**Debugging Code**

**LAB 4**

**SECTION 6**

**Kenneth Schueman**

**SUBMISSION DATE:**

**9/30/2021**

**9/23/2021**

# Problem: Compiler Errors

1. Now do the same for **lab04-1\_2.c**, **lab04-1\_3.c**, **lab04-1\_4.c,** and **lab04-1\_5.c.** Take a look at the output from the compilation errors and try to fix the issues.

# Analysis

Various types of compiler errors that are easy to make when you don’t check your code often.

# Design

Find the code and fix it.

# Testing

Testing one line at a time, corrections can be seen below.

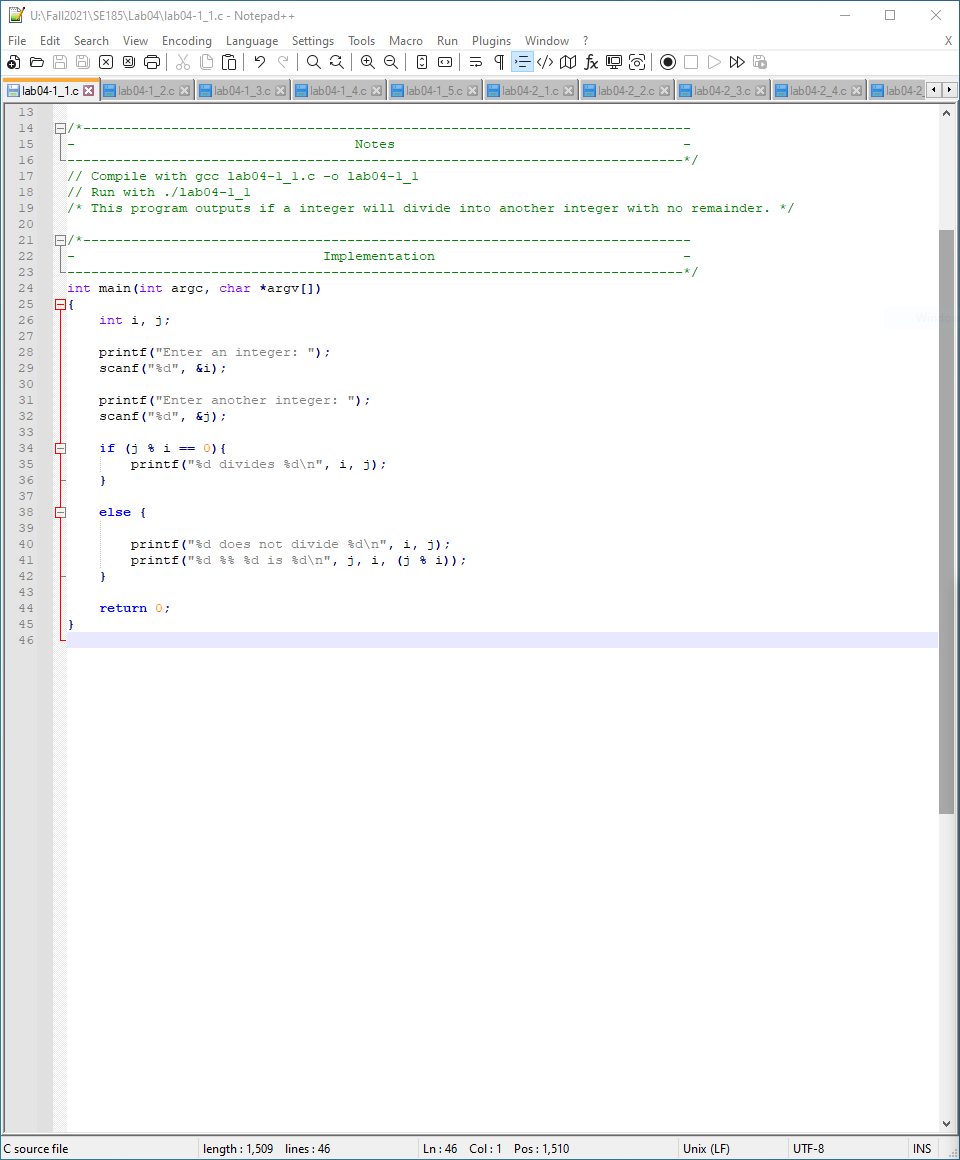
# Comments

The final few gave me issues and program 4 had an arithmetic error.

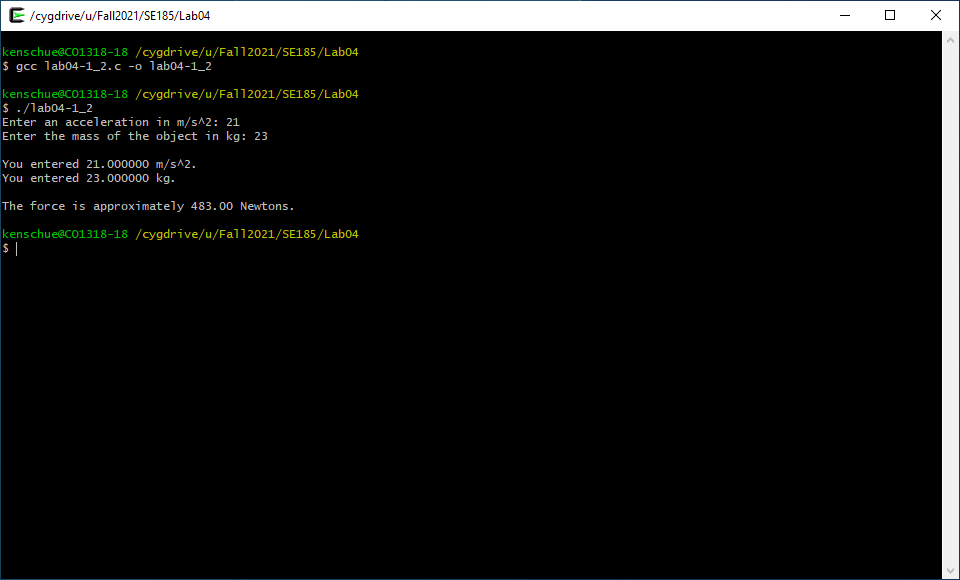
# Screen Shots



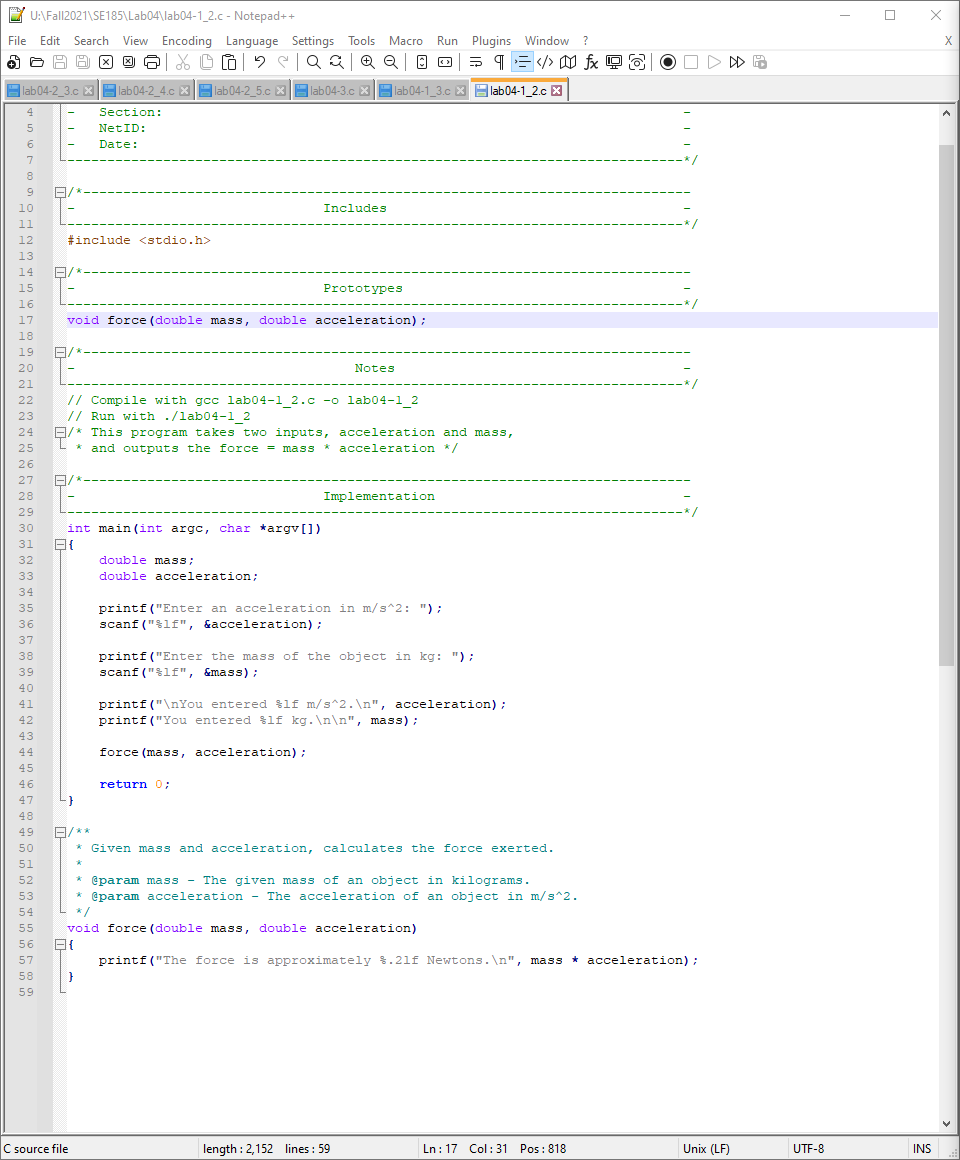
Figure



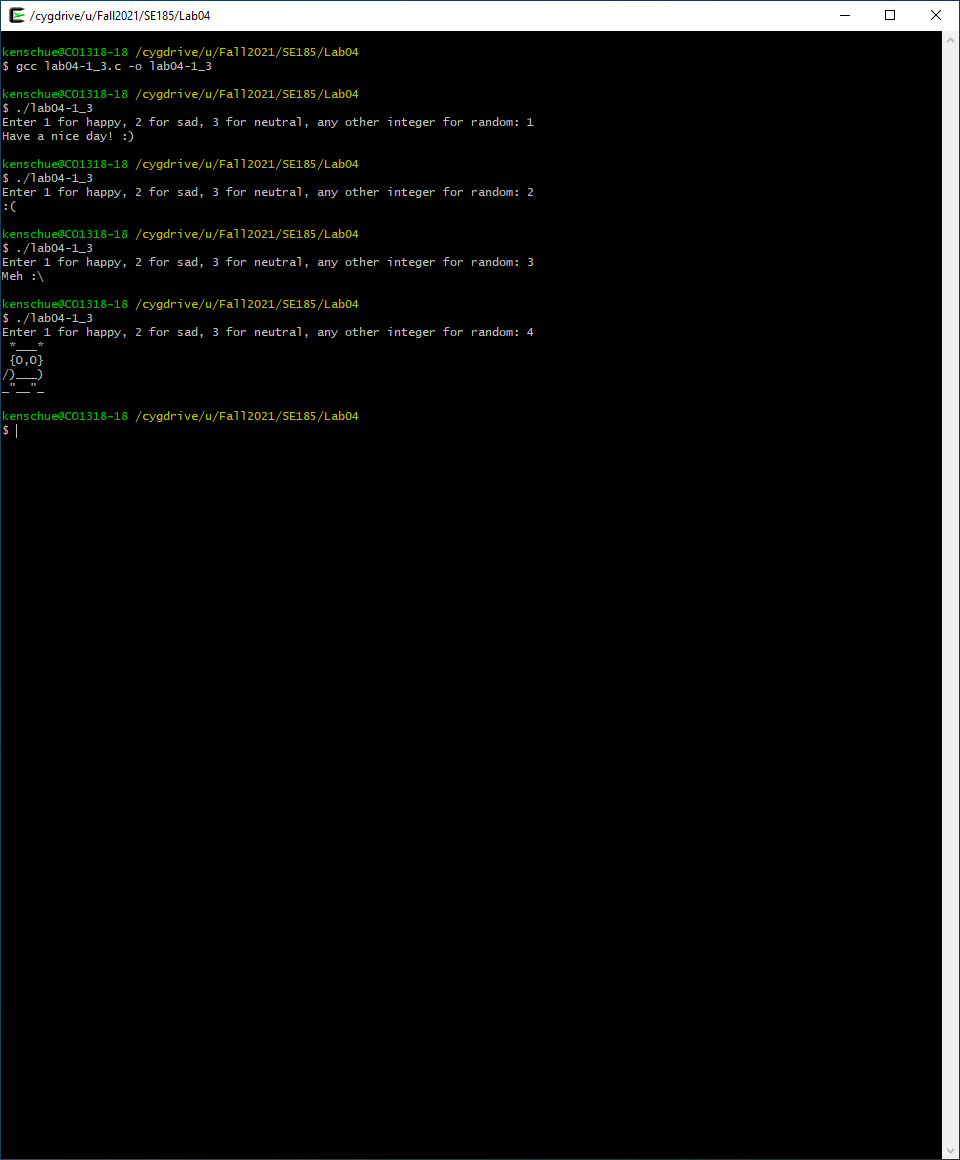
Figure



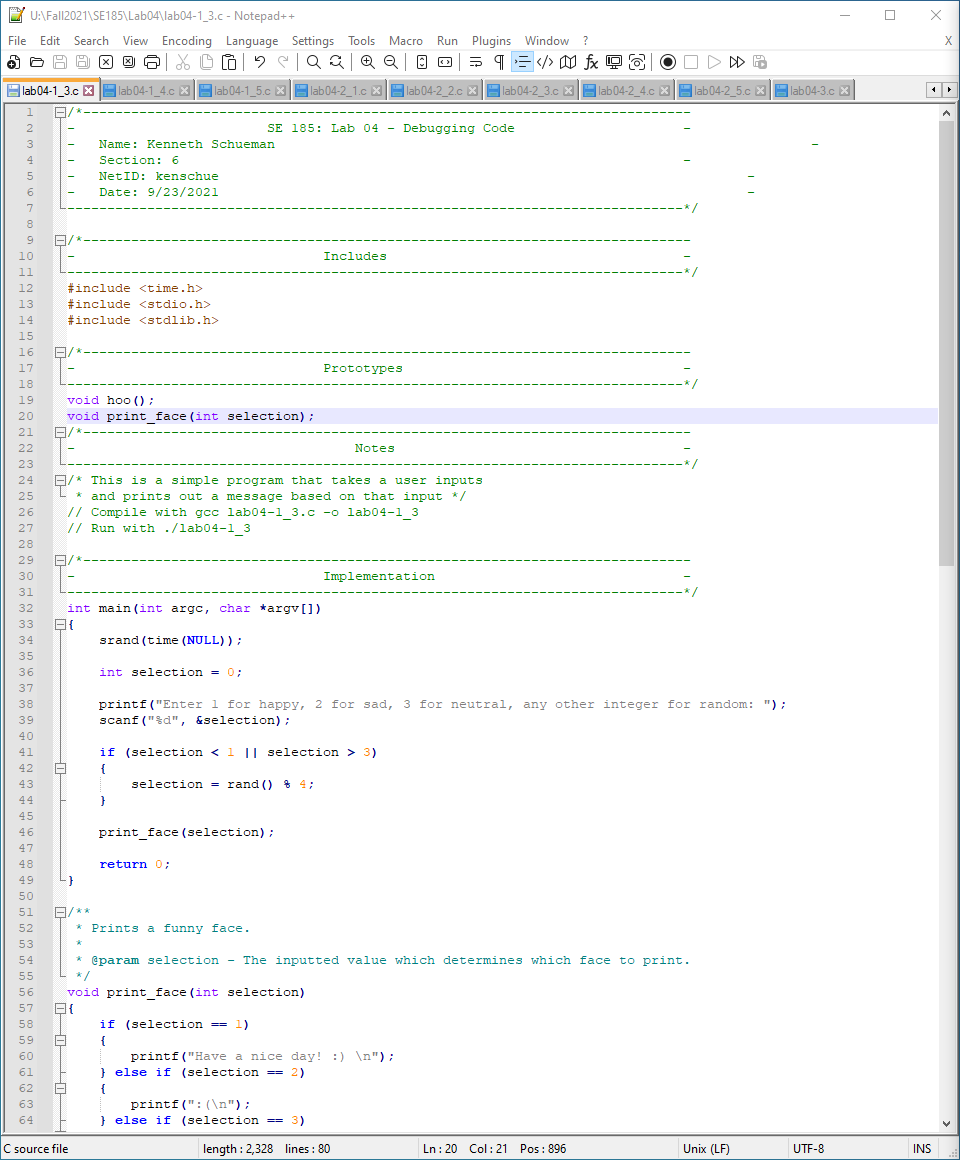
Figure



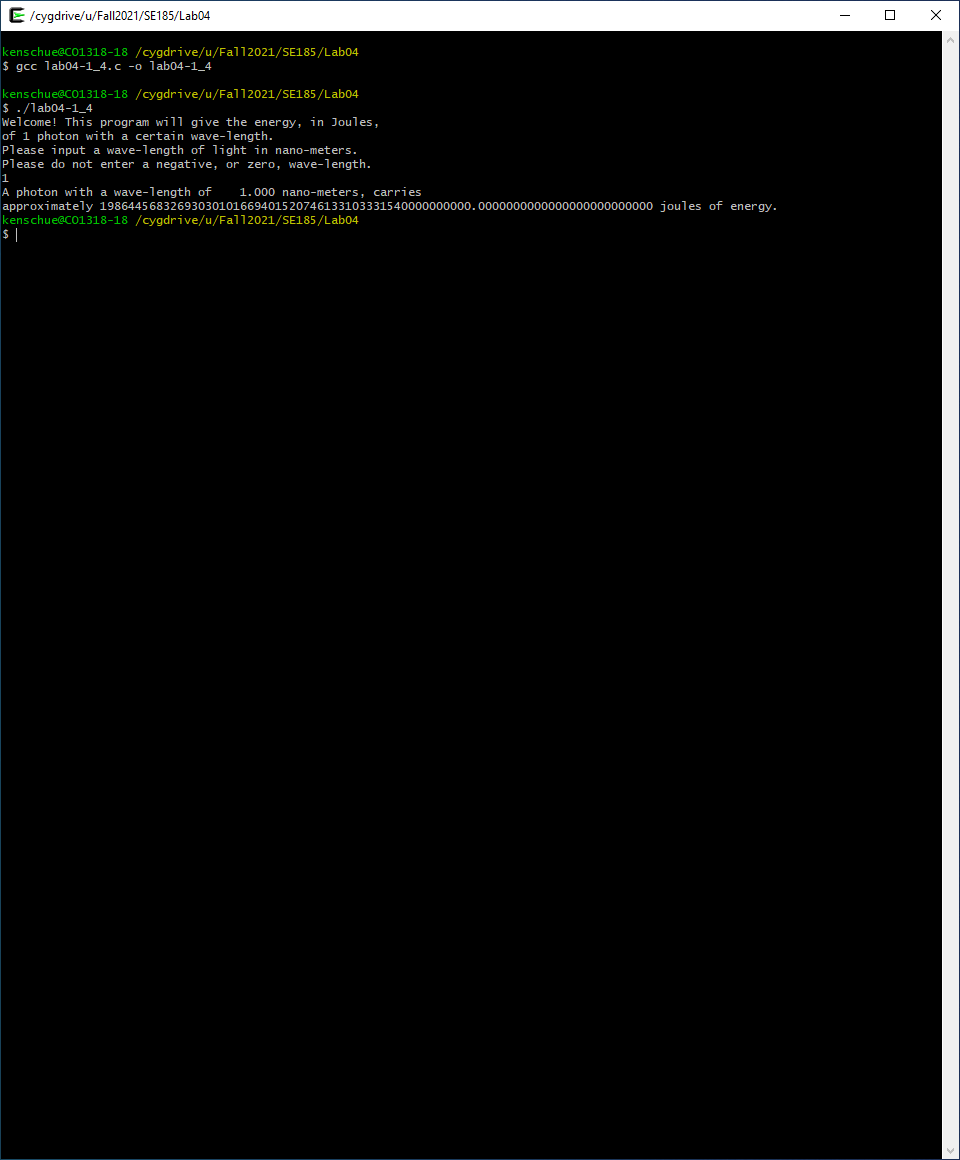
Figure



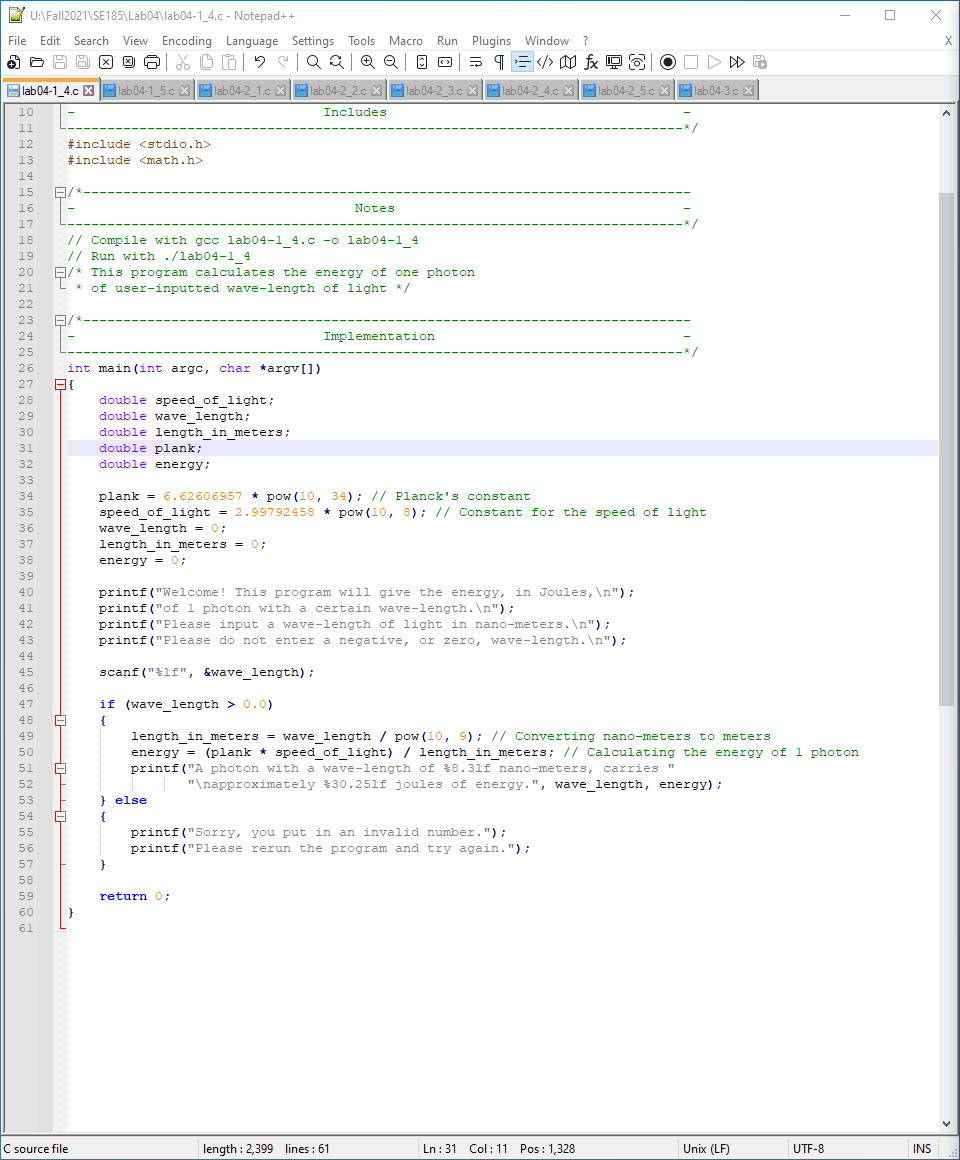
Figure



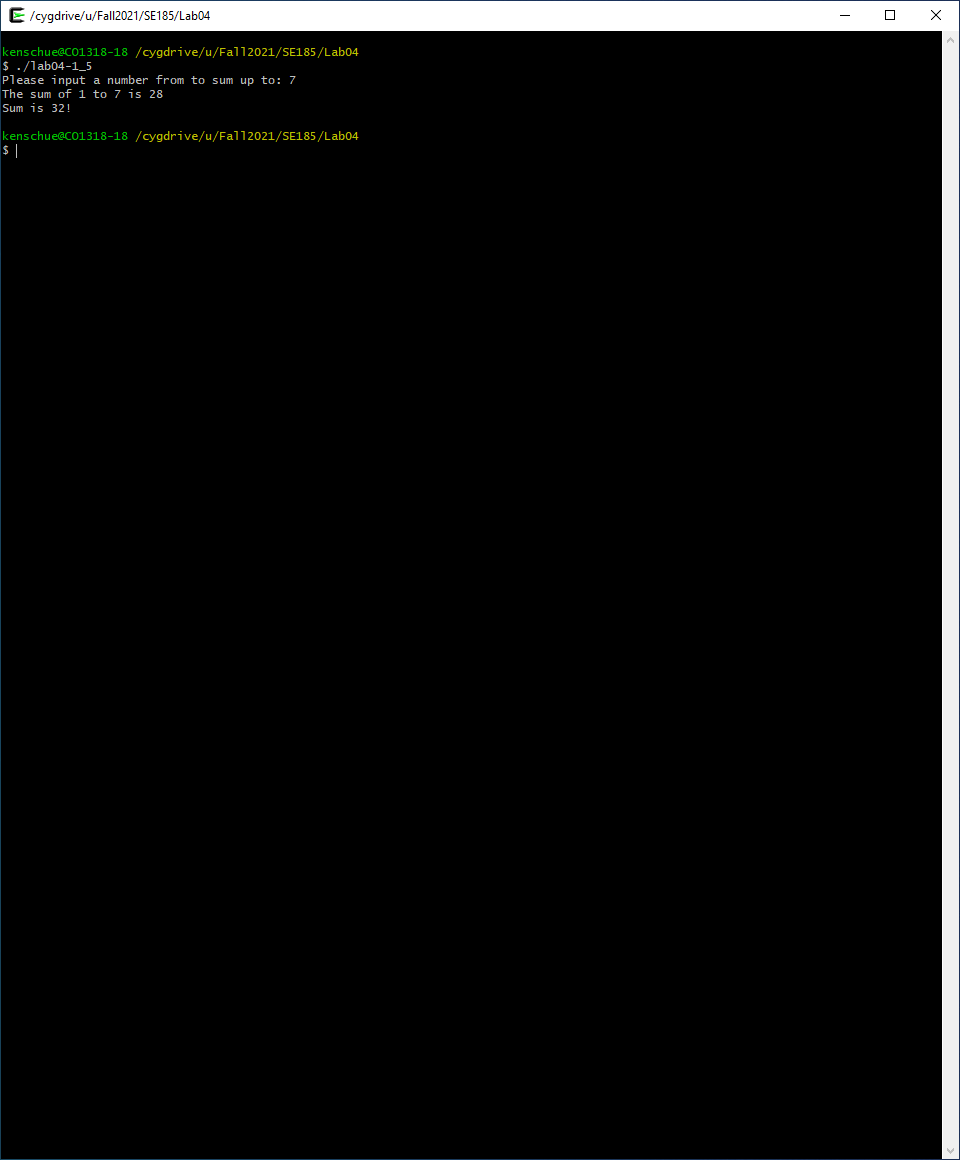
Figure



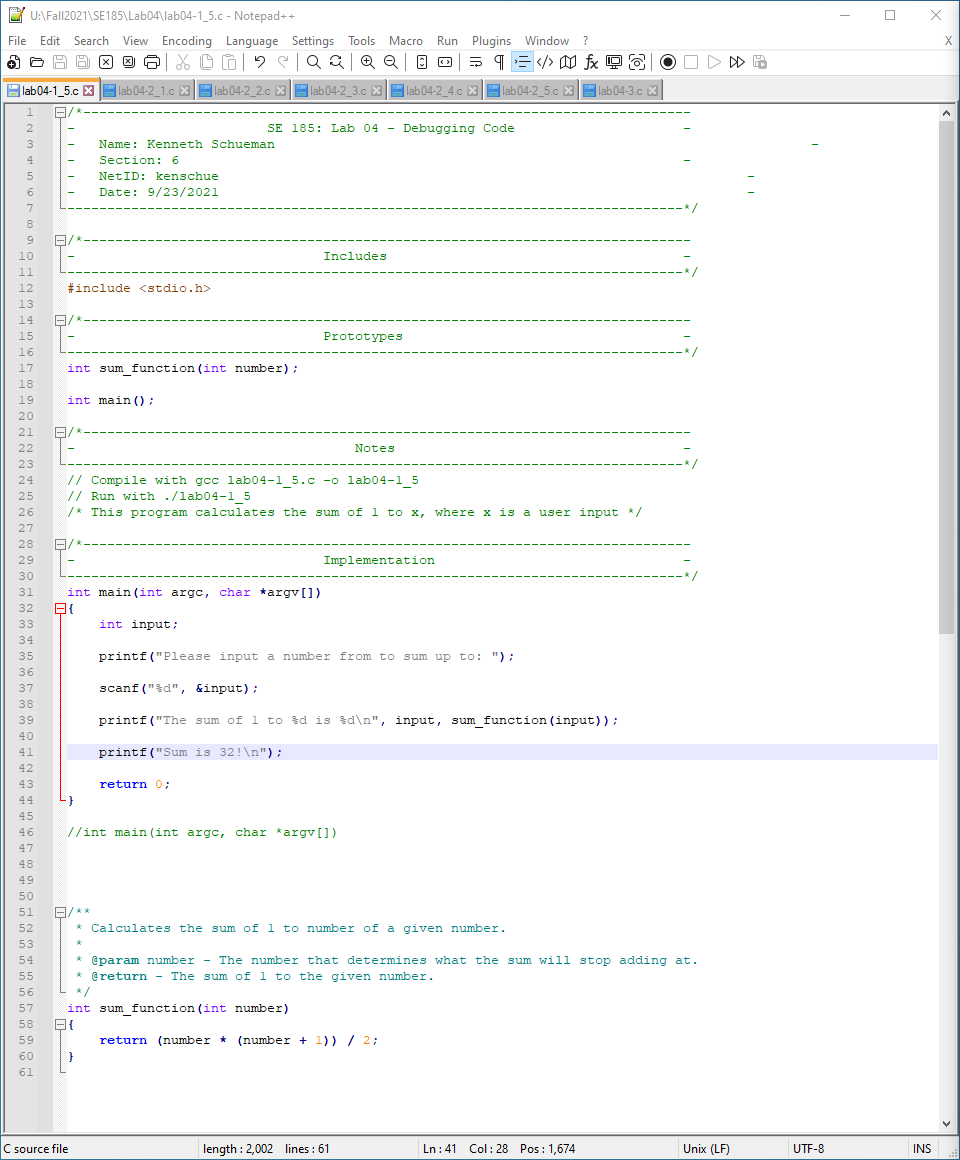
Figure



Figure



Figure



Figure

# Problem: Unintended Results

1. Sometimes your program will compile successfully, but it doesn’t behave or give you the output you may expect. These kind of problems are a bit more difficult to find and fix.
2. Compile and run **lab04-2\_1.c**. If you run it a few times, you may notice the output isn’t always correct for the number you input. Open **lab04-2\_1.c** and try to find where the problem is coming from. Once you find the problem(s), comment out the line and add the corrected line of code below the commented line. Compile, run, and make sure the program has been corrected.
3. Do the same for **lab04-2\_2.c, lab04-2\_3.c, lab04-2\_4.c,** and **lab04-2\_5.c.** There are comments at the top of each program that describe what the program should do.

# Analysis

These set of problems were much more difficult then the first set but once the error was found it was only a matter of testing to make sure the math checks out.

# Design

Most problems only had one error hence only one test and math check.

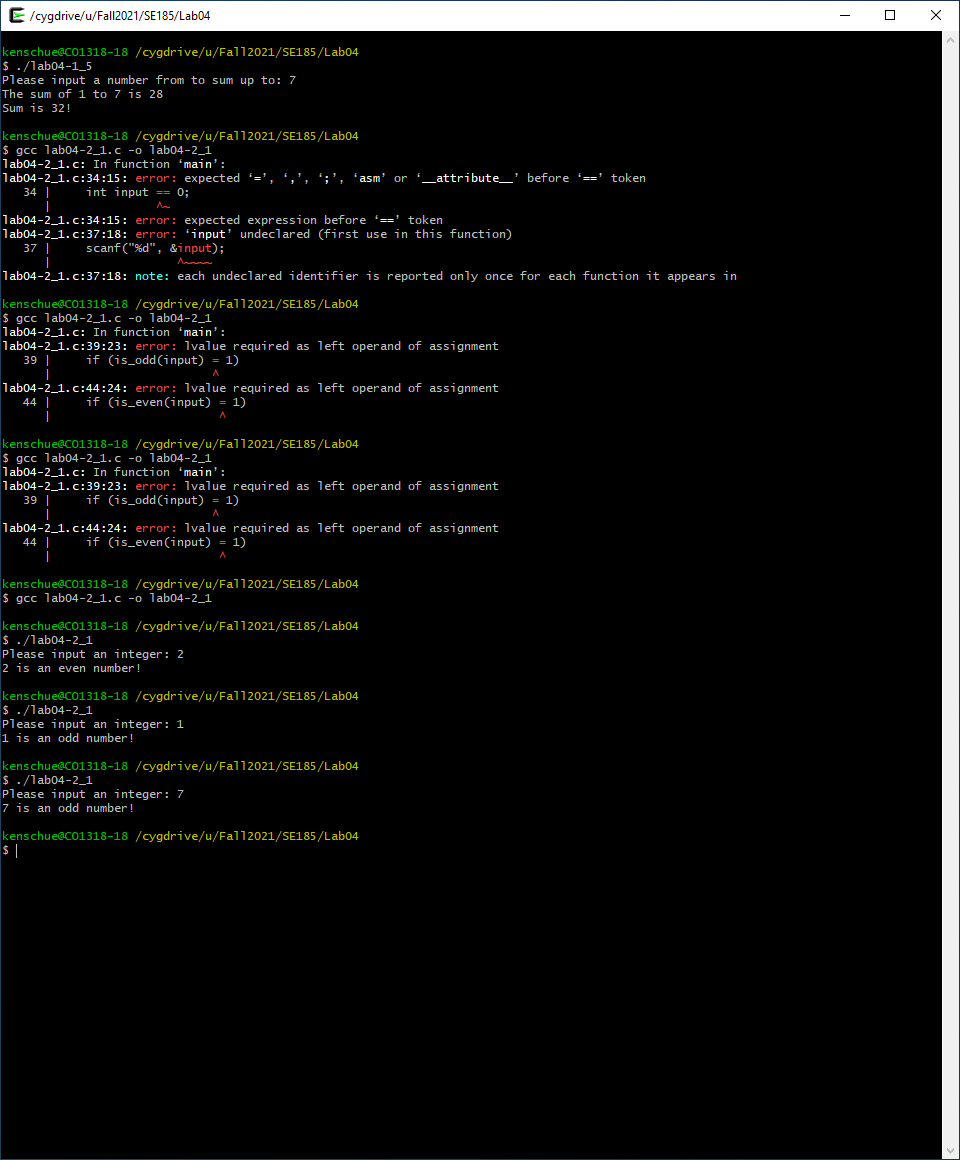
# Testing

Compile and check the math.

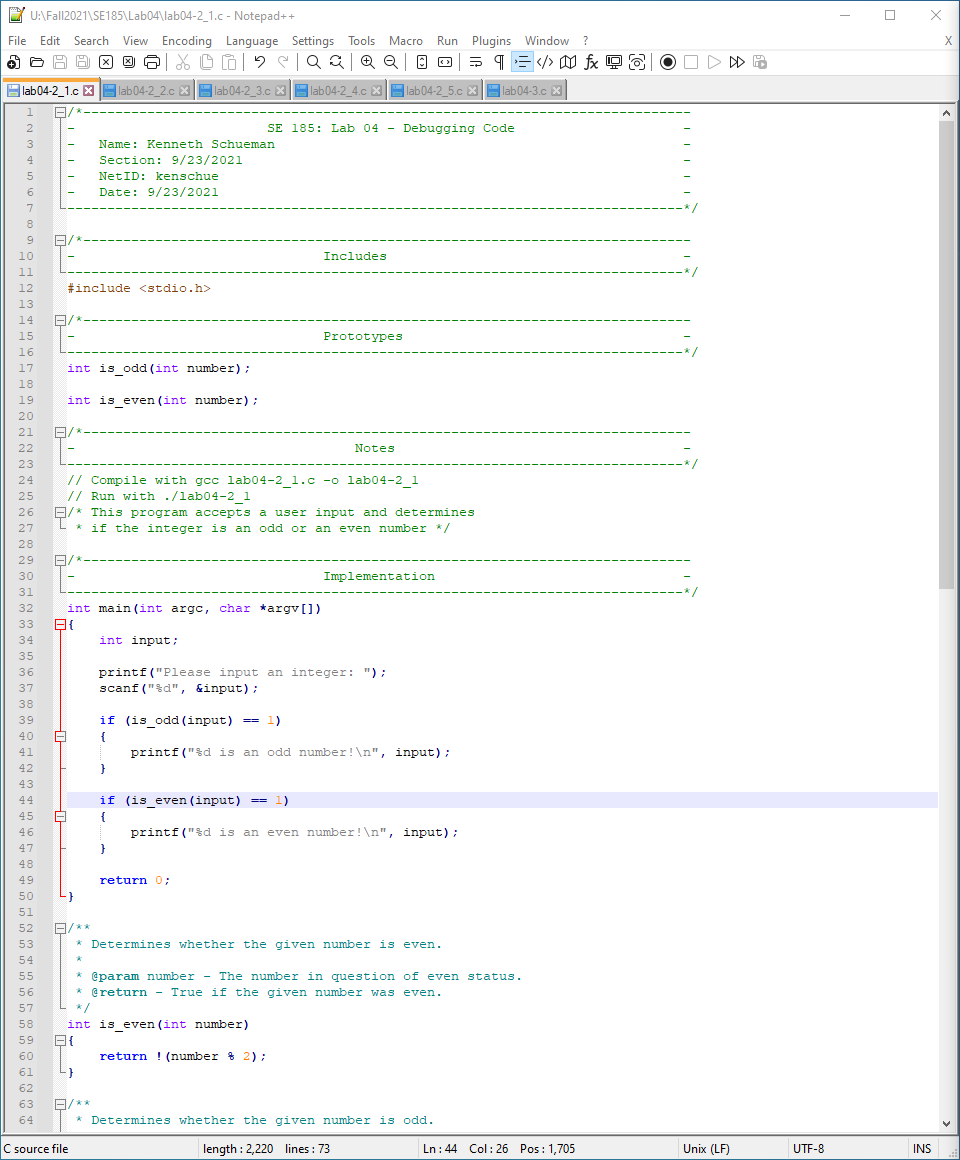
# Comments

The final few gave me issues and but one of TA’s was able to help.

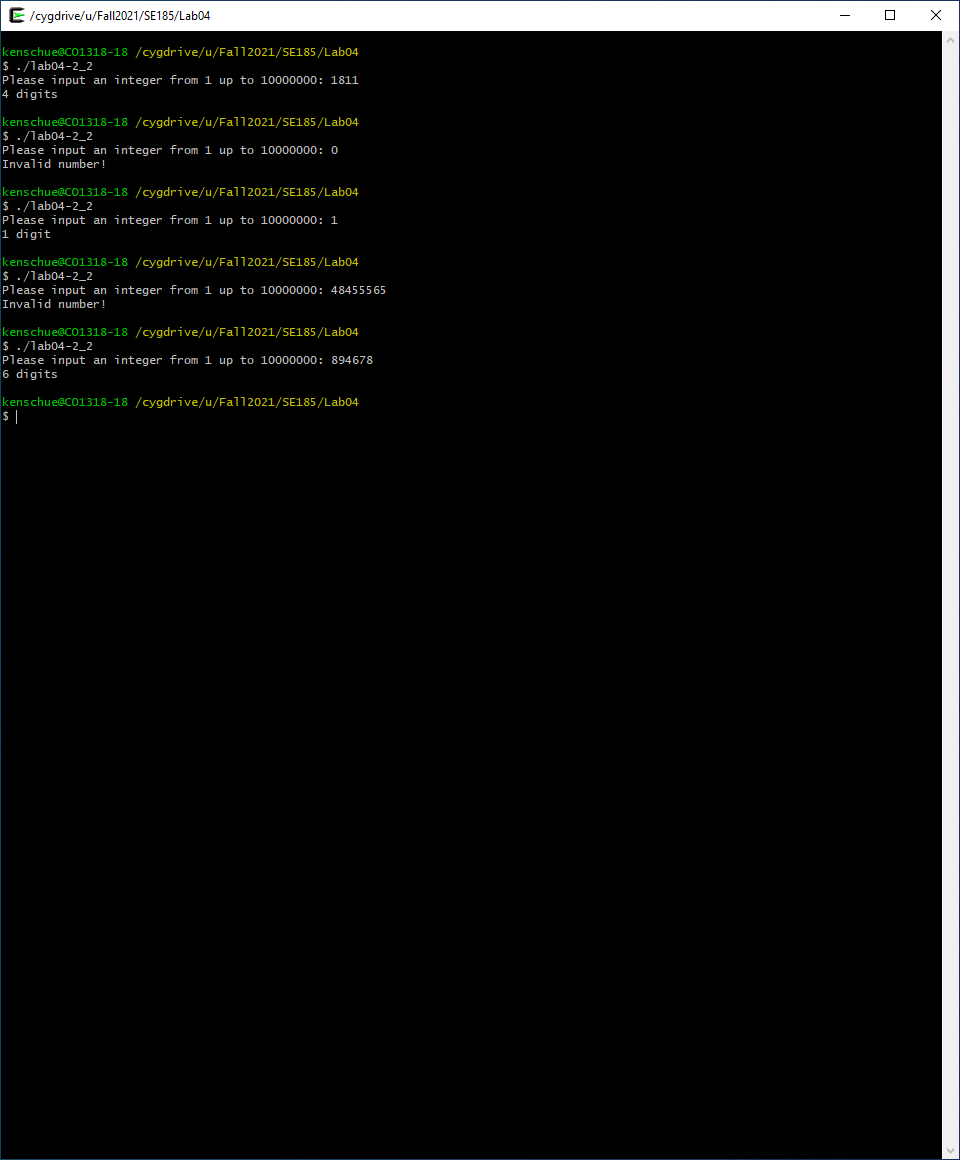
# Screen Shots



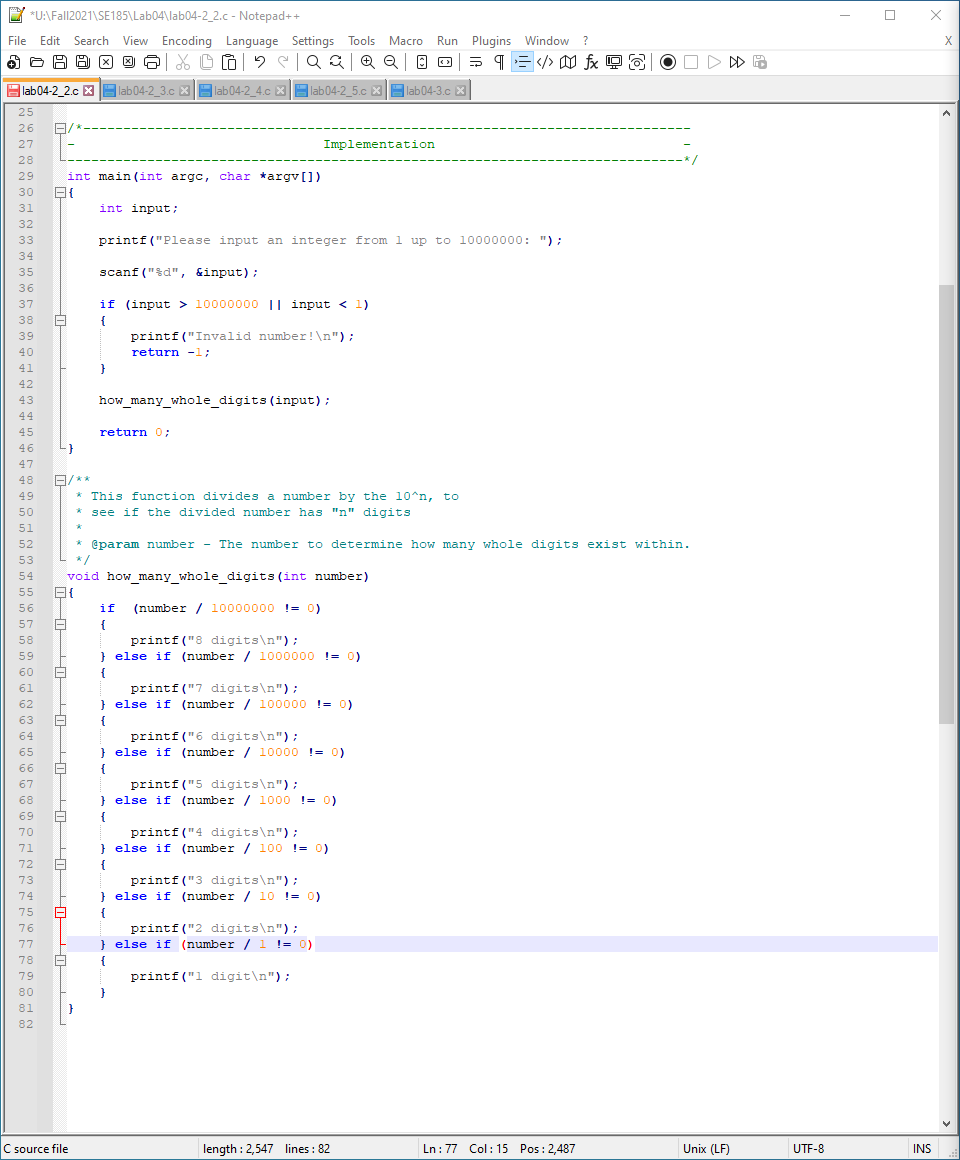
Figure



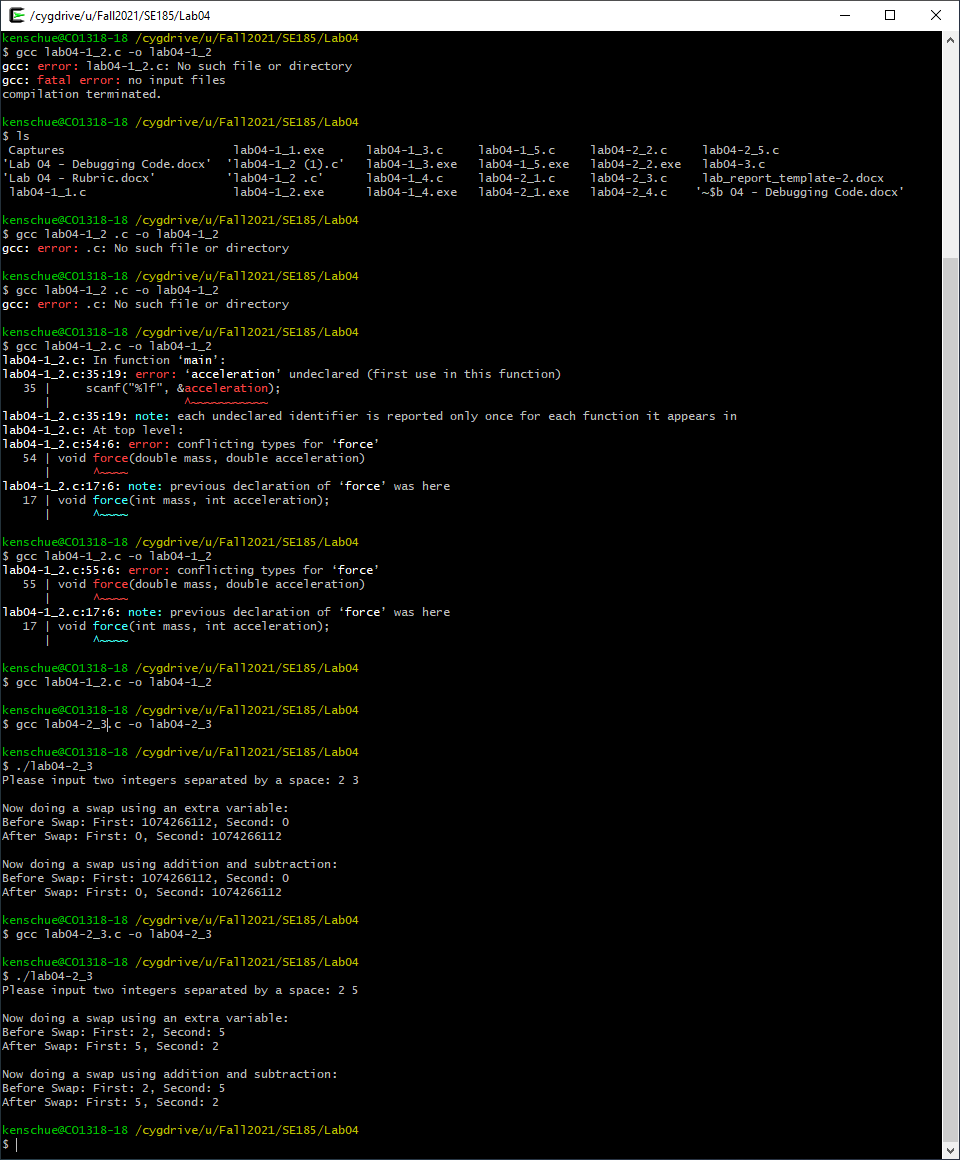
Figure



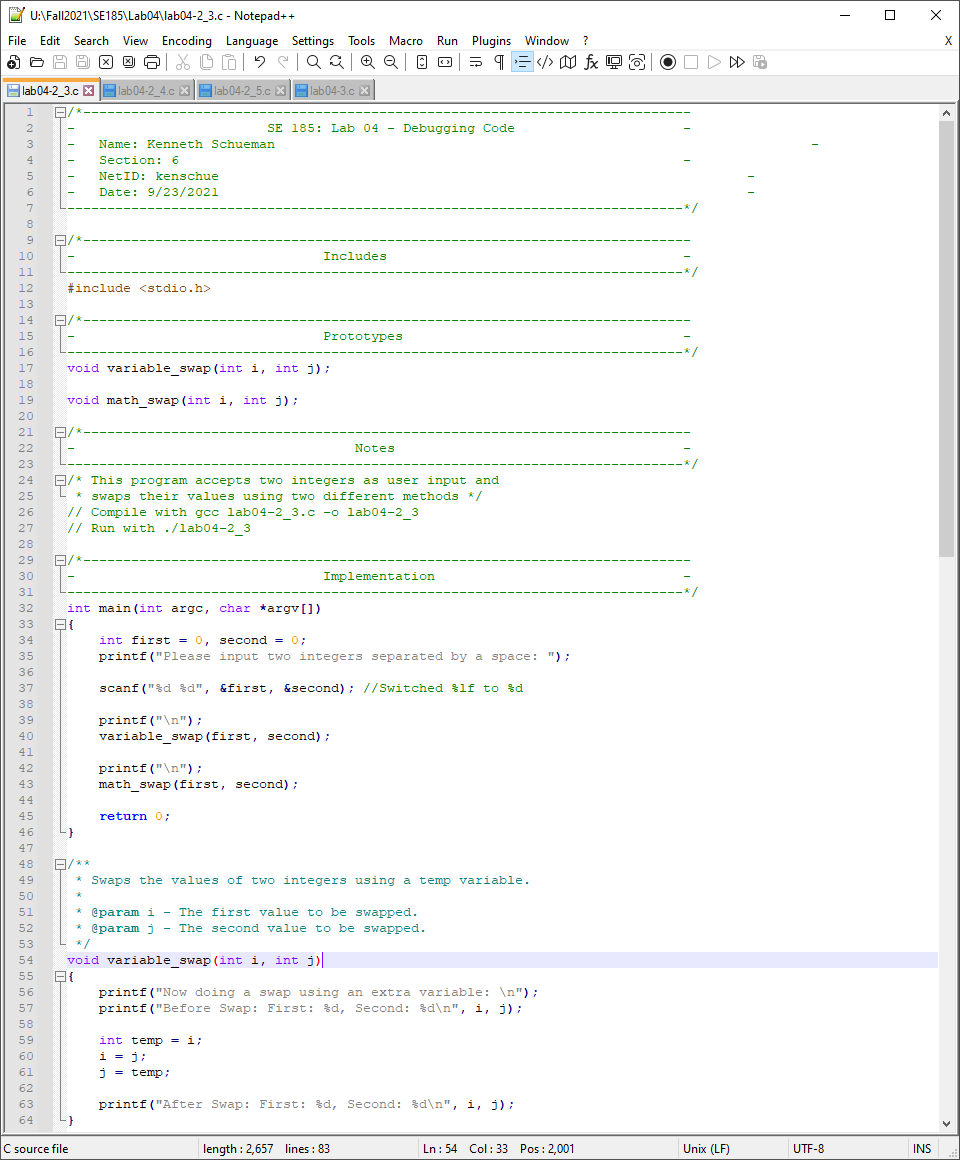
Figure



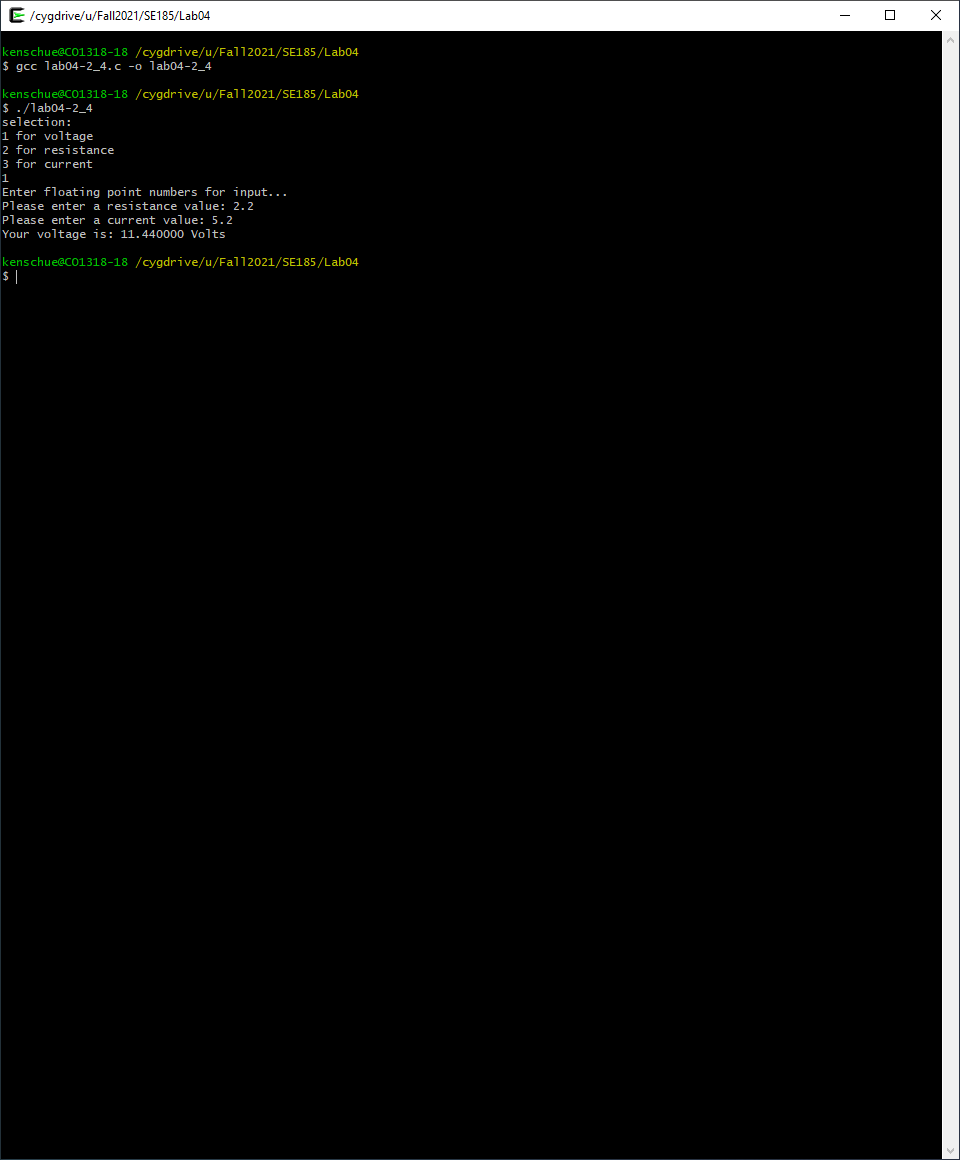
Figure



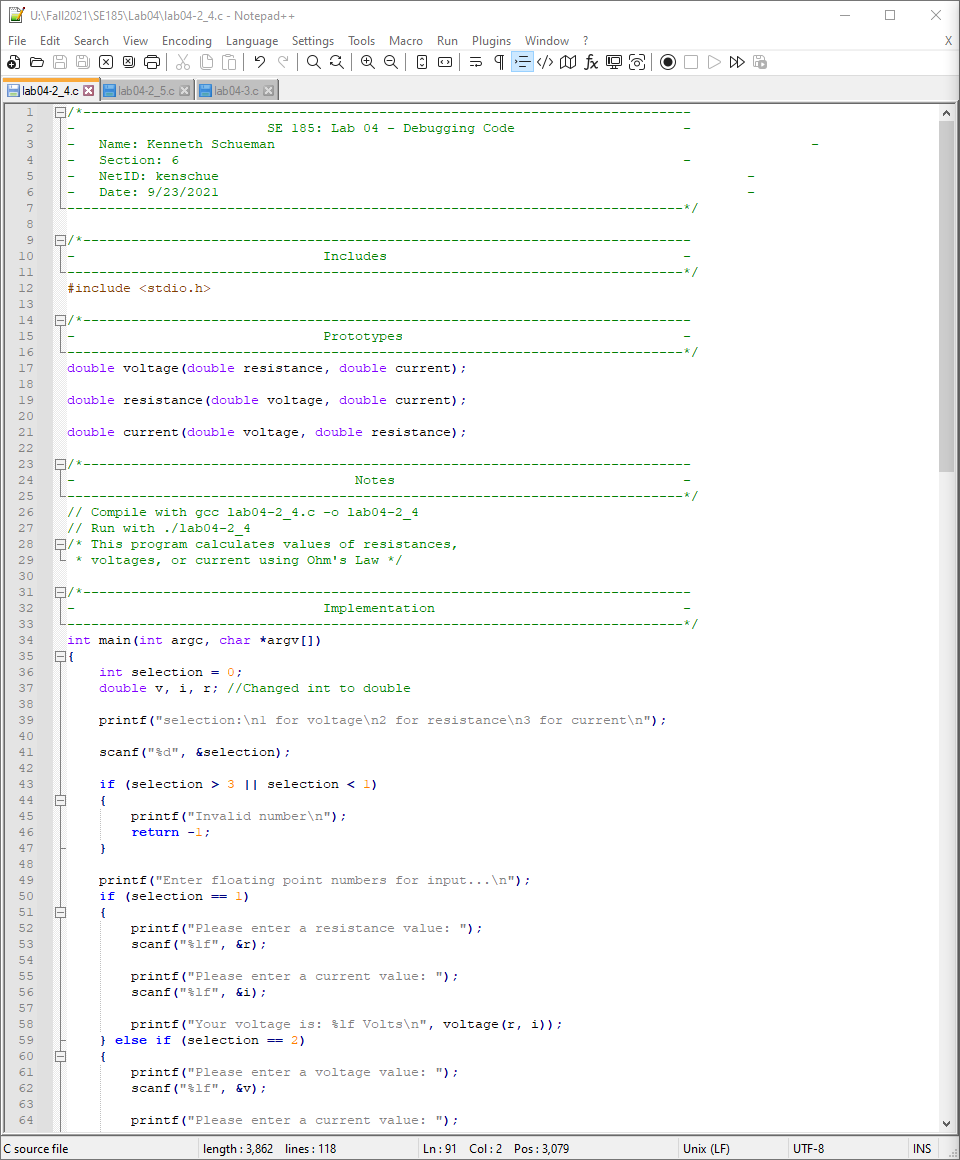
Figure



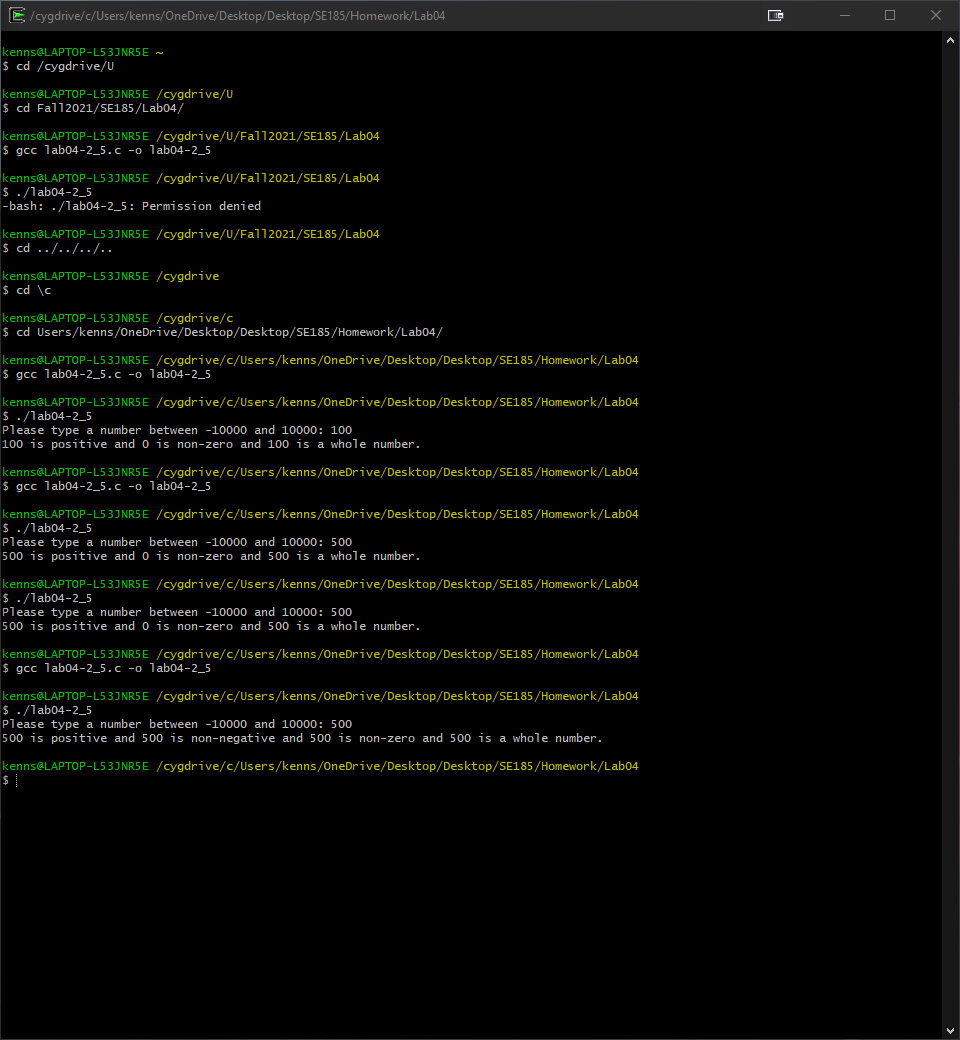
Figure



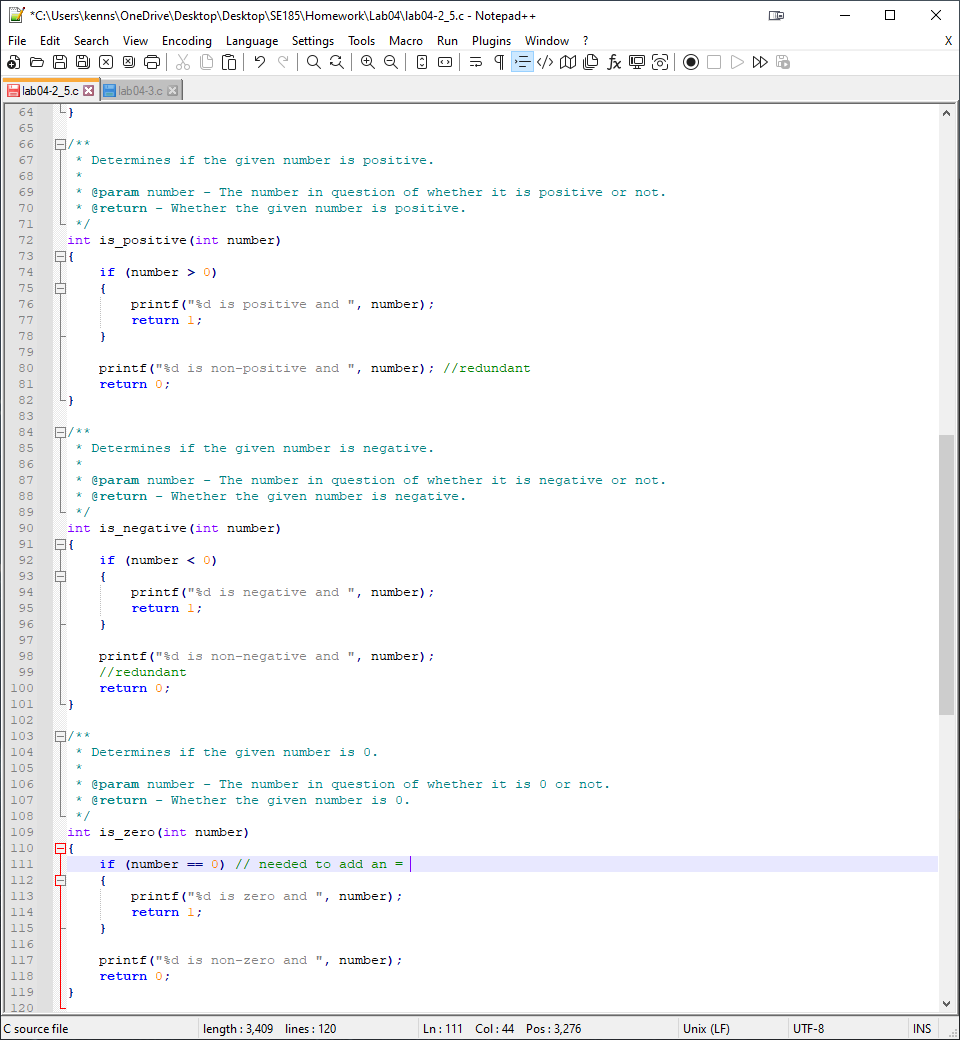
Figure



Figure



Figure



Figure

# Problem: Putting It All Together

**For each** program in **Part 1 and 2** that you fixed, answer the following:

1. What changes did you have to make to fix the program? Please list the line number with the changes that you made.
2. What kind of issue caused this problem?
   1. Example: missing semicolon, wrong variable types, missing brackets, etc.

**For Part 3:** What is the purpose of the **“-Wall”** flag? Do you have to fix all of the messages that it gives you?

# Analysis

There were a lot of small mistakes that caused a lot of issues for me to find

# Design

The person who made this problem was evil but once added the correct declarations and syntax the program ran well.

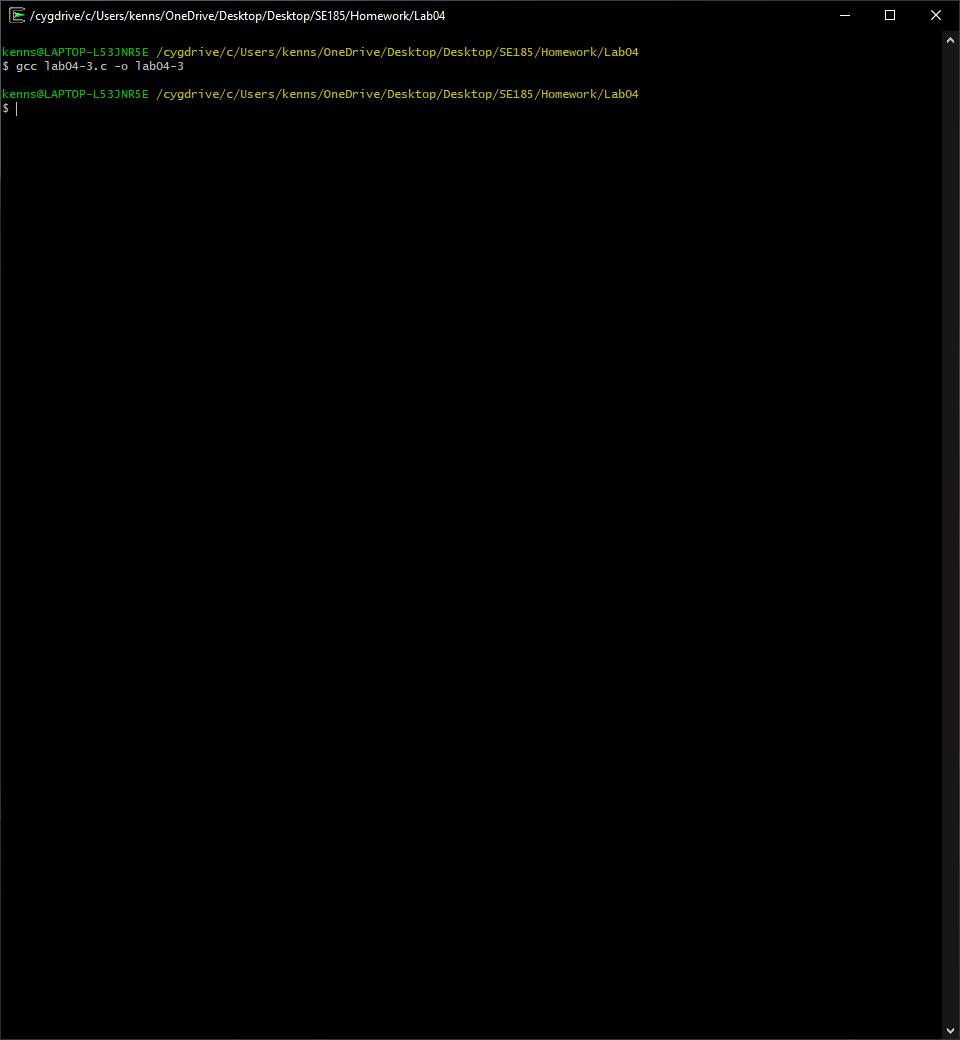
# Testing

-Wall was very useful to find the multiple errors in the program.

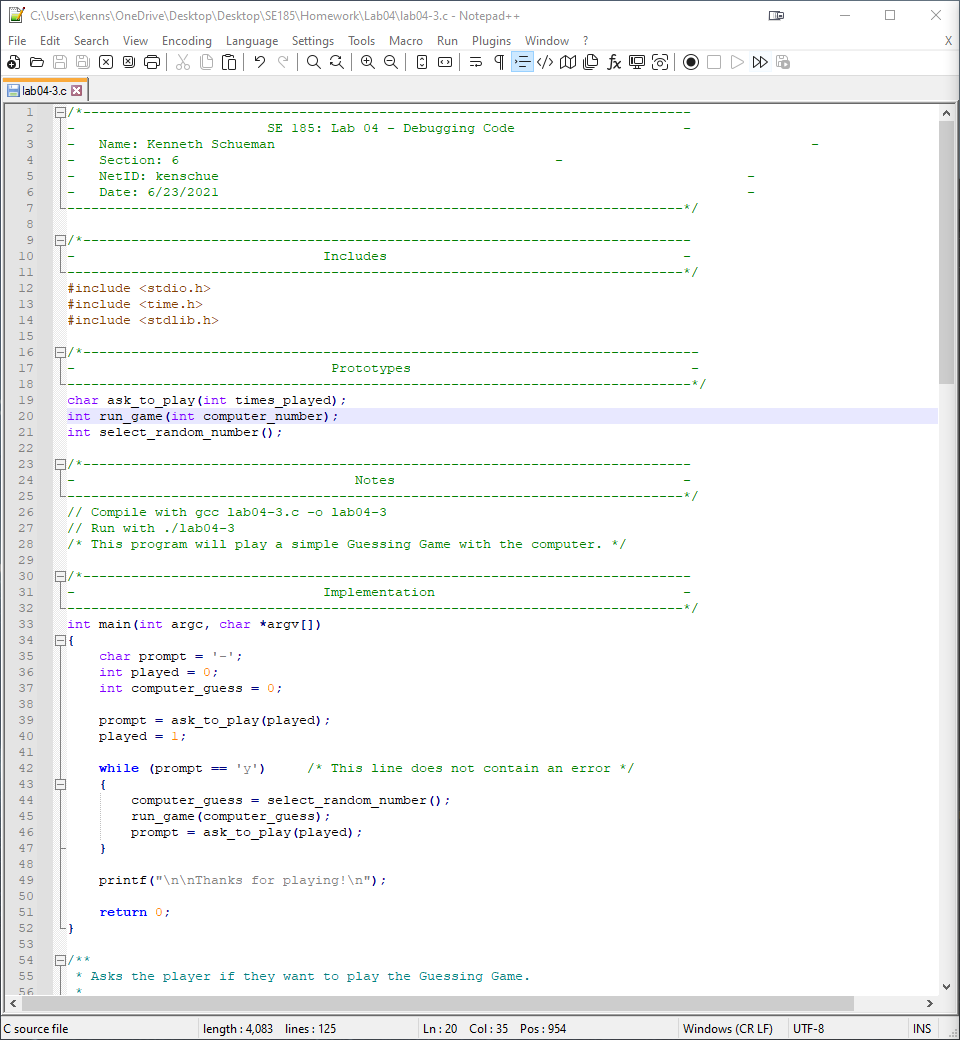
# Comments

This problem took me a very long time.

# Screen Shots



Figure



Figure