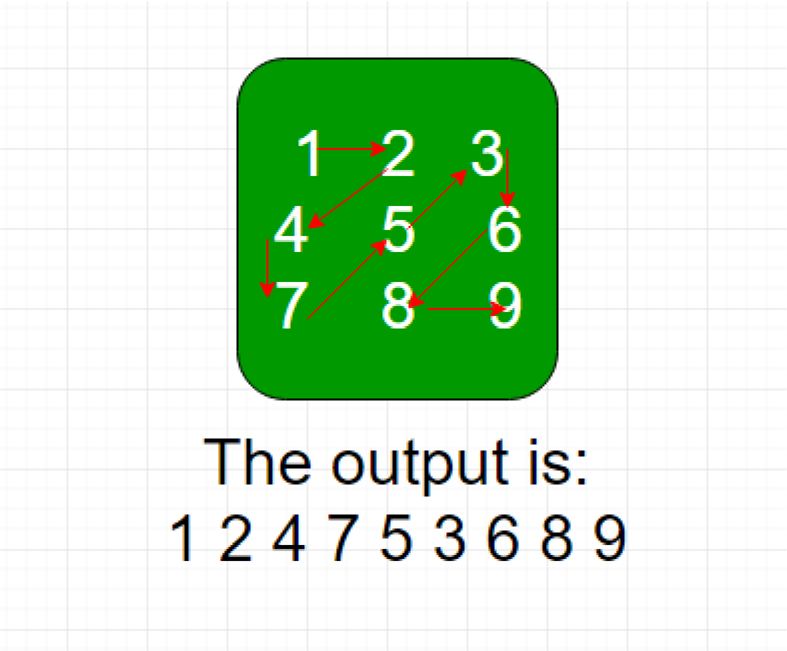
Print matrix in diagonal pattern

Given a square matrix **mat[][]** of **n*n** size, the task is to determine the **diagonal pattern** which is a linear arrangement of the elements of the matrix as depicted in the following example:



Example 1:

```
Input:
n = 3
mat[][] = {{1, 2, 3},
           {4, 5, 6},
           {7, 8, 9}}
Output: {1, 2, 4, 7, 5, 3, 6, 8, 9}
Explaination:
Starting from (0, 0): 1,
Move to the right to (0, 1): 2,
Move diagonally down to (1, 0): 4,
Move diagonally down to (2, 0): 7,
Move diagonally up to (1, 1): 5,
Move diagonally up to (0, 2): 3,
Move to the right to (1, 2): 6,
Move diagonally up to (2, 1): 8,
Move diagonally up to (2, 2): 9
There for the output is {1, 2, 4, 7, 5, 3, 6, 8, 9}.
```

Example 2:

```
Input:
n = 2
mat[][] = {{1, 2},
           {3, 4}}
Output: {1, 2, 3, 4}
Explaination:
Starting from (0, 0): 1,
Move to the right to (0, 1): 2,
Move diagonally down to (1, 0): 3,
Move to the right to (1, 1): 4
There for the output is {1, 2, 3, 4}.
```

Your Task:

You only need to implement the given function **matrixDiagonally()** which takes a matrix **mat[][]** of size **n*n** as an input and returns a list of integers containing the matrix diagonally. Do not read input, instead use the arguments given in the function.

Expected Time Complexity: O(n*n).
Expected Auxiliary Space: O(1).
Constraints:
1 <= n <= 100
-100 <= elements of matrix <= 100