Get account with free plan

See how Codility works from recruiter's point of view.

closed

Demo ticket

Session
ID: demoZR8RYC-CBY
Time limit: 120 min.

Status: closed
Started on: 2014-01-05 06:51 UTC

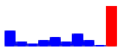
Score:

100

of 100

1. FrogJmp

score: 100 of 100



Count minimal number of jobs from position X to Y.

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

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).

Solution

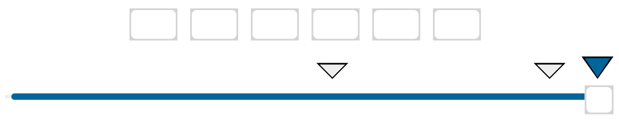
Programming language used: C#

Total time used: 2 minutes (?)

Effective time used: 1 minutes (?)

Notes: correct functionality and scalability

Task timeline What is it?



06:51:14 06:52:27

Code: 06:52:27 UTC, cs, final, score: 100.00

```
using System;
// you can also use other imports, for example:
// using System.Collections.Generic;
class Solution {
        public int solution(int X, int Y, int D)
        {
            // write your code in C90
            if (X < 0 || Y < 0 || D < 0 || X > 1000000000 || Y > 1000000000 || D > 1000000000) throw new ArgumentOutOfRangeException();
            if (X > Y) throw new InvalidOperationException();

            return (int)Math.Ceiling(((Y - X) / (D * 1.0M)));
        }
}
```

Analysis

Detected time complexity:
O(1)

test	time	result
example example test	0.080 s.	**OK**
simple1 simple test	0.080 s.	**OK**

simple2	0.080 s.	**OK**
extreme_position no jump needed	0.080 s.	**OK**
small_extreme_jump one big jump	0.080 s.	**OK**
many_jump1 many jumps, D = 2	0.080 s.	**OK**
many_jump2 many jumps, D = 99	0.080 s.	**OK**
many_jump3 many jumps, D = 1283	0.080 s.	**OK**
big_extreme_jump maximal number of jumps	0.080 s.	**OK**
small_jumps many small jumps	0.080 s.	**OK**

Get acco