CSC 2720: Data Structures
Lab 3

Instructor: Shiraj Pokharel

Due : 48 hours after after official Lab session
Late Submission deadline (with 25% penalty) 24 hours after due date

Answer the below questions.
You may use whatever IDEs / editors you like, but you must submit your responses on iCollege as .java files.
Failure to comply with this simple requirement will result in a score of Zero. Please, be careful not to be assigned a Zero score this way.

Few Rules to be followed, else will receive a score of ZERO

(1) Your submissions will work exactly as required.

(2) Your files shall not be incomplete or worse corrupted such that the file does not compile at all. Make sure you submit a file that compiles.

(3) Your submission will show an output. Should you recive a Zero for no output shown do not bother to email me with "but the logic is perfect" !

Note that your program's output must **exactly** match the specs(design , style) given here for each problem to pass the instructor's test cases .

Design refers to how well your code is written (i.e. is it clear, efficient, and elegant).
Style refers to the readability of your code (commented, correct indentation, good variable names).

Image rotation is a fundamental image-processing operation required to be performed in the areas of Computer Graphics , Computer Vision(Artificial Intelligence/Machine Learning) and Visual Analytics (Data Science).

Here we will explore on ways to do a clock-wise rotation of an example bit-map representation of an image *aka* A Matrix. Below is the matrix :

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Our objective is to write (a) function(s) to rotate the above matrix by 90-degrees clockwise which produces the matrix as below:

$$\begin{bmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

We will solve the problem in two ways:-

(1) [50 points] Implement the function by allocating a new n X n 2D array. Then write the rotation to it by writing the rows of the original matrix to the columns in the solution matrix such that they fit the solution requirement. Then copy the new matrix exactly the same to the original matrix so that you know for sure you have updated the original matrix to look modified.

(2) [50 points] Implement the function without allocating a new n X n 2D array.
Hint :
first exchange elements in top row to bottom row, then swap the symmetry elements at [i][j] with [j][i] where i represents rows and j represents columns.