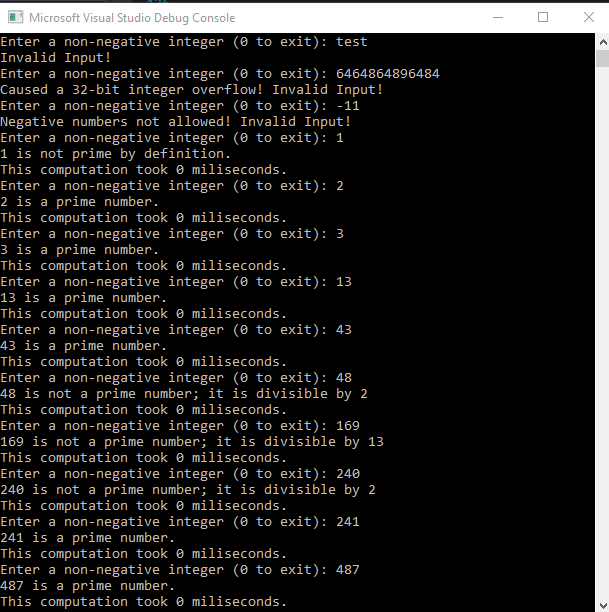
Brandon Hough

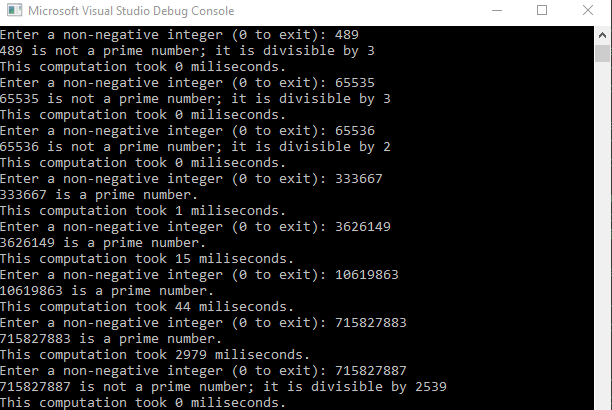
CPEN 3710

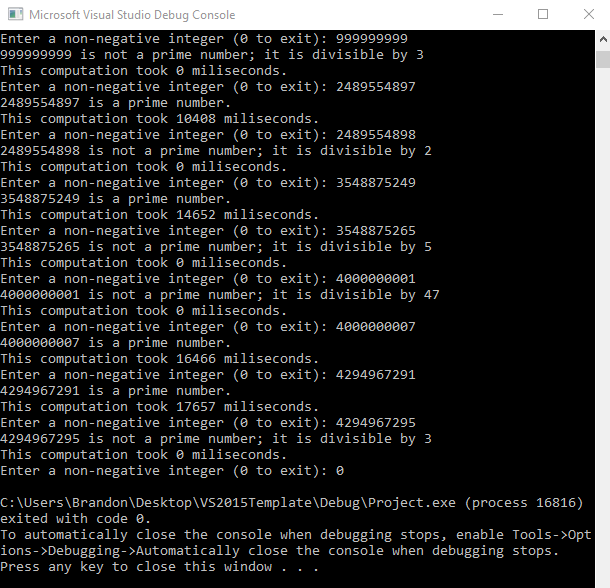
LQF316

Lab Exercise #9

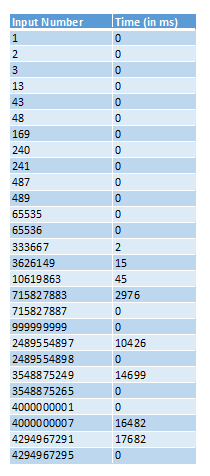
**Console Output:**







**Valid Input Times:**



**Graph of data:**

**Explanation:**

When running several valid inputs ranging from 1 to 232-1 I noticed that each number entered that was composite would run very quickly (0-1ms) no matter how small or large the value entered was. This is true because when a divisor can divide evenly into the input, you know the input number is composite. This is being done by dividing the input number by a divisor (starting at 2) and checking for a remainder. If the number is divisble by 2, then only one iteration will have be done. Computers can do these divison and comparsion instructions very fast which results in the quick times.

The run time is linearly proportional to the size of the number, but only with prime numbers. If the value entered was prime, the larger the value entered the longer it took to process the number to determine if it was prime. This is because you will have to check every divisor, starting from 2, up till the input number. When the divisor is equal to the input number then the number is prime since it does not evenly divide into any number but itself. With a large value, such as 4000000007, you will have to check every value from 2 up to 4000000007 and that large amount of computations is what makes this program take longer when larger prime values.

Instructor’s signature:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Staple your program listing, results, performance analysis, and this signed sheet together. Submit these items by the date and time specified above. Late submissions will be penalized substantially.