

CodeCheck Report: training9VS684-D98

[Check out Codility training tasks](#)

Test Name:

Summary Timeline

Tasks summary

Task	Time spent	Score
FrogJump Java 8	5 min	100%

Total score



Tasks Details

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

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

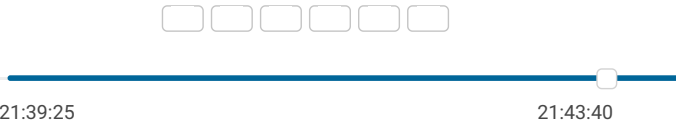
- X, Y and D are integers within the range [1..1,000,000,000];
- $X \leq Y$.

Copyright 2009–2023 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

Programming language used:	Java 8
Total time used:	5 minutes
Effective time used:	5 minutes
Notes:	not defined yet

Task timeline



Code: 21:43:40 UTC, java, final, score: 100 [show code in pop-up](#)

```
1 class Solution {
2     public int solution(int X, int Y, int D) {
3         return (Y - X + D - 1) / D;
4     }
5 }
```

Analysis summary

The solution obtained perfect score.

Analysis

O(1)

Detected time complexity:

collapse all		Example tests	
▼	example		✓ OK
example test			
1.		0.004 s	OK
collapse all		Correctness tests	
▼	simple1		✓ OK
simple test			
1.		0.004 s	OK
2.		0.004 s	OK
▼	simple2		✓ OK
1.		0.008 s	OK
2.		0.004 s	OK
▼	extreme_position		✓ OK
no jump needed			
1.		0.004 s	OK
2.		0.004 s	OK
▼	small_extreme_jump		✓ OK
one big jump			
1.		0.004 s	OK
collapse all		Performance tests	
▼	many_jump1		✓ OK
many jumps, D = 2			
1.		0.004 s	OK
▼	many_jump2		✓ OK
many jumps, D = 99			
1.		0.008 s	OK
▼	many_jump3		✓ OK
many jumps, D = 1283			
1.		0.004 s	OK
▼	big_extreme_jump		✓ OK
maximal number of jumps			
1.		0.004 s	OK
▼	small_jumps		✓ OK
many small jumps			
1.		0.004 s	OK