codility

Training center

Check out Codility training tasks

Candidate Report: Anonymous

Test Name:

SUMMARY TIMELINE

Test Score

100 out of 100 points

100%

Tasks in Test

	Time Spent	Task Score
FrogJmp Submitted in: C++	3 min	100%

TASKS DETAILS

1. FrogJmp

EASY

Count minimal number of jumps from position X to Y.

Task Score	Correctness	Performance
100%	100%	100%

Task description

A small frog wants to get to the other side of the road. The frog is currently located at position X and wants to get to a position greater than or equal to Y. The small frog always jumps a fixed distance, D.

Count the minimal number of jumps that the small frog must perform to reach its target.

Write a function:

```
int solution(int X, int Y, int D);
```

Solution

Programming language used:	C++
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

that, given three integers X, Y and D, returns the minimal number of jumps from position X to a position equal to or greater than Y.

For example, given:

```
X = 10
Y = 85
D = 30
```

the function should return 3, because the frog will be positioned as follows:

- after the first jump, at position 10 + 30 = 40
- after the second jump, at position 10 + 30 + 30 = 70
- after the third jump, at position 10 + 30 + 30 + 30 = 100

Assume that:

- X, Y and D are integers within the range [1..1,000,000,000];
- X ≤ Y.

Complexity:

- expected worst-case time complexity is O(1);
- expected worst-case space complexity is O(1).

Task timeline

14:59:37 15:02:33

Code: 15:02:33 UTC, cpp, final, score: **100** show code in pop-up

```
1  // you can use includes, for example:
2  // #include <algorithm>
3  #include <math.h>
4  // you can write to stdout for debugging
5  // cout << "this is a debug message" << e
6
7  int solution(int X, int Y, int D) {
8      // write your code in C++14 (g++ 6.2.
9      return ceil ((double)(Y-X)/D);
10 }
```

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶	example example test	✔ OK
expand all	**Correctness tests**	
▶	simple1 simple test	✔ OK
▶	simple2	✔ OK
▶	extreme_position no jump needed	✔ OK
▶	small_extreme_jump one big jump	✔ OK
expand all	**Performance tests**	
▶	many_jump1 many jumps, D = 2	✔ OK
▶	many_jump2 many jumps, D = 99	✔ OK
▶	many_jump3	✔ OK

	many jumps, D = 1283	
▶	big_extreme_jump maximal number of jumps	✔ OK
▶	small_jumps many small jumps	✔ OK