|  |  |
| --- | --- |
| **Rotate Image in C++** | |
| #include <iostream>  #include <vector>  using namespace std;  void rotate(vector<vector<int>>& matrix) {  int n = matrix.size();  int m = matrix[0].size();  // Transpose the matrix  for (int i = 0; i < n; i++) {  for (int j = i; j < m; j++) {  swap(matrix[i][j], matrix[j][i]);  }  }  // Reverse each row  for (int i = 0; i < n; i++) {  int sp = 0;  int ep = m - 1;  while (sp < ep) {  swap(matrix[i][sp], matrix[i][ep]);  sp++;  ep--;  }  }  }  void print2DArray(const vector<vector<int>>& array) {  for (size\_t i = 0; i < array.size(); i++) {  for (size\_t j = 0; j < array[i].size(); j++) {  cout << array[i][j] << " ";  }  cout << endl;  }  }  int main() {  vector<vector<int>> matrix = {  {1, 2, 3},  {4, 5, 6},  {7, 8, 9}  };  cout << "Original matrix:" << endl;  print2DArray(matrix);  rotate(matrix);  cout << "Rotated matrix:" << endl;  print2DArray(matrix);  return 0;  } | Input Matrix: Original matrix:  1 2 3  4 5 6  7 8 9 🔄 Step 1: ****Transpose the matrix**** Transposing means swapping matrix[i][j] with matrix[j][i] for j > i.   | **i** | **j** | **matrix[i][j]** | **matrix[j][i]** | **Action** | | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 4 | Swap → 2 ↔ 4 | | 0 | 2 | 3 | 7 | Swap → 3 ↔ 7 | | 1 | 2 | 6 | 8 | Swap → 6 ↔ 8 |   🔁 After transpose:  1 4 7  2 5 8  3 6 9 🔄 Step 2: ****Reverse each row**** Reverse each row of the transposed matrix:   | **Row Before** | **Row After** | | --- | --- | | 1 4 7 | 7 4 1 | | 2 5 8 | 8 5 2 | | 3 6 9 | 9 6 3 |  ✅ Final Output: Rotated matrix:  7 4 1  8 5 2  9 6 3 |
| Original matrix:  1 2 3  4 5 6  7 8 9  Rotated matrix:  7 4 1  8 5 2  9 6 3 | |