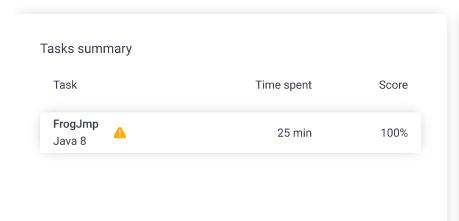
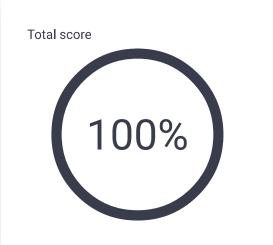
# Codility\_

## CodeCheck Report: trainingZ22PE6-WYV

Test Name:

Summary Timeline Check out Codility training tasks





#### **Tasks Details**



### Task description Solution

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:

class Solution { public int solution(int X, int Y, int D); }

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.

Programming language used: Java 8 Total time used: 25 minutes Effective time used: 25 minutes Notes: not defined yet Task timeline 0

For example, given:							_	7				
X = 10												0
Y = 85												17:51:10
D = 30	How I	ikely a	re you	to rec	omme	nd Co	dility to	o your	friend	ls and	$\times$	
the function should retu as follows:	collea	gues?										show code in pop-up
<ul><li> after the first jur</li><li> after the second</li></ul>	0	1	2	3	4	5	6	7	8	9	10	For example:
70	Not at a	ıll likely								Extreme	ely likely	debugging purposes,

• after the third jump, at position 10 + 30 + 30 + 30 = 100

Write an efficient algorithm for the following assumptions:

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

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

```
// System.out.println("this is a debug message");
6
7
    class Solution {
8
         public int solution(int X, int Y, int D) {
9
             int delta = Y - X;
             if(delta == 0)
10
11
                 return 0;
12
13
             int steps = (delta % D == 0) ? delta / D :
14
15
             return steps;
16
        }
17
     }
```

### Analysis summary

The solution obtained perfect score.

### Analysis

Detected time complexity: O(1)

expar	nd all	Example tests		
•	example example test	<b>~</b>	,	OK
expar	nd all C	orrectness test	S	
•	simple1 simple test	V	′	OK
•	simple2	<b>√</b>	1	OK
•	extreme_position no jump needed	<b>~</b>	,	OK
•	small_extreme_jur one big jump	np 🗸	′	OK
expar	nd all Pe	erformance tes	ts	
•	many_jump1 many jumps, D = 2	V	′	OK
•	many_jump2 many jumps, D = 99	<b>~</b>	,	OK
•	many_jump3 many jumps, D = 1283	<b>~</b>	,	OK
•	big_extreme_jump maximal number of jur	•	′	ок
•	small_jumps many small jumps	<b>~</b>	′	OK

How likely are you to recommend Codility to your friends and X colleagues?

O 1 2 3 4 5 6 7 8 9 10

Not at all likely

Extremely likely