

C++ Prime Number Finder

Time required: 30 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

Pseudocode

1. Write pseudocode for the exercise
2. Submit with the assignment

Prime Numbers

A prime number (or a prime) is a whole number greater than 1 that is divisible only by 1 and itself.

Prime numbers are the numbers that have only two factors, that are, 1 and the number itself. Consider an example of number 5, which has only two factors 1 and 5. This means it is a prime number. Let us take another example of the number 6, which has more than two factors, i.e., 1, 2, 3, and 6. This means 6 is not a prime number. Now, if we take the example of the number 1, we know that it has only one factor. So, it cannot be a prime number as a prime number should have exactly two factors. This means 1 is neither a prime nor a composite number, it is a unique number.

How It Works

The idea to solve this problem is to iterate the val from start to end using a for loop and for every number, if it is greater than 1, check if it divides n. If we find any other number which divides, print that value.

Tutorial 1 - Prime Number Finder

This is a hard coded program that finds the prime numbers or factors from 2 to 9.

```

1 /**
2  * Filename: PrimeNumberFinder.cpp
3  * Written by:
4  * Written on: Find prime numbers in a range
5  * Purpose: Find prime numbers in a range
6  * A number is prime if it is only divisible by itself and 1
7  */
8 #include <iostream>
9 int main()
10 {
11     bool isPrime = true;
12
13     // Hard coded range of numbers to check for prime numbers, 2-9
14     for (int n = 2; n < 10; n++)
15     {
16         isPrime = true;
17         // For each number to test for prime,
18         // test all numbers from 2 to n
19         for (int x = 2; x < n; x++)
20         {
21             // If n is divisible by x, it's not a prime number
22             if (n % x == 0)
23             {
24                 // Not a prime number
25                 // Divide to show the factors
26                 std::cout << n << " equals " << x << " * " << n / x << std::endl;
27                 isPrime = false;
28                 break;
29             }
30         }
31         // If the for loop runs to completion for a number
32         // n is a prime number (flag == true)
33         if (isPrime == true)
34         {
35             // If n was not divisible by any x, it is a prime number
36             std::cout << n << " is a prime number" << std::endl;
37         }
38     }
39     return 0;
40 }

```

Example run:

```

2 is a prime number
3 is a prime number
4 equals 2 * 2
5 is a prime number
6 equals 2 * 3
7 is a prime number
8 equals 2 * 4
9 equals 3 * 3

```

Requirements

Ask the user to enter a range of numbers to check for prime numbers. Calculate and display the prime numbers.

1. Add a creative program title
2. Get range input from user
3. Calculate and print the prime numbers or number and factors
4. Display the results
5. Add a menu loop that allows the user to run the program again

Example run:

```
+-----+
|           Python Prime Number Finder           |
| Find the prime numbers within a range          |
+-----+
First Number: 20
Last Number: 25
20 equals 2 * 10
21 equals 3 * 7
22 equals 2 * 11
23 is a prime number
24 equals 2 * 12
25 equals 5 * 5
Run again? (1)-Y (2)-N 1
Enter a range of numbers to test for prime
First Number: 1001
Last Number: 1005
1001 equals 7 * 143
1002 equals 2 * 501
1003 equals 17 * 59
1004 equals 2 * 502
1005 equals 3 * 335
Run again? (1)-Y (2)-N 2
Bye ;')
```

Assignment Submission

1. Attach the pseudocode.
2. Attach the program files.
3. Attach screenshots showing the successful operation of the program.
4. Submit in Blackboard.