



## HP Forum Archive 15

[ [Return to Index](#) | [Top of Index](#) ]

### HP-15C Mini-Challenge: RSUM, CSUM

Message #1 Posted by [Valentin Albillo](#) on 10 June 2005, 5:00 a.m.

Hi, all

As an appetizer before the next S&SMC (#11), here you are, a new, *very easy*, small, and potentially useful mini-challenge specifically for the HP-15C.

#### Tha facts

Many handhelds and calculators that have built-in matrix operations capabilities do include a couple of matrix functions, usually called RSUM/CSUM that return the sum of all rows/columns of any given matrix. This is the case, for instance, for the HP41C's Advantage ROM, where according to its manual:

```
CSUM finds sums of columns and puts them in a row matrix
RSUM finds sums of rows and puts them in a column matrix
```

i.e., if some matrix A is 4x5, say, CSUM would return a 1x5 matrix, B, where each element would be the sum of the corresponding column in A. Likewise, RSUM would return a 4x1 matrix, B, where each element would be the sum of the corresponding row in A.

Now, the HP-15C does not include such matrix operations as built-in functions, so this is the

#### Challenge

You must write two *separate* small routines (LBL ... RTN) that will implement the RSUM and CSUM functionality described above. Your routines should assume that there exists an MxN matrix called A, and upon being called must return a matrix B to the X register, containing the respective sums. For the case of the RSUM routine, the resulting matrix B must be Mx1, while for the CSUM routine, it must be 1xN.

Your must optimize your routines for speed and space, the faster and shorter the better. Here's an example. Given:

$$A = \begin{array}{c|ccccc|} & 1 & 7 & -3 & 2 & 5 & | \\ & 0 & 4 & 5 & -6 & 3 & | \\ & 4 & 8 & -3 & 1 & 7 & | \\ & -8 & 3 & -6 & 12 & 6 & | \end{array}$$

CSUM must return:  $B = \begin{vmatrix} -3 & 22 & -7 & 9 & 21 \end{vmatrix}$

RSUM must return:  $B = \begin{vmatrix} 12 \\ 6 \\ 17 \\ 7 \end{vmatrix}$

### Epilogue

My original solutions, not counting the mandatory LBL and RTN, are 8 steps for RSUM and 9 steps for CSUM. Both can return the sum of all rows/columns of a 6x8 matrix in 5 seconds flat.

Now it's your turn ! Don't be afraid to try, it's actually very easy. Let's see your solutions ! :-)

Best regards from V.

### Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #2 Posted by [Thibaut.be](#) on 10 June 2005, 6:57 a.m.,  
in response to message #1 by Valentin Albillo

Hi Valentin,

I don't have a 15C in my drawer here, but is the solution like multiplying with a [1 1 1 1] matrix for Csum and [1 1 1 1 1] for rsum ?

Just very old memories from long time ago math classes.

### Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #3 Posted by [Arnaud Amiel](#) on 10 June 2005, 7:32 a.m.,  
in response to message #2 by Thibaut.be

So now the challenge is to solve MC11 efficiently without the use of Matrices...

I am now going to reread the 15C manual about matrices. They are some much easier with a graphical calc.

Arnaud

**Re: HP-15C Mini-Challenge: RSUM, CSUM**

Message #4 Posted by **Valentin Albillo** on 10 June 2005, 8:09 a.m.,  
in response to message #3 by Arnaud Amiel

Hi, Arnaud:

Arnaud posted:

"So now the challenge is to solve MC11 efficiently without the use of Matrices.."

Say what !?

Best regards from V.

**Re: HP-15C Mini-Challenge: RSUM, CSUM**

Message #5 Posted by **Arnaud Amiel** on 10 June 2005, 2:55 p.m.,  
in response to message #4 by Valentin Albillo

I mean when the MC11 comes out, I will certainly try it with matrices as it seems from your post that it will help, but I will also try without.

Arnaud

**Re: HP-15C Mini-Challenge: RSUM, CSUM**

Message #6 Posted by **Namir** on 10 June 2005, 12:26 p.m.,  
in response to message #1 by Valentin Albillo

Your challenge seems to be inspired by MATLAB's sum functions that work on rows and columns of matrices????

Namir

**Re: HP-15C Mini-Challenge: RSUM, CSUM**

Message #7 Posted by **Eamonn** on 10 June 2005, 3:00 p.m.,  
in response to message #1 by Valentin Albillo

Hi Valentin,

Here is my solution:

```
LBL A      ; Row Sum
f RESULT B ; Store the result in B
RCL DIM A  ; X = #columns in A, Y = #rows in A
1          ;
f DIM C    ; C is a #columns in A x 1 matrix
STO MATRIX C ; All entries in C are set to 1
RCL MATRIX A ;
RCL MATRIX C ;
*          ; B = A * C
RTN
```

```
LBL B      ; Column sum
f RESULT B ;
RCL DIM A  ; X = #columns in A, Y = #rows in A
X<>Y      ; X = #rows in A, Y = #columns in A
1          ;
f DIM C    ; C is a #rows in A x 1 matrix
STO MATRIX C ; All entries in C are set to 1
RCL MATRIX A ;
RCL MATRIX C ;
f MATRIX 5 ; B = Transpose(A) * C
RTN
```

The Row Sum requires 10 steps including LBL and RTN and the Column Sum requires 11 steps including LBL and RTN.

For the Row Sum, the routine creates a matrix, C, of size (#columns in A) x 1 and sets each element of C equal to 1. Matrix A is multiplied by the matrix C. The result is in matrix B and the matrix descriptor for B is returned on the stack.

For the Column Sum, the routine creates a matrix, C, of size (#rows in A) x 1 and sets each element of C equal to 1. The transpose of matrix A is multiplied by the matrix C, using the fMATRIX 5 command. The result is again in matrix B and the matrix descriptor for B is again returned on the stack.

Best Regards,

Eamonn.

### Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #8 Posted by [Eamonn](#) on 11 June 2005, 12:05 a.m.,  
in response to message #7 by Eamonn

Valentin,

I just re-read your challenge and realized that in my haste to post a solution, I didn't notice that the result for the CSUM function should be a 1 x (#columns in A) matrix. My solution returns a (#columns in A) x 1 matrix.

However, re-ordering a couple of lines in my original solution fixes this defect. Here is my amended solution...

```
LBL B          ; Column sum
f RESULT B
RCL DIM A      ; X = #columns in A, Y = #rows in A
X<>Y          ; X = #rows in A, Y = #columns in A
1
;
f DIM C        ; C is a #rows in A x 1 matrix
STO MATRIX C   ; All entries in C are set to 1
RCL MATRIX C   ; *** These two lines have been swapped
RCL MATRIX A   ; ***
f MATRIX 5     ; B = Transpose(C) * A
RTN
```

Best Regards,

Eamonn.

### Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #9 Posted by [Thibaut.be](#) on 11 June 2005, 1:44 a.m.,  
in response to message #8 by Eamonn

Well, I'm sorry I had no 15C available yesterday, but this confirms that my approach was correct. I deserve a chocolate medal as I haven't calculated matrixes for 15 years now. Congratulations for posting this solution !

### Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #10 Posted by [Arnaud Amiel](#) on 11 June 2005, 1:25 p.m.,  
in response to message #7 by Eamonn

I haven't tried yet, but it looks to me that something like:

```
LBL B
MATRIX 4      ;transpose input to have problem A
GSB A
```

```
MATRIX 4 ;transpose output  
RTN
```

Would work, maybe more slowly but it is shorter?

Arnaud

## Re: HP-15C Mini-Challenge: RSUM, CSUM

Message #11 Posted by [Valentin Albillo](#) on 14 June 2005, 9:17 a.m.,  
in response to message #1 by Valentin Albillo

Hi, all:

Congratulations to all of you who posted (or suggested) correct solutions. These are my original solutions, that are nearly identical to Eamonn's:

For columns:

```
01 LBL A  
02 RCL DIM A  
03 RDOWN  
04 1  
05 DIM C  
06 STO MATRIX C  
07 RESULT B  
08 RCL MATRIX C  
09 RCL MATRIX A  
10 MATRIX 5  
11 RTN
```

For rows:

```
01 LBL A  
02 RCL DIM A  
03 1  
04 DIM C  
05 STO MATRIX C  
06 RESULT B  
07 RCL MATRIX A  
08 RCL MATRIX C  
09 *  
10 RTN
```

I'm a little too busy to comment further, again thanks for your interest, and

Best regards from V.

---

[ [Return to Index](#) | [Top of Index](#) ]



[Go back to the main exhibit hall](#)