Time Complexity: Primality

by blondiebytes

Problem | Submissions | Leaderboard | Discussions | Editorial

Check out the resources on the page's right side to learn more about asymptotic analysis. The video tutorial is by Gayle Laakmann McDowell, author of the best-selling interview book Cracking the Coding Interview.

A *prime* is a natural number *greater than* 1 that has no positive divisors other than 1 and itself. Given p integers, determine the primality of each integer and print whether it is `Prime` or `Not prime` on a new line.

Note: If possible, try to come up with an $O(\sqrt{n})$ primality algorithm, or see what sort of optimizations you can come up with for an $O(n)$ algorithm. Be sure to check out the *Editorial* after submitting your code!

Input Format

The first line contains an integer, p, denoting the number of integers to check for primality.
Each of the p subsequent lines contains an integer, n, you must test for primality.

Constraints

- $1 \leq p \leq 30$
- $1 \leq n \leq 2 \times 10^9$

Output Format

For each integer, print whether n is `Prime` or `Not prime` on a new line.

Sample Input

```
3
12
5
7
```

Sample Output

```
Not prime
Prime
Prime
```

Explanation

We check the following $p = 3$ integers for primality:

1. $n = 12$ is divisible by numbers other than 1 and itself (i.e.: 2, 3, 6), so we print `Not prime` on a new line.
2. $n = 5$ is only divisible 1 and itself, so we print `Prime` on a new line.
3. $n = 7$ is only divisible 1 and itself, so we print `Prime` on a new line.

Submissions: 4526
Max Score: 30
Difficulty: Medium

Rate This Challenge:

Need Help?

8:37

Running Time and Complexity

Download problem statement

Download sample test cases

Suggest Edits

Collapse

Current Buffer (saved locally, editable) C#

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
class Solution {

     static string VerificarPrimo(int n)
        {
            if (n < 2) return "Not prime";
            if (n == 2) return "Prime";
            if (n % 2 == 0) return "Not prime";

            int sqr = (int)Math.Sqrt(n);

            for (int i = 3; i <= sqr; i += 2)
            {
                if (n % i == 0)
                {
                    return "Not prime";
                }
            }
            return "Prime";
        }


        static void Main(string[] args)
        {
            int p = Convert.ToInt32(Console.ReadLine());
            for (int a0 = 0; a0 < p; a0++)
            {
                int n = Convert.ToInt32(Console.ReadLine());
                Console.WriteLine(VerificarPrimo(n));
            }
           // Console.ReadLine();
        }
}

```

Line: 34 Col: 14

Upload Code as File Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

- ✔ Test Case #0
- ✔ Test Case #1
- ✔ Test Case #2
- ✔ Test Case #3
- ✔ Test Case #4
- ✔ Test Case #5
- ✔ Test Case #6
- ✔ Test Case #7
- ✔ Test Case #8
- ✔ Test Case #9

Next Challenge