

Summary

Timeline

Tasks summary

Task	Time spent	Score
FrogJump Java 8	25 min	100%

Total score

100%

Tasks Details

Easy	1. FrogJump	Task Score	Correctness	Performance
	Count minimal number of jumps from position X to Y.	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:

```
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.

For example, given:

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

the function should return 3 as follows:

- after the first jump - the frog will be at position 40
- after the second jump - the frog will be at position 70
- after the third jump - the frog will be at position 100

Solution

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

Task timeline



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

0 1 2 3 4 5 6 7 8 9 10

Not at all likely Extremely likely

- 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];
- $X \leq Y$ .

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

```
5 // 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;
10        if(delta == 0)
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)**

expand all	Example tests
▶ example	✓ OK
example test	
expand all	Correctness tests
▶ simple1	✓ OK
simple test	
▶ simple2	✓ OK
▶ extreme_position	✓ OK
no jump needed	
▶ small_extreme_jump	✓ OK
one big jump	
expand all	Performance tests
▶ many_jump1	✓ OK
many jumps, D = 2	
▶ many_jump2	✓ OK
many jumps, D = 99	
▶ many_jump3	✓ OK
many jumps, D = 1283	
▶ big_extreme_jump	✓ OK
maximal number of jumps	
▶ small_jumps	✓ OK
many small jumps	

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

0

1

2

3

4

5

6

7

8

9

10

Not at all likely

Extremely likely