Codility_

CodeCheck Report: trainingBUYTCG-HYS

Test Name:

Check out Codility training tasks

Summary | Timeline

Tasks summary

Task	Time spent	Score
MinPerimeterRectangle C#	3 min	100%

Total score

Tasks Details

Easy

1. MinPerimeterRectangle

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

Task Score	Correctness	Performance
N/A	Not assessed	Not assessed

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.

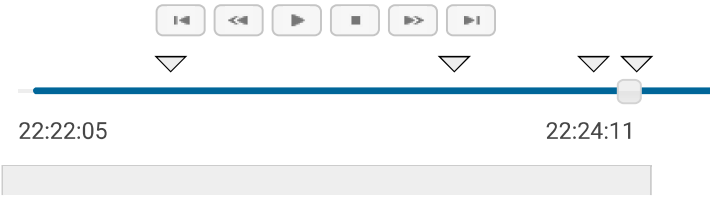
Write a function:

```
class Solution { public int solution(int N); }
```

Solution

Programming language used:	C#
Total time used:	3 minutes
Effective time used:	3 minutes
Notes:	*not defined yet*

Task timeline



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.

Write an **efficient** algorithm for the following assumptions:

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

Code: 22:24:11 UTC, cs, final, score: show code in pop-up

```
using System;

/**
 * 10.2 - Min Perimeter Rectangle
 * Paulo Santos
 * 15.Dec.2022
 */
class Solution {
    public int solution(int N) {

        /*
         * Math gives that the min
         * perimeter for a given area N
         * is Sqrt(N).
         */
        var root = (int)Math.Sqrt(N);
        if (Math.Pow(root, 2) == N)
            return (4 * root);
        if ((root * (root + 1)) == N)
            return (4 * root + 2);

        var min = int.MaxValue;
        for(var i = 1; i <= root; i++)
            if ((i * (N / i)) == N)
                min = Math.Min(min, 2 * (i + (N / i
                
        return min;
    }
}
```

The submission is being evaluated.