



# CodeCheck Report: training7J2YWR-FS7

[Check out Codility training tasks](#)

Test Name:

Summary

Timeline

Mail Status: Not Applicable

- Finished: 2021-10-22 01:10 UTC
- Started: 2021-10-22 00:36 UTC
- Invitation Created: 2021-10-22 00:36 UTC

## Tasks Details

Easy

### 1. StoneWall

Cover "Manhattan skyline" using the minimum number of rectangles.

Task Score	Correctness	Performance
100%	100%	100%

### Task description

You are going to build a stone wall. The wall should be straight and N meters long, and its thickness should be constant; however, it should have different heights in different places. The height of the wall is specified by an array H of N positive integers. H[l] is the height of the wall from l to l+1 meters to the right of its left end. In particular, H[0] is the height of the wall's left end and H[N-1] is the height of the wall's right end.

The wall should be built of cuboid stone blocks (that is, all sides of such blocks are rectangular). Your task is to compute the minimum number of blocks needed to build the wall.

Write a function:

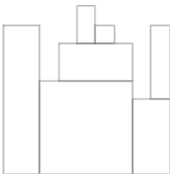
```
int solution(vector<int> &H);
```

that, given an array H of N positive integers specifying the height of the wall, returns the minimum number of blocks needed to build it.

For example, given array H containing N = 9 integers:

H[0] = 8	H[1] = 8	H[2] = 5
H[3] = 7	H[4] = 9	H[5] = 8
H[6] = 7	H[7] = 4	H[8] = 8

the function should return 7. The figure shows one possible arrangement of seven blocks.



### Solution

Programming language used: C++

Total time used:	35 minutes	?
Effective time used:	35 minutes	?

Notes: not defined yet

### Task timeline



Code: 01:10:56 UTC, cpp, [show code in pop-up](#)  
final, score: 100

```
1 // you can use includes, for example:
2 // #include <algorithm>
3
4 // you can write to stdout for debugging purposes
5 // cout << "this is a debug message" << endl;
6 #include <stack>
7
8 int solution(vector<int> &H) {
9     int ladrillo = 0;
10    stack<int> muro;
11
12    for (auto i:H) {
13        while (!muro.empty() && i < muro.top())
```

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [1..100,000];
- each element of array H is an integer within the range [1..1,000,000,000].

Copyright 2009–2021 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Test results - Codility

```
14         muro.pop();
15     }
16     if (muro.empty() || i > muro.top()) {
17         muro.push(i);
18         ++ladrillo;
19     }
20 }
21 return ladrillo;
22 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: <b>O(N)</b>		
expand all	Example tests	
▶ example	✓ OK	
expand all	Correctness tests	
▶ simple1	✓ OK	
▶ simple2	✓ OK	
▶ simple3	✓ OK	
▶ simple4	✓ OK	
▶ boundary_cases	✓ OK	
expand all	Performance tests	
▶ medium1	✓ OK	
▶ medium2	✓ OK	
▶ medium3	✓ OK	
▶ medium4	✓ OK	
▶ large_piramid	✓ OK	
▶ large_increasing_decreasing	✓ OK	
▶ large_up_to_20	✓ OK	
▶ large_up_to_100	✓ OK	
▶ large_max	✓ OK	