

Math 3300 Programming Assignment 6

Instructions: Work on the following 2 programs and submit your source code to me via Blackboard. Send me 2 cpp files.

1. Write a program in which the input is a single positive integer n . The program will then calculate and display the n^{th} prime number.

Recall that a prime number is a positive integer that has exactly 2 factors, the number and 1. The first prime number is 2 (only has factors 1 and 2). The second prime number is 3 (only factors are 1 and 3). The third prime number is 5 (only factors are 1 and 5).

So if the user enters 3, the output would be 5 since 5 is the 3rd prime number.

To help with this program, create a **function** called **isPrime** whose input is an integer. The function should return the integer 1 if the input is a prime number, and the function should return the integer 0 if the input is not prime.

Hint: 17 is prime because 2,3,4 do not divide into it. (so isPrime(17) returns 1)

107 is prime because 2,3,4,5,6,7,8,9,10 do not divide into it. (so isPrime(107) returns 1)

15 is not prime because 3 divides into it. (so isPrime(15) returns 0)

In each example above, think about why I only needed to check whether the listed numbers were factors, or don't think about it, and just check whether any number less than the integer divides into it.

Use your isPrime() function inside a while loop in **main** to count how many prime numbers you have found until you get to the desired number.

As an additional requirement, detect and allow the user to correct invalid input without restarting the program. (i.e. if the user enters a negative number or 0, explain what is wrong and allow them to enter a valid number without restarting the program). Save your program as **nthprime.cpp**.

2. Write a program whose input is a single integer, and which will create a file containing a "square" of that size on the screen consisting of alternating symbols of the user's choice. The user should be able to specify any filename, and you should perform the usual checks on any file created.

For example, if the user enters * and +,

and the user enters 5, the output would be:

```
*  +  *  +  *
+
*
+
*  +  *  +  *
```

Notice that there are 5 rows and 5 columns.

If the user enters 6, the output would be:

```
*  +  *  +  *  +
+
*
+
*
+
+  *  +  *  +  *
```

Notice that there are 6 rows and 6 columns.

To do this, create a **function** which can calculate the character that needs to be displayed based on the row number and column number.

As an added requirement, allow the user to repeat the program without restarting. Save your program as **square.cpp**.