# Codility\_

# CodeCheck Report: trainingEER35X-6DH

Test Name:

Summary

Timeline 👜 Al Assistant Transcript

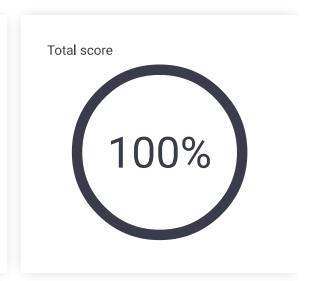
Tasks summary

Task

Time spent Score

FrogJmp
C#

12 min 100%



Check out Codility training tasks

## **Tasks Details**

1. FrogJmp 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

## Solution

Programming language u	sed: C#	
Total time used:	12 minutes	0
Effective time used:	12 minutes	•
Notes:	not defined yet	
sk timeline		
$\overline{}$		

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 ≤ Y.

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

```
Code: 16:17:26 UTC, cs, final,
                                    show code in pop-up
score: 100
     class Solution {
2
         public int solution(int X, int Y, int D) {
3
             int distance = Y - X;
             int minimalJumps = distance / D;
4
 5
             if (distance % D != 0) {
 6
                 minimalJumps++;
 7
             }
8
             return minimalJumps;
9
         }
10
     }
```

# Analysis summary

The solution obtained perfect score.

## Analysis

# Detected time complexity: O(1)

expa	nd all	Example tests
•	example example test	✓ OK
ехра	nd all Co	orrectness tests
•	simple1 simple test	✓ OK
<b>&gt;</b>	simple2	√ OK
•	extreme_position no jump needed	√ OK
•	small_extreme_jum	np ✓ OK
expa	nd all Pe	rformance tests
•	many_jump1 many jumps, D = 2	√ OK
<b>&gt;</b>	many_jump2 many jumps, D = 99	√ OK
•	many_jump3 many jumps, D = 1283	√ OK
•	big_extreme_jump maximal number of jum	✓ <b>OK</b>
•	small_jumps many small jumps	√ OK