Factorial Project Report

Donovan Byler – March 22, 2019

CS23001 – Computer Science 2

**Program Function**

This program is designed to compute the factorial of potentially massive numbers using linked list nodes to store digits as the result becomes too complex to handle with primitive data types. The factorial program is bottlenecked by the amount of memory on the system performing the operation. The purpose for creating this program is to exercise and demonstrate my ability to work with data structures.

**Design**

My factorial program uses three functions alongside the main to calculate any number of factorials. The main function runs a while loop that ensures that the program continues to run so long as the user continues to enter a positive number. It creates an instance of the class, computes the factorial, reverses the numbers, and prints the output to a file.

The compute function uses a for loop to ensure that every number of the factorial is properly calculated. The numbers are stored in nodes comprised of eight digits. When a node become bigger than 8 digits, a new node is created with those digits. The nodes are stored in such a way that the head of the stack is what is normally considered the end of the number. This ensures easy access for the next round of the factorial. Further technical explanations of this process can be found in the code comments.

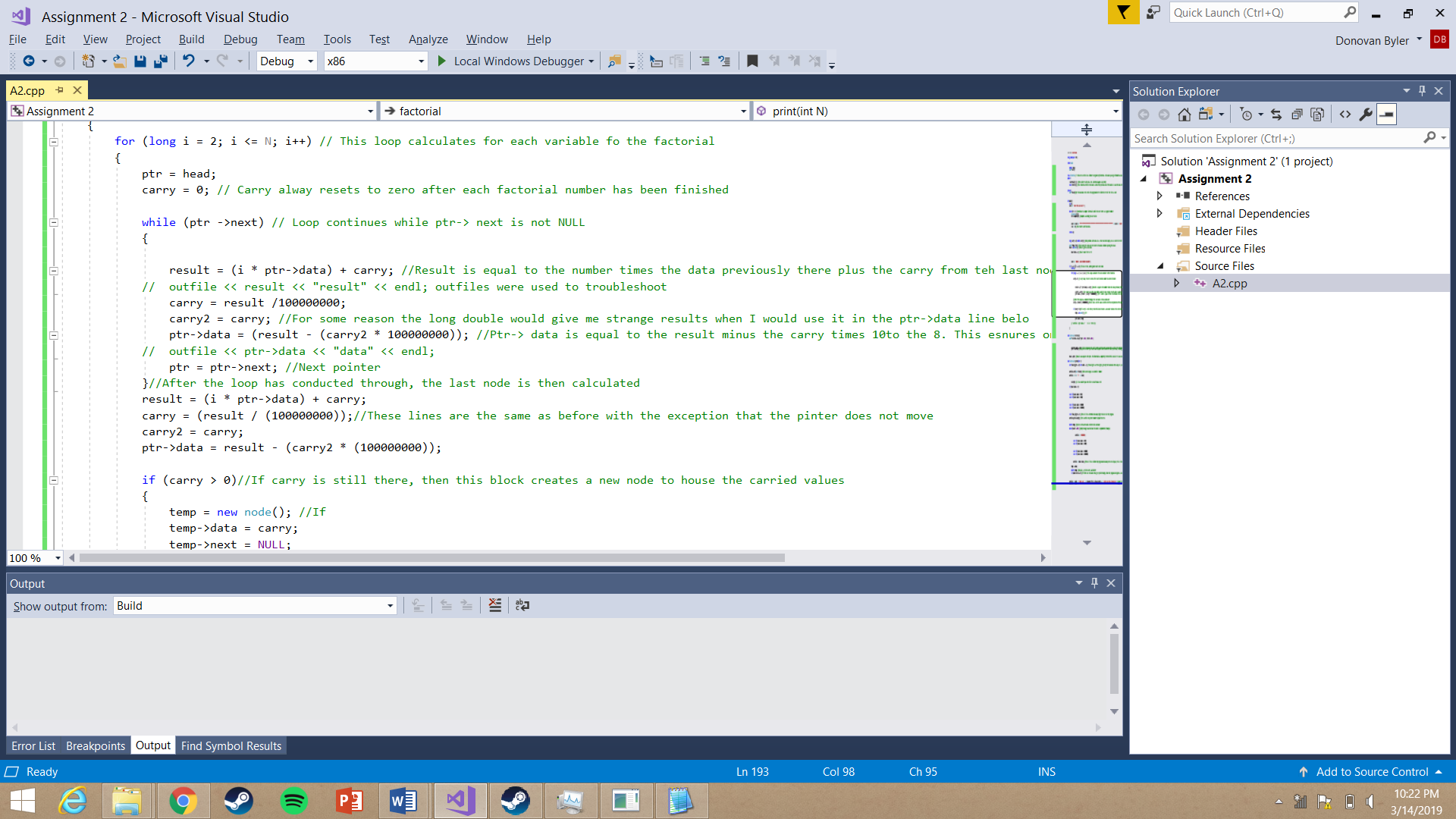


Figure 0.1: Compute Code

The reverse function reverses order of the nodes so that the nodes can displayed in a sensible manner. I used three pointers to leapfrog my way backward into the code, switching the next pointer as I went. I also counted each node as I reversed this.

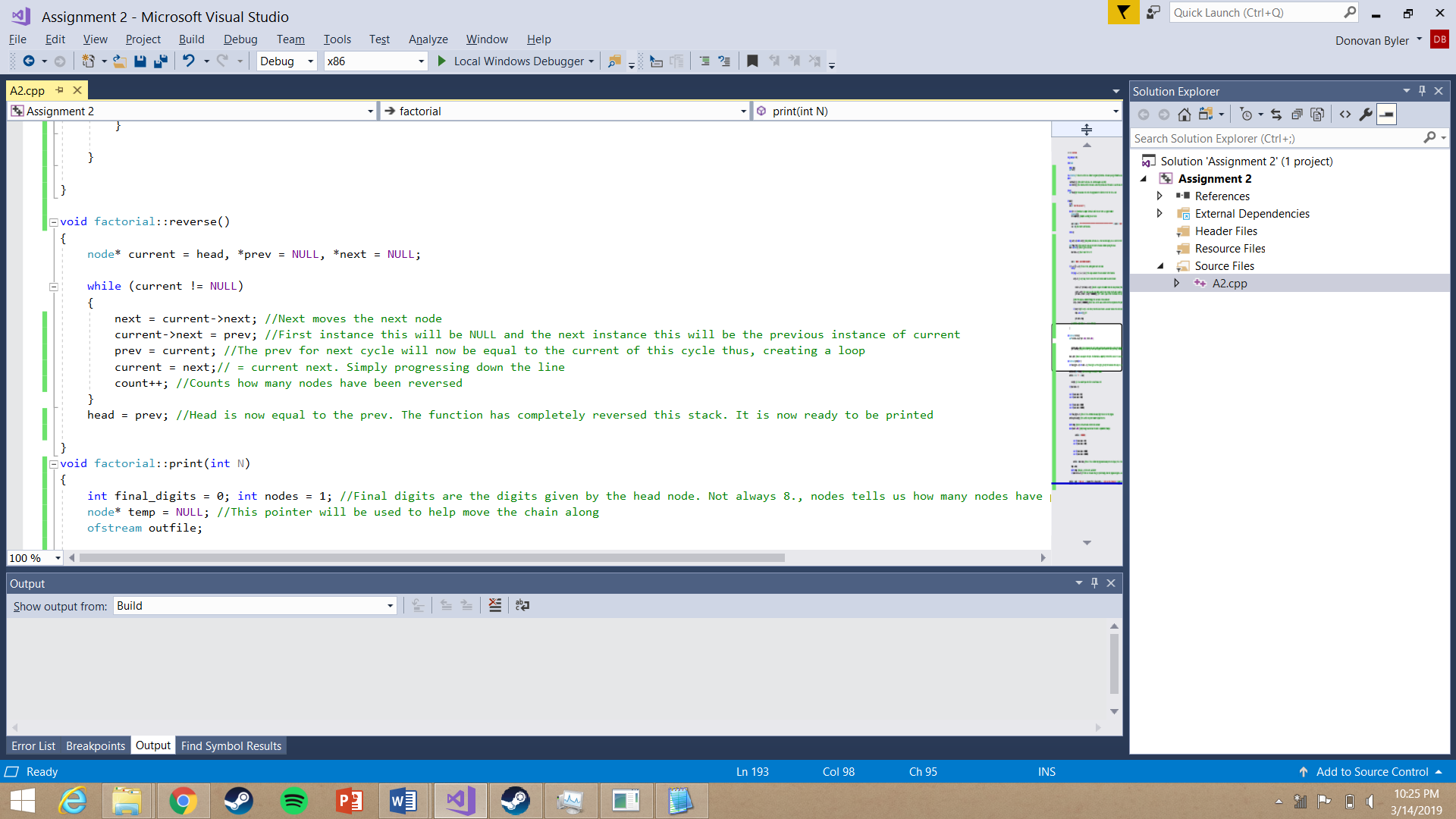


Figure 0.2: Reverse code

The final function is the print function. I used a series of if-else statements to ensure that a proper 8 digits was printed for every node, excluding the first. This accounted for nodes which began with a zero. After printing each node and moving toward the next, I deleted the printed node. After printing 10 nodes, I created a new line to ensure that only 80 characters were displayed per line. In addition, I also created a formula which displayed the number of digits within the solution. The following images are proof that my program successfully calculated these factorials.

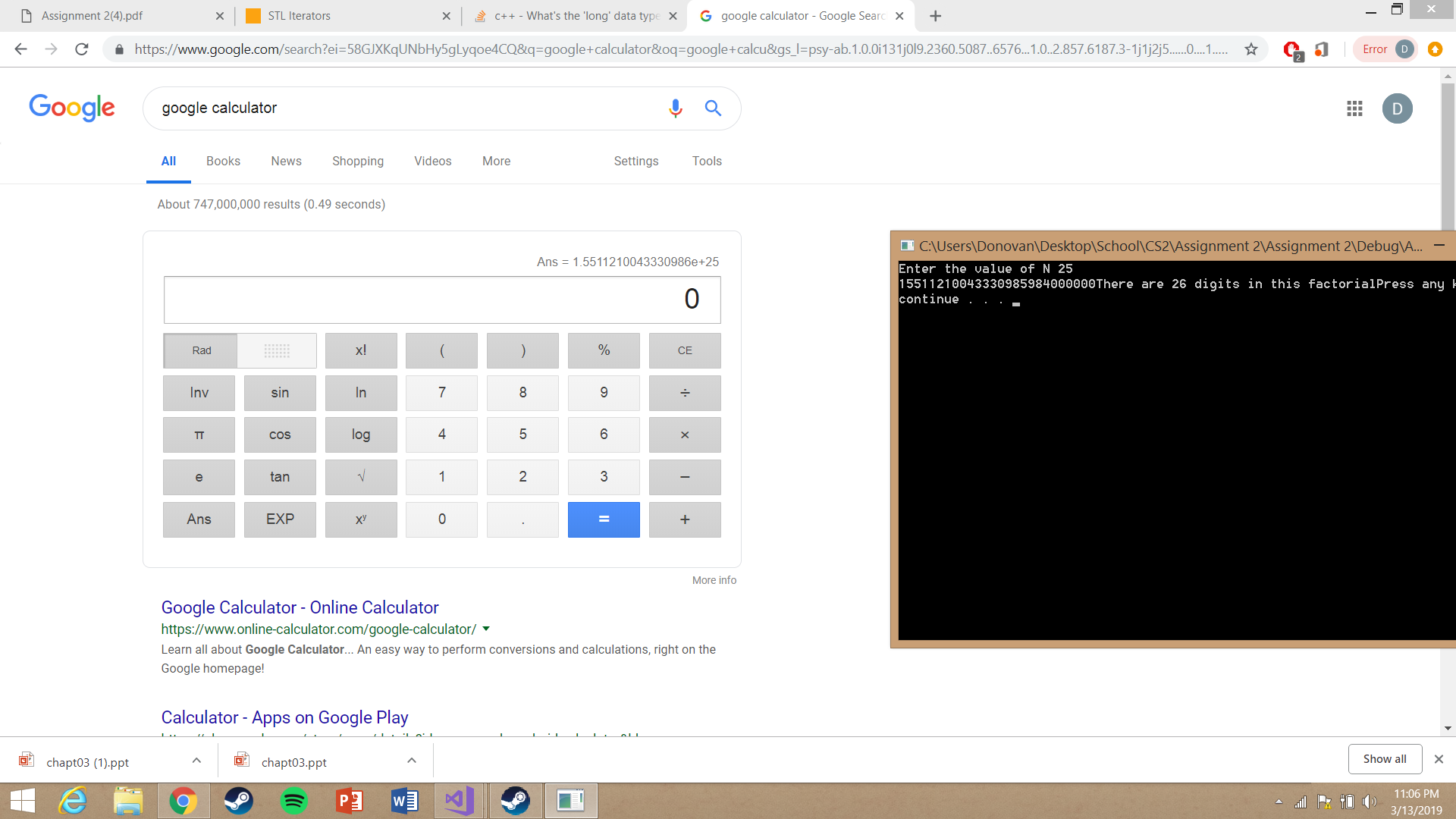


Figure 1

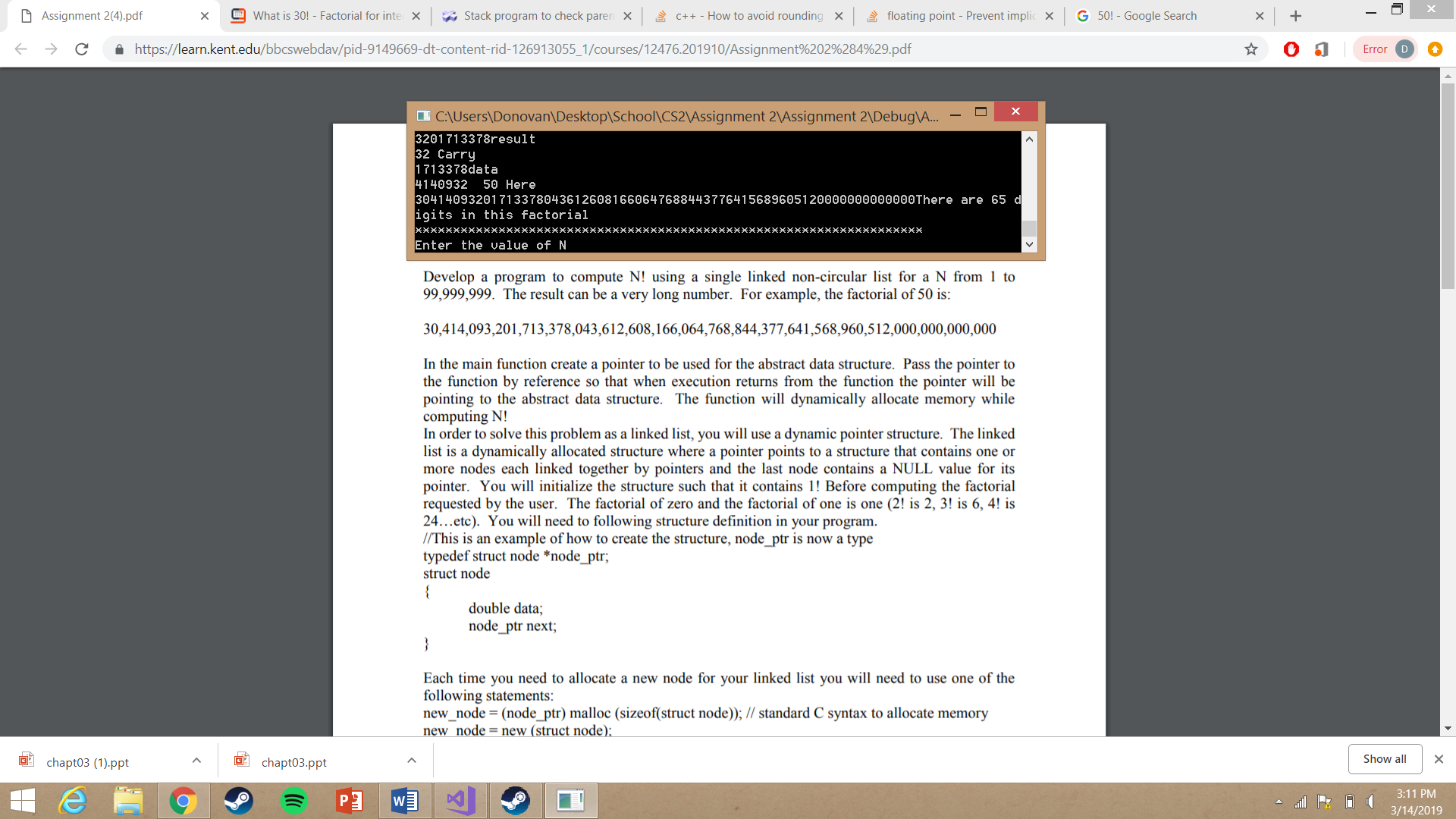


Figure 2

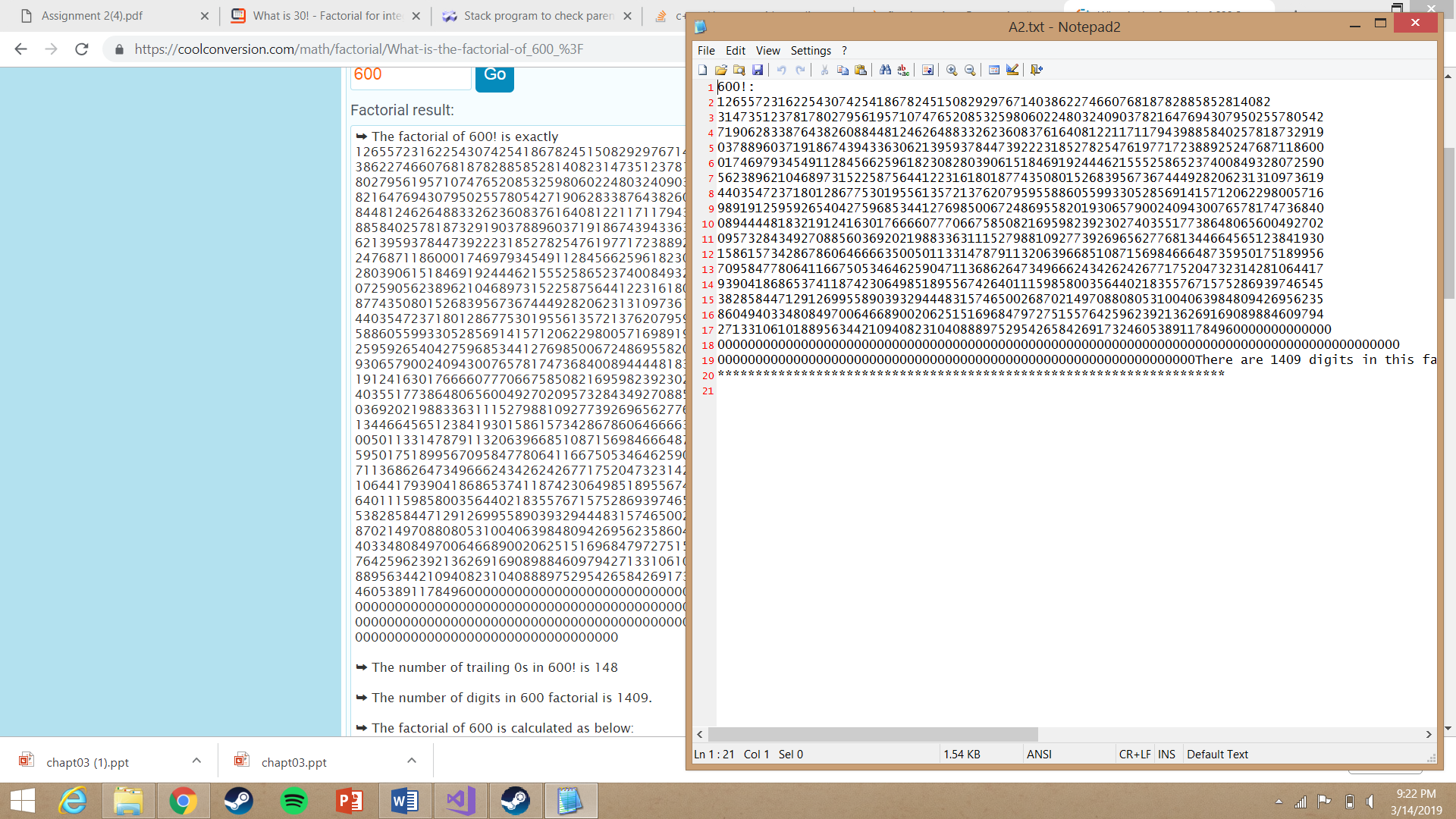


Figure 3: Figures 1-3 are examples of how my program successfully calculated large factorials.

The largest factorial I computed was 2,000,000. This took nearly ten hours. I suspect my program could calculate a larger number, but I simply do not have the time to sit and wait for it to do so. Each of my computations are found in the Factorial.out file with each result appended to the next.

