14.06 Assignment Instructions

Instructions: Write a program to calculate and count the quantity of prime numbers between a given range.

Prime numbers and Cryptography:

The story of the quest for an unbreakable asymmetric cipher was solved by three MIT computer scientists in 1977: Ron Rivest, Adi Shamir, and Leonard Adelman. This team developed a one-way modular function (known as RSA encryption) that is beyond the scope of this course, but the general principle is not difficult.

Imagine Alice picks two prime numbers, ***p*** and ***q***, and calculates their product, such that ***N = p x q***. Assume that she chose ***p*** = 17,159 and ***q*** = 10,247, giving ***N*** a value of 175,828,273. N is now Alice's public encryption key that she distributes to anyone who wants it.

Bob looks up Alice's public key (N = 175,828,273) and uses it in the RSA one-way function, which is also public. Bob uses the one-way function specific to Alice's public key, encrypts the message, and sends it to her.

The RSA one-way function turns out to be reversible if ***p*** and ***q*** are known. Since Alice is the only one who knows the two secret prime numbers, she is the only one who can decrypt the message.

Are you skeptical at this point? If everybody knows ***N*** and the one-way formula, surely Eve or anyone else can deduce ***p*** and ***q***! The trick, of course, is to pick large prime numbers, say on the order of 10^{65}, which is 1 followed by 65 zeroes! Is that big enough for you? The product of two such prime numbers would be 10^{130} and it would take a standard desktop computer roughly 50 years to factor a prime number that big.

On the other hand, a hundred million personal computers working in tandem could factor a number as big as 10^{130} in approximately … 15 seconds. Since paranoia prevails among cryptographers, important transactions like banking business tend to use values of N that are at least 10^{308}. It is estimated that 100 million personal computers working in parallel would take more than a thousand years to crack a value of ***N*** that large. That should convince you that if you mind (the values of) your ***ps*** and ***qs***, RSA is impregnable.

Note: Numeric examples are excerpted from Simon Singh's, *Code Book.*

The Program:

1. The design and implementation details of this program are completely up to you. Do use object-oriented programming style? Make use of the `this` keyword when defining your implementation class.
2. The program should allow the user to enter a lower and an upper limit (e.g., 1 to 100 or 25 to 500).

3. All prime numbers within the lower and upper limits should be calculated.
4. Output needs to clearly communicate to the user what they are seeing. Neatly display all prime numbers within the given range. Also, provide a count of how many prime numbers were found.
5. Include documentation throughout the program to communicate each section's purpose.

Print