**Lab5.Ogbondah**

**LAB 5**

**SECTION D**

**Chimzim Ogbondah**

**SUBMISSION DATE: 10.9.2018**

**DATE: 10.9.2018**

# Problem

1. The purpose of part one of this lab was to understand how to fix syntax errors within the code based on the helpful hints the compiler gave you when you tired running the code as well as understanding what the different compiling errors meant.
2. The purpose of part two of this lab was to understand how to fix errors in our code that wouldn’t cause our code to not compile.
3. The purpose of part 3 of this lab was to understand how to fix both syntax errors that our causing our code to not compile as well as errors that our causing the code to output the wrong thing occasionally.

# Analysis

1. The problem for part one of this lab was to compile and run parts 1.1-1.5. Based on what errors the complier gave and the helpful hints for where the source of the problem was, we were then tasked with fixing all the errors until the code compiled error free, while commenting what was fixed inside the code, preferably next to the fixed line.
2. The problem for part 2 of this lab was to run parts 2.12-2.5 over and over to see how the outputs differed not giving the correct output every time. From This we were tasked with going back into the code and finding the error, commenting out that line and writing the correct line of code below it.
3. The problem for part 3 of the lab was to take the knowledge from parts 1.1-1.5 as well as 2.1-2.5 and put it together, so we could fix the errors within part 3 of the lab allowing it to compile and run correctly giving the right output.

# Design

1. Our problem was to run and compile the code and read into the compiler errors as well as the helpful tips to give us an idea of where the code isn’t being compiled at. This process was to be repeated until the code successfully compiled. I used a step by step process to help me solve this problem
   1. I would myself look over the code to see if I could catch any errors
   2. Then I would run and compile the code, reading into the helpful tips
   3. Using the helpful tips, I would correct the first error the compiler suggested
   4. I would then comment what I changed to fix the error on the same line
   5. I then ran and compiled the code again
   6. I repeated the process of fixing one error, commenting, and then recompiling until it successfully compiled and ran

Using the basic outline above I met the criteria and successfully compiled 1.1-1.5. To ensure I did this correctly I looked I ran the successfully compiled code over and over.

1. Our problem was to run the code several times, so we could see how occasionally, the wrong output was happening and where. From there code was to be checked for the error, until it was fixed, and the code ran correctly. I used a step by step process to complete this part.
   1. I would run the code 6 or 7 times with different inputs
   2. I then would observe where the error in the output would happen
   3. Then I would look over the entire code, and more specifically at where I thought the problem was coming from
   4. Comment out the incorrect line
   5. Then I would fix what I thought this issue was within the code
   6. Run the code again
   7. I would repeat this process until the code ran correctly

Using the outline above I completed 2.1-2.5. to ensure I did this correctly I ran the program multiple times making sure that I was getting the correct output.

1. Our problem to run the code for 5.3c, and use our tools and knowledge gained from the first two parts of the lab to effectively. I used a step by step process to work through this problem.
   1. I would first look over the code to see if I could catch any syntax errors
   2. Compile and run the code
   3. Go back and fix errors
   4. Once all the errors were fixed I read into the code to see what it did
   5. Fixed things that would cause the code to run improperly
   6. Compile and run to see if it worked
   7. Repeated steps e & f until code ran fine

Using the outline above I completed the problem for 5.3c, and to ensure that I did this correctly I ran and compiled the code several times giving different outputs

# Testing

1. To make sure the results of the solution were correct, I compiled to make sure I didn’t get any errors, and then I went back and ensured that I added comment to every line that I fixed to make the code work, and then I ran the program to make sure it worked effectively again.
2. To ensure the results of part 2.1-2.5 were correct, I compiled and ran, and I looked at the code and the task specifically, and then chose values that I knew the correct output for, so I could determine that the program was working effectively.
3. To make sure the results of 5.3c were correct, I compiled and looked at what the code was supposed to do. I then gave wrong inputs to make sure it worked, along with the correct inputs.

# Comments

Part 1

1. 1.1- syntax errors. On line 16 I added a semicolon, on 20 I added a closing bracket, and on 22 I added an opening bracket. On line 23 I added an n to spell printf correctly, on line 12 I added “ to finish the printf() statement
   1. Semicolon, missing bracket, missing bracket, missing “ mark, misspelling of the word printf
2. 1.2- wrong variable declaration. Line 12 I defined mass to a double value, 15 & 18 I changed %d to %lf so it matched up with, 29 defined mass to a double value
   1. Missing declaration, wrong variable def
3. 1.3- syntax errors. Line 10 I defined the print\_face function prototype, commented out srand(time(null)) on line 12 because it wasn’t being used inside the function.
   1. Missing function prototype declaration
4. 1.4-syntax errors. On line 13, 19, 28 & 32 changed – to an \_, on line 14, 20, 33 I deleted the weird symbol in front of length\_in\_meters so it became a valid variable declaration, on line 15, 21 & 33 I deleted the zero that was before energy, on line 17 & 18 I defined the two constants with const
   1. Wrong variable types, wrong variable declaration
5. 1.5- On line 11 & 25 rename the function so there weren’t conflicting calls, on line 26 moved the print statement inside the function
   1. Conflicting function calls print statement was not inside the function
6. 2.1- on line 18 I removed one of the equal signs in input, one line 23 inside the if statement I added another equal sign, on line 27 added and else if, and then added an extra equal sign inside the statement
   1. Missing equal signs, too many equal signs, two if statements in a row (possibility to print out both things)
7. 2.2- on line 36, 40, 44, 48, 52, 56, 60, 64 I deleted the double call inside of the if and else if statements
   1. Confliction variables inside the function (double vs int)
8. 2.3- on line 20 I changed %lf to %d because the scanf statements corresponded with int values
   1. Conflicting variables inside the scanf() statement (int vs double %lf vs %d)
9. 2.4-on line 20 I changed v, I, and r to double calls because they would conflict inside the functions
   1. Variable definition and function call confliction
10. 2.5 on line 15, 17, 19 changed the functions from taking int to double so the last function worked properly, this also changed the functions on lines 47, 61, and 75. On lines 26, 40, 50, 55, 64, 69, 78, and 83 I changed all the %d to %lf so they matched up with the new call of n and num since it is a double value. On line 28 I added another | symbol to complete the or statement, and on line 34 I added another & symbol to complete the and statement
    1. Missing & and | in if else statements, changed all %d to %lf for double call, changed all int calls to double so the functions worked properly
11. 5.3c-

Part 1.1

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program outputs if a integer will divide into another integer with no remainder\*/

int main**()** **{**

int i**,** j**;**

printf**(**"Enter an integer: "**);** //there was a missing semicolon

scanf**(**"%d"**,** **&**i**);**

printf**(**"Enter another integer: "**);** //added a closing " mark

scanf**(**"%d"**,** **&**j**);** //there was a missing semi colon

**if** **(**j **%** i **==** 0**)** **{**

printf**(**"%d divides %d\n"**,** i**,** j**);**

**}** //there was a missing end bracket

**else** **{** //missing opening bracket

printf**(**"%d does not divide %d\n"**,** i**,** j**);** //added a n to correctly spell printf

printf**(**"%d %% %d is %d\n"**,** j**,** i**,** **(**j **%** i**)** **);**

**}**

**return** 0**;**

**}**

Part 1.2

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program takes two inputs acceleration

and mass, and ouputs the force = mass\*acceleration \*/

void force**(**double m**,** double accel**);**

int main**()** **{**

double mass**,** accel**;** //accel variable was not defined

printf**(**"Enter an acceleration: "**);**

scanf**(**"%lf"**,** **&**accel**);** // changed %d to lf since variable is a double value

printf**(**"Enter the mass of the object: "**);**

scanf**(**"%lf"**,** **&**mass**);** //changed %d to lf since variable is a double value

force**(**mass**,** accel**);**

printf**(**"You entered %lf m/s^2\n"**,** accel**);**

printf**(**"You entered %lf kg\n"**,** mass**);**

**return** 0**;**

**}**

void force**(**double m**,** double accel**)** **{**

double mass **=** m **/** 1000**;** //mass in the function wasn't defined so I defined as double

printf**(**"The force is %lf milliNewtons\n"**,** mass **\*** accel**);**

accel **=** accel**\***1000**;**

printf**(**"The force is %lf Newtons\n\n"**,** mass **\*** accel**);**

**}**

Part 1.3

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <time.h>

#include <stdio.h> //added in so scanf() could be used

/\* This is a simple program that takes a user inputs and prints out a message based on that input \*/

void hoo**();**

void print\_face**(**int opt**);** // defined the function above

int main**()** **{**

// srand(time(NULL)); causing for an error and not being used in code

int option **=** 0**;**

printf**(**"Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: "**);**

scanf**(**"%d"**,** **&**option**);**

**if** **(**option **<** 1 **||** option **>** 3**)** **{**

option **=** rand**()** **%** 4**;**

**}**

print\_face**(**option**);**

**return** 0**;**

**}**

void print\_face**(**int opt**)** **{**

**if** **(**opt **==** 1**)** **{**

printf**(**"Have a nice day! :) \n"**);**

**}**

**else** **if** **(**opt **==** 2**)** **{**

printf**(**":(\n"**);**

**}**

**else** **if** **(**opt **==**3**)** **{**

printf**(**"meh :\\ \n"**);**

**}**

**else** **{**

hoo**();**

**}**

**}**

void hoo**()** **{**

printf**(**" \*\_\_\_\*\n {O,O}\n/)\_\_\_)\n\_\"\_\_\"\_\n"**);**

**}**

Part 1.4

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

#include <math.h>

/\* This program calculates the energy of one photon of uder inputed wavelength

\* of light \*/

int main**()**

**{**

double wave\_length**=**0**;** //changed the - to an underscore and updated all other occurances

double length\_in\_meters**=**0**;** //deleted weird symbol infront of length\_in\_meters and updated all other occurances

double energy **=**0**;** //deleted all zeros to match up with energy call in line 36 in print statement

const planck\_const **=** **(**6.62606957**)\*(**pow**(**10**,-**34**));** //Planck's constant (let them be known as constants and deleted conflicting decleration above)

const speed\_light **=** **(**2.99792458**)\*(**pow**(**10**,**8**));** //Constant for the speed of light (let them be known as constants and deleted conflicting dec above)

printf**(**"Welcome! This program will give the energy, in Joules,\n"**);**

printf**(**"of 1 photon with a certain wavelength.\n"**);**

printf**(**"Please input a wavelength of light in nano-meters.\n"**);**

printf**(**"Please do not enter a negative, or zero, wavelength.\n"**);**

scanf**(**"%lf"**,** **&**wave\_length**);**

**if** **(**wave\_length **>** 0.0**)**

**{**

length\_in\_meters **=** wave\_length **/** pow**(**10**,**9**);** //Converting nano-meters to meters

energy **=** **(**planck\_const**\***speed\_light**)** **/** **(**length\_in\_meters**);** //Calculating the energy of 1 photon

printf**(**"A pha wavoton with elength of %8.3lf nano-meters, carries\n%30.25lf joules of energy."**,** wave\_length**,** energy**);**

**}**

**else**

**{**

printf**(**"Sorry, you put in an invalid number."**);**

printf**(**"Please rerun the program and try again."**);**

**}**

**return** 0**;**

**}**

Part 1.5

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This progam calculates the sum of 1 to x, where x is a user input \*/

int sum\_funct**(**int n**);**

int sum\_main**();**

void main**()** **{** //moved the curly bracket up

int input**;**

printf**(**"Please input a number from to sum up to: "**);**

scanf**(**"%d"**,** **&**input**);**

printf**(**"The sum of 1 to %d is %d\n"**,** input**,** sum\_funct**(**input**));**

**}**

int sum\_main**(){** //renamed function to sum name to avoid conflicting call

printf**(**"Sum is 32!\n"**);** //moved the print statement inside of the fucntion

**}**

int sum\_funct**(**int n**){** //moved the curly bracket infront of sum\_fraction

**return** **(**n**\*(**n**+**1**))/**2**;**

**}**

Part 2.1

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This progam accepts a user input and determines if the integer is

\* an odd or an even number \*/

int isOdd**(**int i**);**

int isEven**(**int i**);**

int main**()**

**{**

int input**;**

//input == 0; removed one equal sign

input **=** 0**;**

printf**(**"Please input an integer: "**);**

scanf**(**"%d"**,** **&**input**);**

// if(isOdd(input)=1) added a second equal sign

**if(**isOdd**(**input**)==**1**)** **{**

printf**(**"%d is an odd number!\n"**,** input**);**

**}**

//if(isEven(input)=1) added a second equal sign

**else** **if(**isEven**(**input**)==**1**){**

printf**(**"%d is an even number!\n"**,** input**);**

**}**

**return** 0**;**

**}**

int isOdd**(**int i**)** **{**

**if(**i **%** 2**)**

**{**

**return** 1**;**

**}**

**else**

**{**

**return** 0**;**

**}**

**}**

int isEven**(**int i**)** **{**

**if(!(**i **%** 2**))**

**{**

**return** 1**;**

**}**

**else**

**{**

**return** 0**;**

**}**

**}**

Part 2.2

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program calculates the number of digits in a number from 1 to 100000\*/

void digits**(**int n**);**

int main**()**

**{**

int input**;**

printf**(**"Please input a number from 1 up to 10000000: "**);**

scanf**(**"%d"**,** **&**input**);**

**if(** input **>** 10000000 **||** input **<** 1**)**

**{**

printf**(**"Invalid number!\n"**);**

**return** **-**1**;**

**}**

digits**(**input**);**

**return** 0**;**

**}**

/\* This function divides a number by the 10^n, to see if the divided number

\* has "n" digits \*/

void digits**(**int n**)**

**{**

//if((double)n/10000000!=0) took out double

**if(**n**/**10000000**!=**0**)** **{**

printf**(**"8 digits\n"**);**

**}**

// else if((double)n/1000000!=0) again took out double

**else** **if** **(**n**/**1000000**!=**0**)** **{**

printf**(**"7 digits\n"**);**

**}**

// else if((double)n/100000!=0) took out the double

**else** **if(**n**/**100000**!=**0**)** **{**

printf**(**"6 digits\n"**);**

**}**

//else if ((double)n/10000!=0) took out the double

**else** **if(**n**/**10000**!=**0**)** **{**

printf**(**"5 digits\n"**);**

**}**

//else if ((double)n/1000!=0) took out the double

**else** **if(**n**/**1000**!=**0**)** **{**

printf**(**"4 digits\n"**);**

**}**

// else if ((double)n/100!=0) took out the double

**else** **if(**n**/**100**!=**0**){**

printf**(**"3 digits\n"**);**

**}**

// else if ((double)n/10!=0) took out the double

**else** **if(**n**/**10**!=**0**)** **{**

printf**(**"2 digits\n"**);**

**}**

// else if ((double)n/1!=0) took out the double

**else** **if(**n**/**1**!=**0**){**

printf**(**"1 digit\n"**);**

**}**

**}**

Part 2.3

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program accepts two integers as user input and swaps their values

\* using two different methods \*/

void var\_swp**(**int i**,** int j**);**

void math\_swp**(**int i**,** int j**);**

int main**()**

**{**

int first **=** 0**,** second **=** 0**;**

printf**(**"Please input two integers separated by a space: "**);**

// scanf("%lf %lf", &first, &second); frist and second were defined as int and so changed %lf to %d in scanf statement

scanf**(**"%d %d"**,** **&**first**,** **&**second**);**

printf**(**"\n"**);**

var\_swp**(**first**,** second**);**

printf**(**"\n"**);**

math\_swp**(**first**,** second**);**

**return** 0**;**

**}**

void var\_swp**(**int i**,** int j**)**

**{**

printf**(**"Now doing a swap using an extra variable: \n"**);**

printf**(**"Before Swap: First: %d, Second: %d\n"**,** i**,** j**);**

int k **=** i**;**

i **=** j**;**

j **=** k**;**

printf**(**"After Swap: First: %d, Second: %d\n"**,** i**,** j**);**

**}**

void math\_swp**(**int i**,** int j**)**

**{**

printf**(**"Now doing a swap using addition and subtraction: \n"**);**

printf**(**"Before Swap: First: %d, Second: %d\n"**,** i**,** j**);**

i **=** i **+** j**;**

j **=** i **-** j**;**

i **=** i **-** j**;**

printf**(**"After Swap: First: %d, Second: %d\n"**,** i**,** j**);**

**}**

Part 2.4

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program calculates values of resistances, voltages, or current

\* using Ohm's Law \*/

double voltage**(**double resistance**,** double current**);**

double resistance**(**double voltage**,** double current**);**

double current**(**double voltage**,** double resistance**);**

int main**()**

**{**

int select **=** 0**;**

// int v, i, r; changed the declartion of the variable to a double since it was everywhere else

double v**,** i**,** r**;**

printf**(**"Select:\n1 for voltage\n2 for resistance\n3 for current\n"**);**

scanf**(**"%d"**,** **&**select**);**

**if(**select **>** 3 **||** select **<** 1**)**

**{**

printf**(**"Invalid number\n"**);**

**return** **-**1**;**

**}**

printf**(**"Enter floating point numbers for input...\n"**);**

**if(**select **==** 1**)**

**{**

printf**(**"Please enter a resistance value: "**);**

scanf**(**"%lf"**,** **&**r**);**

printf**(**"Please enter a current value: "**);**

scanf**(**"%lf"**,** **&**i**);**

printf**(**"Your voltage is: %lf Volts\n"**,** voltage**(**r**,** i**));**

**}**

**else** **if(**select **==** 2**)**

**{**

printf**(**"Please enter a voltage value: "**);**

scanf**(**"%lf"**,** **&**v**);**

printf**(**"Please enter a current value: "**);**

scanf**(**"%lf"**,** **&**i**);**

printf**(**"Your Resistance is: %lf Ohms\n"**,** resistance**(**v**,** i**));**

**}**

**else** **if(**select **==** 3**)**

**{**

printf**(**"Please enter a resistance value: "**);**

scanf**(**"%lf"**,** **&**r**);**

printf**(**"Please enter a voltage value: "**);**

scanf**(**"%lf"**,** **&**v**);**

printf**(**"Your current is: %lf Amps\n"**,** current**(**v**,** r**));**

**}**

**return** 0**;**

**}**

double voltage**(**double resistance**,** double current**)**

**{**

**return** resistance **\*** current**;**

**}**

double resistance**(**double voltage**,** double current**)**

**{**

**return** voltage **/** current**;**

**}**

double current**(**double voltage**,** double resistance**)**

**{**

**return** voltage **/** resistance**;**

**}**

Part 2.5

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

/\* This program takes in a number from the user and checks if it is

\* a whole number. It also should print if the number is a

\* postive, negative, or zero number.

\* Ex.

\* input: num = 5

\* output: 5 is a postive and 5 is non-negative and 5 is non-zero and 5 is a whole number.\*/

// int isPositive(int n); changed all to double call since num is used in all

double isPositive**(**double n**);**

// int isNegative(int n); changed all to double call since num is used in all

double isNegative**(**double n**);**

// int isZero(int n); changed all to double call since num is used in all and needs to be zero for last function to work right

double isZero**(**double n**);**

int main**()**

**{**

// int num; changed num to a double value

double num**;**

printf**(**"Please type a number between -1000 and 1000: "**);**

// scanf("%d", &num); changed %d to %lf for double callo

scanf**(**"%lf"**,** **&**num**);**

// if(num > 1000 | num < -1000) added another | so the or statement was complete

**if(**num **>** 1000 **||** num **<** **-**1000**)** **{**

printf**(**"Number is out of range!\n"**);**

**return** **-**1**;**

**}**

// if( ( isPositive(num) & !isNegative(num) ) | isZero(num) ) added another & to complete the and statement and another |

**if(** **(**isPositive**(**num**)** **&&** **!**isNegative**(**num**)** **)** **||** isZero**(**num**)** **)** **{**

printf**(**"%lf is a whole number.\n"**,** num**);** //changed %d to %lf

**}**

**else**

**{**

printf**(**"%lf is non-whole number.\n"**,** num**);** //changed %d to %lf

**}**

**return** 0**;**

**}**

// int isPositive(int n) changed to match with first prototype call

double isPositive**(**double n**){**

**if(**n**>**0**)**

**{**

printf**(**"%lf is postive and "**,** n**);** //changed %d to %lf

**return** 1**;**

**}**

**else**

**{**

printf**(**"%lf is non-postive and "**,** n**);** //changed %d to %lf

**return** 0**;**

**}**

**}**

// int isNegative(int n)changed to match first call

double isNegative**(**double n**){**

**if(**n**<**0**)**

**{**

printf**(**"%lf is negative and "**,** n**);** //changed %d to %lf

**return** 1**;**

**}**

**else**

**{**

printf**(**"%lf is non-negative and "**,** n**);** //changed %d to %lf

**return** 0**;**

**}**

**}**

// int isZero(int n) changed to match first call

double isZero**(**double n**){**

**if(**n**==**0**)**

**{**

printf**(**"%lf is zero and "**,** n**);** //changed %d to %lf

**return** 1**;**

**}**

**else**

**{**

printf**(**"%lf is non-zero and "**,** n**);** //changed %d to %lf26

**return** 0**;**

**}**

**}**

5.3 c

/\*-----------------------------------------------------------------------------

- SE 185 Lab 05

- Developed for 185-Rursch by T.Tran and K.Wang

-----------------------------------------------------------------------------\*/

#include <stdio.h>

#include <stdlib.h> //included the this library so it understands rand and srand

#include <time.h>

/\* Prototypes \*/ // added a star to complete the comment

char AskToPlay**(**int playedBefore**);** //changed the name to line up with the function

int SelectRandNum**();** //defined the function prototype above

void RunGame **(**int computerNum**);**

int main**(){**

char prompt **=** '-'**;**

int played **=** 0**,** computerGuess **=** 0**;**

prompt **=** AskToPlay**(**played**);**

played **=** 1**;**

**while(**prompt **==** 'y'**)** **{** /\* This line does not contain an error \*/

computerGuess **=** SelectRandNum**();** //deleted one equal sign so it isn't saying computer guesses is slecetrandNum

RunGame**(**computerGuess**);**

prompt **=** AskToPlay**(**played**);**

**}**

printf**(**"\n\nThank you for playing.\n"**);** //added a semicolon

**return** 0**;**

**}**

char AskToPlay**(**int playedBefore**){**

char yesNo**;**

**if** **(**playedBefore **==** 0**)** **{**

printf**(**"Do you want to play a game?\n ->"**);**

scanf**(**" %c"**,** **&**yesNo**);** //added & infornt of the variable

**}**

**else{**

printf**(**"Do you want to play again?\n ->"**);**

scanf**(**" %c"**,** **&**yesNo**);**

**}**

**return** yesNo**;**

**}**

int SelectRandNum**()** **{**

// int c\*mpGuess = 0; changed the \* to an o

int compGuess **=** 0**;**

srand**((**int**)**time**(**0**));**

compGuess **=** **((**rand**()** **%** 100**)** **+** 1**);**

**return** compGuess**;**

**}**

void RunGame**(**int computerNum**){**

int number **=** 0**,** correct **=** 0**;**

printf**(**"\nYou are guessing a number. The options are 1 through 100.\n\n"**);**

printf**(**"What is your guess on what number I will select?\n ->"**);**

scanf**(**"%d"**,** **&**number**);**

**while** **((**number **<**1**)** **||** **(**number **>**100**))** **{** /\* This line does not contain an error \*/

printf**(**"\nYour number is not within the correct range of numbers. Guess again\n ->"**);**

scanf**(**"%d"**,** **&**number**);**

**}**

**while** **(**correct **==** 0**){** /\* This line does not contain an error \*/

**if** **(**number **==** computerNum**){**

printf**(**"\nYou guessed the number correctly!\n"**);**

printf**(**"The number was %c\n\n"**,** computerNum**);**

correct **=** 1**;**

**}**

**else** **if** **(**number **<** computerNum**)** **{** //removed a semicolon because those don't belong in these statements

printf**(**"\nYou guessed too low. Enter another guess.\n ->"**);**

scanf**(**"%d"**,** **&**number**);**

**}**

**else** **{**

printf**(**"\nYou guessed too high. Enter another guess.\n ->"**);**

scanf**(**"%d"**,** **&**number**);**

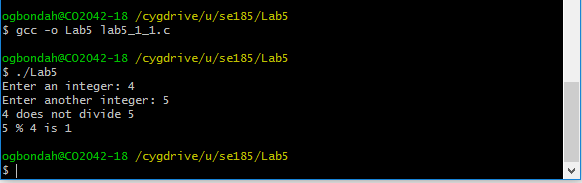
**}**

**}**

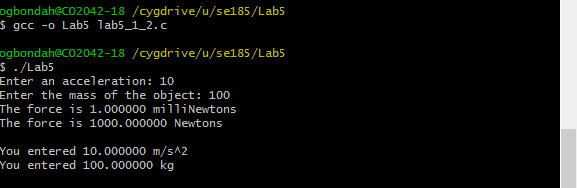
**}**

Screenshots

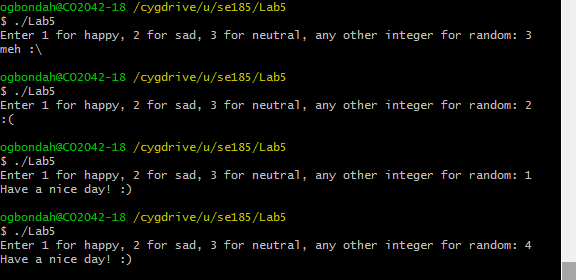
1.1



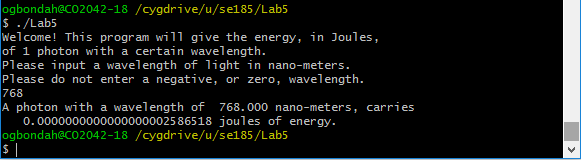
1.2



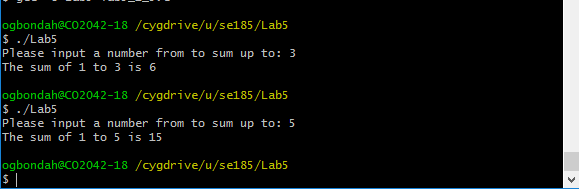
1.3



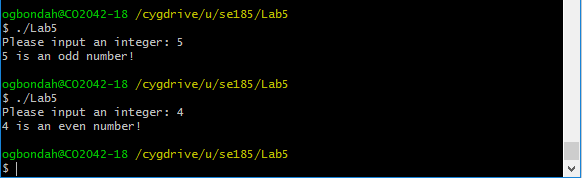
1.4



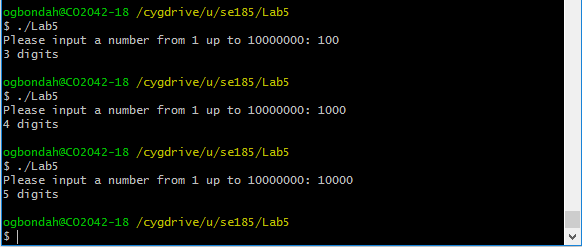
1.5



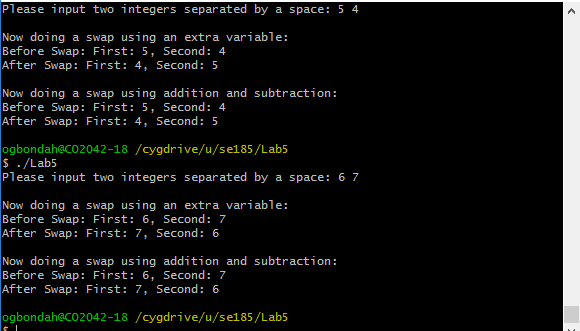
2.1



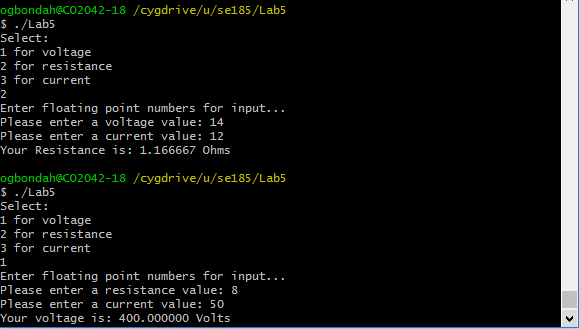
2.2



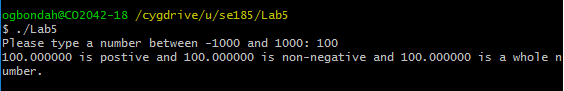
2.3



2.4



2.5



5.3c

