CS 1301 Homework 3

Homework – Next Prime
Due: Friday Sep 17th, before 6pm

THIS IS AN INDIVIDUAL ASSIGNMENT!

You should work individually on this assignment. You may collaborate with other students in this class. Collaboration means talking through problems, assisting with debugging, explaining a concept, etc. Students may only collaborate with fellow students currently taking CS 1301, the TA's and the lecturer. You should not exchange code or write code for others. For individual assignments, each student must turn in a unique program. Your submission must not be substantially similar to another student's submission. Collaboration at a reasonable level will not result in substantially similar code.

Scored out of 50 points
Your Submission file must be named:
prime.py

If you need help, we have several resources to assist you successfully complete this assignment:
- The TA Helpdesk – Schedule posted on class website.
- Email the TA's
- Jay's office hours

Notes:
• **Don't forget to include the required comments and collaboration statement (as outlined on the course syllabus).**
• **Do not wait until the last minute to do this assignment in case you run into problems.**
• If you find a significant error in the homework assignment, please let a TA know immediately.

Finding the next prime number: (50 points)

Firstly, what is a prime number? A prime number, strictly speaking, is a number greater than or equal to 2 that is only divisible by 1 and itself (0 can be divided by infinitely many numbers, and 1 can only be divided by itself. Negative numbers are typically not included). For example, the first few prime numbers are 2, 3, 5, 7, 11, 13, and 17.

For your assignment:

Write a function named **isPrime(num)** that accepts a single argument (num) and **returns** True if the argument is prime, and False otherwise.

Example: isPrime(2) should return True. isPrime(5) should return True as well. isPrime(4) should return False. You may assume that the input **num** will be a non-negative integer. (0, 1, 2, etc...)

Next, write a second function named **nextPrime(num)** that will *return* the next prime number following the corresponding number that is the argument. It is likely that your **nextPrime(num)** function can make use of the **isPrime(num)** function to make it's job easy!

For example: nextPrime(5) should *return* (NOT PRINT) the integer 7, and nextPrime(100) should *return* the integer 101. You may assume that only non-negative integers will be used as input.

Save and submit your two functions in a single file called **prime.py**

Grading Criteria for Part 1:

isPrime

Shows correct use of conditionals and loops:	-	10 pt
Produces correct results for all inputs:	-	10 pt
Function named correctly and returns a boolean (instead of printing)	-	5 pt

nextPrime

Shows correct use of conditionals and loops:	-	10 pt
Produces correct results for all inputs	-	10pt
Function named correctly and returns an int (instead of printing)	-	5 pt

NextPrime Originally Written By: Jason Tilley, Fall 2009