CSCI 1300
Summer 2018
Assignment 2

Task 1

An integer is called “prime” if it is divisible only by itself and 1

- For example: 5 is prime, but 4 is not.

Your task is to write a program that decides whether or not an integer input by the user is prime or not. Furthermore, after the user enters a number, the program will ask whether the user wishes to enter a new number and continue if the user types a `'y'` (stops if user enters `'n'`).
Name your file `task1.cpp`

Hints:

- **1 is not prime**
- **No need to be clever with this. You can just use a “brute-force” method to test divisibility.**

Example run (user input in red):

```
Please enter an integer: 5
5 is a prime number!
Would you like to enter another number? (y/n) y
Please enter an integer: 6
6 is not a prime number!
Would you like to enter another number? (y/n) n
Exiting...
```

Task 2

Consider the following code snippet

```
#include <iostream>
using namespace std;

int main() {

      int numbers[5] = {1, 2, 3, 4, 5};
      bool is_prime[5]; // uninitialized

      // YOUR CODE GOES HERE
      return 0;
}
```

Your task is to write code that assigns either a **`true/false`** to each element of the `is_prime` array depending on whether or not each element of the `numbers` array is prime or not. You will then print the contents of the array. **Call your program `task2.cpp`**

Task 3 (Extra Credit -- 20pts)

Testing for a prime number by testing divisibility is a bit inefficient. One method to **efficiently** find prime numbers is to use what is called a "prime sieve" algorithm. Check out the following link to learn more about a famous instance of such an algorithm. **Call your program task3.cpp**

https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

The Sieve of Eratosthenes is a method for finding all prime numbers up to a given number.

Your task is to write a program that defines an array of 100 boolean values. Your goal is to implement the Sieve of Eratosthenes, which will take this array of 100 values and set all the **prime indices** to **true** and **false** otherwise.

For example, let **A** be our array. Then **A[2]** will be **true** and **A[4]** would be **false,** etc.

Note: You must be able to fully explain the algorithm/code that you've written to receive any points for this task.

Advice

- Finish task 1 as early as possible.
- Ask questions. Notice that I didn't go into that much detail regarding Task 2.