

```
#include <iostream>
using namespace std;

int min_cost2(int** input, int m, int n) {
    int ** dp = new int*[m];
    for (int i = 0; i < m; i++) {
        dp[i] = new int[n];
    }
    dp[m - 1][n-1] = input[m-1][n-1];
    for (int i = m - 2; i >= 0; i--) {
        dp[i][n - 1] = dp[i + 1][n-1] + input[i][n-1];
    }

    for (int j = n - 2; j >=0; j--) {
        dp[m - 1][j] = dp[m - 1][j + 1] + input[m-1][j];
    }

    for (int i = m - 2; i >=0; i--) {
        for (int j = n - 2; j >=0 ; j--) {
            dp[i][j] = input[i][j] + min(dp[i+1][j], min(dp[i+1][j+1], dp[i][j + 1]));
        }
    }
    return dp[0][0];
}

int min_cost(int** input, int si, int sj, int ei, int ej) {
    if (si == ei && sj == ej) {
        return input[ei][ej];
    }
    if (si > ei || sj > ej) {
        return INT_MAX;
    }
    int option1 = min_cost(input, si + 1, sj, ei, ej);
    int option2 = min_cost(input, si + 1, sj + 1, ei, ej);
    int option3 = min_cost(input, si, sj + 1, ei, ej);
    return input[si][sj] + min(option1, min(option2, option3));
}

int main() {
    int ** input = new int*[3];
    input[0] = new int[3];
    input[1] = new int[3];
    input[2] = new int[3];
    input[0][0] = 4;
    input[0][1] = 3;
    input[0][2] = 2;
    input[1][0] = 1;
    input[1][1] = 8;
```

```
input[1][2] = 3;
input[2][0] = 1;
input[2][1] = 1;
input[2][2] = 8;

cout << min_cost(input, 0,0,2,2) << endl;
cout << min_cost2(input,3,3) << endl;
delete [] input[0];
delete [] input[1];
delete [] input[2];
delete [] input;
}
```