Codility_

CodeCheck Report: trainingKYFQ9K-9X4

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
CountFactors C++ ⚠	2 min	100%

Total score

Tasks Details

	1. CountFactors Count factors of given number n.	Task Score	Correctness	Performance
Easy		100%	100%	100%

Task description

A positive integer D is a *factor* of a positive integer N if there exists an integer M such that N = D * M.

For example, 6 is a factor of 24, because M = 4 satisfies the above condition (24 = 6 * 4).

Write a function:

```
int solution(int N);
```

that, given a positive integer N, returns the number of its factors.

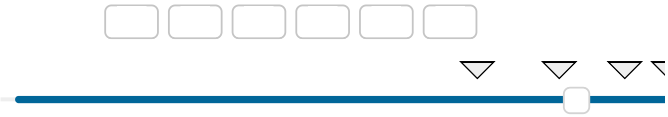
For example, given N = 24, the function should return 8, because 24 has 8 factors, namely 1, 2, 3, 4, 6, 8, 12, 24. There are no other factors of 24.

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

Solution

Programming language used:	C++	
Total time used:	2 minutes	?
Effective time used:	2 minutes	?
Notes:	*not defined yet*	

Task timeline ?



- N is an integer within the range [1..2,147,483,647].

11:16:47 11:18:22

Code: 11:18:22 UTC, cpp, final, score: **100**

show code in pop-up

```cpp
// you can use includes, for example:
// #include <algorithm>
#include <cmath>
// you can write to stdout for debuggin
// cout << "this is a debug message" <<

int solution(int N) {
    int result = 0;
    int sqrtN = static_cast<int>(sqrt(N

    for (int i = 1; i <= sqrtN; ++i) {
        if (N % i == 0) {
            if (i == N / i) {
                result += 1;
            } else {
                result += 2;
            }
        }
    }

    return result;
}
```

Analysis summary

The solution obtained perfect score.

Analysis

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

expand all	Example tests	
▶ example1 example test (N=24=4!)		✔ OK
expand all	Correctness tests	
▶ squares N=16, N=36		✔ OK
▶ tiny N <= 10		✔ OK
▶ simple1 N=41(prime), N=42		✔ OK
▶ simple2 N=69, N=64, N=120=5!		✔ OK
▶ simple3 N=720=6!, N=1111		✔ OK
▶ simple4 N=5,040=7!, N=12,345		✔ OK
▶ simple5 N=34,879, N=40,320=8!		✔ OK

▶ extreme_one ✔ OK

N=1

expand all **Performance tests**

▶ medium1 ✔ OK

N=362,880=9!, N=1,948,102

▶ medium2 ✔ OK

N=3,628,800=10!, N=5,621,892, N=4,999,696

▶ big1 ✔ OK

N=27,043,111, N=39,916,800=11!, N = 39,992,976

▶ big2 ✔ OK

N=97,093,212, N=2^28

▶ big3 ✔ OK

N=479,001,600=12!, N=780291637(prime), N=449,991,369

▶ extreme_maxint ✔ OK

N=1,000,000,000, N=MAX_INT, N=2147,395,600