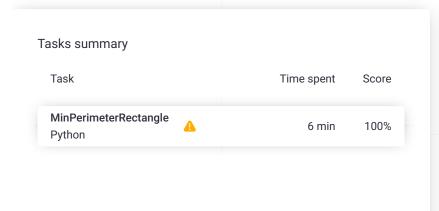
# Codility\_

### CodeCheck Report: trainingRX6BUF-YDX

Test Name:

Summary Timeline

Check out Codility training tasks





### **Tasks Details**

MinPerimeterRectangle
Find the minimal perimeter
of any rectangle whose area equals N.

Task Score
Correctness
Performance
100%
100%

### Task description

An integer N is given, representing the area of some rectangle.

The area of a rectangle whose sides are of length A and B is A  $^{\star}$  B, and the perimeter is 2  $^{\star}$  (A + B).

The goal is to find the minimal perimeter of any rectangle whose area equals N. The sides of this rectangle should be only integers.

For example, given integer N = 30, rectangles of area 30 are:

- (1, 30), with a perimeter of 62,
- (2, 15), with a perimeter of 34,
- (3, 10), with a perimeter of 26,
- (5, 6), with a perimeter of 22.

### Write a function:

def solution(N)

that, given an integer N, returns the minimal perimeter of any rectangle whose area is exactly equal to N.

For example, given an integer N = 30, the function should return 22, as explained above.

# Programming language used: Python Total time used: 6 minutes Effective time used: 6 minutes Notes: not defined yet Task timeline 13:50:16 13:56:02 Code: 13:56:02 UTC, py, final, score: 100

# you can write to stdout for debugging purpo

# print("this is a debug message")

1 von 2

Write an efficient algorithm for the following assumptions:

• N is an integer within the range [1..1,000,000,000].

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

```
3
    import math
4
5
    def solution(N):
6
        # Implement your solution here
7
        # pass
8
        min_perimeter = float('inf')
9
        sqrt_N = int(math.sqrt(N))
10
         for i in range(1, sqrt_N + 1):
11
             if N % i == 0:
12
                 factor_1 = i
13
14
                 factor_2 = N // i
                 perimeter = 2 * (factor_1 + facto
15
16
                 min_perimeter = min(min_perimeter
17
18
         return min_perimeter
19
```

### Analysis summary

The solution obtained perfect score.

### Analysis

## Detected time complexity: O(sqrt(N))

expand all	Example tests
example example test	<b>✓</b> OK
expand all Correctness tests	
<pre>extreme_min N = 1 test</pre>	<b>∠</b> OK
simple1 N = 36 test	<b>∠</b> OK
simple2 N = 48 test	<b>∠</b> OK
simple3 N = 101 test	<b>∠</b> OK
► small N = 1,234 test	<b>∠</b> OK
expand all Performance tests	
► medium N = 4,564,320 test	<b>∠</b> OK
► prime1 N = 15,486,451 tes	<b>∨</b> OK
► square N = 100,000,000 te	✓ OK  est
► prime2 N = 982,451,653 te	✓ OK  est
extreme_max N = 1,000,000,000	✓ OK test

2 von 2