#include <iostream>

#include <vector>

using namespace std;

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

int N = matrix.size();

if (N == 0) return 0;

vector<vector<int>> dp(N, vector<int>(N, 0));

dp[0][0] = matrix[0][0];

for (int i = 1; i < N; ++i) {

dp[i][0] = dp[i-1][0] + matrix[i][0];

dp[0][i] = dp[0][i-1] + matrix[0][i];

}

for (int i = 1; i < N; ++i) {

for (int j = 1; j < N; ++j) {

dp[i][j] = min(dp[i-1][j], dp[i][j-1]) + matrix[i][j];

}

}

return dp[N-1][N-1];

}

int main() {

vector<vector<int>> matrix = {

{1, 3, 1},

{1, 5, 1},

{4, 2, 1}

};

cout << shortestPathSum(matrix) << endl;

return 0;

}