CMPUT 275 - Tangible Computing
Interview Problem: Twin Primes

Description

Given a single query of the form "p n" or "t k", where n and k both integers values (and p or t are actual characters) you must do the following.

- `isPrime` query

  If the query takes the form of "p n" with $0 \leq n \leq 100,000$, call a function `bool isPrime(int n);` that returns true if n is **prime**.

  A **prime** is any integer $p \geq 2$ that is divisible only by one and itself.

  Your implementation of `isPrime` must have running time $O(n)$ (or better) to be considered for full credit.

  As an **optional challenge** try to write an implementation that runs in $O(\sqrt{n})$ time. A hint towards this faster version is the following: for any three positive integers a, b, n such that $n = a \cdot b$, it must be either $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.

- `twinPrimes` query

  If the query takes the form of "t k" with $1 \leq k \leq 1,000$, call a function `void twinPrimes(int k);` that prints the first k **twin prime pairs**, in order, one per line (see the sample output below).

  A **twin prime pair** is a pair of two **primes** of the form $(p,\ p+2)$.
  e.g. (79559, 79561) is the 1000th twin prime pair.

  The function `twinPrimes` can call `isPrime`; a solution with a hard-coded array of twin prime pairs will not be accepted. Your implementation should work for larger k in principle. For fun, try to see how large you can make k while maintaining a reasonable running time. There is not target $O()$ running time bound for this problem, it just has to finish running within a few seconds for all possible inputs $1 \leq k \leq 1,000$.

To get full credit, your function signatures must be exactly written as above (eg. `isPrime` returns a `bool` and accepts only a single parameter that is of type `int`). You can use additional functions if needed, but you must have at least `isPrime` and `twinPrimes` with the declaration and functionality described above.

Interesting Note: Determining if there are infinitely many twin primes is still an open problem.

Input

Input will consist of a **single** query either of the form "p n" or "t k" with $0 \leq n \leq 100,000$ and $1 \leq k \leq 1,000$.

Output

In response to a query "p n" you must output "prime" if n is prime and "not prime" oth-

erwise. In response to a query "t k" you must output the first k twin prime pairs, one per line.

Submission Instructions
Submit only a single source code file `twin_primes.cpp` containing your solution to this interview problem. Use exactly this file name and do not zip your solution.

Sample Input 1

```
p 3
```

Sample Output 1

```
prime
```

Sample Input 2

```
p 6
```

Sample Output 2

```
not prime
```

Sample Input 3

```
t 4
```

Sample Output 3

```
3 5
5 7
11 13
17 19
```

Grading Comments
Despite the fact this appears similar to a morning problem, it will be graded like a weekly exercise. In particular:

- Style matters. Use appropriate comments, proper indentation, etc. Consult the style guide on eClass.
- You must use the function signatures `bool isPrime(int n)` and `void twinPrimes(int k)`. The variable names n and k are not important, but their type is as is the return type of these functions. Deviating from this will result in a deduction.
- You must adhere exactly to the output specification: for example, if you misspell prime or print extra whitespaces then you will receive a deduction. The test centre must accept the output without any presentation error.

- You were only give a few test cases in the test centre files on eClass. We will test your solution on additional test cases that adhere to the input specification.
- Partial credit may be obtained if your solution works on some inputs but not all inputs in the described range.
- Adhere closely to the submission instructions for the weekly exercise.