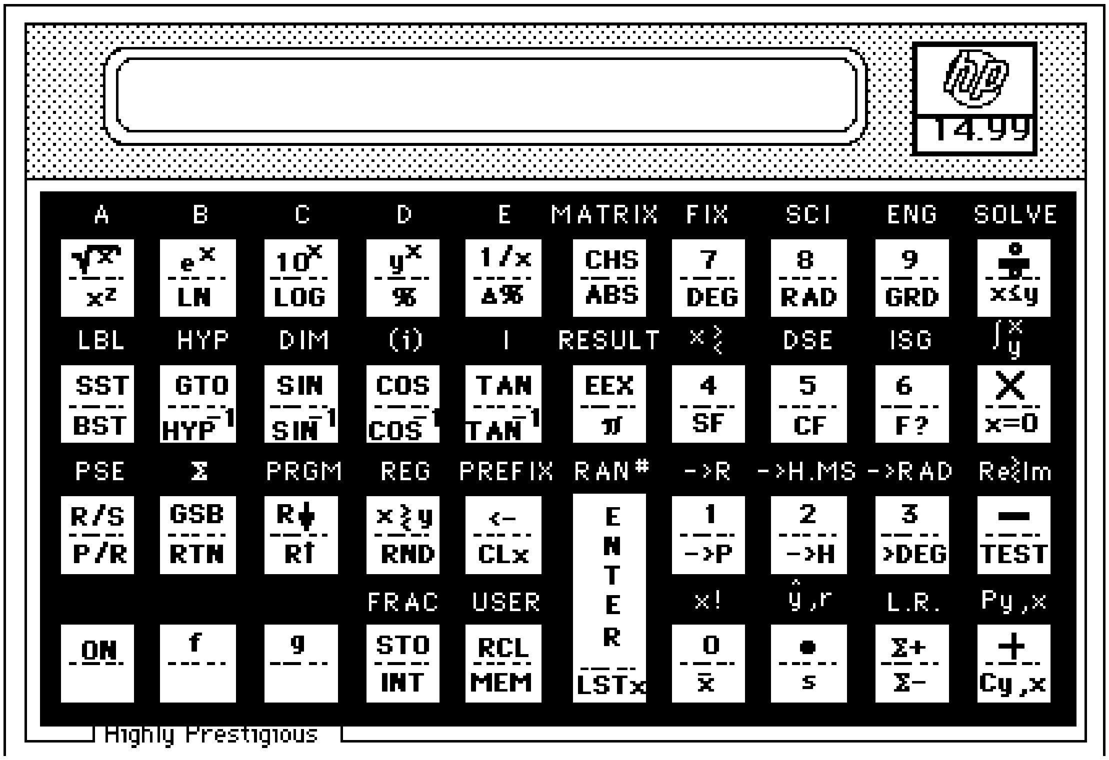
Fifteen-C

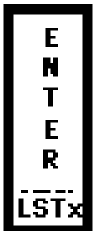
INRESCO

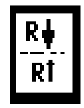


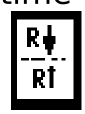
Function:

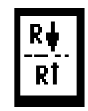
HP-15C Calculator emulation. HP hand held calculators, if not the first to use reverse Polish notation (RPN), certainly advanced the use and familiarity of this paradigm.

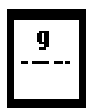
RPN, or post fix operator notation, works this way: Key in a number, when you are finished keying in the number, press the enter key. This terminates the enter mode and prepares the calculator for the next operation, whether it be entering the next number or performing an arithmetic operation(addition, subtraction, multiplication or division) or a trigonometric operation(sine, cosine or tangent). When the enter key,

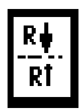


, is pressed, the number in the display is entered into the stack, the name of a convenient device, where the numbers are stored in preparation to their use in some arithmetic operation. The arrangement of the stack is such that the number viewed in the display is in the stack register designated X. Three other stack registers exist, designated Y, Z and t. These are named for the three axes of the Cartesian coordinate system, and time. To view the contents of the stack, use the roll down button 

. Pressing the roll-down button copies the contents of the X stack register into a temporary register, copies the contents of the Y register into the X register, copies the Z register into Y, copies time into Z, and takes the former contents of X, stored in temp, and places it in t. Pressing 

three more times repeats this process three more times, restoring the state of the stack to its state before 

was pressed the first time. Pressing 



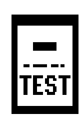
(g selects the lower function on the button face) performs a similar function, except that t goes into X, Z goes into t, Y goes into Z and X goes into Y.

As an example, to subtract the number 2.3 from 4.2, first key in the number 4.2, press enter, 4.20000000 will appear if the display is set to display 8 digits past the decimal point.

t ????
Z ????
Y 4.20000000
X 4.20000000

When the enter key is pressed, 4.2 is copied into Y, and remains displayed in X. Next enter the number 2.3. The stack now looks like this:

t ????
Z ????
Y 4.20000000
X 2.3

The 4.20000000 in the display is overwritten by 2.3. Press the minus key 

. The subtraction is performed, subtracting the value in X from that value in Y and rolling the stack down, to display:

t ????
Z ????
Y ????
X 1.90000000

Some operations can be performed without the use of the enter key. For example to square a number, 2.4, first enter it:

t ????
Z ????
Y 1.90000000
X 2.4

then press [g] [√x / x²]

t ????
Z ????
Y 1.90000000
X 5.76000000.

If you wish to enter the number "24.68024", and you plan to use the key clicks [2 / ->H] [4 / SF] [• / s] [6 / F?] [8 / RAD] [0 / x̄] [2 / ->H] [4 / SF], and find that after you have pressed the last 4, that some how you pressed the [9 / GRD] key instead of the [0 / x̄] key and the actual keypress sequence must have been [2 / ->H] [4 / SF] [• / s] [6 / F?] [8 / RAD] [9 / GRD] [2 / ->H] [4 / SF] and your display looks like this:

t ????
Z 1.90000000
Y 5.76000000
X 24.68924

You can use the [<- / CLx] key to correct your mistake. Press it once:

t ????
Z 1.90000000
Y 5.76000000
X 24.6892

Press [← / CLx] again:

t ????
Z 1.90000000
Y 5.76000000
X 24.689

And again:

t ????
Z 1.90000000
Y 5.76000000
X 24.68

Then key in correctly [0 / x̄] [2 / ->H] [4 / SF], providing the corrected result.

t ????
Z 1.90000000
Y 5.76000000
X 24.68024

Features:

The programmability works. [g] turns on the program enunciator. Clicking on a button or using keyboard equivalents puts code in the display. The codes are the row and column numbers of the buttons. The rows are 1 at the top and 4 at the bottom. The columns are 1 at the left and 0 at the right. The numbers are represented by themselves and not by row and column notation. Establish a label by the sequence [f] [LBL SST / BST] [A √x / x²] and have that be a starting point for a routine. Key in the program steps and use [g] [GSB / RTN] to end the routine. Start the program by keying in [f] [A √x / x²], or in USER mode, [A √x / x²]. The calculator will go to the label

[A √x / x²] and begin executing instructions. Establish the label [• / s] [6 / F?] with the sequence

[f] [LBL SST / BST] [• / s] [6 / F?] and follow with instructions. Terminate with [g] [GSB / RTN]. Use

[GSB / RTN] [• / s] [6 / F?] and program flow will go to the subroutine [• / s] [6 / F?] and execute

instructions until the [g] [GSB / RTN] is reached, at which point program flow will return to the

instruction following the [GSB / RTN] [• / s] [6 / F?] instruction and continue from there.

Program clearing: In program mode, click [f] [PRGM R↓ / R↑].

Program movement: In program mode, move forward one instruction by clicking [SST / BST].

Move backward one instruction by clicking [g] [SST / BST]. To move to the start of program

memory, in run mode(toggled by clicking [g]) click [f] [PRGM R↓ / R↑]. To move to a

particular line in the program, use the sequence [GTO / HYP⁻¹] [CHS / ABS] n n n, where n n n represents the

line number in the program. This works in run mode or in program mode.

Program editing: To insert instructions in the middle of a program, single step in program mode to the instruction preceding the point where you wish to add instructions and begin to key in instructions. Instructions following will be pushed down in program memory to make room. To delete program instructions, single step in program mode to the instruction you wish

to delete and use the [← / CLx] button. Program steps following will move up to fill in the space.

Program branching and control: Branching occurs two ways, with subroutines, GSB, and go to's, GTO. Both of these operate in both direct and indirect. In the direct mode, a label is used, for example [GSB / RTN] [A / $\sqrt{x}$ / x^2] or [GTO / HYP^{-1}] [• / s] [6 / F?]. The GSB requires that there be a [g] [GSB / RTN] todirect the flow of the program back to the instruction following the GSB instruction.

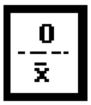
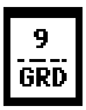
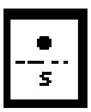
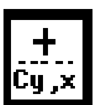
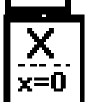
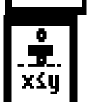
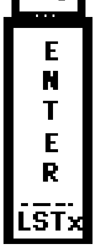
The GTO instruction has no return requirement. In the indirect mode, use [GSB / RTN] [I / TAN / TAN^{-1}] or [GTO / HYP^{-1}] [I / TAN / TAN^{-1}]. The I above the TAN button stands for Index. You could use the sequences [GSB / RTN] [f] [I / TAN / TAN^{-1}] or [GTO / HYP^{-1}] [f] [I / TAN / TAN^{-1}] but the f is understood and not necessary. The integer portion of the number stored in the Index register is used for branching. If the number is negative, the branch will be to the line number corresponding to the absolute value of the integer portion of the Index register contents. If the number is positive, the branch will be to a label according to the following table:

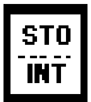
contents of RI	transfer to
0	[f] [LBL / SST / BST] [0 / $\bar{x}$]
1	[f] [LBL / SST / BST] [1 / ->P]
2	[f] [LBL / SST / BST] [2 / ->H]
3	[f] [LBL / SST / BST] [3 / >DEG]
4	[f] [LBL / SST / BST] [4 / SF]
5	[f] [LBL / SST / BST] [5 / CF]
6	[f] [LBL / SST / BST] [6 / F?]

7	f	LBL SST / BST	7 / DEG	
8	f	LBL SST / BST	8 / RAD	
9	f	LBL SST / BST	9 / GRD	
10	f	LBL SST / BST	• / s	0 / $\bar{x}$
11	f	LBL SST / BST	• / s	1 / ->P
12	f	LBL SST / BST	• / s	2 / ->H
13	f	LBL SST / BST	• / s	3 / >DEG
14	f	LBL SST / BST	• / s	4 / SF
15	f	LBL SST / BST	• / s	5 / CF
16	f	LBL SST / BST	• / s	6 / F?
17	f	LBL SST / BST	• / s	7 / DEG
18	f	LBL SST / BST	• / s	8 / RAD
19	f	LBL SST / BST	• / s	9 / GRD
20	f	LBL SST / BST	A $\sqrt{x}$ / x^2	

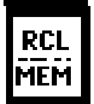
21	f	LBL SST/BST	B e^x/LN
22	f	LBL SST/BST	C 10^x/LOG
23	f	LBL SST/BST	D y^x/%
24	f	LBL SST/BST	E 1/x/Δ%

The keypad works for 30 of the 39 buttons. Use mouse clicks in the window for the rest of them. The keyboard equivalents are as follows:

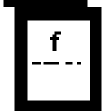
	Buttons	Keyboard equivalents
		0 on the keypad
	• • •	• • •
		9 on the keypad
		. on the keypad
		+ on the keypad
		- on the keypad
		* on the keypad
		/ on the keypad
		ENTER on the keypad



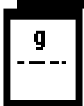
s or S



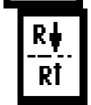
r or R



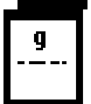
f or F

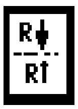


g or G

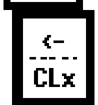


down cursor arrow

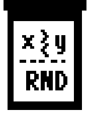




up cursor arrow

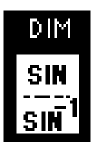


left cursor arrow

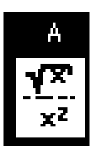


right cursor arrow

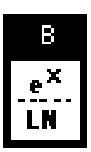
To simplify matrix operations these buttons affecting matrix operations can be accessed by the keyboard equivalents as follows:



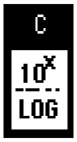
i or I for dimension



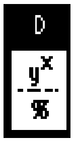
a or A, so recall matrix A would be: r m a



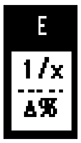
b or B, so to dimension matrix B as 2 x 3 would be: 2 enter 3 f i b



c or C



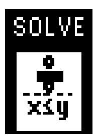
d or D



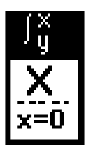
e or E

m or M

combinations of keyboard equivalents work, for example, f r turns USER mode on.

The SOLVE function, 

, is not implemented, and probably won't be.

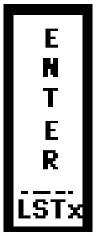
The numerical integration, 

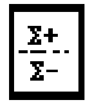
, is not implemented, and probably won't be.

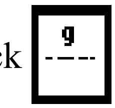
Matrix inversion doesn't work.

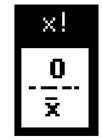
Matrix addition, subtraction and multiplication work.

Complex addition, subtraction, multiplication and division works, as well as complex number square root.

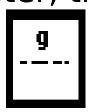
Linear Regression works.To find a regression line, key in data pairs. Key in the Y value first, click the 

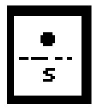
button, then key in the X value. Click the 

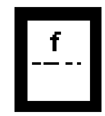
key. The cumulative numbers of data pairs, n, appears in the x-register. Repeat this sequence for as many data pairs as you have. The mean and standard deviation of both the X- and Y- values are available. To get the mean, click 

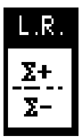


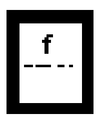
.

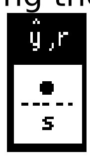
The mean of X is in the X-register, the mean of Y is in the Y-register. To get standard deviations, click the sequence 



. The standard devition of X is in the X-register, the standard deviation of Y is in the Y-register. To find the regression line, key the sequence 



. This computes, by the method of least squares, the slope, A, and the y-intercept, B, of the linear equation: y = Ax + B. B is found in the X-register, and A is in the Y-register. To find the linear estimate for y, after collecting the X- and Y- values as described above, enter a proposed value of x and press 



. The linear estimate, y^ is in the X- register, and the correlation coeficient, r, is in the Y- register.

The memory manager works.

The memory consists of 67 registers, R0 through R65 and the Index Register, RI. R0 and R1 are fixed but R2 through R65 can be reallocated for different purposes. In the physical HP-15C hand held calculator, the memory bytes can be allocated to store program instructions, but in the software emulation that I have created, the programmability has not been implemented yet so that function of memory is ignored.

Memory can be allocated two ways: To be used as registers for storing numbers, or for use as storage for matrix elements. This is described below.

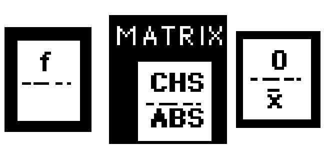
You can dimension a matrix up to 8 x 8.

Mathematical knowledge concerning matrix operations and complex number operations is assumed.

Operation:

To facilitate understanding of the following operations a menu has been added called windows. This menu has two items, matrices and registers. Selecting the Matrices menu item causes windows to appear when the matrices are created, which happens when a dimension of 1 x 1 or greater is given to a particular matrix as described below. The size of the window is appropriate to the size of the matrix. The windows are named Matrix A, Matrix B and so on. Selecting the Register menu item causes a window to appear which reports how the registers are used for the matrix functions. This works as follows: When a matrix is dimensioned, the registers are allocated for that matrix, starting at register 65 and counting downward to the limit of the partition, which is variable. When the registers are allocated, a window appears with as many cells as are needed appropriate to the size of the matrix, and filled with zeros. If another matrix is dimensioned, more cells are added in the window in one column until 20 cells are displayed. For more than 20 cells, a second column is added, for more than 40 a third, for more than 60 a fourth. If and when values are assigned to the matrix elements they appear in the appropriate matrix window and also in the register window. If, after dimensioning and assigning values to all of the five matrices, one dimensions the third matrix created to zero by zero, thereby releasing the memory used, and the contents of the array used, the memory manager moves the values from the lower numbered array locations into the array locations freed in the process, thus compacting the applications heap and preventing memory fragmentation.

To clear the contents of the matrix elements and set the dimensions of all of them to 0 by 0, click the following button sequence:



.

At this time, all the matrices A, B, C, D and E have the capabilities described in the following:

To dimension a matrix:

1) enter the number of rows into the display.

2) click the [ENTER / LSTx] button.

3) enter the number of columns into the display.

4) click the sequence [f] [DIM: SIN / SIN⁻¹] and

[A: √x / x!] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

to dimension the matrices A or B or C or D or E respectively

To display the dimensions of a matrix:

Method A:

click the sequence [RCL / MEM] [MATRIX: CHS / ABS] followed by the label A, B, C, D or E.

ie [A: √x / x!] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

Method B:

click the sequence [RCL / MEM] [DIM: SIN / SIN⁻¹] followed by the label A, B, C, D or E.

the # rows will be in the Y-register.
the # columns will be in the X-register.

To store and recall matrix elements:

Method A:

Storage Registers R0 and R1 are used to store the row and column numbers of a matrix element respectively.

1) To set R0 and R1 to 1 and 1 click [f] [MATRIX CHS/ABS] [1/->P]

2) Activate USER mode clicking [f] [USER RCL/MEM]. This allows for automatic incrementing of the values in R0 (rows) and R1 (columns) so that the values are placed in the appropriate memory locations corresponding to the row and column indices.

3) If you are storing elements, key the value to be stored in row r column c where r and c are initially 1 and 1.

4) Click [STO/INT] followed by [A √x/x²] or [B eˣ/LN] or [C 10ˣ/LOG] or [D yˣ/%] or [E 1/x/Δ%]

5) To recall matrix elements,

Click [RCL/MEM] followed by [A √x/x²] or [B eˣ/LN] or [C 10ˣ/LOG] or [D yˣ/%] or [E 1/x/Δ%]

6) Repeat steps 3 & 4 to store all elements of the matrix. The values of r and c are automatically incremented.

or

Repeat step 5 for all elements of the matrix. The values of r and c are automatically incremented.

Method B:

With USER mode turned off,

Store the row and column numbers in R0 and R1 as described above.

To recall an element value

Click [RCL / MEM] followed by [A: √x / x²] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

To store an element value, key the value into the display using mouseclicks or the numeric keypad.

Click [STO / INT] followed by [A: √x / x²] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

Method C:

To recall an element value, enter the row number and column number into the stack in that order, then:

Click [RCL / MEM] [g] followed by [A: √x / x²] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

To store an element value, enter the value on the stack followed by the row number then the column number, then:

Click [STO / INT] [g] followed by [A: √x / x²] or [B: eˣ / LN] or [C: 10ˣ / LOG] or [D: yˣ / %] or [E: 1/x / Δ%]

How the memory manager works in this context is actually transparent to the user, but a description follows for the dedicated technoweenie. As an example I describe the process of successively dimensioning matrices: First, dimension [C: 10ˣ / LOG] as 2 x 2. This allocates the registers R65 through R62. Then dimension [E: 1/x / Δ%] as 2 x 2. This allocates the registers R61 through R58. Then dimension [A: √x / x²] as 3 x 3. This allocates the registers R57 through R49. Then dimension [D: yˣ / %] as 4 x 4. This allocates the registers

R48 through R33. Then dimension [B e^x / LN] as 2 x 2. This allocates the registers R32 through R29. As the HP-15C comes from the factory, and as I have implemented it, registers through R19 are allocated such that you can store 20 different numbers as described in the following section. Thus after dimensioning 5 matrices, 37 registers are used, leaving 9 unallocated. Matrix [B e^x / LN], the last one allocated, I would like to re-dimension to 4 x 4 for a total of 16 registers. What is available is the unallocated 9 plus the 4 that would be recovered when the memory allocated to [B e^x / LN] is released, sums to 13, 3 less than the 16 needed. If I re-dimensioned matrix [E 1/x / Δ%] to 0 x 0 this would release another 4 registers, but if I do so, the heap is fragmented, and the 17 registers that are free are not contiguous, as is needed for allocation for a matrix in the model for memory that I have implemented. But it works, as I have designed the memory manager to automagically compact the heap.

Memory Reallocation

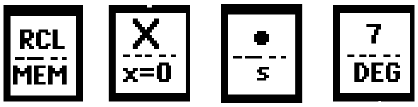
There are 21 registers available R0 through R9 and R.0 through R.9 and RI

Data can be stored and recalled using mouse clicks or the numeric keypad.

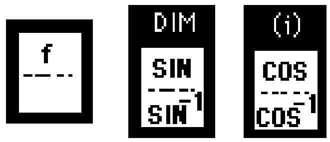
Click [STO / INT] [5 / CF] to store in register 5 the contents of the display.

Click [RCL / MEM] [• / s] [9 / GRD] to recall the value stored in register 19.

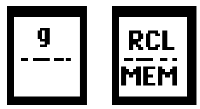
Click [STO / INT] [+ / Cy,x] [6 / F?] to add the contents of the display to the value stored in register 6.

Clicking 

multiplies the value in the display by the contents of register 17.

The memory that is available can be reallocated to allow up to 67 registers, including the previously mentioned 21. To do this, click 

after first placing the number, dd, of the highest data storage register in the display. dd suffers the constraint 1 <= dd <= 65.

To test the value of dd, click the sequence 

. The calculator will display briefly the memory configuration in the form:

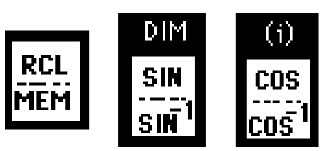
dd uu pp-b

where

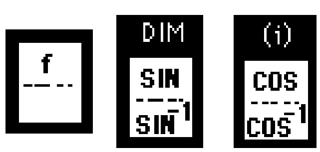
dd = the number of the highest numbered register in the data storage pool.

uu = the number of uncommitted registers in the common pool.
(currently unimplemented)

pp = the number of registers containing program instructions.
(currently unimplemented)

(Pressing 

will display the value of dd.)

(Pressing 19 

will redimension the memory space so that dd = 19.)

Significance:

Registers, complex functions and matrix manipulations are affected by each other.

Once additional registers have been allocated, they can be accessed by direct and indirect addressing. This uses [I: TAN / TAN^{-1}] and [(i): COS / COS^{-1}] .

There are several ways to manipulate the value of the INDEX Register RI.

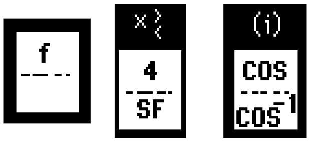
You can store (or recall) directly, as with [STO / INT] (or [RCL / MEM]) [I: TAN / TAN^{-1}] . This places the value in the display in RI (or recalls the value of RI into the display).

You can also do register arithmetic as in [STO / INT] [+ / Cy,x] [I: TAN / TAN^{-1}] , which adds the display value to the contents of RI.

Similarly, [STO / INT] (or [RCL / MEM]) [(i): COS / COS^{-1}] stores the display value into (or recalls into the display the value) in the register (from the register) corresponding to the absolute value of the integer portion of the number in the index register RI.

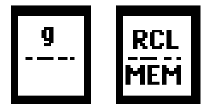
You can also do indirect register arithmetic. For example, [RCL / MEM] [÷ / x≤y] [(i): COS / COS^{-1}] divides the display value by the value contained in the register corresponding to the absolute value of the integer part of the number in RI.

Clicking [f] [x≷: 4 / SF] [I: TAN / TAN^{-1}] exchanges the contents of the x-register (the display) and the INDEX register, RI.

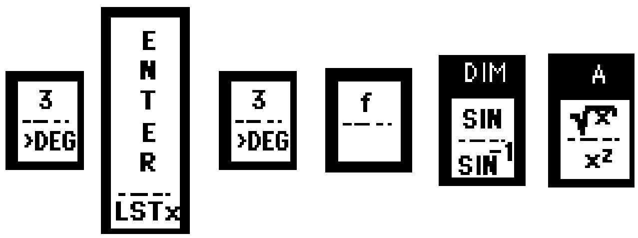
Clicking 

exchanges the contents of the x-register (the display) and the data storage register addressed by the number(0 to 65) stored in the INDEX register, RI.

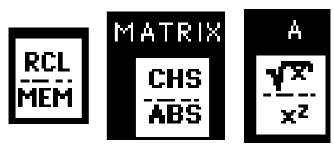
Significance of the value dd and uu

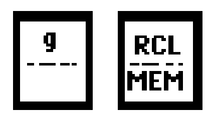
With the initial values 19 46 0-0 (check this using the mouse clicks 

, if there is some other value, see Note 1 below), see what happens when a matrix is dimensioned.

Click the sequence 

to dimension the matrix A to be a 3 by 3 matrix.

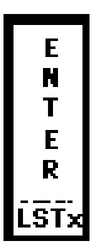
Check its dimension by clicking the sequence 

. You should see in the display "A 3 3". Now check the memory allocation with the mouse click sequence 

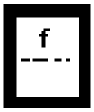
. The display should read "19 37 0-0" . This demonstrates that registers for matrix elements are allocated from the "uu" field.

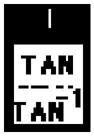
Working with complex numbers:

Key(with mouseclicks) in the real part of the number into the display

Click 

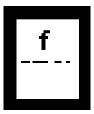
Key the imaginary part of the number into the display

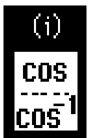
Click 

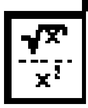


to enable the imaginary stack and store the imaginary part in it.

After performing arithmetic operations on complex numbers the display contains the real part of the sum.

Click 



to display the imaginary part of the number. Use this to retrieve the results of clicking  to compute the square root of a complex number.

Some of the other functions of this app are obvious, some are not.

Error Conditions

Error 0: Improper Mathematics Operation

Error 1: Improper Matrix Operation

Error 2:

Error 3:

Error 4:

Error 5:

Error 6:

Error 7:

Error 8:

Error 9:

Error 10: Insufficient Memory

Meaning:
There is not enough memory to perform a given operation.

Error 11: Improper Matrix Argument

Inconsistent or improper matrix arguments for a given matrix operation:

Note 1:

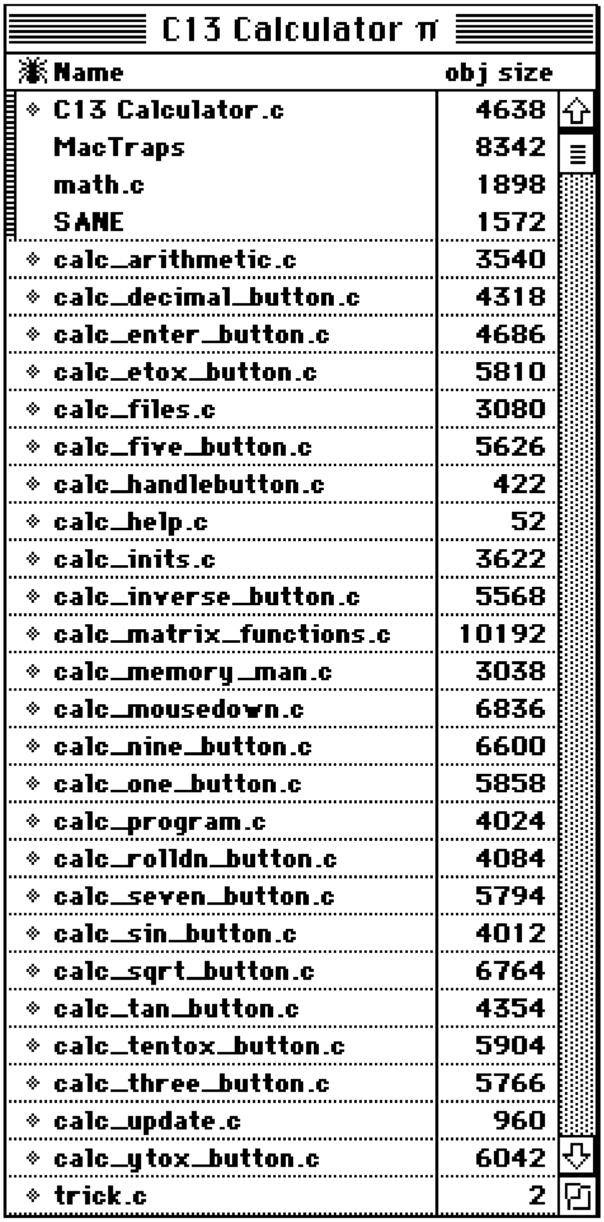
This documentation is far from complete, as is the functionality of the application. Copies of the original Hewlett-Packard documentation may be obtained for a copyright fee to HP and photocopy charges.

Send bug reports, praise, flames, money or job offers to:

James Ullrey
INRESCO
AOL: JamesU9
AppleLink: ULLREY
BMUG RBBS:Conferences:Macintosh:Programming:James C. Ullrey
PRODIGY:dbpt67a

Rev 13 features:

In order that the matrix functions be more visible I have implemented a menu selection with items Matrices and Registers, which, when the Matrix item is selected, displays matrix windows for those matrices which have been dimentioned. The Register menu item causes the contents of the registers to be displayed. This causes problems on machines such as the Mac Plus and the Classic, which have small screens compared to other models. Thus I have implemented a Transmogrifier function, which is activated by clicking on the HP logo box while holding down the command key. This hides the calculator window and displays a smaller version. Both calculator faces are draggable by clicking and dragging in the HP logo box. The system is polled at launch time for the presence of color in the main display and displays a color splash screen on color displays and a black and white splash screen on black and white displays.



C13 Calculator π

Name	obj size
C13 Calculator.c	4638
MacTraps	8342
math.c	1898
SANE	1572
calc_arithmetic.c	3540
calc_decimal_button.c	4318
calc_enter_button.c	4686
calc_etox_button.c	5810
calc_files.c	3080
calc_five_button.c	5626
calc_handlebutton.c	422
calc_help.c	52
calc_inits.c	3622
calc_inverse_button.c	5568
calc_matrix_functions.c	10192
calc_memory_man.c	3038
calc_mousedown.c	6836
calc_nine_button.c	6600
calc_one_button.c	5858
calc_program.c	4024
calc_rolldn_button.c	4084
calc_seven_button.c	5794
calc_sin_button.c	4012
calc_sqrt_button.c	6764
calc_tan_button.c	4354
calc_tentox_button.c	5904
calc_three_button.c	5766
calc_update.c	960
calc_ytox_button.c	6042
trick.c	2

Disclaimer

The author makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose, regarding Fifteen-C. The author does not warrant, guarantee or make any representations regarding the use or the results of the use of Fifteen-C in terms of its correctness, accuracy, reliability, currentness or otherwise. The entire risk as to the results and performance of Fifteen-C is assumed by you.

In no event will the author be liable to you for any consequential, incidental or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use Fifteen-C even if the author has been advised of the possibility of such damages.