Problem Set 2: Working with Prime Numbers

For this assignment, you will use Python and Visual Studio Code to create a computer program. Submit this Python program script file (file extension is py) in Blackboard by Sunday at 11:59 PM. Using the information provided in the course up to this point, you will use a generate-and-test program that systematically generates solutions to a problem. Then you will use a sequence of tests to determine if the solution is valid. Sometimes this type of programming incorporates random numbers or probability to get to the solution, but quite often, it is more efficient to use a systematic approach.

Script Requirements

When writing and saving your programming files, ensure that you heed the instructions. For all programming coursework, specific things must be placed in the programming code. The beginning of the script file must include the assignment name, the purpose of the programming code, the date it was written or revised, and who wrote the code. These comments can be written as line comments or a commenting block. Figure 1 depicts an example of what this might look like in your file. When saving the program for the first problem, name the file **`assignment_2_1.py`**. The second problem shall be saved in a file named **`assignment_2_2.py`**. When you have completed and tested your script file, submit all of the script files for this assignment in Blackboard.

All analysis begins with a problem to solve, a question to answer, or something to prove (you might call this a hypothesis). The following information establishes the requirements of the program and some hints for completing this work. There are two problems that you need to solve for this assignment.

Figure 1

Content Needed in Every Script File

```
"""
Assignment 1: this is a basic input and output file
September 22, 2022
John Smith
"""
```

Note. This contains the name of the assignment (Assignment 1), purpose of this assignment (this is a basic input and output script), date (September 22, 2022), and the name of the developer (John Smith).

Problem One

The first problem has three requirements. First, write a program that will find the 450th prime number. After every 50 prime numbers in the search for the 450th, print the number of prime numbers found up to that point to the console. Include an explanation of what the iteration means (i.e., **`150 prime numbers found so far`**). When you have identified the 450th prime number, print a statement to the console that indicates what this program was looking for and the solution. (Not sure what a prime number is? You can read the explanation provided by Wikipedia.) Walk through how you might accomplish this task before you attempt to write the programming code.

All prime numbers are odd, other than the number two. This eliminates a lot of candidates. The first step is to create an array of odd numbers, then test whether they are prime or not. Do this while you identify the remaining 449 prime numbers. You could use the modulo to test the values. How many divisors need to be checked before ruling a number out? You can review this list of prime numbers to ensure that you have identified the correct prime number, you can review this list of prime numbers. It is very important to note that submissions that solely find prime numbers are unlikely to earn any points. A plethora of code snippets on the internet accomplish any of the objectives for this assignment. However, copying their work does not demonstrate that you understand it. To be successful, you have to do the work.

Problem Two

The second problem is built on the foundation of the first problem. It was established that the sum of the logarithms of prime numbers between 2 and `n` will converge on the value of `n`. For example, the sum of the logarithms of all prime numbers between 2 and 5000 is approximately 4912. The sum of the logarithms of all prime numbers between 2 and 50,000 is approximately 49,732. The problem is that some may contest that this is a valid theory. To solve this problem, write a program that evaluates the numbers between 2 and `n` to determine if the number is a prime number. If it is, find the logarithm of the prime number. Add all logarithms of the prime numbers between 2 and `n`. After iterating over the values between 2 and `n`, print out the sum, the number `n`, and the ratio of these two quantities. The ratio of the sum and `n` should converge on the number one. Evaluate the results with multiple values for `n` to demonstrate this convergence. To use the Python function `log()`, add `from math import *` to the script file immediately following the block comments that include the assignment name, assignment purpose, date, and your name at the top of the script file.