Solved Problem - Check Prime

In this lesson, we'll look at an efficient way to perform the primality test.

We'll cover the following

Problem statement

Given a number, N, check if it's prime or not. Do this for T test cases.

Input format

The first line contains a positive integer T $(1 \leq T \leq 10^3)$.

The following T lines contain a positive integer N $(1 \leq N \leq 10^8)$.

Output format For each integer, N, print `yes` if N is prime, otherwise print `no` on a new line.

Sample

Input

```
5
34
29
11
14
6
```

Output

```
no
yes
yes
no
no
```

Brute force

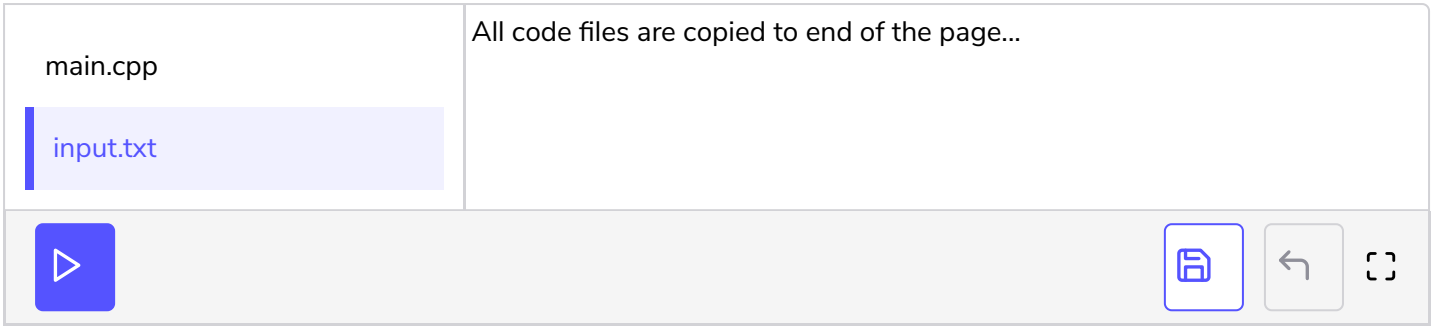
The brute force solution would be to check whether N has any factor between 2 and $N - 1$ (*inclusive*).

The time complexity would be $O(N)$ for one test case. So the overall time complexity would be $O(T * N)$.

Optimization

Similar to the factorization problem. We only need to iterate up to $\sqrt{N}$ to find factors.

This reduces our time complexity to $O(T * \sqrt{N})$, which is good enough for given the constraints.



In the next lesson, we'll learn about arithmetic progression.

Code Files Content !!!

⤋ ⤋

```
------------------------------------------------------------------------------
|  main.cpp [1]
------------------------------------------------------------------------------


#include
#include
#include
#include
using namespace std;

string is_prime(int N) {
  for (int i = 2; i * i <= N; i ++)
    if (N % i == 0) {
      return "no";
    }
  return "yes";
}

int main() {
  ifstream cin("input.txt");
  int T, N;

  cin >> T;
  while (T--) {
    cin >> N;
    cout << is_prime(N) << "\n";
  }
  return 0;
}


------------------------------------------------------------------------------
|  input.txt [1]
------------------------------------------------------------------------------


5
34
29
11
14
6




********************************************************************************
```