Simple-Sum Function Exercises

PDF document due on 11:59:50 pm on Tuesday, May 5, 2015 via email, Subject: Simple-Sum Function Exercises.

Got a new programming job at the new accounting company “**Sumer, Inc”** and you are assigned to create three user-defined functions (1) **sumColumn, (2) sumRow, (3) sumAll** that adds up the elements of columns, rows and all elements in the array. For instance, for the **t** array will return the result shown below. Assume that **t** has **n x m** (row x column) elements.

					sumRow=	
t=	1	6	11	**=**	**18**	
	2	7	12	**=**	**21**	
	3	8	13	**=**	**24**	
	4	9	14	**=**	**27**	
	5	10	15	**=**	**30**	
	‖	‖	‖			
sumColum =	**15**	**40**	**65**			

For example, the input to the function is the **t** vector **above** and the **sumColumn** function returns the array [15, 40, 65] and the **sumRow** returns the array [18,21,24,27,30]. The **sumAll** functions will return the sum of all elements, which for the above array is [120].

You can use any other MATLAB library function within your function but I recommend you, to maximize learning, to code them using either loops or array/matrix operators (vectorization).

Hint: Use the **numel** and **size** library functions to determine the number of column, rows and elements in **t**. You can’t use the **sum** library function.

You are required to submit function code and sample running (output). Works not following instructions won’t be graded, no exceptions.

SOLUTION

```
function [s] = sumRow(t)
       %  Computes the sum of all the elements in a ROW of the t array element-by element

[n,m] =size(t); s(1)=0;
  for ii=1:1:n
    s(ii)=0;
    for jj=1:1:m
      s(ii)=s(ii)+t(ii,jj);
    end
   end
end

function [s] = sumColumn(t)
       %  Computes the sum of the columns of the t array element-by element

       [n,m] =size(t);

       for jj=1:1:m
           s(jj)=0;
           for ii=1:1:n
             s(jj)=s(jj)+t(ii,jj);
           end
       end
end

function [s] = sumAll(t)
       %  Computes the sum of all the elements of the t array

       n =numel(t); s=0;
       for ii=1:1:n
              s=s+t(ii);
       end
end
```

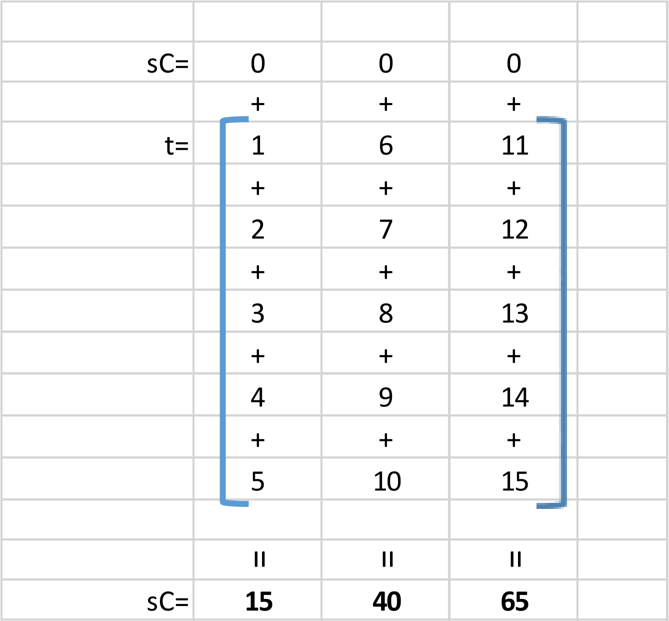
Alternative-2 to the Above Solutions:

```
function [sC]=sumColumn2(t)
        % This function sums up the values of the column in an array.

        sC=[0,0,0];        % First three elements on top (accumulators) are zero

        [r,c]=size(t);
        for ii=1:1:r
                sC=sC+t(ii,:);      % Sum occurs simultaneously in each column
        end
end
```

<u>Below the mechanism of this algorithm</u>:



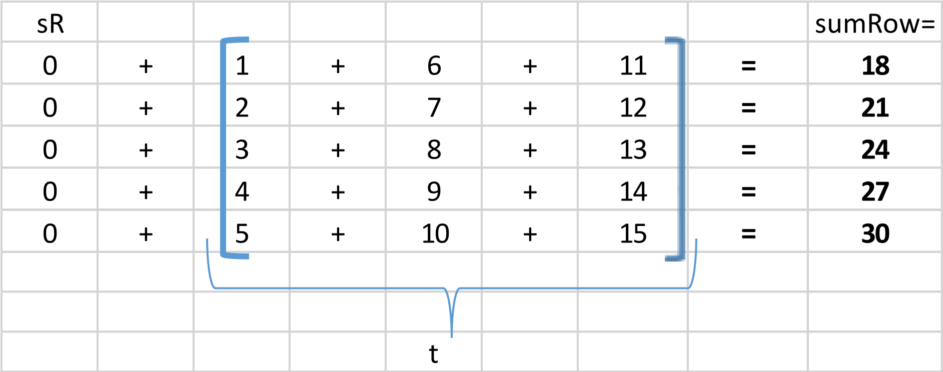
sC=	0	0	0	
	+	+	+	
t=	1	6	11	
	+	+	+	
	2	7	12	
	+	+	+	
	3	8	13	
	+	+	+	
	4	9	14	
	+	+	+	
	5	10	15	
	ıı	ıı	ıı	
sC=	**15**	**40**	**65**	

```
function [sR]=sumRow2(t)
%This function sums up the values of the rows in an array.

        [r,c]=size(t);

        sR=zeros(r,1);    % A column of zeros (i.e., rx1 array of zeros)
          for jj=1:1:c
                sR=sR+t(:,jj);
          end
end
```

Below the mechanism of this algorithm:



sR								sumRow=
0	+	1	+	6	+	11	=	**18**
0	+	2	+	7	+	12	=	**21**
0	+	3	+	8	+	13	=	**24**
0	+	4	+	9	+	14	=	**27**
0	+	5	+	10	+	15	=	**30**
				t				

Alternative-3. In this problem it was prohibited to use the sum library function. Below is a solution using the sum function in vectorized code. You better read the documentation of this function so you can understand the algorithm.

```
function [sC]=sumColumn3(t)

        %This function sums up the values of the column in an array.

        sC=sum(t);  % A row vector with the sum over each column

end
```

NOTE: If t is a matrix, SC is a row vector with the sum over each column (see function documentation, i.e., type >>help sum in the Command Window. Therefore, this is just "not a very smart" name change.

```
function [sR]=sumRow3(t)
%This function sums up the values of the rows in an array.

        [r,c]=size(t);
        for ii=[1:1:r]
                sR(ii,1)=sum(t(ii,:));
        end
end
```