Day 25: Running Time and Complexity!

Welcome to Day 25! Check out a video review of running time, or just jump right into the problem.

In this challenge, you will determine if a given number X is prime or not. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. You will be given N numbers and for each, you will print out "Prime" if the number is prime or "Not prime" if the number is not prime.

If this is too easy, create a method that decides if X is prime or not in $O(\sqrt{X})$ time. Think modulos and square root! If you are having trouble, try creating an $O(X)$ time algorithm and see whether it solves the problem or not.

To review Big-O Notation, remember...

- Big-O "is used in Computer Science to describe the performance or complexity of an algorithm."
- Big-O "specifically describes the worst-case scenario, and can be used to describe the execution time required or the space used (e.g. in memory or on disk) by an algorithm."
- Read more here

Good luck!

Input Format

The first line of the input is **T**, total number of test cases. Each of the next line contains an integer **N**.

Constraints

- $1 \le T \le 20$
- $1 \le N \le 2 \times 10^{9}$

Output Format

For each test case print *Prime* if **N** is prime, otherwise print *Not prime*.

Sample Input

```
3
12
5
7
```

Sample Output

```
Not prime
Prime
Prime
```

Explanation

There are three testcases 12, 5, and 7. 12 is not prime, 5 is prime, and 7 is prime.