

















Dashboard > Tutorials > 30 Days of Code > Day 25: Running Time and Complexity

Day 25: Running Time and Complexity





by blondiebytes

Tutorial

Problem

Submissions

Leaderboard

Discussions

Editorial 🖴

Objective

Today we're learning about running time! Check out the Tutorial tab for learning materials and an instructional video!

A prime is a natural number greater than 1 that has no positive divisors other than 1 and itself. Given a number, n, determine and print whether it's Prime or Not prime.

Note: If possible, try to come up with a $O(\sqrt{n})$ primality algorithm, or see what sort of optimizations you 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, T, the number of test cases. Each of the T subsequent lines contains an integer, n, to be tested for primality.

Constraints

- $1 \le T \le 30$
- $1 \le n \le 2 \times 10^9$

Output Format

For each test case, print whether **n** is **Prime** or **Not prime** on a new line.

Sample Input

3

12

5

Sample Output

Not prime

Prime

Prime

Explanation

Test Case 0: n = 12.

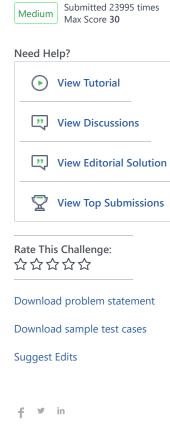
12 is divisible by numbers other than 1 and itself (i.e.: 2, 3, 6), so we print Not prime on a new line.

Test Case 1: n = 5.

5 is only divisible 1 and itself, so we print Prime on a new line.

Test Case 2: n = 7.

7 is only divisible 1 and itself, so we print Prime on a new line.



```
Current Buffer (saved locally, editable) &
                                                                                 Java 8
                                                                                                               Ö
1 ▼ import java.util.Scanner;
2
3 ▼ public class Solution {
4
5
        static boolean isPrime(int n) {
             //check if n is a multiple of 2, knowing if even nums are prime is trivial
6
7
            if (n % 2 == 0) return false;
8
             //then check the odds
9
             for (int i = 3; i * i <= n; i += 2) {
10
                 if (n \% i == 0)
                     return false;
11
12
13
            return true;
14
15
16 ▼
        public static void main(String[] args) {
17
18
            Scanner sc = new Scanner(System.in);
19
             int input = sc.nextInt();
20
21 ▼
            int[] nums = new int[input];
22 ▼
            for (int i = 0; i < input; i++) {
23
                 nums[i] = sc.nextInt();
24
25
            sc.close();
26
27
             for (int i :
28 ▼
                     nums) {
29
                 if (isPrime(i)) {
30
                     System.out.println("Prime");
31
                 } else {
32
                     System.out.println("Not prime");
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature