
Table of Contents

.....	1
.....	2
INITIALIZATION	2
.....	2
CALCULATIONS	2
.....	3
COMMAND WINDOW OUTPUT	3
.....	3
ACADEMIC INTEGRITY STATEMENT	3

```
function [newArray] = PS10_sort_koike(array)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This program is designed to rearrange an user inputted array by
% identifying the arrays dimensions and then setting the lowest value
% at the top left corner of the array the following lowest number at
% the next column in the same row and this continues on to the next
% row
% until the last row last column.
%
% Function Call
% PS10_sort_koike(array);
%
% Input Arguments
% 1. Array: the user inputted array
%
% Output Arguments
% 2. newArray: the rearranged array
%
% Assignment Information
%   Assignment:      PS 10, Problem 2
%   Author:          Tomoki Koike, koike@purdue.edu
%   Team ID:         002-08
%   Contributor:     Name, login@purdue [repeat for each]
%   My contributor(s) helped me:
%       [ ] understand the assignment expectations without
%           telling me how they will approach it.
%       [ ] understand different ways to think about a solution
%           without helping me plan my solution.
%       [ ] think through the meaning of a specific error or
%           bug present in my code without looking at my code.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
%probing the dimensions of the array
[rows, cols] = size(array);

%setting a new zero matrix with the same dimesions as the input
newArray = zeros(rows, cols);
```

CALCULATIONS

```
%sorting the new array
for rowCount = 1:rows
    for colCount = 1:cols
        %finding the minimum value in each column
        minValCol = min(array);
        %finding the minimum value in the array
        minVal = min(minValCol);
        %find the multiplicity of the minimum value
        sumMinVal = sum(array(:)==minVal);
        %preallocate the zero vectors from the top left corner
        newArray(rowCount,colCount) = minVal;

        if sumMinVal > 1
            %creating a new vector with the multiplicity of the
            %minimum value found in the previous step
            minValVec = zeros(1, sumMinVal-1);
            minValVec(:) = minVal;
            %replace the values in the input array with the vector
            %created in the previous step concatenated with the values
            %in the original array which values are larger
            array = array(:)';
            array = sort(array);
            array = array((sumMinVal+1):(numel(array)));
            array = [minValVec, array];
        else
            %Replace the input array with a
            %vector that has all the values
            %greater than the minimum value
            array = array(:)';
            array = sort(array);
            array = array((sumMinVal+1):(numel(array)));
        end
    end
end

%return the sorted array
disp('The sorted array is the answer below');
```

```
disp(array)
```

The sorted array is the answer below

COMMAND WINDOW OUTPUT

```
% array = [100 -72 14 30 27];
% PS10_sort_koike(array)
% The sorted array is the answer below
%
% ans =
%
%      -72      14      27      30      100

% array = [2 0.5 -5 3 6; -5 4 -3 4 6; 8 2.5 1 -2 -1];
% PS10_sort_koike(array)
% The sorted array is the answer below
%
% ans =
%
%      -5.0000      -5.0000      -3.0000      -2.0000      -1.0000
%       0.5000       1.0000       2.0000       2.5000       3.0000
%       4.0000       4.0000       6.0000       6.0000       8.0000
```

ACADEMIC INTEGRITY STATEMENT

```
PS07_academic_integrity_koike("Tomoki Koike")
```

I am submitting code that is my own original work. I have not used source code, either modified or unmodified, obtained from any unauthorized source. Neither have I provided access to my code to any peer or unauthorized source. Signed,
<Tomoki Koike>

Published with MATLAB® R2018a