
Table of Contents

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

```
function [sortedArray] = PS09_sort_rbehl(userArray)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% The program takes in a user inputed array and sorts it so that the
% smallest number is in the upper left corner of the array, the
% next
% smallest value moves to the first row second column, and so on until
% the largest value
% is in the lower right corner of the array.
% Function Call
%[sortedArray] = PS09_sort_rbehl(userArray)
%
% Input Arguments
% userArray = The input array
%
% Output Arguments
% sortedArray = The sorted array
%
% Assignment Information
% Assignment:      PS 09, Problem 3
% Author:         Ranjan Behl, rbehl@purdue.edu
% Team ID:        008-14
% 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

```
dim = size(userArray); % finding the dimensions of the input array
tempArray = zeros(dim); % creating a matrix of zeros that has the same
dimension as the input array
```

CALCULATIONS

nested for loops that run the sorting algo

```
for rowindex = 1:1:dim(1)
    for colindex = 1:1:dim(2)
        minval = min(userArray(:));
        minValPos = find(userArray == minval);
        num_minval = numel(minValPos);
        tempArray(rowindex,colindex) = minval;
        if (num_minval > 1)
            vector = zeros(1,num_minval - 1);
            vector(:, :) = minval;
            vector = transpose(vector);
            largeVector = userArray(userArray > minval);
            userArray = vertcat(vector,largeVector);
        else
            largerVector = userArray(userArray > minval);
            userArray = largerVector;
        end
    end
end
sortedArray = tempArray;
```

COMMAND WINDOW OUTPUT

Test case 1

```
%{
[sortedArray] = PS09_sort_rbehl([100,-72,14,30,27])
```

```
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,
    <Ranjan Behl>
sortedArray =
```

```

    -72    14    27    30    100
%}

% Test case 2
%{
[sortedArray] =
    PS09_sort_rbehl([2,0.5,-5,3,6;-5,4,-3,4,6;8,2.5,1,-2,-1])

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,
    <Ranjan Behl>
sortedArray =
    -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_integrity_rbehl("Ranjan Behl");
```

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,

<Ranjan Behl>

```
sortedArray =
    -72    14    27    30    100
```

Published with MATLAB® R2018b