Problem Statement **Suggest Edit**

You are given a square matrix of non-negative integers of size 'N x N'. Your task is to rotate that array by 90 degrees in an anti-clockwise direction without using any extra space.

For example:

```
For given 2D array :

    [    [ 1,  2,  3 ],
         [ 4,  5,  6 ],
         [ 7,  8,  9 ]  ]

After 90 degree rotation in anti clockwise direction, it will become:

    [   [ 3,  6,  9 ],
        [ 2,  5,  8 ],
        [ 1,  4,  7 ]   ]
```

Input Format :

```
The first line of input contains an integer 'T' representing the number of the
test case. Then the test case follows.

The first line of each test case contains an integer 'N' representing the size of
the square matrix ARR.

Each of the next 'N' lines contains 'N' space-separated integers representing the
elements of the matrix 'ARR'.
```

Output Format:

```
For each test case, return the rotated matrix.
```

Note:

```
You do not need to print anything; it has already been taken care of. Just
implement the given function.
```

Constraints:

```
1 ≤ T ≤ 50
1 ≤ N ≤ 100
1 ≤ ARR[i][j] ≤ 10^9
```

Time Limit: 1 sec

Sample Input 1:

```
2
3
1  2  3
4  5  6
7  8  9
4
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
```

Sample Output 1:

```
3  6  9
2  5  8
1  4  7
4  8 12 16
3  7 11 15
2  6 10 14
1  5  9 13
```

Explanation Of Input 1:

(i) The array has been rotated by 90 degrees in an anticlockwise direction as the first row is now the first column inverted and so on for second and third rows.

(ii) The array has been rotated by 90 degrees in an anticlockwise direction as the first row is now first column inverted and so on for second, third and fourth rows.

Sample Input 2:

```
2
3
7  4  1
8  5  2
9  6  3
4
13  9  5  1
14  10  6  2
```

```
15 11 7 3
16 12 8 4
```

Sample Output 2:

```
1  2  3
4  5  6
7  8  9
1  2  3  4
5  6  7  8
9 10 11 12
13 14 15 16
```