4/18/2018 HackerRank



PRACTICE

COMPETE

JOBS LEADERBOARD

Q Search

Sashu1231 V

 $\label{eq:practice} \mbox{Practice} \mbox{ > Tutorials} \mbox{ > 30 Days of Code} \mbox{ > Day 25: Running Time and Complexity}$

5 more challenges to get your gold badge!

<u>Learn more</u>



38% 25/30

Day 25: Running Time and Complexity ☆



Problem

Submissions

Leaderboard

Discussions

Editorial A

Tutorial

Objective

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

Task

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 $m{T}$ subsequent lines contains an integer, $m{n}$, 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

_

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.

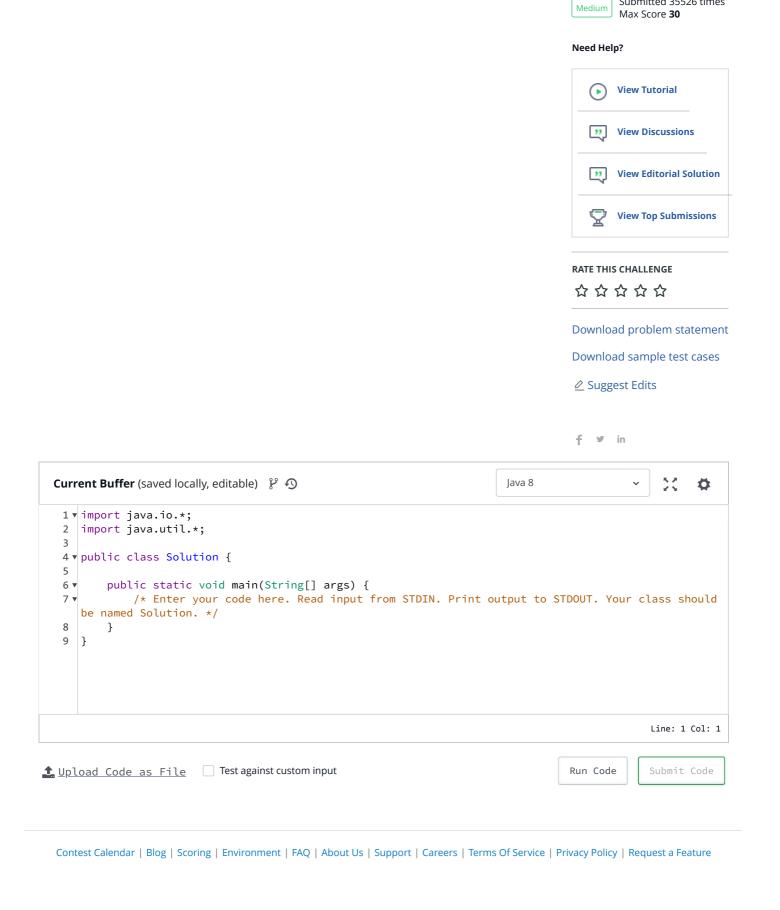
Test Case 1: n = 5

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

4/18/2018 HackerRank

Test Case 2: n = 7.

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



Submitted 35526 times