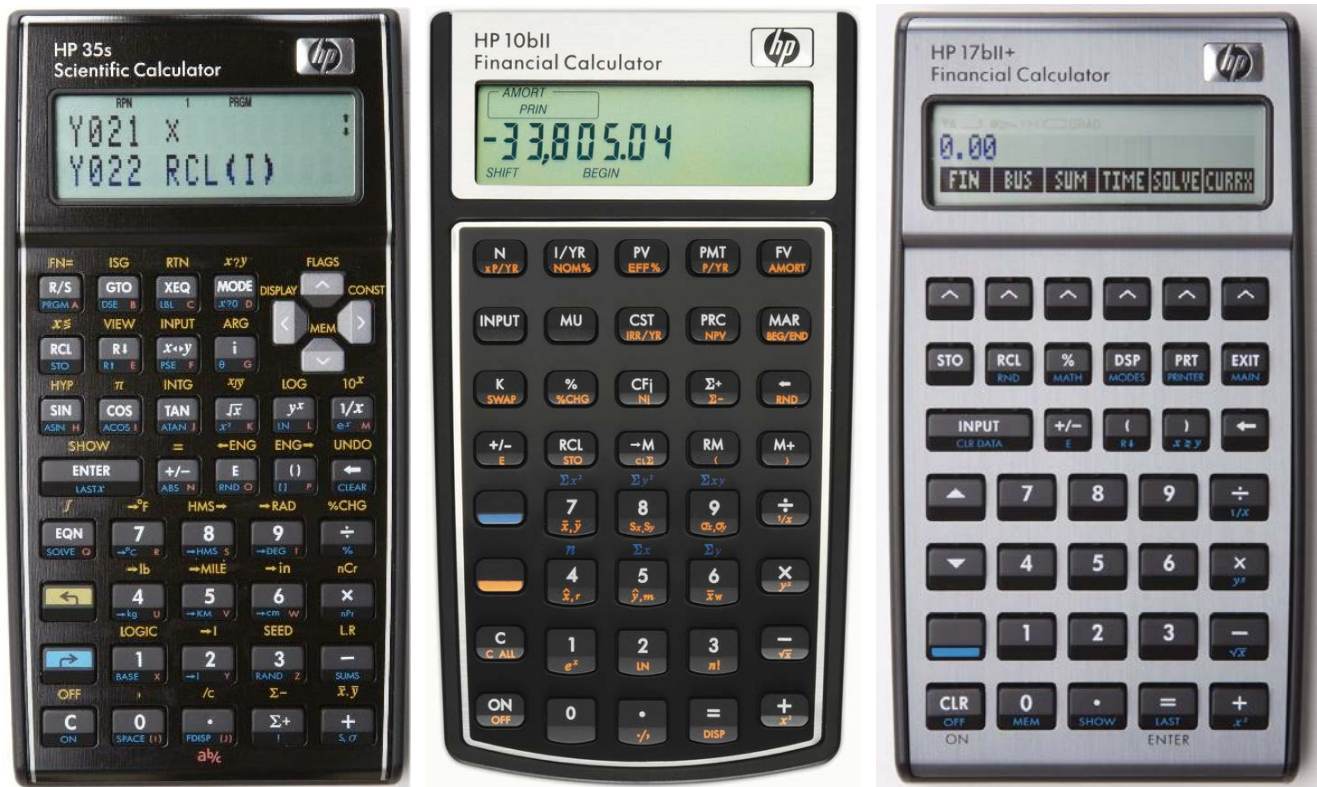


Fig. 6 - HP35s scientific calculator (left), HP 10bII financial machine (center), and the HP 17bII+ financial calc (right), all with keys which permit shifted functions to be printed on the front of the keys.

1. Since the key arrangement and screen of the current HP17BII+ mimics that of the original 42S, migrate the functionality into that physical case;
2. Reduce keystroke counts for some commonly-used functions by utilizing a second shift key and shifted key plane, allowing for elimination of some menus;
3. Adopt the HP41/HP35S ALPHA scheme with letters on each key front; and
4. Increase the RAM capacity to 32K since the HP42S operating system could address that much despite HP's original decision to restrict it to 8K.

An image of the original and revised "42S+" is shown in figure 7. First, let us discuss the HP17BII+ physical case and keys. The keyboard arrangement and two-line dot-matrix LCD matches that of the older Pioneer series, and thus would be a perfect enclosure for a 42S successor. Also with its slanted-front keys, it would be a natural for two shifts and two shifted-key planes. In order to retain the up- and down-arrow keys (originally in the lower left) along with the two shift keys, some relocation was required in order to maintain the column orientation of these two key pairs. Since the locations of the shifts have traditionally been in that lower left-hand position over the past many years, it was decided to move the arrow keys up to the middle of the second and third rows. Along with this, since an additional key was needed for the second shift, the roll-down and x-exchange-y stack manipulation functions were combined onto the same key just below the ENTER. (The other shifted position on this key also became a convenient place to include the roll-up function as well.)



Fig. 7 - The original HP42S (left) and a proposed “HP42S+” (right) ported to the current HP17BII+ case.

Adding Common Functions to the Keyboard

For the remainder of the new second shifted key plane, the goal was to bring out commonly-used functions and programming aids, many of which appeared on the HP41’s faceplate. As a result, virtually all of the functions in the FLAGS, BASE, PROB and CONVERT menus were able to be relocated onto the blue shifted positions, along with the “LBL”, “RTN” and “ISG” programming functions. In addition, keys to activate the “x-and-zero” and “x-and-y” comparison-test menus were promoted to the keyboard from originally having only been embedded as submenus inside the “PGM.FCN” menu. A total of 39 items were added, representing 1,690 function combinations averaging 4.532 keystrokes per function in the new layout versus 5.736 for the embedded ones in the original 42S. Of those 39, thirty are functions needing no parameters and they average 2.300 keystrokes per function for the new layout versus 4.433 for the original machine. (The only reason this no-parameters average is not exactly 2.000 is due to the hyperbolic trig functions which require the blue-shift “HYP” ahead of the trig keys themselves.) For these functions, the advantage is not only the reduction in keystrokes, but also that they are directly visible on the keyboard and thus may be located much more quickly.

Changing the ALPHA Scheme

By bringing the alphabetic capital letters (along with a few punctuation symbols) out onto the keyboard,

ALPHA entry returns to the HP41 method, which averages one keystroke per character (not counting keystrokes for entering and exiting ALPHA mode itself). On the HP41, the numeric and arithmetic symbols plus the “ASTO”, “ARCL” and “CLA” functions were reached via shifted key presses; and the same would go for our revised “42s+” calculator with the gold shift. The blue-shifted key plane would produce the lower-case letters, %, pi and “AVIEW” on the appropriate keys. Also, where in the original 42S entering ALPHA mode turned on the A-Z soft-key menu, the new method would directly activate the original second ALPHA-menu page – the top-level symbols menu with brackets, braces, math, punctuation and other miscellaneous characters. However, since the top key row has A to F and a to f in its primary and blue-shifted positions, the gold-shifted keys would select the menu soft-keys. Pressing gold-shift followed by a top-row key therefore activates the selected lower-level individual-symbols menu and associates it with the primary top-row key presses, switching capital A-F to the gold-shifted plane and leaving the lower-case a-f to the blue-shifted plane. Then when a key for a symbol was pressed, it would revert the top row back to having A-F as primary and the top-level symbols menu returned as gold-shifted (with lower-case a – f remaining blue shifted).

Moving Pi and Percent

On the original HP42S, the “%” and “pi” functions were gold-shifted on the second row of keys, above “RCL” and “R↓”, respectively. This permitted direct 2-key access to the “%” and “pi” characters in ALPHA mode as well, without requiring any menus. With the new 42S+, two changes required moving these functions elsewhere. First, since the RCL key now contains “H” as the primary ALPHA function, “ARCL” as the gold-shifted function and “h” as the blue-shifted function, no other position remains available in ALPHA mode. Secondly, the “R↓” function was moved off the next key to be replaced by the up-arrow primary and “BST” gold-shifted function. It therefore became necessary to move “%” and “pi” to keys which had an ALPHA function position open. The spots chosen were above the decimal point and the “+”, since their primary ALPHA characters were not alphabetic, leaving the blue-shifted positions available.

Eliminating the BASE Menu

On the original HP42S, the six-key BASE menu would activate the four base-mode keys, plus the “A - F” submenu (for the hex digits) along with a “LOGIC” submenu key for bitwise logic functions in any base. Also, while the BASE menu was activated, the arithmetic and change-sign keys would perform the 36-bit “BASE+”, “BASE-”, “BASE*”, “BASE/” and “BASE +/-” functions. On our updated machine, the four base-mode functions plus the “LOGIC” menu were brought out on the blue-shifted functions on the keyboard fourth and fifth row. Pressing any of the four base-mode keys would activate that particular base mode, plus it would redefine the arithmetic and change-sign keys as before. In addition, choosing “HEXM” would also activate the “A – F” menu for hex input. Finally, choosing the LOGIC menu would activate the most-recent base menu mode as well as turning on the LOGIC soft keys.

Tallying the Changes

Overall, the 5,955 function combinations had their keystroke totals reduced in the revised 42s+ from 32,104 to 29,515, causing a reduction in the average keystrokes per function from 5.392 to 4.956, or almost half a keystroke. However, digging a bit deeper reveals more significant differences. A total of five soft-key menus were able to be eliminated. With the 39 additional commands added to the keyboard, the number of function combinations accessible directly from the keyboard increased from 2,562 to 4,260 with a similar drop in the number of menu-based functions. Those new keyboard-accessible functions averaged over 1.2 fewer keystrokes per function than when they were embedded in menus. For the thirty

commands which take no parameters, the average keystrokes per function dropped by 2.1. In addition, due to the new ALPHA entry method, the commands whose most-efficient access were through spelling out via ALPHA had their average keystrokes per function reduced from 12.694 to 10.618, or by around two keystrokes each. With these changes, a very worthy successor to the HP42S (and higher-powered companion to the HP35S) could be realized.

Epilog: For More Information

A collection of comprehensive keyboard metrics collected on roughly forty HP non-RPL calculators from the early 1970s through to the early 1990s may be found at <http://www.pahhc.org/keyboards.htm>.

Additionally, there are studies as to how clutter and keystroke counts would be affected by experimentally adjusting the number of keyboard keys, the number of shift keys, the number of soft-key menus and the number of single-row menu pages. And finally, there are detailed metrics for our fictitious “42S+” along with discussions of a few earlier “wish-list” machines based on previously-published articles in 2008 in Britain’s Handheld and Portable Computer Club (HPCC) group’s *Datafile* newsletter.

Copies of those articles are available at <http://www.pahhc.org/>. For additional information on the HPCC group, consult <http://www.hpcc.org>. By no means are these particular “fantasy” machines the only ones ever proposed in detail by the user community; for instance, one should seek out the great work done by Walter Bonin, who has been proposing logical and exciting enhancements to HP machines for several years now and (with Paul Dale) is currently engaging in a serious repurposing project for the HP30b financial calculator which would transform it into a powerful RPN-programmable scientific machine. Additional information along with the latest version of the 50-plus-page user manual may be obtained at <http://sourceforge.net/projects/wp34s/>.

About the Author



Jake Schwartz has been an HP calculator fan since 1971 after first trying the HP9810A desktop RPN machine at a co-op job at RCA in New Jersey. He has owned most of the scientific top-of-the-line handhelds since the HP35A in the early 1973, joined the PPC Calculator Club in 1977 and contributed to many of the clubs since, including serving as Peripheral Routines coordinator for the PPC ROM project for the HP41 in 1980. Currently working at Lockheed-Martin as a software engineer, Jake has been presenting at and videotaping the annual U.S. HP calculator conferences for more than two decades.

Better Problem Solving Part II

HP Solve #23 page 29

[← Previous](#) **Article** – [Next →](#)

Better Problem Solving Part II

Richard J. Nelson

Introduction

In the last issue of *HP Solve* I suggested that I am convinced that it is still possible to improve the common functions that we use with our calculators. The premise is that calculators still survive in the market place amongst so many other devices such as smart phones and mini-laptops because of their convenience and low cost. The calculator must continually improve or become obsolete. Ever increasing problem solving efficiency is vital for an advanced calculator.

Four “improved” functions were discussed in Part I.

Problems with Two Possibilities – $\Delta\%$


Problems With Two Possibilities – Function & Inverse

Problems with Two Possibilities – conversions

Problems With Two Possibilities – Function & Inverse alternate

Part I concluded with: “Better Problem Solving Part II will suggest a method of implementing a function that requires a single key press and three inputs. The function then decides which of 19 solutions is the correct one and then calculates four outputs. If this function is added to all mid-range and high end scientific calculators it would save hundreds of hours of involved calculations for just about every technical user. The problem is a very common everyday type problem.”

Solving Triangles

The problem I am referring to is what I call triangle solving. Surveyors are probably the most well-known triangle solvers, but nearly every engineer, technician, contractor, or hobbyist has had the need to solve, usually for a side, a triangle,  problem. If you Google “Solving Triangles” you will get nearly 2 million hits. Obviously this is an important problem.

Most HP calculator manuals will have Triangle solutions included. Calculators with solvers will address the problem using the solver. All of these fall short of a good efficient built-in automatic Triangle Solving Environment - to use an RPL term.

Why hasn't there ever been a **good efficient** Triangle Solver on a calculator?

Understanding the Problem

The critical issue here is automatic. If you research triangles you will find long discussions on triangle properties (e.g. the sum of the angles = 180°), Law of sines (in various forms), Law of cosines (in various forms), special case of a right triangle, and the dreaded cases.

If you think about the possibilities you will find that there are 20 possible cases (six combinations taken three at a time) wherein you know three of the six unknowns and you want to calculate the remaining three. Of these 20 possible cases the case of knowing all three angles is meaningless because there are an infinite number of possible triangles. These cases are often grouped and described using three letters denoting S for side and A for angle i.e. AAA, AAS (side not between an angle), ASA, SAS, SSS, and SSA.

In order to solve a triangle problem you will need three knowns, and one of them must be a side.

Why must I draw a triangle, label it, and look up the various equations in order to get the answer I need?

Why shouldn't the calculator do this work for me?

Since Triangle Solving is a common problem HP calculator users have addressed this "automatic" solution issue with programs. The math, method, and solution has been programmed on the HP-97, HP-67, and, HP-41C, by John Kennedy. See *PPC Calculator Journal* April 1980, V7N3P14. Also see similar programs in the *PPC Calculator Journal* May 1980 V7N4P30. Several others contributed ideas especially those of Keith Jarrett and George Donaldson.

John Kennedy's analysis is especially important because he used selected equations to solve for the unknowns based on the knowns. He never used a calculated value to calculate another unknown. This maintained the highest possible accuracy. The automated aspect is that the calculator identifies the 19 possible cases and selects the appropriate solution method based on the input knowns.

There are two additional considerations for an automated triangle solver.

- 1) The Side-Side-Angle, SAS, case is special. See Fig. 1. This represents a solution that is ambiguous. There are two possibilities for this in the 19 cases. The answer is to calculate both solutions. John Kennedy investigated this and that is exactly what his programs do. It should be obvious that the selection logic of which equations are used on the inputs is very important.

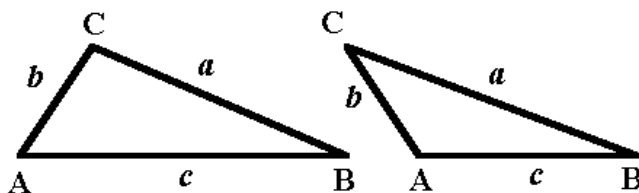


Fig. 1 – Two solutions for SSA case, a , b , & B known.

- 2) In addition to calculating the various unknown angles and sides of the triangle you may also want to know the area.

At least one HP Calculator manual suggested the use of the Solver to solve triangles. While this is a good approach it is not quite the ideal solution. In years past it was the speed of the solver that was the primary issue. Today's machines are fast enough. Perhaps the ideal solution using today's technology is to wrap a triangle solver user interface around the solver.

When people keep talking about cases as if the user needs to be even aware of them, they don't understand the problem. All that thinking must be automated. Just put in your unknowns and get all the answers.

The only inputs (case information) the user need be concerned with is:

- (1) One of the inputs must be a side.
- (2) The relative relationship of the sides and angles A , a ; B , b ; C , c . This is insured by the labeled input diagram and the menu/key labels.

How a Calculator Triangles Solver Should Work

The purpose of a modern calculator is to provide low cost fast and efficient answers to numerical problems. If you need to solve a triangle problem at your computer connected to the internet you could use one of the many "solvers" that are on the web such as:

<http://www.handymath.com/cgi-bin/irregangle8.cgi?submit=Entry>

This program does not provide a high accuracy answer (5D), but it does handle the ambiguous cases and it provides two sets of answers. This website nicely illustrates how a Triangle Solver should work. It does not include the area.

If you want to visualize the relationships of a triangle see:

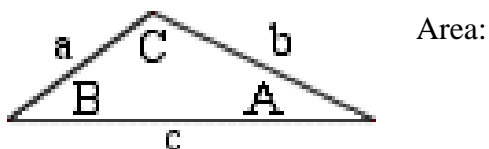
<http://www.geogebra.org/en/upload/files/english/fauxmanchu/visualizetriangleinequalities9.html>

If, however, you want to use a Calculator Triangles Solver it must be as efficient and easy to use as possible.

1. One key press should open the triangle environment.
2. Three numbers should then be entered. Upon the termination of the third number the machine should immediately (and automatically) calculate the remaining three unknowns and the area.

Exactly how this is done will depend on the calculator and its features. The angular mode would be the same as the calculator is set – degrees, radians or grads.

- A. A graphing calculator could display the triangle nomenclature such as:



A soft menu could be labeled to match the diagram and the appropriate key pressed after keying in the value. When the third value is entered the program automatically solves. When the answers are calculated they could put on the diagram. The Area would also be indicated. If a value is changed enter with a shift menu key and a new triangle is calculated in a similar manner as the solver works.

- B. An entry level scientific calculator could not provide as nice a user efficient interface as a graphing calculator because of its limited display. One solution could be to use data registers. R1-R3 = Sides A, B, & C. R4-R6 angles a, b, & c. R0 could be the area.

Key strokes are important and the user should not have to enter “place holder” values such as zero.

How often could you use a Triangle Solver?

If you have ideas on improving calculator function efficiency email me at:

Email *HP Solve* at: hpsolve@hp.com

About the Author



Richard J. Nelson has written hundreds of articles on the subject of HP's calculators. His first article was in the first issue of *HP 65 Notes* in June 1974. He became an RPN enthusiast with his first HP Calculator, the HP-35A he received in the mail from HP on July 31, 1972. He remembered the HP-35A in a recent article that included previously unpublished information on this calculator. <http://hhuc.us/2007/Remembering%20The%20HP35A.pdf> He has also has an article published on HP's website on HP Calculator Firsts. See <http://h20331.www2.hp.com/Hpsub/cache/392617-0-0-225-121.html>.

What Is FOCAL?

HP Solve #23 page 33

[← Previous](#) **Article** – [Next →](#)

What Is FOCAL?

Richard J. Nelson

One of the most popular calculators HP has ever made is the HP-41 introduced in 1979 – see Issue # 16. The HP-41 was programmable and many thousands of programs have been written for this machine. The programming “language” was simple, powerful, easy to learn, and a very high percentage of users wrote programs for their machines. What was the programming language of the HP-41 called?

FOCAL is the name of a computer language and the name of the HP-41 programming language.

FOCAL is/was a trademark of Digital Equipment Corporation and is referenced in:

http://www.bitsavers.org/pdf/dec/pdp11/rt11/v2c_Jan76/DEC-11-ORDDA-A-D_V2C_DocDir.pdf

The above link is (a PDF file) to an RT-11 Document that is described with the following Abstract. “This document lists and describes the software manuals for the RT-11 Operating System, Version 2C (V2C), including the Monitor, system programs, FORTRAN, and BASIC.”

A list of DEC Trademarks is provided in the front of the document and is shown below.

Copyright © 1976 by Digital Equipment Corporation

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-10
DECCOMM		TYPESET-11

Fig. 1 – List of Digital Equipment Trademarks related to their computer systems – as listed by DEC.

A search of the USPTO found the following. There are many other US and foreign FOCAL trademarks listed including several from FOCAL Corporation.

TARR Status	ASSIGN Status	TDR	TTAB Status	(Use the "Back" button of the Internet Browser to return to TESS)
Typed Drawing				
Word Mark	FOCAL			
Goods and Services	(EXPIRED) IC 016. US 038. G & S: BOOKLETS AND MANUALS CONTAINING A COMPUTER LANGUAGE AND INSTRUCTIONS FOR PROGRAMMING. FIRST USE: 19680216. FIRST USE IN COMMERCE: 19680216			
Mark Drawing Code	(1) TYPED DRAWING			
Serial Number	72293484			
Filing Date	March 18, 1968			
Current Filing Basis	1A			
Original Filing Basis	1A			
Registration Number	0879976			
Registration Date	November 4, 1969			
Owner	(REGISTRANT) DIGITAL EQUIPMENT CORPORATION CORPORATION MASSACHUSETTS 146 MAIN ST. MAYNARD MASSACHUSETTS 01754			
Type of Mark	TRADEMARK			
Register	PRINCIPAL			
Affidavit Text	SECT 15.			
Live/Dead Indicator	DEAD			

Fig. 2 – Digital Equipment Corporation registered FOCAL trademark details. The trademark has long expired.

The acronym meaning of the DEC FOCAL computer language is Formula CALculator language.

FOCAL as the HP name for the HP-41 calculator programming language is the result of a naming contest conducted by *HP Key Notes* editor Henry Horn published in HP's calculator newsletter in the July-August 1982, V6 N3 issue on page 16. He mentioned the contest in the context of a letter from a reader asking about the HP-41 programming language. See the last part of the text in Fig. 3. The details of the issue are shown in Fig. 4.

HP Key Notes continued to be published for another year until the August 1983, Vol 7 No. 2 issue. This was the last issue and still the contest winning name was not announced. The text is shown in Fig. 5.

It is unclear as to when the contest was decided, what the official name was, and who won. It is commonly believed that the name is FOCAL and that the meaning is Forty-One CALculator Language.

An Internet search hasn't resolved the issue, but the search has not been completely exhaustive.

If the reader is able to provide a hard source to answer this question I would appreciate receiving it.

A search of the various user community newsletters should be made. The search should start after August 1983.

The following references describe the HP-41 version of "FOCAL."

<http://en.wikipedia.org/wiki/Focal>

[http://en.wikipedia.org/wiki/Focal_\(HP-41\)](http://en.wikipedia.org/wiki/Focal_(HP-41))

[http://www.absoluteastronomy.com/topics/Focal_\(HP-41\)](http://www.absoluteastronomy.com/topics/Focal_(HP-41))

Here is an HP Museum reference to FOCAL.

Re: FOCAL

Message #12 Posted by [Frank Wales](#) on 15 Sept 2004, 5:24 p.m.,
in response to message #11 by Mike (Stgt)

FOCAL actually got associated with the HP-41 via a contest that Henry Horn ran through 'HP Key Notes' to come up with a name for the 41 user code programming language, around about 1982, I think. I remember, because I entered it (and, in fact, I'd also submitted 'FOCAL' as a name, which I suggested could be short for 'Forty-One Contra-Algebraic Language', rather than the more straightforward 'Forty-One Calculator Language' that eventually won).

There was a protracted period when they said that they'd picked a winner, but were going through some legal checking before they could announce it, and I guess it was because the name had been used before for a language, and they were doing due diligence for trade marks or assessing the likelihood of being sued, or something like that.

So, honored KEY NOTES readers, Dr. Metcalf has *his* reward; now which one of you is going to win our contest to "Name That Language"? The rules are very simple, and everyone except Hewlett-Packard employees and their families may enter.

1. The "Language" name must be a clever acronym such as: radar, BASIC, scuba, FORTRAN, sonar, COBOL.
2. The acronym must relate to the HP-41.
3. The acronym must be original (new).
4. Only one entry per person is permitted.

Fig. 3 – FOCAL Contest Details.

HP KEY NOTES

June-August 1982 Vol. 6 No. 3

Programming and operating tips, answers to questions, and information about new programs and developments concerning Hewlett-Packard handheld computers. Published quarterly. See page 15. Reader comments or contributions are welcomed. Please send them to one of the following addresses.

Fig. 4 - Issue Details of Fig. 3.

HP-41 Language Contest

In V6N3, we initiated a contest to name the "language" used by the HP-41. And, by the way, the contest was long ago closed, so please do not submit more entries. Since the end of the contest, we've tried to legally clear several names you seemed to favor, but they already were registered trademarks. Also, it is very difficult to do this on a worldwide basis.

However, be patient. We are confident we have a winner in process, and we will notify that person soon. We haven't forgotten you.

Fig. 5 – Last Issue contest report.

If you know of, or have a reference, to answer the question of when, the name, and who won the contest I would love hearing from you. The big question to me is how did HP announce this to the world? The news letter was defunct. The Fig. 5 newsletter text implies that the winner would be notified. If that was the case how did the HP User Community find out about it?

Some HP enthusiasts I have asked even suggest that this could be an “urban legend.”

Do you have any information on this?

X < > Y,

Richard

Email me at: hpsolve@hp.com

Upgrading the HP-41

HP Solve #23 page 37

[← Previous](#) **Article** – [Next →](#)

Upgrading the HP-41

Monte J. Dalrymple

Early in 2004 I discovered that HP had, at some point in the past, released many of the internal specifications for the chips and software used in the HP-41 calculator⁽¹⁾. I design microprocessors for a living, and it was fascinating to see how some relatively simple hardware, combined with some very clever software, could implement the marvel that is the HP-41. Just for fun I decided to try to create a clone of the NUT CPU, which was the processor used in the HP-41.

At the time, my main goal was just to demonstrate that designing such a CPU using modern design tools wasn't really that difficult. In fact it only took me about six weeks, working in my spare time, to finish the design. But once the design was done, I realized that the internal construction of the HP-41 might allow me to actually replace the "brains" of the calculator. This is because, in all but the last iteration of the HP-41, the CPU and memory is contained on a small daughterboard that connects to the main calculator circuit board via a press-fit connector.

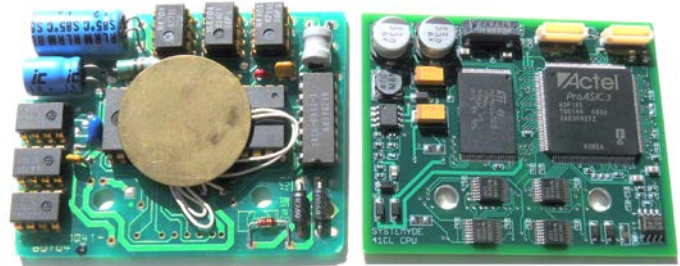


Fig. 1 – Old and new CPU.

This is how the NEWT⁽²⁾ (which stands for Nut, Extended, With Turbo) came to be. I didn't want to just replace the CPU in the HP-41, I wanted to try to extend the design to take advantage of the advances in technology since the time of the HP-41. I targeted two areas for improvement.

The first area of improvement was the speed of the machine. The CPU in the original HP-41 ran at just 360KHz, which might seem ludicrous given the multi-GHz clocks of today's computers, but it was more than sufficient for most uses of the calculator. However, since there are some cases where it would be nice to have the machine run faster, the NEWT is designed to run at up to fifty times normal speed, using the "Turbo" mode.

But I couldn't just bump up the clock speed of the CPU without considering the rest of the HP-41 system. For example, anything plugged into a port has to see only the original 360 KHz clock and bus signals. So while the NEWT is running faster than normal it continues to drive the Port signals at regular speed. It also automatically throttles back to normal speed for operations that have timing requirements, such as communicating with the display or driving the beeper.

The second area of improvement was the memory space available. There are literally dozens of plug-in modules that I might like to have available at a time, but trying to carry them around with the calculator just isn't feasible. So I designed a way for the NEWT to access a much larger memory, while remaining 100% backwards-compatible with the original HP-41 software, by using a Memory-Management Unit (MMU).

As I soon found out, designing the circuitry to do this was just a fraction of the effort that would be required to put together a replacement CPU board, and it wasn't until 2010 that I actually had enough time to devote to complete the project.

The replacement CPU board, which I call the 41CL⁽³⁾ (I assumed that the numbering, from 41C, to 41CV to 41CX was partly Roman numerals) is finally ready for prime time. This board, which is a drop-in replacement for the CPU board in an HP-41C has the following features:

- All of the features of an HP-41CX except for the timer. If you need the time functionality you'll need a Time Module plugged into one of the Ports.
- Full 600-register Extended Memory is built-in.
- Over 100 plug-in module images are built in.
- Turbo mode with 2X, 5X, 10X, 20X and 50X speed options.
- 56 empty pages (4K in length) in non-volatile Flash memory available for user programs.
- 58 pages (4K in length) of RAM available. All RAM is continuously powered.
- Full bus-compatibility on the Ports, allowing the use of any peripheral designed for the HP-41 system.
- New built-in functions to manage the expanded memory.
- A full-duplex serial port is available with an optional connector.

Clearly most of these new features are not supported by any existing software, so in addition to the hardware design I also had to write a set of functions that will allow the typical user access to these new features. These functions, which I call Y-Functions and which Catalog as YFNS -1A, aim to provide an intuitive way for the user to manage the module images present in the Flash memory on the new CPU board as well as control the new Turbo operating speed modes. In addition, for the advanced user, there are functions to PEEK and POKE memory, move entire 4k blocks of memory, access the on-chip peripheral registers, and do PUT and GET operations with the on-board serial port.

Probably the most significant new feature provided by the 41CL is the wide range of module images available in the on-board Flash memory. These images can be "plugged into a Port" just as a physical module would be plugged into a Port. First, the four-character module identifier is entered into the Alpha register and the appropriate PLUG command is executed from the keyboard. Twelve PLUG commands are available: PLUG1, PLUG1L and PLUG1U control Port 1 on the calculator, with corresponding commands for the other three ports.

The PLUG commands parse the module identifier and verify that the selection is valid for the Port selected. Just like the original physical modules, some module images can only be used in particular ports, and if an invalid configuration is attempted a DATA ERROR message will result. Once the selection is validated, the software then automatically programs the MMU so that accesses for addresses in that Port will be redirected to the module image in the Flash memory. At this point a Catalog will verify that the module is "plugged into the Port" and available for use. No power-down of the calculator is required. While a module image is plugged into a port the physical port is not accessed by the CPU, so physical modules may be present in the Port without any electrical conflict.

In addition to the module images in Flash memory, blocks of RAM memory may also be plugged into a Port. This allows the on-board RAM memory to be used as HEPAX memory or MLDL memory, for example. An experienced user may want to copy a module image to RAM, modify it, and then plug this modified image into a Port. I had to do this while debugging the design, as I had some errors in several of the new functions that I had to patch before they could be used.

Unplugging a module image from a Port is just as simple. There are twelve UNPLUG commands available: UPLUG1, UPLUG1L and UPLUG1U control Port 1, with corresponding commands for the

other three ports. Unlike the PLUG commands, no checking for configuration is performed. The software merely invalidates the appropriate MMU entries so that the CPU resumes access of the physical Port. The user is responsible for not unplugging half of a module image! Table 1 shows the module images available and the corresponding module identifiers.

Module Image	Mnemonic	Module Image	Mnemonic	Module Image	Mnemonic
Advantage Pac 1B	A41P	ES41 Module	E41S	PPC Melbourne	PMLB
Advantage Apps	AADV	ESMLDL 7B	ESML	PPC Module	PPCM
ADVENTURE #1	ADV1	Extended I/O 1A	EXIO	Pride ROM	PRIQ
ADVENTURE #2	ADV2	SKWID EXT-IL	EXTI	Rampage ROM	RAMP
AECROM	AECR	Financial Pac	FINA	Real Estate Pac 1A	REAL
AFDC1 ROM	AFDE	FUNSTUFF ROM	FUNS	ROAM-0A	ROAM
AFDC2 ROM	AFDF	Games Pac 1A	GAME	ROMSV01	ROMS
Autofinance	AFIN	GMAC 2	GMAS	Sandmath 12K	SANA
Astrology ROM	ALGY	GMAC 3	GMAT	Sandbox	SBOX
ALPHA ROM	ALPH	Hydracomp ROM	HCMP	Securities Pac 1A	SECY
AMC-OSX ROM	AOSX	HEPAX RAM	HEPR	SGS GAS	SGSG
Assembler 4	ASM4	HEPAX 1D	HEPX	SIM Module	SIMM
Assembler 3	ASMB	Home Management	HOME	Simplex ROM	SIMP
ASTRO-2010	ASTT	ICODE ROM	ICOD	SKWID ROM	SKWD
ASTRO-2010 UI	ASTU	JMB-MATH	JMAT	Speed Machine II	SMCH
HP Autostart	AUTO	JMB-MATRIX	JMTX	Simplex Module	SMPL
Beechcraft ROM	AV1Q	LABELS ROM	LBLS	Sandmath 8K	SMTS
Aviation Pac 1A	AVIA	LANDNAV	LAND	HP Standard	STAN
Boeing-B52	B52B	Math Pac 1D	MATH	Statistics Pac 1B	STAT
BMW ROM	BCMW	Machine Design Pac	MCHN	Stress Analysis	STRE
Buffer ROM	BUFR	MDP1 ROM	MDP1	Structural Analysis	STRU
CCD Plus Module	CCDP	MDP2 ROM	MDP2	SUP-R-ROM	SUPR
CCD Module 1B	CCDR	Melbourne ROM	MELB	Survey Pac 1B	SURV
CCD OS/X	CCDX	Mil Engineering	MILE	Thermal Pac 1A	THER
Chemistry User	CHEM	MLROM	MLRM	Toms ROM	TOMS
CHES ROM	CHES	MCTEST ROM	MTST	Toolbox ROM	TOOL
Circuit Analysis Pac	CIRC	Muecke ROM	MUEC	TREKKIES ROM	TREK
Clinical Lab Pac	CLIN	Navigation Pac 1B	NAVI	83Trinh ROM	TRIH
CurveFit ROM	CURV	NFCROM	NFCR	UNITCONV ROM	UNIT
CVPAK ROM	CVPK	Navpac ROM	NPAC	USPS Module	USPS
Disasm 4C	DA4C	NAVCOM 2	NVCM	alternate YFNS	YFNS
Data Acquisition	DACQ	Oilwell ROM	OILW	default YFNS	YFNZ
Disasm 4C	DASM	P3B/C Aviation Pac	P3BC	HP41Z ROM	Z41Z
David Assembler	DAVA	Paname ROM	PANA	Zenrom ROM	ZENR
HP-IL Development	DEVI	ProtoPARIO ROM	PARI	Zeprom ROM	ZEPM
HP- IL Diagnostic	DIIL	ProtoCODER2	PCDE	4K block of mem	-RAM
Diamond ROM	DMND	PCODER 1A	PCOD	16K block of mem	-16K
Dyerka ROM	DYRK	Petroleum Pac 1A	PETR		
ES41 Module	E41S	Plotter Pac 1A	PLOT		

Table 1 - Module images and mnemonics

Control of the various Turbo modes is similar. The TURBO2, TURBO5, TURBO10, TURBO20 and TURBO50 commands select the corresponding Turbo clock speed. The TURBOX command disables the Turbo mode and returns to 1x operation. The TURBO? command returns the current value of the Turbo mode in the X register. All of the module images, as well as the majority of the operating system

software run at the selected speed. However, there are several timing loops in the operating system that will always automatically run at 1x.

The commands above will probably be sufficient for the vast majority of users, and that's why I made them as simple as possible to use. But advanced (or adventurous) users can take advantage of the remainder of the Y-Functions and the features of the NEWT microprocessor itself.

Most of these advanced commands require one or more memory addresses, and in designing the command operation I chose to use the Alpha register for the user to input this information (because it is hexadecimal). The command is then executed, and if data is returned the value in the Alpha register is modified appropriately. Some commands also return a status in the X register (0 for success and 1 for failure). Most of the commands allow both a logical address, which is the 16-bit HP-41 address plus the bank select information, or a physical address, which is the native NEWT 24-bit address. See Figure 2 for the requirements for the various commands.

Command	Address format
YMCLR	PPPPPP-DDDD or LLLL-B-DDDD
YMCPY	PPP>PPP or PPP>L-B or L-B>PPP or L-B>L-B
YPEEK & YPOKE	PPPPPP-DDDD or LLLL-B-DDDD or R---DDDD
YFERASE	PPPPPP
YFWR	PPP-PPP

Table 2 - Command address format

Four basic commands for dealing with memory are available. The YMCLR command initializes a 4k block of memory to a specified value, while the YMCPY command copies from one 4k block of memory to another. As I mentioned previously, this command is handy for moving a module image to RAM for editing. This copy command accepts either logical (what the CPU uses) or physical memory addresses for both source or destination. These commands do not check for an attempt to write to Flash memory though, so be careful.

The YPEEK and YPOKE commands read or write a specific word to or from memory (remember that the NEWT memory is 16-bits wide) or an on-chip peripheral port. The YPOKE command must be used with care, however, because the entire address space of the NEWT microprocessor is available. This means that a user can write directly to a word in the X-, Y-, Z- or T register, or to the variables used by the operating system or directly to an MMU register.

The last two memory commands deal with the Flash memory, and I hesitated including them because they require even more care by the user. Because they operate on the Flash memory, the code must be relocated to RAM before the commands can be used. Both commands check to make sure that they are running from RAM before executing, and will return an error message without executing if they are located in the Flash memory. Both commands will also return an error message without executing if the user attempts to erase or program the area of Flash that holds the operating system. Allowing either operation on the operating system area would turn the 41CL into a brick.

The YFERASE command erases a 32kx16 block of Flash to the default state of 0xFFFF for every word. The YFWR copies a 4kx16 block of RAM to a 4kx16 block of Flash. An attempt to copy from Flash will return an error message without executing, because the only operation allowed during a Flash write is the writes themselves.

The final commands deal with the on-board RS-232 port. Currently I have commands to set the baud rate: BAUD96, BAUD48, BAUD24 and BAUD12 select 9600, 4800, 2400 and 1200 baud respectively. There are also PUT and GET commands to transfer individual bytes, as well as serial block transfer

commands. I plan to offer an optional serial connector that uses a blank Port cover to hold a 2.5mm stereo jack to bring the serial port signals out of the calculator.

Although it has taken forever, the 41CL is almost ready for prime time. In retrospect, I underestimated the effort required to both design the PC board and write the software to control the new features on the NEWT microprocessor. But I think that the result has been worth the effort (and wait). The project has given me a much greater appreciation for the engineering that went into the design of the original HP-41 series!

References

- (1) See <http://www.hp41.org> for examples of the original HP documentation.
- (2) The technically-inclined can find the specification for the NEWT microprocessor at <http://www.systemyde.com/pdf/newt.pdf>
- (3) See <http://www.systemyde.com/pdf/sy41cl.pdf> for the manual for the new 41CL calculator.

About the Author



Monte Dalrymple has been designing integrated circuits for over 30 years. He holds a BSEE and MSEE from the University of California at Berkeley and has fifteen patents. Monte uses an HP-16C, HP-41CV, and HP-41CL routinely. Not limited to things digital, he holds both amateur and commercial radio licenses. Email Monte at: monted@systemyde.com

#6 in Fundamentals of Applied Math Series

HP Solve #23 page 43

[← Previous](#) **Article**

e

Richard J. Nelson

Introduction – What is the excitement over e ?



Which number does not belong? $\sqrt{3}$, π , π^2 , 1, ϕ (golden ratio $\frac{1+\sqrt{5}}{2}$), e , $\sqrt[3]{3}$?

Hint: What kind of number is $\sqrt[3]{3}$?

The previous article in this series discussed Logs and it pointed out that the most commonly used log bases are 2, e , & 10. The base for natural logarithms is e which is given to 56 digits below.

Fig. 1 - HP35s e^x key. $e = 2.71828\ 18284\ 59045\ 23536\ 02874\ 71352\ 66249\ 77572\ 47093\ 69995\ 95749\ \dots$

The 15D value of the number e may be easily called to memory if you can remember:

1. It starts as 2.7. This is 0.3 less than three.
2. The next four digits repeat twice, 1828. What happened in 1828? (*The year Noah Webster copyrights the first edition of his dictionary, and science fiction writer Jules Verne is born.*)
3. Think of a symmetrical right triangle. It has angles of 45, 90, and 45.

This memory aid sequence, 2.7 1828 1828 45 90 45 will give you 15 decimal digits of e .

e is an irrational number whose decimal expansion never terminates and is never periodic.

(1)

What is the number that doesn't belong in the list above? It is the center "one." See *Issue #20, p30, (1)*

Defining e

e is usually defined by:
$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \tag{e1}$$

A curve of the limit converging to e is shown in Fig. 2 below. See example values in table one.

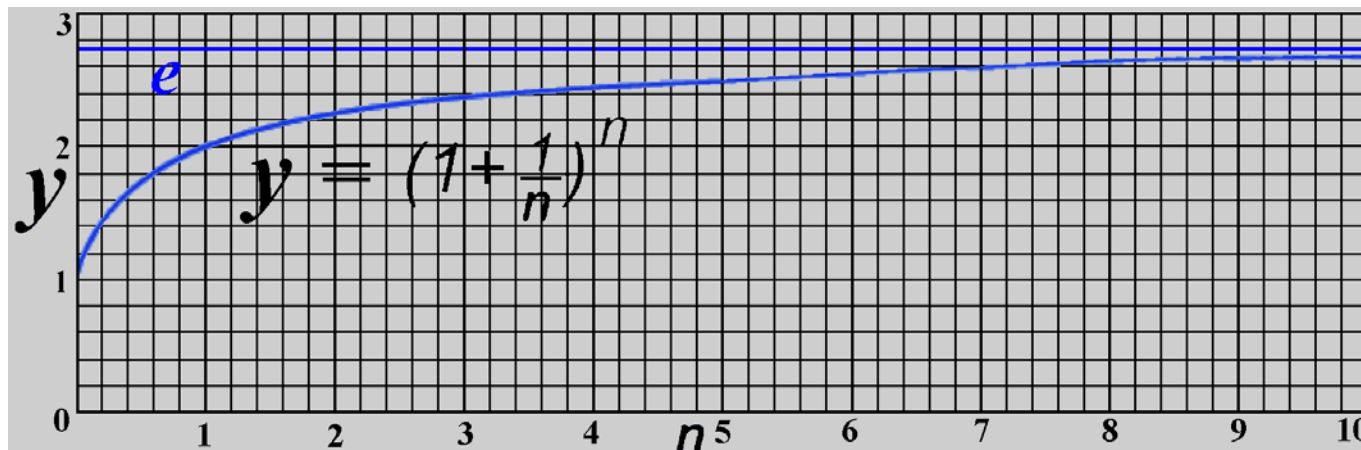


Fig. 2 – Plot of the limit of $y = (1 + 1/n)^n$. Values to eleven digits are given in table 1 below.

Table 1 – Example of e calculated (limit) values to 11 digits

n	$(1 + 1/n)^n$	Difference	d	n	$(1 + 1/n)^n$	Difference	d
1	2	0.71828182846	1	10^3	2.71 692393224	0.00135789622	3
2	2.25	0.46828182846	1	10^4	2.718 145 92683	0.00013590163	4
3	2.37037037035	0.34791145811	1	10^5	2.7182 6823717	0.00001359129	5
4	2.44140625	0.27687557846	1	10^6	2.71828 046932	0.00000135914	6
5	2.48832	0.22996182846	1	10^7	2.718281 69254	0.00000013592	7
7	2.54649969709	0.17178213137	1	10^8	2.7182818 1487	0.00000001359	8
10	2.5937424601	0.12453936836	1	10^9	2.71828182 710	0.00000000136	9
74	2.7 0013967815	0.18142150310	2	10^{10}	2.718281828 32	0.00000000014	10
10^2	2.7 0481382942	0.13467999040	2	10^{11}	2.7182818284 5	0.00000000001	11

Note: d is the number of accurate digits.

The values of Table 1 were calculated using the following RPL program.

```
'EE' << DUP INV 1 + OVER ^ SWAP →TAG 1 EXP OVER - >>
```

Key n , run the program, and level two will have the input value, colon, and the calculated value. Level one will have the difference of the calculated value subtracted from e . These values are the first three values in the table.

Here is how the program works:

The input, n , is copied to level 2 with DUP. The reciprocal of n is calculated with INV. One is added with 1+. A copy of n from level two is brought to level one with OVER and $^$ (Y^X) calculates the function. Remember that n is still on level two. It is swapped to have the two values in the correct order for the $→TAG$ command. Next the value of e is calculated using 1 EXP. The calculated value of the tagged function is still on level two and a copy is brought to level one with over. These are subtracted with the final $-$, and the program terminates.

The Story of a Number

Now that we know the value of e , and we can easily calculate it, we will explore the number that even has a book, published by Princeton University Press, 41 William Street, Princeton, New Jersey, 08540,

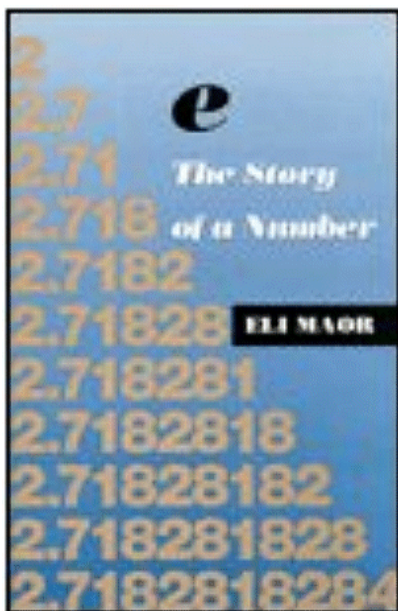


Fig. 3a – 1994 Hardcopy version.



Fig. 3b – 1998 paperback edition.

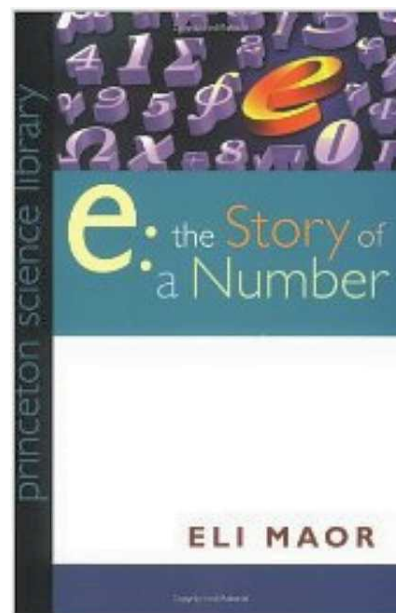


Fig. 3c – 2009 paperback edition.

dedicated to it. This book has been translated into Spanish, Greek, Portuguese (Brazil), Korean, Chinese (traditional), and Japanese. See Fig. 3 above. e is sometimes called Euler's constant and some writers have referred to it as Napier's constant. The name designation, e , is often given as e for Euler or e for Exponent.

$$e \text{ is also (more easily) calculated by: } e = 2 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots \quad (\text{e2})$$

Understanding e

A few numbers are especially important in mathematics. Zero, one, π , and i are examples. e has to be added to this short list. Combining these five numbers into one identity is one of Leonhard Euler's major accomplishments.

$$e^{i\pi} + 1 = 0 \quad \text{Where:} \quad \begin{array}{l} e \text{ is Euler's number, the base of natural logarithms.} \\ i \text{ is the imaginary unit, which satisfies } i^2 = -1 \\ \pi \text{ is pi, the ratio of the circumference of a circle to its diameter.} \end{array} \quad (\text{e3})$$

e as the base of the natural logarithms and is frequently used in financial compound interest and natural growth calculations. These calculations are part of the Time Value of Money, TVM, (solver) environment of HPs financial calculators. The idea of adding the interest to the principal on each compounding time period raises the question of the length of the period. The adage "time is money" is based on this relationship. Interest, and interest compounding, associates the value of money on the time and the interest rate.

Most interest rates are annual interest rates expressed in percentage and usually specified as an APR. Suppose the period is reduced from monthly to weekly. If the interest rate remains the same the principal will grow faster because of compounding interest. What happens when you reduce the interest time period to a day, minute, or a second? Will your principle increase ever faster? Is there a limit? Yes there is, and it is e . This limit is called continuous compounding.

The formula to calculate the future value of a starting principal with interest compounded n times per year is:

$$FV_n = P(1 + r/n)^{Yn} \quad (\text{e4})$$

Where:

FV is the future value after Y years

Y is the number of years

P is the starting principal

r is the annual interest rate.

n is the number of compounding periods

To get to the continuous case we take the limit as the time slices get tiny:

$$FV = \lim_{n \rightarrow \infty} P(1 + r/n)^{Yn}$$

We can simplify the right side by introducing a new variable, defining $m = n/r$

$$FV = \lim_{m \rightarrow \infty} P(1 + 1/m)^{Ymr}$$

$$= P \left[\lim_{m \rightarrow \infty} (1 + 1/m)^m \right]^{Yr}$$

The limit in the square brackets converges to e . The formula may be simplified and becomes

$$FV = Pe^{Yr} \tag{e5}$$

The difference between **e4** and **e5** is that **e5** is based on continuous compounding. The principal at any instant in time is changing at a rate that equals the interest rate times the principal.

Does the discussion above give you a practical sense for the value of e ?

Let's look at the value of e in a different way. Compound interest is based on an interest rate for a time period. The time periods are all equal. In mathematical terms the periods are integers. The amount of money involved is also a number. This number is divided in to a specific number of periods. Fig. 4 shows the number 10 divided into equal parts. Along the horizontal axis. Each part has a Value. This Value is multiplied by its self the number of times as its Part divisor. See Fig. 4. This is an analysis of the number ten. Ten is divided by integer values 1 through 10 as indicated by the "Part" column. Ten divided by the part value is listed in the value column. The Product column is the Value column multiplied by itself by the Value of the Part. Mathematically the Product column is the Value raised to the power of the Part or $Value^{part}$. 10 divided by 3 is 3.3333 and the Product = $3.3333^3 = 37.037$. The number 10 is only used here to illustrate the process. The Value and Product columns are plotted as shown in the red and blue

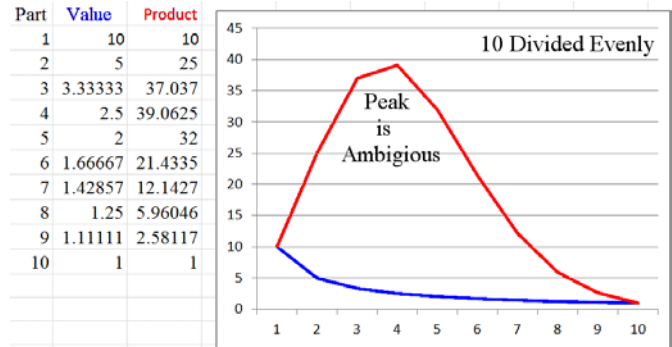


Fig. 4 – Ten divided into equal parts and multiplied.

plots. Note that the Product and Value curves start together and end together. Also note that the product values reach a peak. Fig. 5 is a similar plot for the number 20. The point we are interested is in the peak of the product. To better show the values I only plotted the first 14 of the 20 values. I also plotted the vertical axis using a logarithmic scale. (See #5 in Fundamentals of Applied Math Series in issue 22). By now you should notice the value of the 7th part. It is close to e .

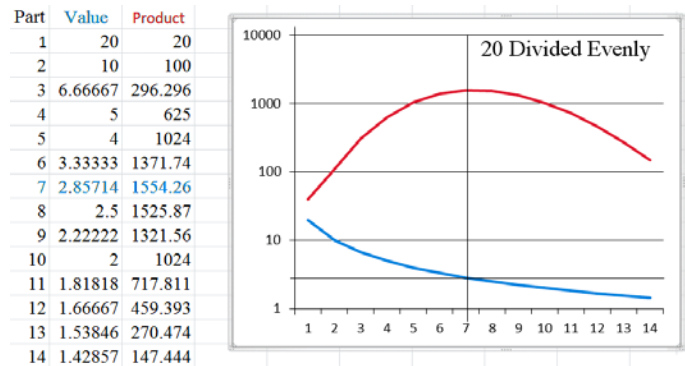


Fig. 5 – Twenty divided into equal parts.

Table 2 list the relevant values for the number 100.

Table 2 – Table of the First 40 Equal Parts of 100.

Part	Value	Product	Part	Value	Product	Part	Value	Product
1	100	100	11	9.0909	3.50E+10	21	4.7619	1.71E+14
2	50	2500	12	8.33333	1.12E+11	22	4.54545	2.92E+14
3	33.3333	37037	13	7.6923	3.30E+11	23	4.34782	4.78E+14
4	25	390625	14	7.14285	8.99E+11	24	4.16666	7.49E+14
5	20	3.20E+06	15	6.66666	2.28E+12	25	4	1.12E+15
6	16.6667	2.14E+07	16	6.25	5.42E+12	26	3.84615	1.62E+15
7	14.2857	1.21E+08	17	5.88235	1.20E+13	27	3.7037	2.25E+15
8	12.5	5.96E+08	18	5.55555	2.54E+13	28	3.57124	3.01E+15
9	11.1111	2.58E+09	19	5.26315	5.05E+13	29	3.44827	3.89EE15
10	10	1.10E+10	20	5	9.53E+13	30	3.33333	4.85E+15
						31	3.2258	5.85E+15
						32	3.125	6.84E+15
						33	3.0303	7.74E+15
						34	2.94117	8.50E+15
						35	2.85714	9.07E+15
						36	2.77777	9.39E+15
						37	2.7027	9.47E+15
						38	2.63157	9.29E+15
						39	2.5641	8.88E+15
						40	2.5	8.27E+15

Does this process provide another perspective of the value of e ? If you use ever larger values the Value in the tables approaches e : **2.71828 18284 59045 23536 02874 71352 66249 77572 47093 69995 95749 ...**

If you have an electrical background the number 36 should “pop out.” This number is the “complement” of the time constant. The Product peak value corresponds to the 37th value or 2.7027.

Time constant, $\tau = 1 - 1/e$ (63.2120558829...%) (e6)

Is $100 - 63.2120558829 = 36.7879441171...$ the proportional value corresponding to e of the numbers in Fig. 4, Fig. 5, and Table 2.

Let's evaluate a few Part and Product values for 1,000. $367 = 2.7247$ and $5.8553E159$;
 $368 = 2.7173$ and $5.8614E159$ (Peak)
 $369 = 2.7100$ and $5.85154E159$

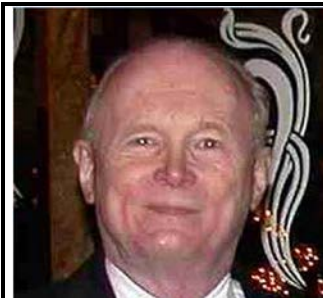
Let's evaluate a few Part and Product values for 10,000. $3677 = 2.719608$ and **Overflow**
 $3678 = 2.718868$ and **Overflow**
 $3679 = 2.718124$ and **Overflow**

We can't see the peak value of the Product because our calculator overflows.

Summary and Conclusion

e is an interesting and exciting number that is fundamental to nature and mathematics. It is an irrational number and a member of an exclusive group of five numbers that make up Euler's famous Identity: $e^{i\pi} + 1 = 0$. e has a decimal expansion that never terminates and is never periodic. In this article e is defined, and its value accurate to 56 digits is provided. Several plots are used to illustrate the nature of this exciting number. e is the base of the natural logarithms and a vital part of interest calculations involving the time value of money, TVM. It is so interesting that there is a book dedicated to it – *e: The Story of A Number*. Finally, a memory aid given in this article helps the reader to memorize and remember 15 digits of e .

About the Editor



Richard J. Nelson is a long time HP Calculator enthusiast. He was editor and publisher of *HP-65 Notes*, *The PPC Journal*, *The PPC Calculator Journal*, and the *CHHU Chronicle*. He has also had articles published in *HP65 Key Note* and *HP Key Notes*. As an Electronics Engineer turned technical writer Richard has published hundreds of articles discussing all aspects of HP Calculators. His work may be found on the Internet and the HCC websites at: <http://hhuc.us> He proposed and published the PPC ROM and actively contributed to the UK HPCC book, *RCL 20*. His primary calculator interest is the User Interface.