g center

lity training tasks

Congratulations

You have completed a Codility training test.

Tweet this!
I scored 100% in #python on @Codility!
https://codility.com/demo/take-sample-test/frog_jmp/

Sign up for our newsletter!

Like us on Facebook!

Training ticket

Session

ID: trainingMRS33T-BX3
Time limit: 120 min.

Status: closed

Created on: 2017-09-05 16:23 UTC
Started on: 2017-09-05 16:23 UTC
Finished on: 2017-09-05 16:41 UTC

Tasks in test	Correctness	Performance	Task score
1 FrogJmp Submitted in: Python	100%	100%	100%

Test score ?

100%

100 out of 100 points

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

Not at all likely *Extremely likely*

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:

```
def solution(X, Y, 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: Python

Total time used: 19 minutes

Effective time used: 19 minutes

Notes: *not defined yet*

Task timeline

16:23:23 16:41:43

Code: 16:41:41 UTC, py, final, score: **100**

show code in pop-up

```
def solution(X, Y, D):
  covered_distance = Y-X
  jumps = covered_distance/D
  if covered_distance%D > 0: jumps +=1
  return jumps
```

Analysis summary

The solution obtained perfect score.

Analysis

Detected time complexity:
O(1)

collapse all Example tests

▼ example ✓ OK

example test

1. 0.020 s **OK**

collapse all **Correctness tests**

▼ simple1 ✓ OK

simple test

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ simple2 ✓ OK

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ extreme_position ✓ OK

no jump needed

1. 0.016 s **OK**
2. 0.016 s **OK**

▼ small_extreme_jump ✓ OK

one big jump

1. 0.016 s **OK**

collapse all **Performance tests**

▼ many_jump1 ✓ OK

many jumps, D = 2

1. 0.016 s **OK**

▼ many_jump2 ✓ OK

many jumps, D = 99

1. 0.016 s **OK**

▼

many_jump3 ✓ OK

many jumps, D = 1283

1. 0.016 s OK

▼ big_extreme_jump ✓ OK

maximal number of jumps

1. 0.016 s OK

▼ small_jumps ✓ OK

many small jumps

1. 0.016 s OK

Training center