

## CodeCheck Report: trainingRX6BUF-YDX

Test Name:

[Check out Codility training tasks](#)

Summary

Timeline

## Tasks summary

Task	Time spent	Score
MinPerimeterRectangle Python	6 min	100%

## Total score



## Tasks Details

Easy	1.				
	<b>MinPerimeterRectangle</b>	Task Score	Correctness	Performance	
	Find the minimal perimeter of any rectangle whose area equals N.	100%	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  $A * B$ , and the *perimeter* is  $2 * (A + B)$ .

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.

## Solution

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

[show code in pop-up](#)

```
1 # you can write to stdout for debugging purpo
2 # print("this is a debug message")
```

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
11     for i in range(1, sqrt_N + 1):
12         if N % i == 0:
13             factor_1 = i
14             factor_2 = N // i
15             perimeter = 2 * (factor_1 + factor_2)
16             min_perimeter = min(min_perimeter, 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	✓ OK
expand all	Correctness tests
▶ extreme_min N = 1 test	✓ OK
▶ simple1 N = 36 test	✓ OK
▶ simple2 N = 48 test	✓ OK
▶ simple3 N = 101 test	✓ OK
▶ small N = 1,234 test	✓ OK
expand all	Performance tests
▶ medium N = 4,564,320 test	✓ OK
▶ prime1 N = 15,486,451 test	✓ OK
▶ square N = 100,000,000 test	✓ OK
▶ prime2 N = 982,451,653 test	✓ OK
▶ extreme_max N = 1,000,000,000 test	✓ OK