cødility

TASKS DETAILS

1.
FrogJmp

EASY

Count minimal number of jumps from position X to Y.

Task Score	Correctness	Performance
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:

```
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];

Solution

Programming language used:	Python
Total time used:	1 minutes
Effective time used:	1 minutes
Notes:	*not defined yet*

Task timeline

02:03:37   02:03:58

Code: 02:03:58 UTC, py, final, score: **100**   show code in pop-up

```
1   # Round 5
2   def solution(X, Y, D):
3       distance = Y - X
4       jumps = distance / D
5       decimal_needs = jumps % 1
6       if decimal_needs == 0:
7           return int(jumps)
8       else:
9           return int(jumps) + 1
```

Analysis summary

- X ≤ Y.

Complexity:

- expected worst-case time complexity is O(1);
- expected worst-case space complexity is O(1).

The solution obtained perfect score.

Analysis ❓

Detected time complexity: O(1)

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