**2D Array**

**Q1**- Write the C++ program to take input in the SQUARE integer matrix (2D Array).

* Display the elements of left diagonal
* Display the elements of right diagonal
* Display the sum of all elements of mid row and mid column.

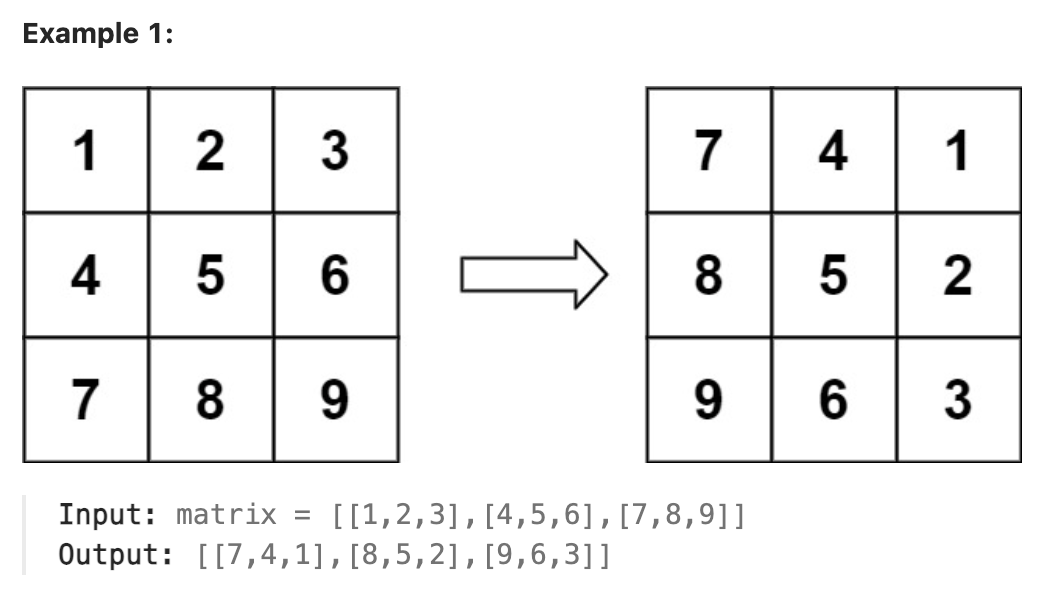
**Q2**- Write the C++ program to take input in the SQUARE integer matrix (2D Array) and display the transpose of the matrix **(without using a temporary array).**

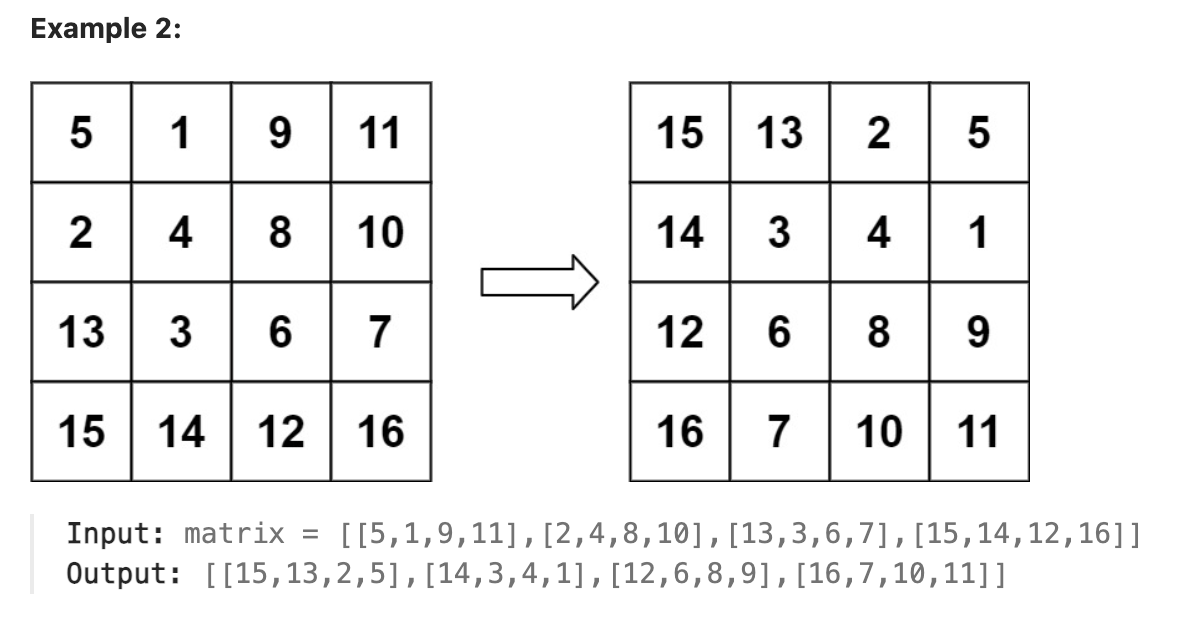
**Q3,4**- Write the C++ program to initialize input two SQUARE integer matrices (2D Arrays).

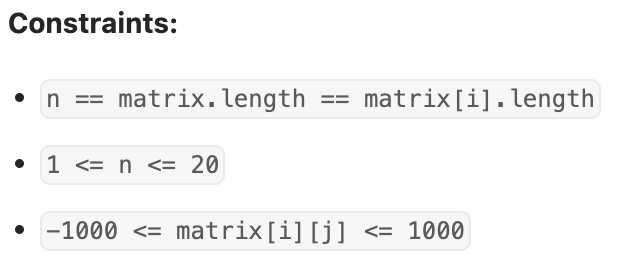
* Sum of two matrices
* Check if two matrices are equal or not

**Q5-** You are given an n x n 2D matrix representing an image, rotate the image by 90 degrees (clockwise).

**You have to rotate the image in place, which means you have to modify the input 2D matrix directly. DO NOT allocate another 2D matrix and do the rotation.**

****

****

****

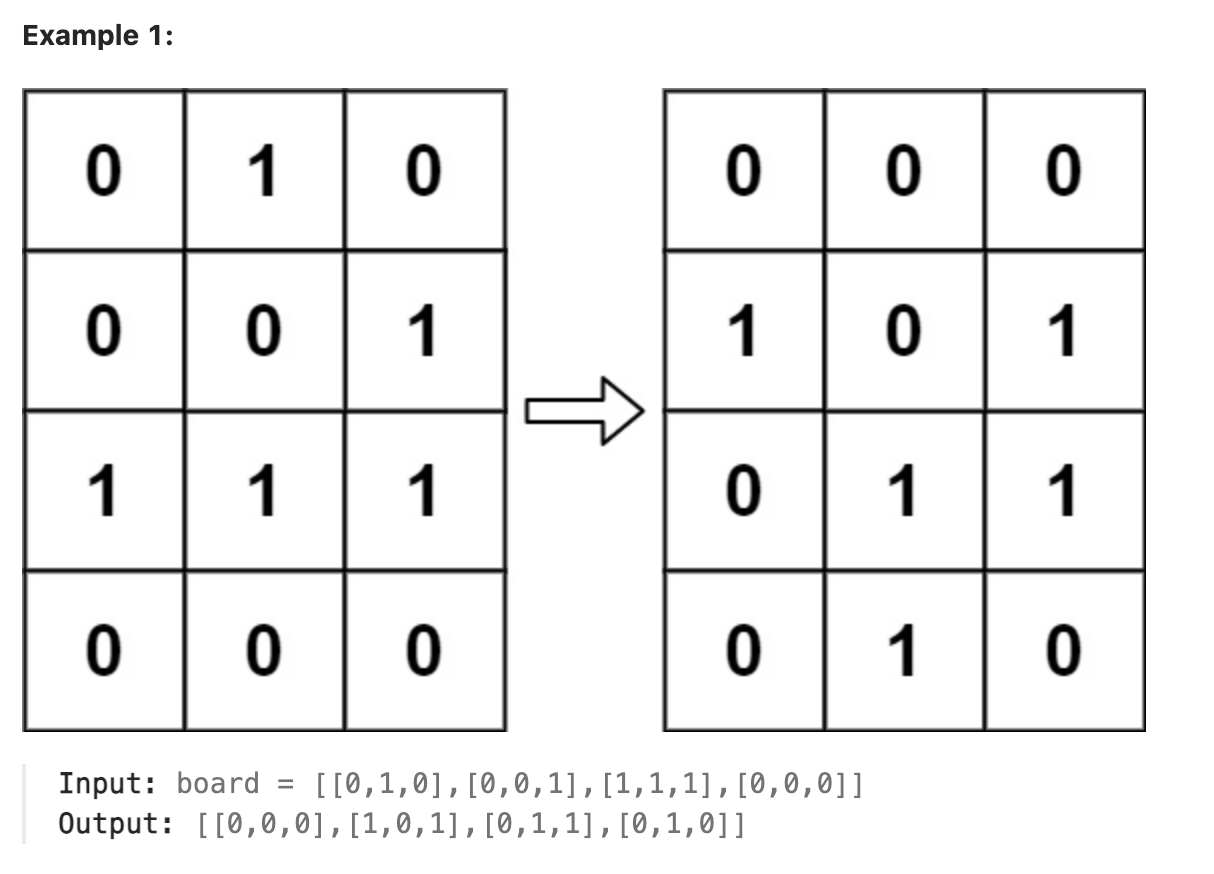
**Q6**- According to [Wikipedia's article](https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life): "The Game of Life, also known simply as Life, is a cellular automaton devised by the British mathematician John Horton Conway in 1970."

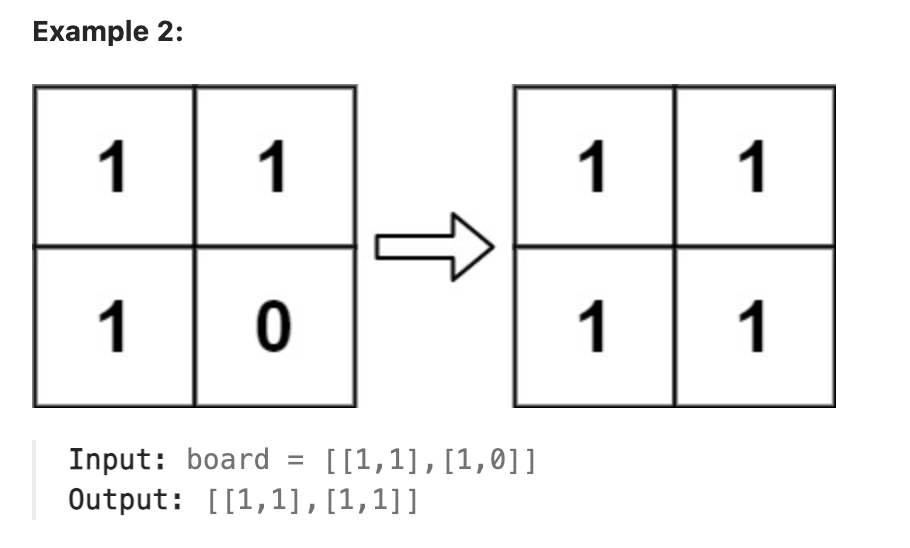
The board is made up of an m x n grid of cells, where each cell has an initial state: live (represented by a 1) or dead (represented by a 0). Each cell interacts with its [eight neighbors](https://en.wikipedia.org/wiki/Moore_neighborhood) (horizontal, vertical, diagonal) using the following four rules (taken from the above Wikipedia article):

1. Any live cell with fewer than two live neighbors dies as if caused by under-population.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

The next state is created by applying the above rules simultaneously to every cell in the current state, where births and deaths occur simultaneously. Given the current state of the m x n grid board, return *the next state*.

**Remember that the board needs to be updated simultaneously: You cannot update some cells first and then use their updated values to update other cells.**

****

****