**TITLE**

**Debugging Code**

**LAB # 04**

**SECTION # 08**

**FULL NAME**

**Adam Jennissen**

**SUBMISSION DATE:**

**09/29/2022**

**DATE**

**09/27/2022**

# Problem

The purpose of this lab was to learn about the C compiler messages, become familiar with various types of compiler errors, and to learn coding practices to help avoid unintentional errors. The lab had us debug a total of 11 programs, 5 had compiler errors, 5 had logic errors, and the last one had a combination of both.

# Analysis

In order to successfully do this lab, I had to figure out what the program was supposed to do and then figure out why it was not doing that. The compiler errors were straight forward, the compiler mostly explains what is wrong. The logic errors require a bit of thought about what it is supposed to do and then what it is currently doing.

# Design

When doing the lab, I had to edit lines to their correct form and figure out why they did not work.

1. 1\_1. On line 33 I added in the second “ mark in order to tell the printf function where it should end. On line 35 I added a ; to the end of the line, so that the compiler knows it is the end of that line. On line 43 I added a { to tell the compiler what is part of the else function. On line 46 I added the n into the pritf to make it printf which is the correct function.
2. 1\_2. On line 35 I added double acceleration; I added this so that the acceleration variable was declared.
3. 1\_3. On line 13 I added #include <stdio.h> so that the standard input and output library can be used. On line 14 I added #include <stdlib.h> so that the rand function can be used. On line 20 I added void print\_face(int selection); so that the main function can call the print\_face function.
4. 1\_4. I had to change the names of 3 of the variables throughout the entire program. I changed speed\_of\_light! to speed\_of\_light because variable names cannot include !. I changed wave-length to wave\_length, because variable names cannot include -. I changed ~length\_in\_meters to length\_in\_meters, because ~ cannot be used in variable names. I changed 0energy to energy, because variable name cannot start with a number. On line 35 I had to move const to before double, because that is the proper way to declare a constant variable.
5. 1\_5. On line 19 I removed the prototype for the main function, because it is not needed. On line 44-47 I removed the function, because you can only have one main function.
6. 2\_1. On line 35 I removed the second =, because you use a single = to assign a value. On line 40 and 46 I added an = because you use == to check if 2 values are equal.
7. 2\_2. On lines 57, 62, 67, 72, 77, 82, 87, and 92, I removed the (double), because you do not need to keep the variable as an int in order to figure out if it has a remainder or not.
8. 2\_3. On line 38 I changed both of the %lf to %d, so that it took the correct type of input.
9. 2\_4. On line 38 I changed the in to double, so that it used the correct type of variable.
10. 2\_5. On line 56 I added a second & and | because you need to use 2, because that is the proper syntax for or and and. On lines 81, 101, and 124, I added an else statement, so that it only runs if the if statement is false.
11. 3. On line 13 I added #include <stdlib.h> so that I can use the rand function. On line 16 I added an \* so that the comment is done properly. On line 23 I added void run\_game(int computer\_number); so that the function can be called in the main function. On line 34 I added a / to end the comment. On line 48 I added an e so that played was spelled correctly. On line 72 I added an &, because it is required when taking input for anything other than strings. On line 93 I added a +1 so that the rand function returns a value within 1-100. On line 105, I declared the variable correct with a type of int. On line 110 I changed the input variable type specifier from %c to %d. On line 119 I added an = sign because you use == when checking if 2 things are equal. On line 127 I removed the ; because you do not use one at the end of an else if statement. On line 133, I changed it to an else if statement so that I was able to check certain parameters.

# Testing

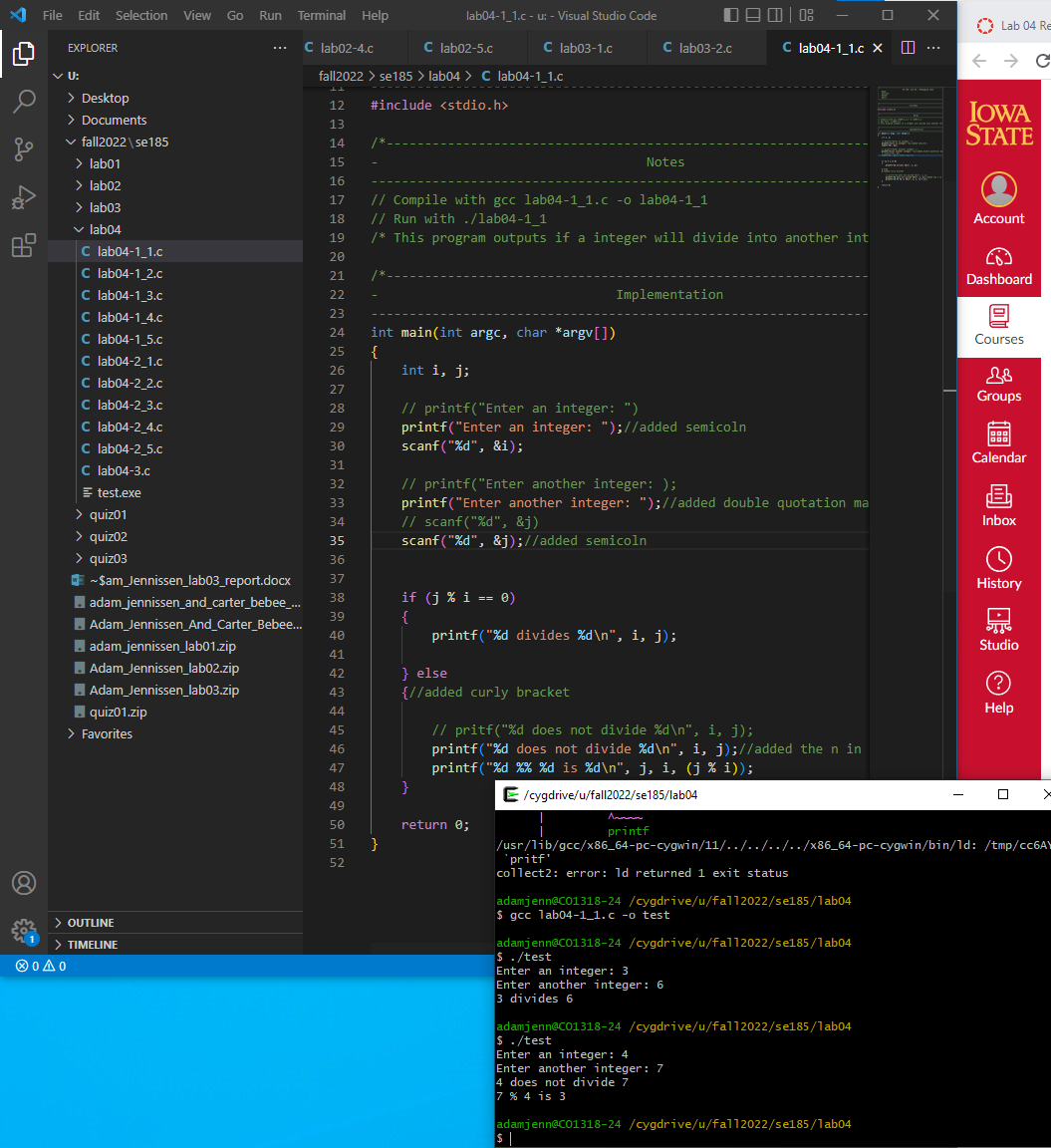
When going through the lab I made sure to compile and run after changing one or two lines, to see if what I changed had fixed the problem. I also made sure that when I was running the problem I tested multiple inputs, both within the range and outside of the input range, to make sure the program handled them correctly. I also had to check after every change I made on the last 6 problems, because the errors in those ones were logic based, I had to be extra sure that they were running correctly.

# Comments

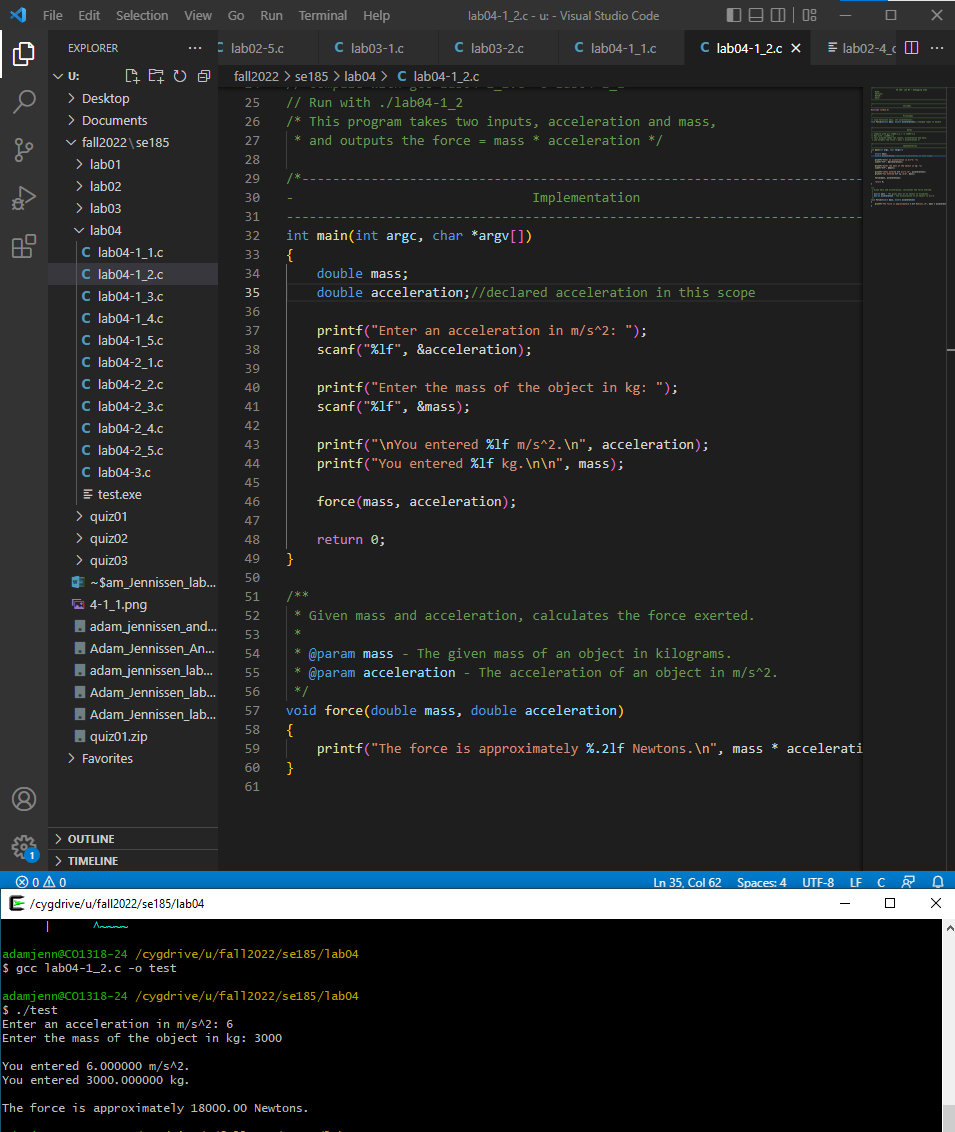
I learned that using the -wall flag when compiling will tell you about things that could give unintended results. I got a much better understanding of what kind of error the compiler will notice, and what kind of errors will run, but give you improper results.

# Screen Shots

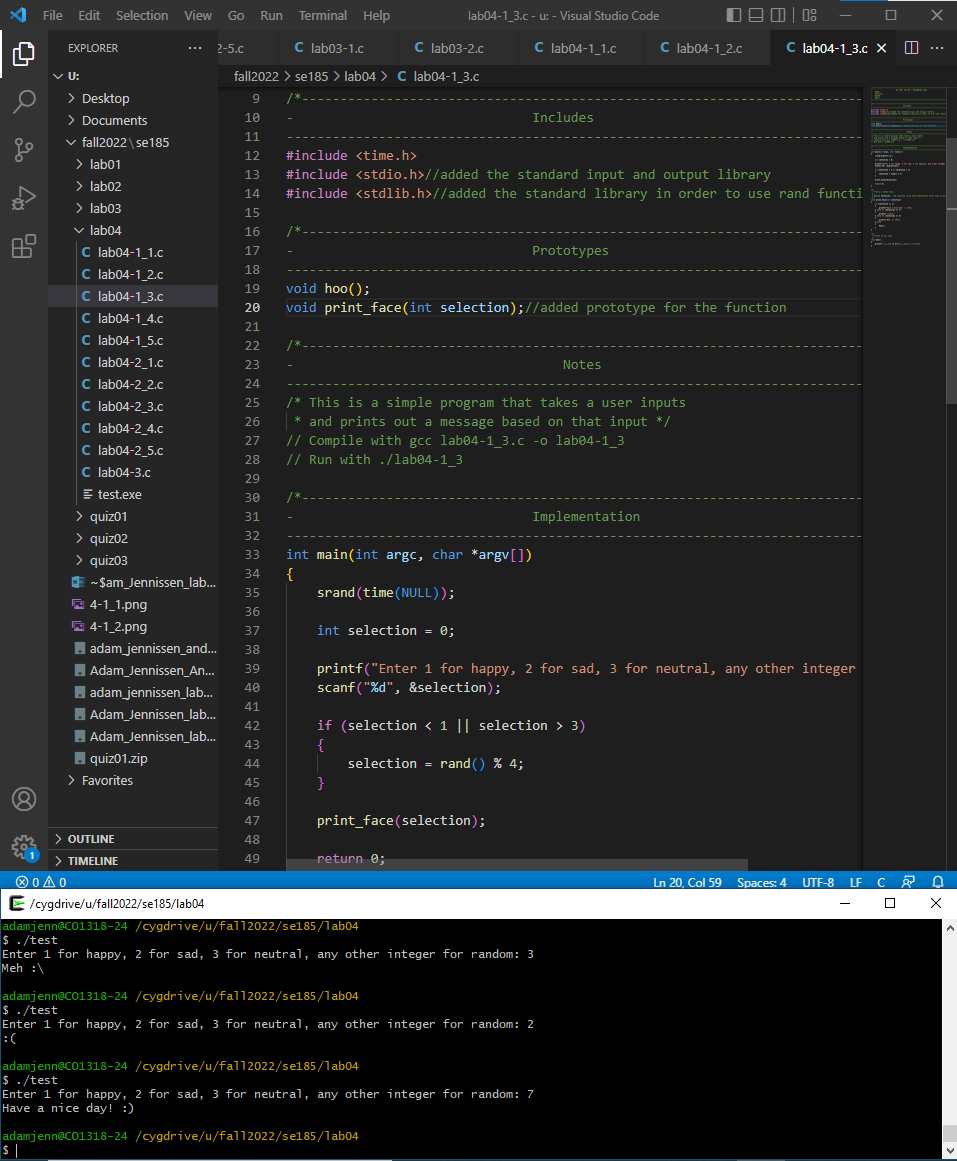
1.



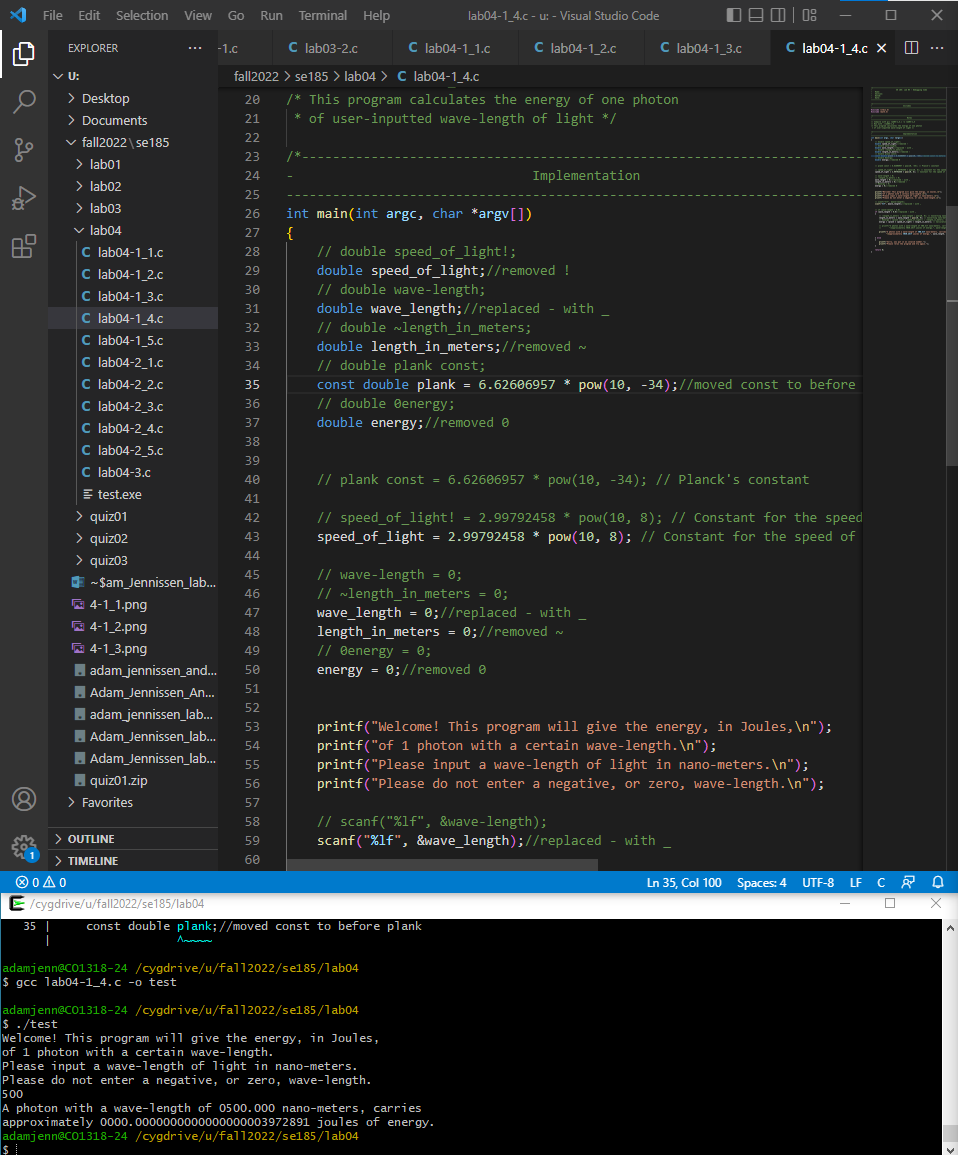
2.



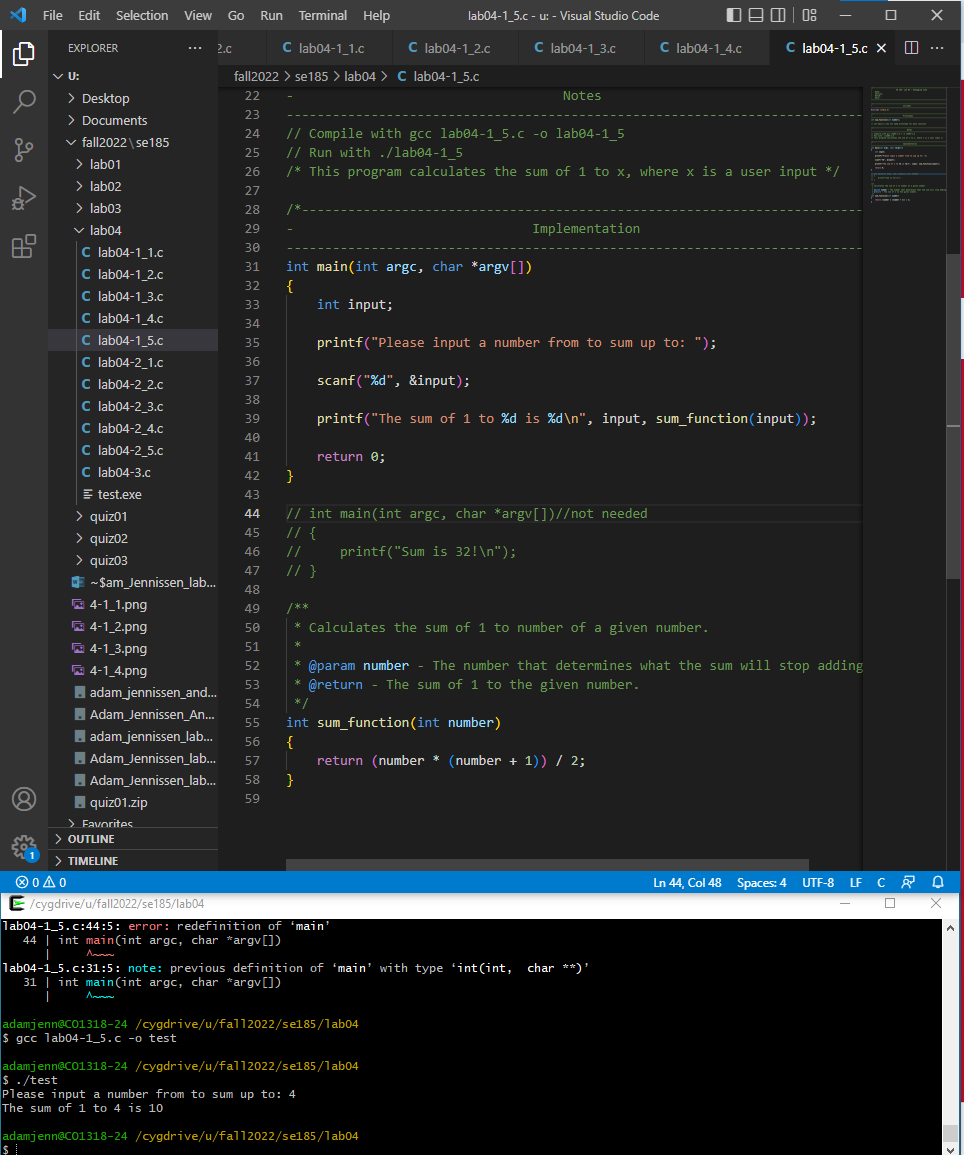
3.

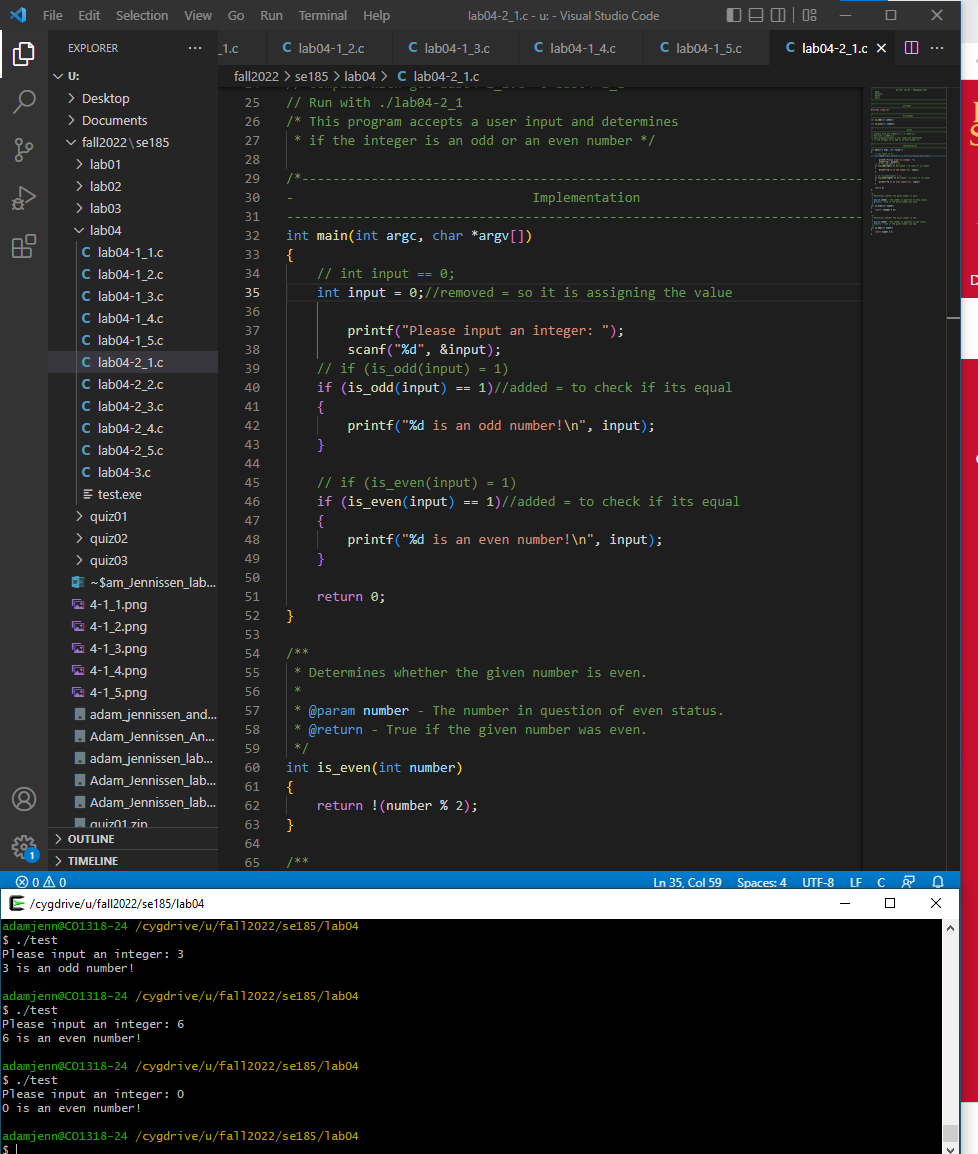


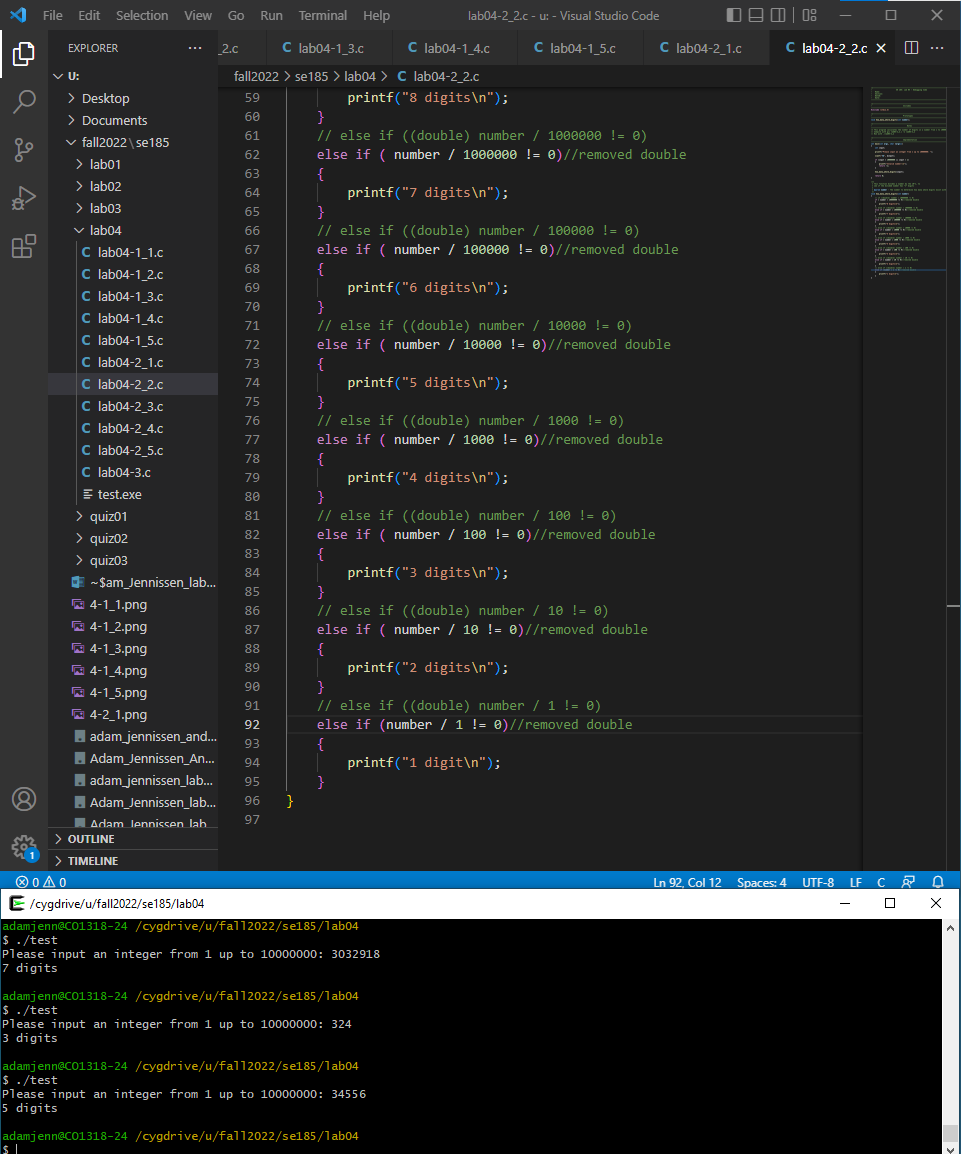
4.

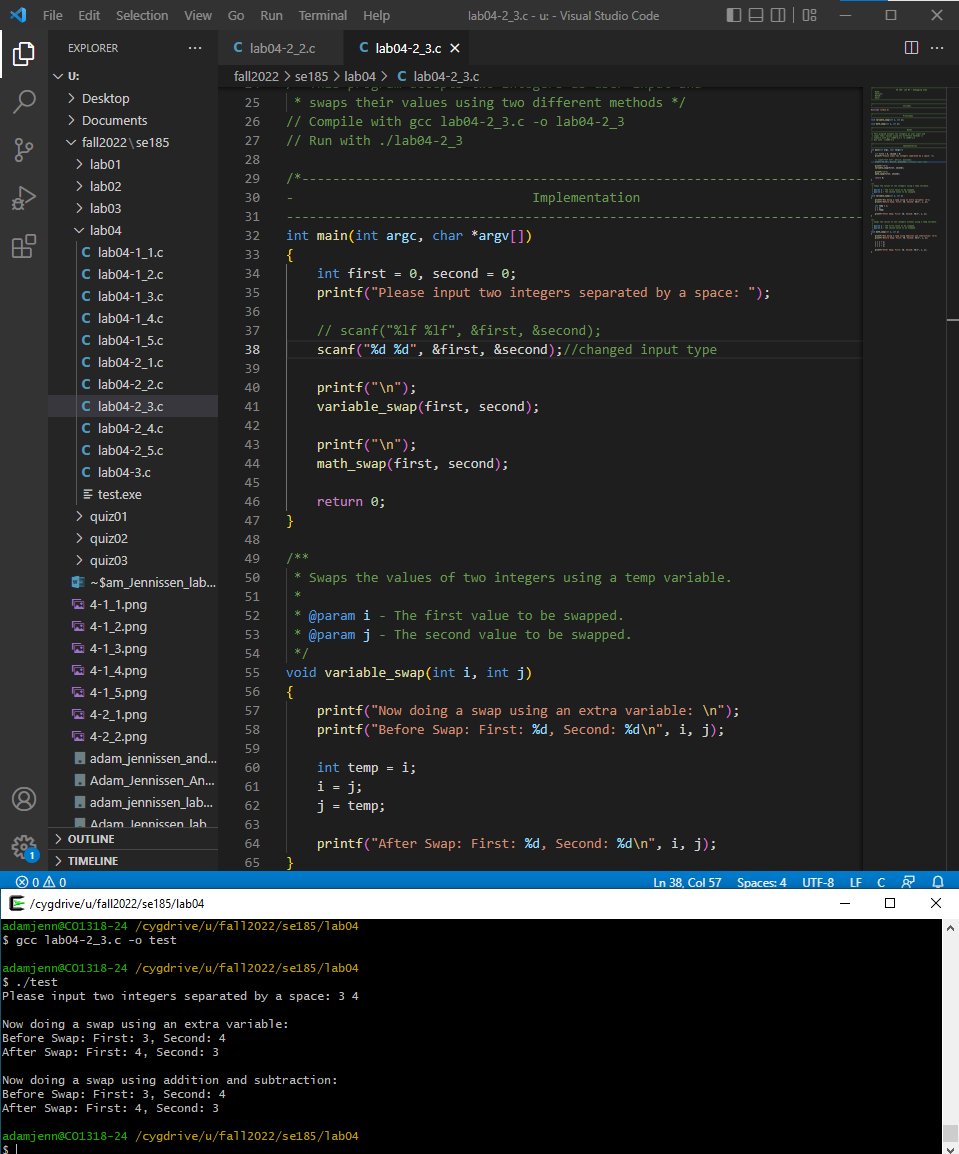


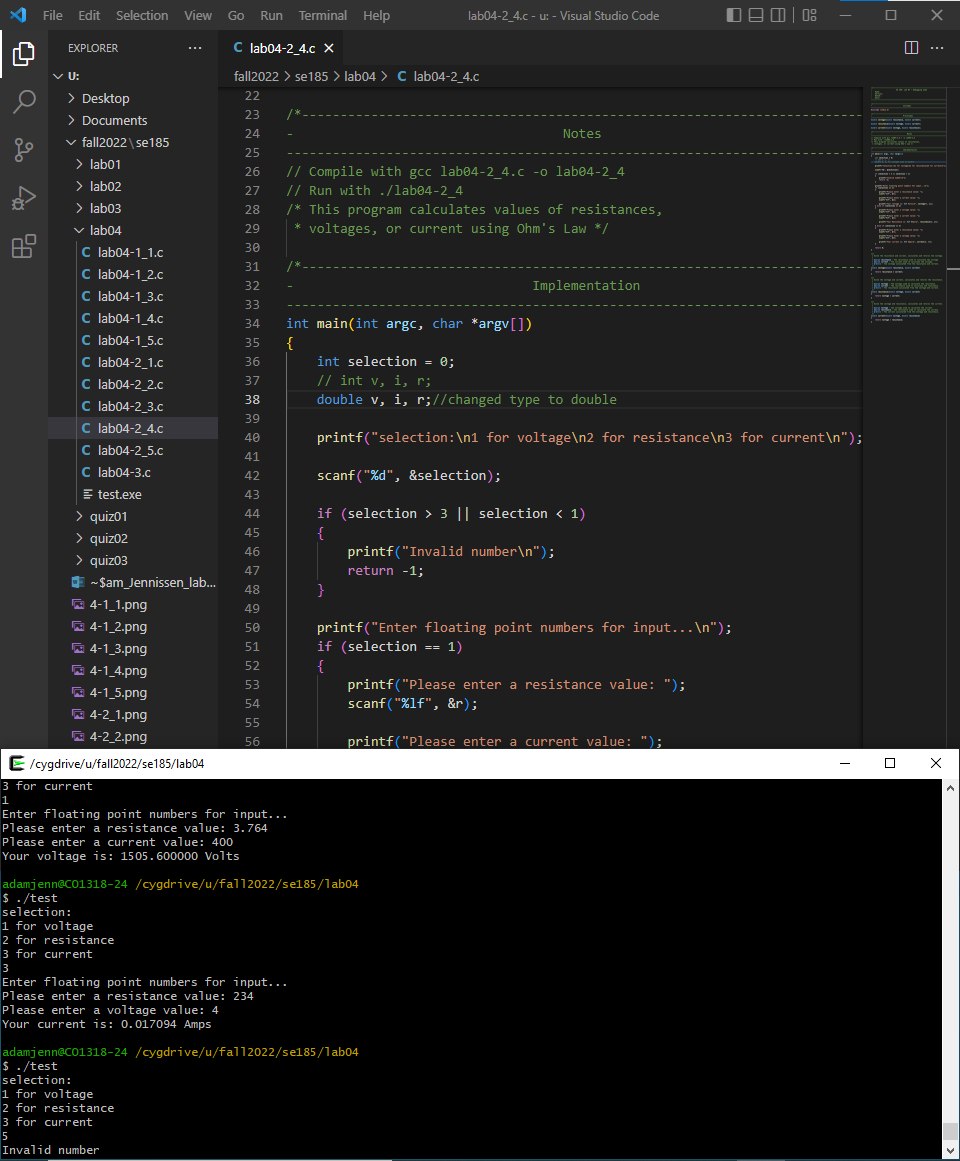
5.

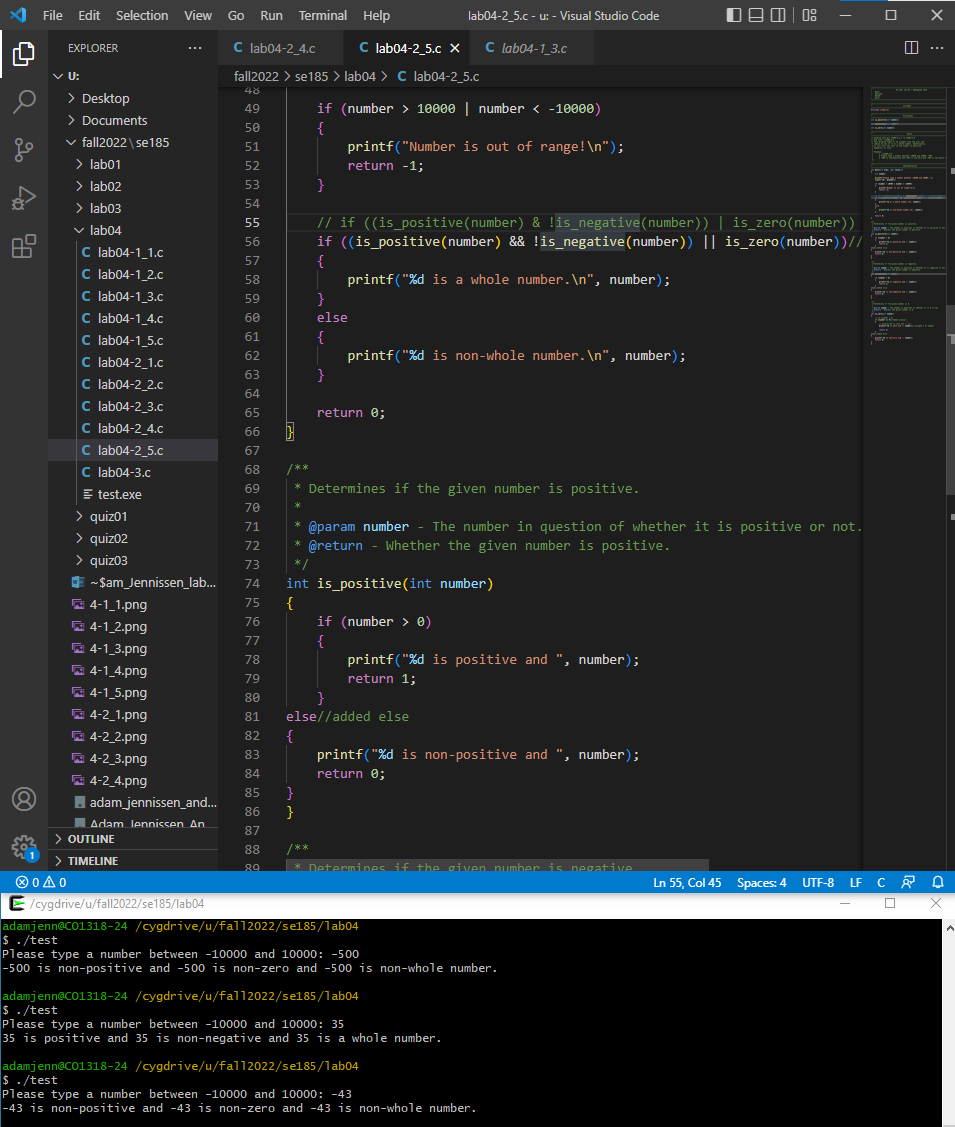


6.

7.

8.

9.

10.

11.