**Debugging Code**

**LAB 4**

**SECTION 2**

**RYAN AUN SHERN CHIN**

**SUBMISSION DATE:**

**2/21/24**

# Problem

In this lab, we were tasked to learn about the C compiler messages, become familiar with various types of Compiler errors and learn coding practices to help avoid unintentional errors. Therefore, we were given 11 C files that were riddled with errors that we had to debug in order for the program to compile successfully with intentional results.

# Analysis

In ascending order of given C files from lab04-1\_1.c to lab04-1\_5.c for Part 1, we noticed that the compiler failed to compile these files into a working program. The compiler also outputted messages that identified potential errors in certain lines of the code in the program. We therefore concluded these files from lab04-1\_1.c to lab04-1\_5.c had syntax errors that included mistakes like missing semicolons, incorrect statement syntaxes or incorrect data type declarations and correcting identifiers.

From C files, lab04-2\_1.c to lab04-2\_5.c for Part 2, we noticed that these files manage to compile successfully but with unintentional results that did not align with the purpose of the program given in the comments at the top of each program in these files. We concluded that this might be a case of logic errors, sematic errors, or bugs in the code that the original owner of these files may have unknowingly coded.

For the last C file, lab04-3.c for Part 3, we realized that this file had a combination of compiler errors and logic errors. Therefore, we had to practice what we had done in Part 1 and Part 2 again to identify and resolve errors in order for the program to compile and run successfully with intended results.

# Design

In identifying the errors in all of the C files provided in the lab manual, I religiously looked at the outputted message from compiler and fixed the errors one by one, given in ascending order of lines in the code and compiled each time I fixed an error to fully make sure I corrected the error. In the event of no compiler errors but unintended results of the program, I made sure to go through each line of the source code to identify errors or specifically traced back where the error could have originated from in the source code upon inspection of the unintended result from the compiler output. If any or all errors can’t be identified, I used to the “-Wall” compiler flag to tell the compiler to warn me of potential issues that might not cause compile errors.

Every time I fixed an error in any lines of code, I either commented out the line which had the error and wrote the corrected code in the line below it or commented what I changed next to the line of corrected code.

# Testing

In order for the programs in Part 1, 2 and 3 to run perfectly as intended, I tested the code by inputting inputs when prompted by the program and verified or checked the output to be the same as what is intended for what the program is supposed to do as given in comments at the beginning of each code file. For lab04-1\_1.c and lab04-1\_2.c that had basic calculation or arithmetic as outputs, I verified and compared the output given in the terminal with a hand calculator.

# Comments

In doing this lab, I realized the importance of double checking my code in order to avoid any errors that may appear when compiling my code. Furthermore, I managed to become familiar with various types of Compiler errors and learn coding practices like unit testing to help avoid unintentional errors.

**Question 1 and 2**

In Part 1, for lab04-1\_1.c, I added a semicolon after the printf() statement in line 29 that had a missing semicolon, a double quotation mark in line 33 that was missing, a missing semicolon after the scanf() statement in line 35, a missing brace after else in line 42 and changed the incorrect pritf() statement to printf() statement in line 45 which did not follow the correct syntax for a printf() statement.

For lab04-1\_2.c, I changed the variable type from int to double for the function prototype in line 18, which were the incorrect variable type. I also added double acceleration; in line 34, which was missing a double type declaration for the variable acceleration.

For lab04-1\_3.c, I added #include <stdio.h> and #include <stdlib.h> in line 13 and line 14 in order for printf(), scanf() and srand statements to work, which needed the missing <stdio.h> and <stdlib.h> libraries. I added void print\_face(int selection); in line 19, which was missing a function prototype for the print\_face function.

For lab04-1\_4.c, I changed the names of the variable. For example, speed\_of\_light! to speed\_of\_light in lines 29, 42, and 64. wave-length to wave\_length in lines 31, 44, 56, 59, 62 and 68. ~length\_in\_meters to length\_in\_meters in lines 33, 46, 62 and 64. plank const to plank\_const in lines 35, 40 and 64. 0energy to energy in lines 37, 48, 64 and 68. All of these naming for variables did not follow the rules for identifiers.

For lab04-1\_5.c, I removed the line “int main();” in line 19 and the lines 44 to 47 since it was not needed in the program that calculated the sum of 1 to x, where x is a user input.

Next, in Part 2, for lab04-2\_1.c, I changed the == to = in line 34. The == caused the variable input to be equal to 0 instead of being assigned with the value of 0 using the = operator. Next, I changed the = to == in lines 39 and 44. The = operator caused an error where the is\_odd(input) and is\_even(input) is assigned to the value of 1 instead of being equal to 1 which can be indicated with the == operator.

For lab04-2\_2.c, I changed double type declaration of variable number to int type for lines 56, 59, 62, 65, 68, 71, 74 and 77. Those lines had wrong type declaration of the variable number.

For lab04-2\_3.c, I changed the data specifier from %lf to %d in line 37’s scanf() statement which had an incorrect data specifier for int type variables.

For lab04-2\_4.c, I changed the variable type for v, i, r, from int to double in line 37, which had incorrect variable type.

For lab04-2\_5.c, I changed the argument from n to number in line 112. The argument n has never appeared once or ever being declared in the source code, which made me believe it was intended to be number instead.

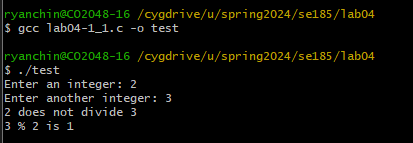
Lastly, for lab04-3.c, I added #include <stdlib.h> in line 13, which had a missing library. I also added a missing \* in line 18 to complete the expression for a multi-line comment, which made some of the following code needed for the program to run to be commented out. The same goes for line 32, where I added a missing /, which was needed to complete the expression for a multi-line comment, which made some of the following code needed for the program to run to be commented out. I added void run\_game(int computer\_number); in line 20, that was missing a function prototype for the function run\_game. I changed ask\_to\_play(playd) to ask\_to\_play(played) in line 45, which had a misspelling of the declared played variable. I added a missing & in line 67, which failed to specify the memory location of the user input. I added int correct = 0; in line 97 which had a missing declaration of the Boolean operator. I changed the format specifier from %c to %d in line 101, which had an incorrect format specifier. I changed = to == in line 108 which had an incorrect operator to specify number to be equal to computer\_number instead of being assigned to it. I removed the semicolon in line 114, which did not adhere to the if, else if, else syntax.

**Question 3**

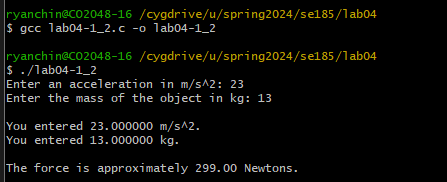
The purpose of the “-Wall” flag is to tell the compiler to warn me of potential issues in my code. It is recommended that I fix all of the messages that it gives me which does not cause compile errors but may cause issues when running my code. These warnings are often the result of unintended results issues in codes, which makes it a good practice to probably run my code with the “-Wall” flag. For example, the “-Wall” flag may warn me when I forget the return value of the scanf() statement.

# Screen Shots

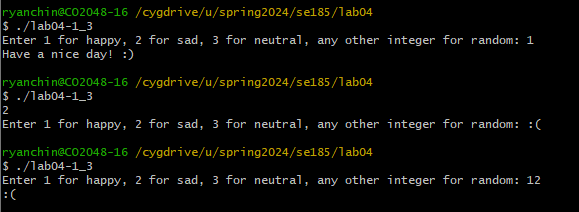
Terminal input & output lab04-1\_1.c



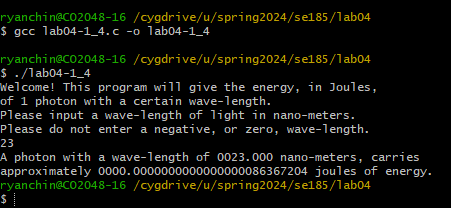
Terminal input & output lab04-1\_2.c



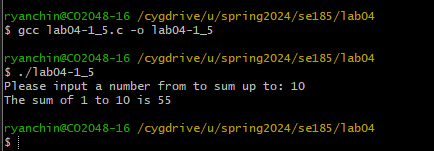
Terminal input & output lab04-1\_3.c



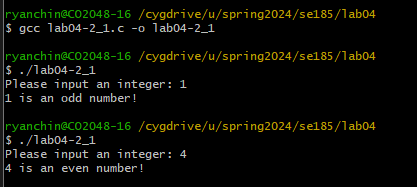
Terminal input & output lab04-1\_4.c



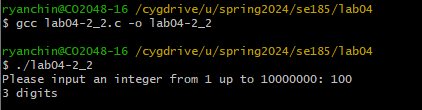
Terminal input & output lab04-1\_5.c



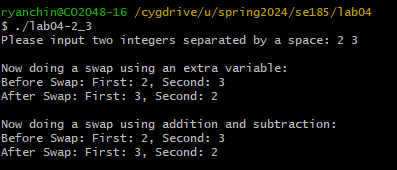
Terminal input & output lab04-2\_1.c



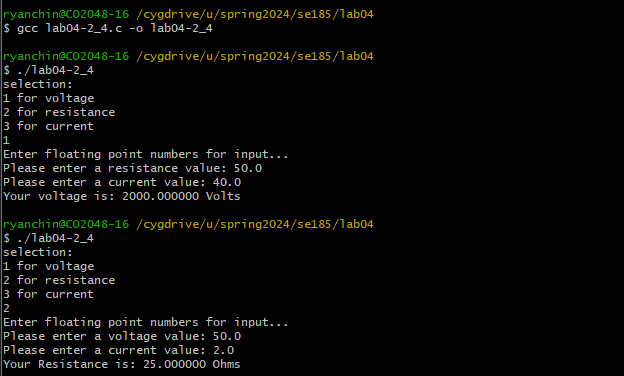
Terminal input & output lab04-2\_2.c



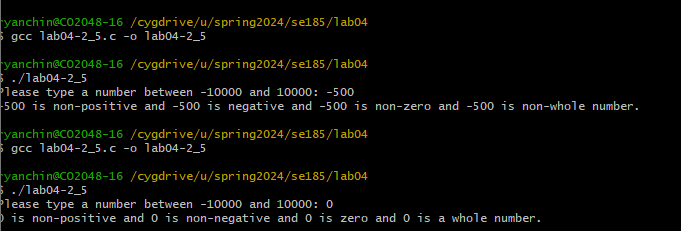
Terminal input & output lab04-2\_3.c



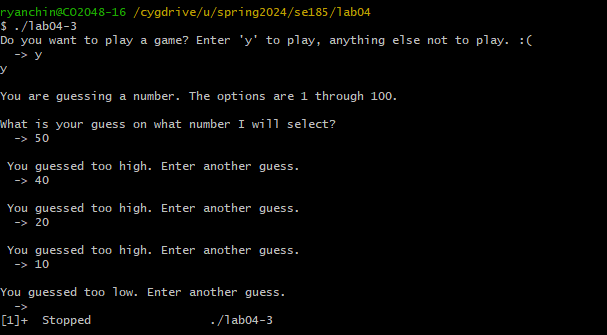
Terminal input & output lab04-2\_4.c



Terminal input & output lab04-2\_5.c



Terminal input & output lab04-3.c



# Fixed Source Code

lab04-1\_1.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Notes -

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

// Compile with gcc lab04-1\_1.c -o lab04-1\_1

// Run with ./lab04-1\_1

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

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

- Implementation -

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

int main**(**int argc**,** char **\***argv**[])**

**{**

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

//printf("Enter an integer: ") //missing semicolon

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

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

//printf("Enter another integer: ); //missing paranthesis

printf**(**"Enter another integer: "**);**

//scanf("%d", &j) //missing semicolon

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

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

**{**

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

**}** //else //missing brace

**else** **{**

//pritf("%d does not divide %d\n", i, j); // incorrect printf() syntax

printf**(**"%d does not divide %d\n"**,** i**,** j**);**

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

**}**

**return** 0**;**

**}**

lab04-1\_2.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

//void force(int mass, int acceleration); //incorrect variable type

void force(double mass, double acceleration);

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

- Notes -

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

// Compile with gcc lab04-1\_2.c -o lab04-1\_2

// Run with ./lab04-1\_2

/\* This program takes two inputs, acceleration and mass,

\* and outputs the force = mass \* acceleration \*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

double mass;

double acceleration; //added acceleration type declaration

printf("Enter an acceleration in m/s^2: ");

scanf("%lf", &acceleration);

printf("Enter the mass of the object in kg: ");

scanf("%lf", &mass);

printf("\nYou entered %lf m/s^2.\n", acceleration);

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

force(mass, acceleration);

return 0;

}

/\*\*

\* Given mass and acceleration, calculates the force exerted.

\*

\* @param mass - The given mass of an object in kilograms.

\* @param acceleration - The acceleration of an object in m/s^2.

\*/

void force(double mass, double acceleration)

{

printf("The force is approximately %.2lf Newtons.\n", mass \* acceleration);

}

lab04-1\_3.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <time.h>

#include <stdio.h> //included this library for printf() scanf()

#include <stdlib.h> //included this library for srand

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

- Prototypes -

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

void hoo();

void print\_face(int selection); //added function prototype for print\_face

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

- Notes -

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

/\* This is a simple program that takes a user inputs

\* and prints out a message based on that input \*/

// Compile with gcc lab04-1\_3.c -o lab04-1\_3

// Run with ./lab04-1\_3

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

- Implementation -

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

int main(int argc, char \*argv[])

{

srand(time(NULL));

int selection = 0;

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

scanf("%d", &selection);

if (selection < 1 || selection > 3)

{

selection = rand() % 4;

}

print\_face(selection);

return 0;

}

/\*\*

\* Prints a funny face.

\*

\* @param selection - The inputted value which determines which face to print.

\*/

void print\_face(int selection)

{

if (selection == 1)

{

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

} else if (selection == 2)

{

printf(":(\n");

} else if (selection == 3)

{

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

} else

{

hoo();

}

}

/\*\*

\* Prints an owl face.

\*/

void hoo()

{

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

}

lab04-1\_4.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

#include <math.h>

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

- Notes -

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

// Compile with gcc lab04-1\_4.c -o lab04-1\_4

// Run with ./lab04-1\_4

/\* This program calculates the energy of one photon

\* of user-inputted wave-length of light \*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

//double speed\_of\_light!;

double speed\_of\_light;

//double wave-length;

double wave\_length;

//double ~length\_in\_meters;

double length\_in\_meters;

//double plank const;

double plank\_const;

//double 0energy;

double energy;

//plank const = 6.62606957 \* pow(10, -34); // Planck's constant

plank\_const = 6.62606957 \* pow(10, -34); // Planck's constant

//speed\_of\_light! = 2.99792458 \* pow(10, 8); // Constant for the speed of light

speed\_of\_light = 2.99792458 \* pow(10, 8); // Constant for the speed of light

//wave-length = 0;

wave\_length = 0;

//~length\_in\_meters = 0;

length\_in\_meters = 0;

//0energy = 0;

energy = 0;

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

printf("of 1 photon with a certain wave-length.\n");

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

printf("Please do not enter a negative, or zero, wave-length.\n");

//scanf("%lf", &wave-length);

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

//if (wave-length > 0.0)

if (wave\_length > 0.0)

{

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

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

//0energy = (plank const \* speed\_of\_light!) / ~length\_in\_meters; // Calculating the energy of 1 photon

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

/\*printf("A photon with a wave-length of %08.3lf nano-meters, carries "

"\napproximately %030.25lf joules of energy.", wave-length, 0energy);\*/

printf("A photon with a wave-length of %08.3lf nano-meters, carries "

"\napproximately %030.25lf 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;

}

lab04-1\_5.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

int sum\_function(int number);

// int main(); //not needed

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

- Notes -

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

// Compile with gcc lab04-1\_5.c -o lab04-1\_5

// Run with ./lab04-1\_5

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

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

- Implementation -

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

int main(int argc, char \*argv[])

{

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\_function(input));

return 0;

}

/\*int main(int argc, char \*argv[])

{

printf("Sum is 32!\n"); //this entire part is not needed

}\*/

/\*\*

\* Calculates the sum of 1 to number of a given number.

\*

\* @param number - The number that determines what the sum will stop adding at.

\* @return - The sum of 1 to the given number.

\*/

int sum\_function(int number)

{

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

}

lab04-2\_1.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

int is\_odd(int number);

int is\_even(int number);

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

- Notes -

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

// Compile with gcc lab04-2\_1.c -o lab04-2\_1

// Run with ./lab04-2\_1

/\* This program accepts a user input and determines

\* if the integer is an odd or an even number \*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

int input = 0; //changed == to =

printf("Please input an integer: ");

scanf("%d", &input);

if (is\_odd(input) == 1) //changed = to ==

{

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

}

if (is\_even(input) == 1) // changed = to ==

{

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

}

return 0;

}

/\*\*

\* Determines whether the given number is even.

\*

\* @param number - The number in question of even status.

\* @return - True if the given number was even.

\*/

int is\_even(int number)

{

return !(number % 2);

}

/\*\*

\* Determines whether the given number is odd.

\*

\* @param number - The number in question of odd status.

\* @return - True if the given number was odd.

\*/

int is\_odd(int number)

{

return number % 2;

}

lab04-2\_2.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

void how\_many\_whole\_digits(int number);

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

- Notes -

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

/\* This program calculates the number of digits in a number from 1 to 10000000 \*/

// Compile with gcc lab04-2\_2.c -o lab04-2\_2

// Run with ./lab04-2\_2

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

- Implementation -

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

int main(int argc, char \*argv[])

{

int input;

printf("Please input an integer from 1 up to 10000000: ");

scanf("%d", &input);

if (input > 10000000 || input < 1)

{

printf("Invalid number!\n");

return -1;

}

how\_many\_whole\_digits(input);

return 0;

}

/\*\*

\* This function divides a number by the 10^n, to

\* see if the divided number has "n" digits

\*

\* @param number - The number to determine how many whole digits exist within.

\*/

void how\_many\_whole\_digits(int number)

{

if ((int) number / 10000000 != 0) //changed double to int

{

printf("8 digits\n");

} else if ((int) number / 1000000 != 0) //changed double to int

{

printf("7 digits\n");

} else if ((int) number / 100000 != 0) //changed double to int

{

printf("6 digits\n");

} else if ((int) number / 10000 != 0) //changed double to int

{

printf("5 digits\n");

} else if ((int) number / 1000 != 0) //changed double to int

{

printf("4 digits\n");

} else if ((int) number / 100 != 0) //changed double to int

{

printf("3 digits\n");

} else if ((int) number / 10 != 0) //changed double to int

{

printf("2 digits\n");

} else if ((int) number / 1 != 0) //changed double to int

{

printf("1 digit\n");

}

}

lab04-2\_3.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

void variable\_swap(int i, int j);

void math\_swap(int i, int j);

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

- Notes -

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

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

\* swaps their values using two different methods \*/

// Compile with gcc lab04-2\_3.c -o lab04-2\_3

// Run with ./lab04-2\_3

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

- Implementation -

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

int main(int argc, char \*argv[])

{

int first = 0, second = 0;

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

scanf("%d %d", &first, &second); //changed data specifier from %lf to %d

printf("\n");

variable\_swap(first, second);

printf("\n");

math\_swap(first, second);

return 0;

}

/\*\*

\* Swaps the values of two integers using a temp variable.

\*

\* @param i - The first value to be swapped.

\* @param j - The second value to be swapped.

\*/

void variable\_swap(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 temp = i;

i = j;

j = temp;

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

}

/\*\*

\* Swaps the values of two integers without using a temp variable.

\*

\* @param i - The first value to be swapped.

\* @param j - The second value to be swapped.

\*/

void math\_swap(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);

}

lab04-2\_4.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

double voltage(double resistance, double current);

double resistance(double voltage, double current);

double current(double voltage, double resistance);

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

- Notes -

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

// Compile with gcc lab04-2\_4.c -o lab04-2\_4

// Run with ./lab04-2\_4

/\* This program calculates values of resistances,

\* voltages, or current using Ohm's Law \*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

int selection = 0;

double v, i, r; //changed variable type from int to double

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

scanf("%d", &selection);

if (selection > 3 || selection < 1)

{

printf("Invalid number\n");

return -1;

}

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

if (selection == 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 (selection == 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 (selection == 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;

}

/\*\*

\* Given the resistance and current, calculates and returns the voltage.

\*

\* @param resistance - The resistance used to calculate the voltage.

\* @param current - The current used to calculate the voltage.

\* @return - The voltage calculated from the resistance and current.

\*/

double voltage(double resistance, double current)

{

return resistance \* current;

}

/\*\*

\* Given the voltage and current, calculates and returns the resistance.

\*

\* @param voltage - The voltage used to calculate the resistance.

\* @param current - The resistance used to calculate the resistance.

\* @return - The resistance calculated from the voltage and current.

\*/

double resistance(double voltage, double current)

{

return voltage / current;

}

/\*\*

\* Given the voltage and resistance, calculates and returns the current.

\*

\* @param voltage - The voltage used to calculate the current.

\* @param resistance - The resistance used to calculate the current.

\* @return - The current calculated from the voltage and resistance.

\*/

double current(double voltage, double resistance)

{

return voltage / resistance;

}

lab04-2\_5.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

int is\_positive(int number);

int is\_negative(int number);

int is\_zero(int number);

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

- Notes -

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

// Compile with gcc lab04-2\_5.c -o lab04-2\_5

// Run with ./lab04-2\_5

/\* This program takes in an integer from the user and

\* checks to see if it is a whole number. Additionally,

\* it will tell the user if the number is positive,

\* negative, or zero.

\*

\* Example:

\* $ ./lab04\_2-5

\* $ Please type a number between -10000 and 10000: -500

\* $ -500 is non-positive and -500 is non-zero and -500 is non-whole number.

\*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

int number;

printf("Please type a number between -10000 and 10000: ");

scanf("%d", &number);

if (number > 10000 | number < -10000)

{

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

return -1;

}

if ((is\_positive(number) & !is\_negative(number)) | is\_zero(number))

{

printf("%d is a whole number.\n", number);

} else

{

printf("%d is non-whole number.\n", number);

}

return 0;

}

/\*\*

\* Determines if the given number is positive.

\*

\* @param number - The number in question of whether it is positive or not.

\* @return - Whether the given number is positive.

\*/

int is\_positive(int number)

{

if (number > 0)

{

printf("%d is positive and ", number);

return 1;

}

printf("%d is non-positive and ", number);

return 0;

}

/\*\*

\* Determines if the given number is negative.

\*

\* @param number - The number in question of whether it is negative or not.

\* @return - Whether the given number is negative.

\*/

int is\_negative(int number)

{

if (number < 0)

{

printf("%d is negative and ", number);

return 1;

}

printf("%d is non-negative and ", number);

return 0;

}

/\*\*

\* Determines if the given number is 0.

\*

\* @param number - The number in question of whether it is 0 or not.

\* @return - Whether the given number is 0.

\*/

int is\_zero(int number)

{

if (number == 0)

{

printf("%d is zero and ", number); //corrected n to number

return 1;

}

printf("%d is non-zero and ", number);

return 0;

}

lab04-3.c

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

- SE 185: Lab 04 - Debugging Code -

- Name: Ryan Aun Shern Chin -

- Section: 2 -

- NetID: ryanchin -

- Date: 2/14/24 -

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

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

- Includes -

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

#include <stdio.h>

#include <stdlib.h> // added library

#include <time.h>

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

- Prototypes -

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

char ask\_to\_play(int times\_played);

void run\_game(int computer\_number); //added prototype

int select\_random\_number();

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

- Notes -

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

// Compile with gcc lab04-3.c -o lab04-3

// Run with ./lab04-3

/\* This program will play a simple Guessing Game with the computer. \*/

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

- Implementation -

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

int main(int argc, char \*argv[])

{

char prompt = '-';

int played = 0, computer\_guess = 0;

prompt = ask\_to\_play(played);

played = 1;

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

{

computer\_guess = select\_random\_number();

run\_game(computer\_guess);

prompt = ask\_to\_play(played); //changed playd to played

}

printf("\n\nThanks for playing!\n");

return 0;

}

/\*\*

\* Asks the player if they want to play the Guessing Game.

\*

\* @param played\_before - Whether the player has played a round of the game before or not.

\* @return - Whether the player wants to play again or not.

\*/

char ask\_to\_play(int played\_before)

{

char yes\_or\_no;

if (!played\_before) /\* This line does not contain an error \*/

{

printf("Do you want to play a game? "

"Enter 'y' to play, anything else not to play. :(\n -> ");

scanf(" %c", &yes\_or\_no); // added &

} else

{

scanf(" %c", &yes\_or\_no);

}

printf("%c", yes\_or\_no);

return yes\_or\_no;

}

/\*\*

\* Generates a random number between 1 to 100, inclusive.

\*

\* @return - A number between 1 and 100, inclusive.

\*/

int select\_random\_number()

{

srand(time(NULL));

return rand() % 100;

}

/\*\*

\* Starts the Guessing Game for you to play!

\*

\* @param computer\_number - The randomly generated number to be used for the game.

\*/

void run\_game(int computer\_number)

{

int number = 0;

int correct = 0; // declared boolean operator

printf("\n\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); //changed format specifier from %c to %d

while (!correct) /\* This line does not contain an error \*/

{

if (number < 1 || number > 100)

{

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

} else if (number == computer\_number) //changed = to ==

{

printf("\nThe number was %d!\n", computer\_number);

printf("\nYou guessed the number correctly!\n\n"

"Do you want to play again? ('y' for yes)\n -> ");

correct = 1;

} else if (number < computer\_number) //removed semicolon

{

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

} else

{

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

}

scanf("%d", &number);

}

}