Problem Statement-Wave Form Traversal

```
#include <iostream>

#include <vector>


void waveTraversal(int n, int m, const std::vector<std::vector<int>>& matrix) {

    for (int j = 0; j < m; ++j) {

        if (j % 2 == 0) {

            // Even columns: top to bottom

            for (int i = 0; i < n; ++i) {

                std::cout << matrix[i][j] << " ";

            }

        } else {

            // Odd columns: bottom to top

            for (int i = n - 1; i >= 0; --i) {

                std::cout << matrix[i][j] << " ";

            }

        }

    }

}


int main() {

    int n, m;

    std::cout << "Enter the number of rows (N) and columns (M): ";

    std::cin >> n >> m;


    std::vector<std::vector<int>> matrix(n, std::vector<int>(m));

    std::cout << "Enter the elements of the matrix:" << std::endl;

    for (int i = 0; i < n; ++i) {

        for (int j = 0; j < m; ++j) {
            std::cin >> matrix[i][j];
        }
    }

    std::cout << "Wave traversal of the matrix:" << std::endl;
    waveTraversal(n, m, matrix);
    std::cout << std::endl;

    return 0;
}
```

Problem Statement-Transpose of a matrix

```
#include <iostream>
#include <vector>

void transposeMatrix(int n, int m, const std::vector<std::vector<int>>& originalMatrix) {
    // The transposed matrix will have dimensions m x n
    std::vector<std::vector<int>> transposedMatrix(m, std::vector<int>(n));

    // Fill the transposed matrix
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            transposedMatrix[j][i] = originalMatrix[i][j];
        }
    }

    // Print the transposed matrix
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cout << transposedMatrix[i][j] << " ";
        }
        std::cout << std::endl;
    }
}

int main() {
    int n, m;
    std::cout << "Enter the number of rows (N) and columns (M): ";
    std::cin >> n >> m;

    std::vector<std::vector<int>> matrix(n, std::vector<int>(m));
    std::cout << "Enter the elements of the matrix:" << std::endl;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            std::cin >> matrix[i][j];
        }
    }

    std::cout << "Transposed matrix:" << std::endl;
    transposeMatrix(n, m, matrix);

    return 0;
}
```

Problem Statement-Spiral Traversal of a Matrix

```
#include <iostream>
#include <vector>

void spiralTraversal(int n, int m, const std::vector<std::vector<int>>& matrix) {
    int top = 0, bottom = n - 1;
int left = 0, right = m - 1;

while (top <= bottom && left <= right) {
    // Traverse top row from left to right
    for (int i = left; i <= right; ++i) {
        std::cout << matrix[top][i] << " ";
    }
    top++;

    // Traverse rightmost column from top to bottom
    for (int i = top; i <= bottom; ++i) {
        std::cout << matrix[i][right] << " ";
    }
    right--;

    // Traverse bottom row from right to left (if valid)
    if (top <= bottom) {
        for (int i = right; i >= left; --i) {
            std::cout << matrix[bottom][i] << " ";
        }
        bottom--;
    }

    // Traverse leftmost column from bottom to top (if valid)
    if (left <= right) {
        for (int i = bottom; i >= top; --i) {
            std::cout << matrix[i][left] << " ";
        }
        left++;
    }
}
```

```
}

int main() {
    int n, m;
    std::cout << "Enter the number of rows (N) and columns (M): ";
    std::cin >> n >> m;

    std::vector<std::vector<int>> matrix(n, std::vector<int>(m));
    std::cout << "Enter the elements of the matrix:" << std::endl;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            std::cin >> matrix[i][j];
        }
    }

    std::cout << "Spiral traversal of the matrix:" << std::endl;
    spiralTraversal(n, m, matrix);
    std::cout << std::endl;

    return 0;
}
```

Problem Statement-Rotate Matrix by 90 degree Clockwise

```
#include <iostream>
#include <vector>
#include <algorithm>

void rotateMatrix(std::vector<std::vector<int>>& matrix) {
    int n = matrix.size();
    // Step 1: Transpose the matrix
    for (int i = 0; i < n; ++i) {
        for (int j = i + 1; j < n; ++j) {
            std::swap(matrix[i][j], matrix[j][i]);
        }
    }

    // Step 2: Reverse each row
    for (int i = 0; i < n; ++i) {
        std::reverse(matrix[i].begin(), matrix[i].end());
    }
}

void printMatrix(const std::vector<std::vector<int>>& matrix) {
    for (const auto& row : matrix) {
        for (int val : row) {
            std::cout << val << " ";
        }
        std::cout << std::endl;
    }
}

int main() {
    int n;
    std::cout << "Enter the size of the square matrix (N): ";
    std::cin >> n;

    std::vector<std::vector<int>> matrix(n, std::vector<int>(n));
    std::cout << "Enter the elements of the matrix:" << std::endl;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            std::cin >> matrix[i][j];
        }
    }

    std::cout << "Original matrix:" << std::endl;
    printMatrix(matrix);

    rotateMatrix(matrix);

    std::cout << "\nRotated matrix:" << std::endl;
    printMatrix(matrix);

    return 0;
}
```