Codility_

CodeCheck Report: trainingWBQXDY-NA4

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
CountDiv C++	11 min	100%

Total score

Tasks Details

Medium

1. CountDiv

Compute number of integers divisible by k in range [a..b].

Task Score	Correctness	Performance
100%	100%	100%

Task description

Write a function:

```
int solution(int A, int B, int K);
```

that, given three integers A, B and K, returns the number of integers within the range [A..B] that are divisible by K, i.e.:

{ i : A ≤ i ≤ B, i **mod** K = 0 }

For example, for A = 6, B = 11 and K = 2, your function should return 3, because there are three numbers divisible by 2 within the range [6..11], namely 6, 8 and 10.

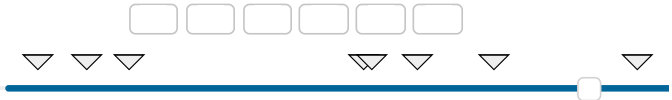
Write an **efficient** algorithm for the following assumptions:

- A and B are integers within the range [0..2,000,000,000];
- K is an integer within the range [1..2,000,000,000];
- A ≤ B.

Solution

Programming language used:	C++
Total time used:	11 minutes
Effective time used:	11 minutes
Notes:	*not defined yet*

Task timeline



06:04:26 06:14:43

Code: 06:14:43 UTC, cpp, final, score: **100** show code in pop-up

```
1  // you can use includes, for example:
2  // #include <algorithm>
3
```

```
4  // you can write to stdout for debugging purpo
5  // cout << "this is a debug message" << endl;
6
7  int solution(int A, int B, int K) {
8    int first = ((A % K) == 0) ? A : (A / K + 1)
9
10   if (first > B) return 0;
11
12   return (B / K) - ((A > 0) ? (A - 1) / K : -1
13 }
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity: **O(1)**

expand all	Example tests	
▶	example A = 6, B = 11, K = 2	✔ OK
expand all	**Correctness tests**	
▶	simple A = 11, B = 345, K = 17	✔ OK
▶	minimal A = B in {0,1}, K = 11	✔ OK
▶	extreme_ifempty A = 10, B = 10, K in {5,7,20}	✔ OK
▶	extreme_endpoints verify handling of range endpoints, multiple runs	✔ OK
expand all	**Performance tests**	
▶	big_values A = 100, B=123M+, K=2	✔ OK
▶	big_values2 A = 101, B = 123M+, K = 10K	✔ OK
▶	big_values3 A = 0, B = MAXINT, K in {1,MAXINT}	✔ OK
▶	big_values4 A, B, K in {1,MAXINT}	✔ OK