// 304-Range\_Sum\_Query\_2D-Immutable.cpp //

#include <iostream>

#include <vector>

using namespace std;

class NumMatrix {

private:

int\*\* arr;

int v = 0;

public:

NumMatrix(vector<vector<int>>& matrix) {

if (matrix.size() == 0)

return;

v = matrix[0].size();

arr = new int\* [matrix.size()];

for (int i = 0; i < matrix.size(); i++)

{

arr[i] = new int[matrix[i].size()];

}

for (int i = 0; i < matrix.size(); i++)

{

for (int j = 0; j < matrix[i].size(); j++)

{

if (i == 0 && j == 0)

{

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

}

else if (j == 0)

{

arr[i][j] = arr[i - 1][matrix[i - 1].size() - 1] + matrix[i][j];

}

else

{

arr[i][j] = arr[i][j - 1] + matrix[i][j];

}

}

}

}

int sumRegion(int row1, int col1, int row2, int col2) {

//cout << arr[row1][col1] << " " << arr[row2][col2] << endl;

int sum = 0;

for (int i = row1; i <= row2; i++)

{

if (i == 0 && col1 == 0)

{

sum = sum + arr[i][col2];

}

else if (col1 == 0)

{

sum = sum + arr[i][col2] - arr[i - 1][v - 1];

}

else

{

sum = sum + arr[i][col2] - arr[i][col1 - 1];

}

}

return sum;

}

};

int main()

{

/\*\*

\* Your NumMatrix object will be instantiated and called as such:

\* NumMatrix\* obj = new NumMatrix(matrix);

\* int param\_1 = obj->sumRegion(row1,col1,row2,col2);

\*/

vector<vector<int>> matrix = {

//int matrix[5][5] = {

{3, 0, 1, 4, 2},

{5, 6, 3, 2, 1},

{1, 2, 0, 1, 5},

{4, 1, 0, 1, 7},

{1, 0, 3, 0, 5}

};

NumMatrix\* obj = new NumMatrix(matrix);

cout << obj->sumRegion(2, 1, 4, 3) << endl; // 8

cout << obj->sumRegion(1, 1, 2, 2) << endl; // 11

cout << obj->sumRegion(1, 2, 2, 4) << endl; // 12

}