2.Medium\12.Rotate_matrix.cpp

```cpp
/*
QUESTION:-

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

Example 1:
Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]
Output: [[7,4,1],[8,5,2],[9,6,3]]

Example 2:
Input: matrix = [[5,1,9,11],[2,4,8,10],[13,3,6,7],[15,14,12,16]]
Output: [[15,13,2,5],[14,3,4,1],[12,6,8,9],[16,7,10,11]]

*/

/*
APPROACH:-

To rotate the image by 90 degrees clockwise in-place, we can follow these steps:

1. Transpose the matrix: Iterate over the matrix and swap each element (i, j) with its corresponding element (j, i). This step transforms rows into columns.

2. Reverse each row: Iterate over each row in the transposed matrix and reverse the elements. This step ensures the rotation in a clockwise direction.

*/

// CODE:

void rotate(vector<vector<int>>& matrix) {
    // Transpose the matrix
    int n = matrix.size();
    int m = matrix[0].size();
    for(int i=0; i<n; i++){
        // note here we move
        for(int j=0; j<i; j++){
            swap(matrix[i][j],matrix[j][i]);
        }
    }

    // Reverse each row
    for(int i=0; i<n; i++){
        reverse(matrix[i].begin(),matrix[i].end());
    }
}

// TIME COMPLEXITY = O(N^2), where N is the size of the matrix.
// SPACE COMPLEXITY = O(1)

```