

Assignment #5

Submission deadline: 5th April, 2020, Sunday till 11:59pm

Submission guidelines:

- Put all your .cpp files in a folder. Zip that folder and name it with your roll number.
- Email the zipped folder at natalia@pucit.edu.pk. You must put the **subject** of your email as: **"A#5"**
- Students who are facing issues with their laptop or compiler can write the code on paper and send the screenshots, following the above guideline. Since your issue will be genuine therefore relaxation will be given accordingly. Don't worry!

Task

10 marks

Consider a 2D matrix. You need to construct a query rectangle. To construct *query* rectangle, you can ask user about *seed point* and *dimension*. You can construct rectangular boundary (query rectangle) using this. Seed point tells the index from where to start constructing query rectangle (i.e. the top left index). Dimension tells the number of row and columns the query rectangle consists of. Apply proper validation to check if the query rectangle lies within the main 2D matrix or index goes out of bound. For example consider a valid query rectangle for a 6x6 square 2D matrix. In the following example, if seed point is (2,3) and dimension is 3 then the query rectangle will look like:

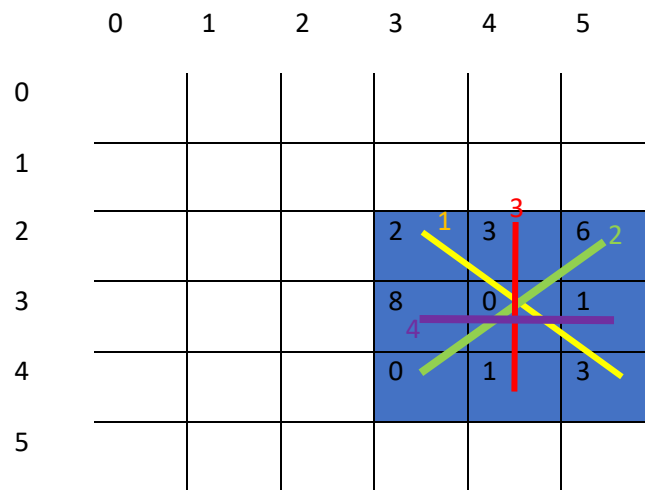
	0	1	2	3	4	5
0						
1						
2						
3						
4						
5						

Print the original matrix and query rectangle.

Bonus part:

Initialize query rectangle. After initializing the query rectangle, find 4 directional sums using query rectangle:

- Directional_1 sum (depicted in yellow) -> 2+0+3
- Directional_2 sum (depicted in green) -> 6+0+0
- Directional_3 sum (depicted in red)....
- Directional_4 sum (depicted in purple)...



Then find and display the sorted directional sums (i.e. first smallest, second smallest, third smallest, and largest (forth smallest)).