



CodeCheck Report: training2HCW3A-M4V

[Check out Codility training tasks](#)

Test Name:

Summary    Timeline    AI Assistant Transcript

Tasks summary

Task	Effective time spent	Score
CountNonDivisible C++	12 min	55%

Total score

55%

Tasks Details

Medium

1. **CountNonDivisible**  
Calculate the number of elements of an array that are not divisors of each element.

Task Score

55%

Correctness

100%

Performance

0%

Task description

You are given an array A consisting of N integers.

For each number A[i] such that  $0 \leq i < N$ , we want to count the number of elements of the array that are not the divisors of A[i]. We say that these elements are non-divisors.

For example, consider integer N = 5 and array A such that:

A[0] = 3  
A[1] = 1  
A[2] = 2  
A[3] = 3  
A[4] = 6

For the following elements:

- A[0] = 3, the non-divisors are: 2, 6,
- A[1] = 1, the non-divisors are: 3, 2, 3, 6,
- A[2] = 2, the non-divisors are: 3, 3, 6,
- A[3] = 3, the non-divisors are: 2, 6,
- A[4] = 6, there aren't any non-divisors.

Write a function:

vector<int> solution(vector<int> &A);

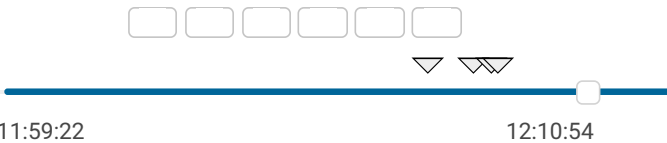
Solution

Programming language used: C++

Time spent on task: 12 minutes

Notes: not defined yet

Task timeline



Code: 12:10:54 UTC, cpp, [show code in pop-up](#)  
final, score: 55

```
1 // you can use includes, for example:
2 // #include <algorithm>
3 #include <math.h>
4 // you can write to stdout for debugging purposes
5 // cout << "this is a debug message" << endl;
```

that, given an array A consisting of N integers, returns a sequence of integers representing the amount of non-divisors.

Result array should be returned as a vector of integers.

For example, given:

A[0] = 3  
A[1] = 1  
A[2] = 2  
A[3] = 3  
A[4] = 6

the function should return [2, 4, 3, 2, 0], as explained above.

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

- N is an integer within the range [1..50,000];
- each element of array A is an integer within the range [1..2 \* N].

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
6
7 bool isDivisor (int target, int divide) {
8     return target%divide != 0;
9 }
10 vector<int> solution(vector<int> &A) {
11     //전체 배열에서 약수가 아닌 수의 개수를 배열로 반환
12     vector<int> answer;
13
14     for (int e : A) {
15         int count = 0;
16         for (int a : A) {
17             if(isDivisor(e, a)) {
18                 count++;
19             }
20         }
21         answer.push_back(count);
22     }
23
24     return answer;
25 }
```

Analysis summary

The following issues have been detected: timeout errors.

Analysis

Detected time complexity: **O(N \*\* 2)**

Example tests	
▶ example	✓ OK
example test	
Correctness tests	
▶ extreme_simple	✓ OK
extreme simple	
▶ double	✓ OK
two elements	
▶ simple	✓ OK
simple tests	
▶ primes	✓ OK
prime numbers	
▶ small_random	✓ OK
small, random numbers, length = 100	
Performance tests	
▶ medium_random	✗ TIMEOUT ERROR
medium, random numbers length = 5,000	
running time: 0.108 sec., time limit: 0.100 sec.	
▶ large_range	✗ TIMEOUT ERROR
1, 2, ..., N, length = ~20,000	
running time: 1.740 sec., time limit: 0.100 sec.	
▶ large_random	✗ TIMEOUT ERROR
large, random numbers, length = ~30,000	
running time: 3.904 sec., time limit: 0.100 sec.	
▶ large_extreme	✗ TIMEOUT ERROR
large, all the same values, length = 50,000	
Killed. Hard limit reached: 6.000 sec.	