A - Rotation of Matrix

Given a 2D square matrix, rotate the matrix by 90 degrees in clockwise manner.  
Note: Try to solve it by first scanning the matrix, then do an in-place rotation and then print the rotated matrix.

**Input Format**

First line of input contains T - number of test cases. First line of each test case contains N - size of the matrix [NxN]. Its followed by N lines each containing N integers - elements of the matrix.

**Constraints**

1 <= T <= 100  
1 <= N <= 100  
-100 <= ar[i][j] <= 100

**Output Format**

For each test case, print the rotated matrix, separated by newline.

**Sample Input 0**

4

1

1

2

1 2

4 3

3

1 2 3

8 9 4

7 6 5

5

-44 25 -52 69 -5

17 22 51 27 -44

-79 28 -78 1 -47

65 -77 -14 -21 -6

-96 43 -21 -20 90

**Sample Output 0**

Test Case #1:

1

Test Case #2:

4 1

3 2

Test Case #3:

7 8 1

6 9 2

5 4 3

Test Case #4:

-96 65 -79 17 -44

43 -77 28 22 25

-21 -14 -78 51 -52

-20 -21 1 27 69

90 -6 -47 -44 -5

#include <iostream>

#include <vector>

using namespace *std*;

void display2DMatrix(*vector*<*vector*<int>>& matrix)

{

for (auto i : matrix)

{

for (auto j : i)

{

*cout* << j << " ";

}

*cout* << *endl*;

}

//cout << endl;

}

void transpose(*vector*<*vector*<int>>& matrix)

{

for (*size\_t* i = 0; i < matrix.*size*(); i++)

{

// SWAP elements of upper triangle with lower triangle

for (*size\_t* j = i; j < matrix[0].*size*(); j++)

{

auto temp = matrix[i][j];

matrix[i][j] = matrix[j][i];

matrix[j][i] = temp;

}

}

}

void reverse(*vector*<*vector*<int>>& matrix)

{

for (*size\_t* i = 0; i < matrix.*size*(); i++)

{

int li = 0; int ri = matrix[i].*size*() - 1;

while (li < ri)

{

auto temp = matrix[i][li];

matrix[i][li] = matrix[i][ri];

matrix[i][ri] = temp;

li++;

ri--;

}

}

}

void fill2DMatrix(*vector*<*vector*<int>>& matrix)

{

int ele;

int rows = matrix.*size*(), cols = matrix[0].*size*();

for (auto i = 0; i < rows; i++)

{

for (auto j = 0; j < cols; j++)

{

*cin* >> ele;

matrix[i][j] = ele;

}

}

}

int main(void)

{

int t; *cin* >> t;

for (auto j = 1; j <= t; j++)

{

int n; *cin* >> n;

*vector*<*vector*<int>> matrix(n, *vector*<int>(n));

fill2DMatrix(matrix);

transpose(matrix);

reverse(matrix);

*cout* << "Test Case #" << j << ":\n";

display2DMatrix(matrix);

}

return 0;

}