EGR: 226 Microcontroller Programming and Applications

Winter 2021

Instructor” Prof. Trevor Ekin

**Lab 1: Programming Refresher**

Gabriel Gasbarre

January 26, 2021

[1. Objectives 2](#_Toc62592272)

[2. Equipment 3](#_Toc62592273)

[3. Introduction 3](#_Toc62592274)

[3.1 Part 1: Resistor Analysis Tool 3](#_Toc62592275)

[3.2 Part 2: Statistics Library 4](#_Toc62592276)

[4. Procedure 4](#_Toc62592277)

[4.1 Part 1: Resistor analysis tool 4](#_Toc62592278)

[4.2 Part 2: Statistics Library 5](#_Toc62592279)

[5. Results 6](#_Toc62592280)

[5.1: Part 1 7](#_Toc62592281)

[5.2: Part 2 7](#_Toc62592282)

[6 Conclusions and Future Work 8](#_Toc62592283)

# Objectives

This lab is intended to provide an opportunity to practice writing C programs in the Code::Blocks console. Key programming practices to be demonstrated during this lab are: use of functions, commenting to ensure program readability, and error checking to avoid unwanted execution failures.

# Equipment

|  |  |  |
| --- | --- | --- |
| Part | Description | Model |
| Code::Blocks | cross-platform IDE that supports multiple compilers | 20.03 |
| EGR:226 Structured Laboratory Activity | C programming refresher guide | N/A |
|  |  |  |

# Introduction

## 3.1 Part 1: Resistor Analysis Tool

For part 1, students were asked to create a C program with function prototypes to prompt the user to input a desired Ohm reading for a resistor. This function should also display the Color-Code scheme featured in table 1 of the lab assignment to the monitor. Next, the program created was to take the input and determine what color code corresponds to the resistor with the Ohm reading that was input by the user. This program should have error checking included.

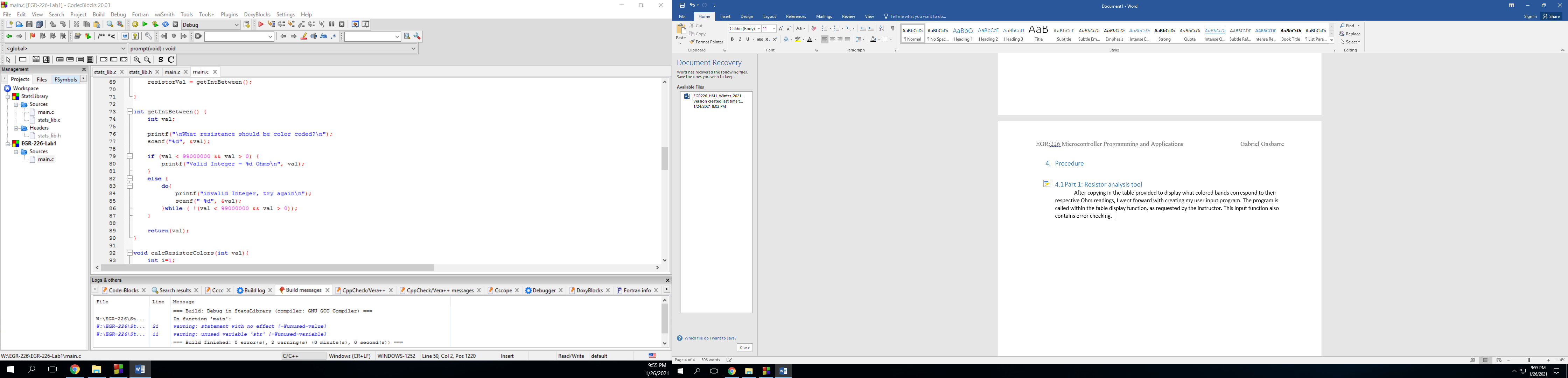
## 3.2 Part 2: Statistics Library

For part 2, students were asked to create a C program that contains an operational library of functions. The source file should contain all of the library functions and the code required to make them execute successfully, while the header file declares all of the library functions. These functions are basic arithmetic calculations that read a text file and output the desired calculations for the user without any input.

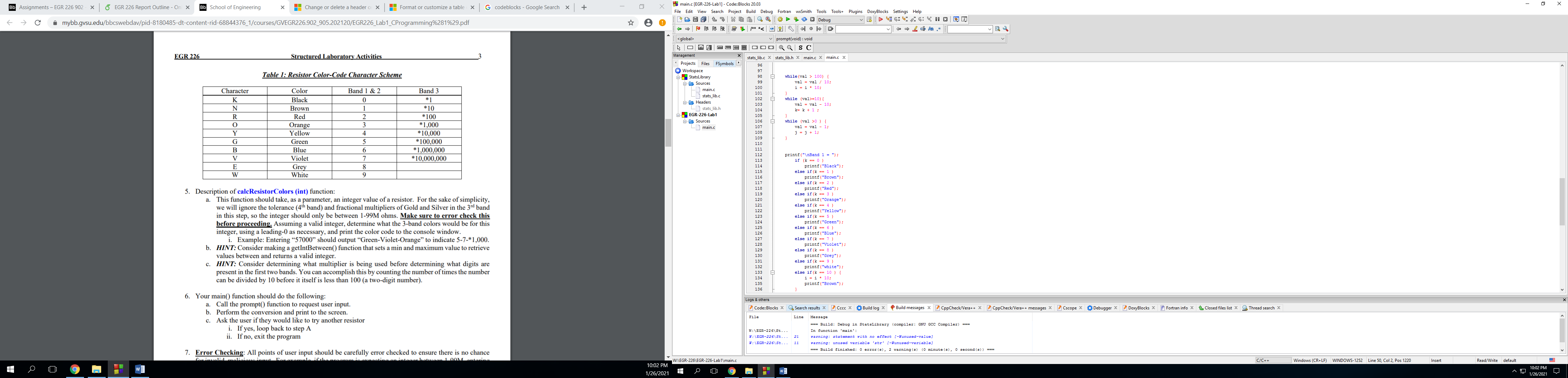
# 4. Procedure

## Part 1: Resistor analysis tool

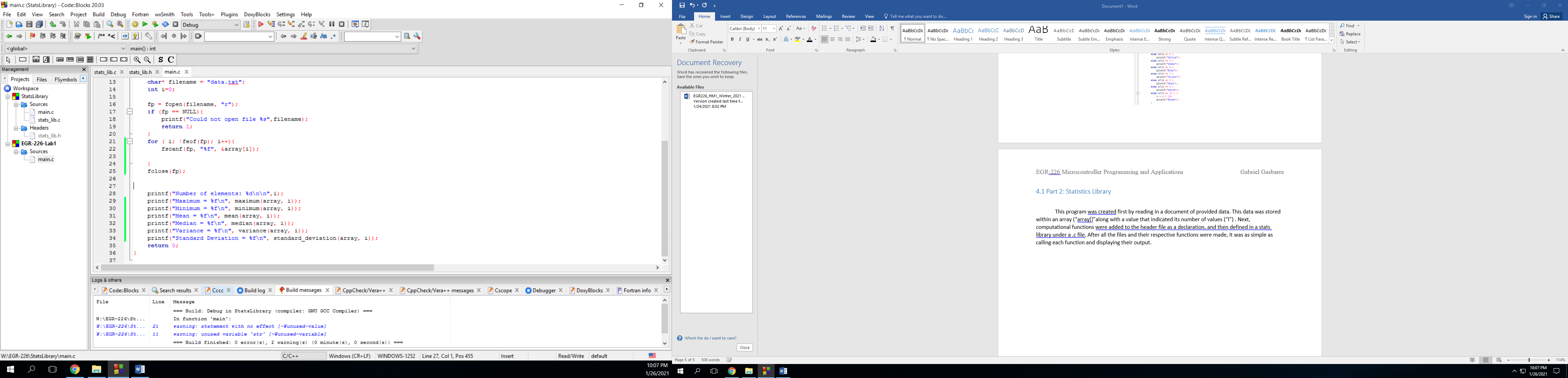
After copying in the table provided to display what colored bands correspond to their respective Ohm readings, I went forward with creating my user input program. The program is called within the table display function, as requested by the instructor. This input function also contains error checking, forcing the user to input a resistor with ohms between 99,000,000 and 0.



Next, that input is read and saved as “resistorVal” which is then passed to the next function, which calculates the corresponding band colors. I figured out the third band multiplier first by dividing by 10 until the value was less than or equal to 100. After that, the first band was calculated by subtracting 10 until it was less than or equal to 10. Finally, the second band was calculated by subtracting 1 until the program reached 0. Each division or subtraction was counted, and saved as a number that would then be used to find the corresponding color code using a simple if-else tree.

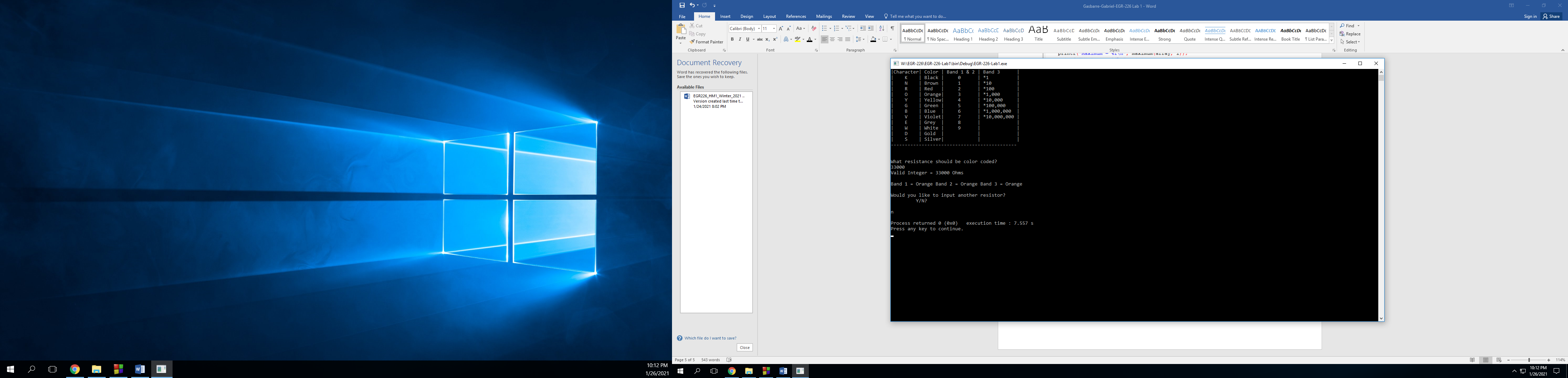


## 4.2 Part 2: Statistics Library

 This program was created first by reading in a document of provided data. This data was stored within an array (“array[]”along with a value that indicated its number of values (“I”) . Next, computational functions were added to the header file as a declaration, and then defined in a stats library under a .c file. After all the files and their respective functions were made, it was as simple as calling each function and displaying their output.

# Results

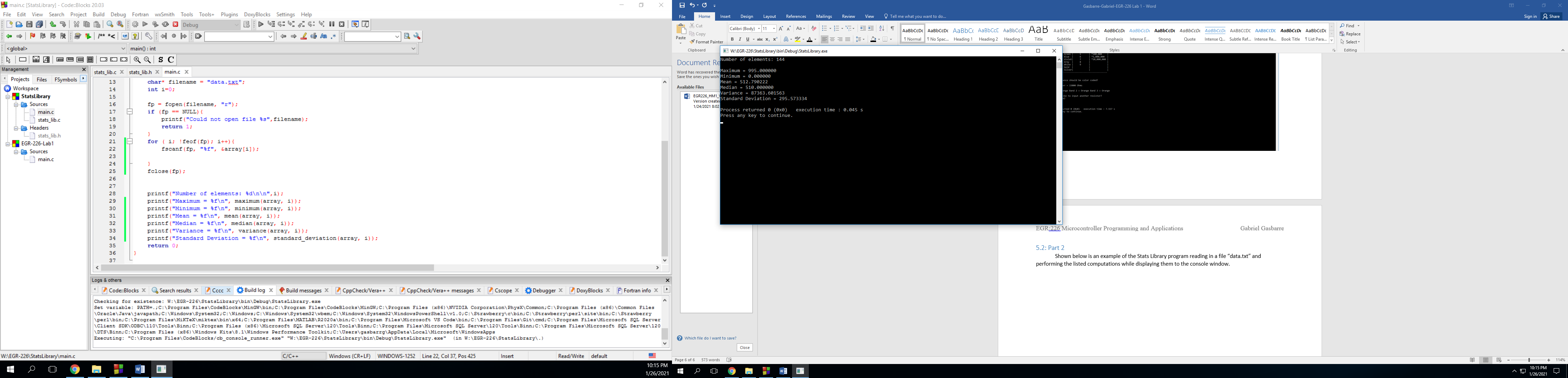
## 5.1: Part 1

Running the resistor analysis tool worked successfully and provided the desired output. Listed below is an example of how the program responds to the input of “33,000” Ohms, without another desired input.

## 

## 5.2: Part 2

Shown below is an example of the Stats Library program reading in a file “data.txt” and performing the listed computations while displaying them to the console window.



# 6 Conclusions and Future Work

I am quite satisfied with the functionality of my programs. Both seem to work flawlessly, however I would like to include more comments in the future for someone reading my program.