**LAB REPORT**

**LAB #4**

**SECTION #2**

**FULL NAME**

**Alvin John Thomas**

**SUBMISSION DATE:**

**9/26/2023**

**DATE**

**9/19/2023**

# Problem

There are 5 programs with compiler errors.

# Analysis

See the errors in the compiler and fix them.

# Design

Using the line and character numbers shown in the compiler, we can find out where the error is.

# Testing

I tried compiling the program each time and then corrected each error until there was none left

# Comments

The corrected programs are given below:

4-1\_1:

\*----------------------------------------------------------------------------

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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: ")

printf("Enter an integer: "); //Added a semicolon

scanf("%d", &i);

//printf("Enter another integer: );

printf("Enter another integer: "); //Added a quotation mark

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

scanf("%d", &j); //Added a semicolon

if (j % i == 0)

{

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

} else

{ //Added a curly brace

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

printf("%d does not divide %d\n", i, j); //Added 'n' in print

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

}

return 0;

}

4-1\_2:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

//void force(int mass, int acceleration);

void force(double mass, double acceleration); //Replaced int with double

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

- 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; //Declared a variable 'acceleration'

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);

}

4-1\_3:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- Includes -

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

#include <time.h>

#include <stdio.h> //Included <stdio.h>

#include <stdlib.h> //Included <stdlib.h>

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

- Prototypes -

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

void hoo();

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

- 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");

}

4-1\_4:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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; //Removed '!'

//double wave-length;

double wave\_length; //Replaced - with \_

//double ~length\_in\_meters;

double length\_in\_meters; //Removed '~'

//double plank const;

//double 0energy;

double energy0; //Moved '0' to end of variable name

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

const double plank = 6.62606957 \* pow(10, -34); // Moved const and initialized variable

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

speed\_of\_light = 2.99792458 \* pow(10,8); //Removed '!'

//wave-length = 0;

wave\_length = 0; //Replaced '-' with '\_'

//~length\_in\_meters = 0;

length\_in\_meters = 0; //Removed '~'

//0energy = 0;

energy0 = 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); //Replaced '-' with '\_'

//if (wave-length > 0.0)

if (wave\_length > 0.0) //Replaced '-' with '\_'

{

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

length\_in\_meters = wave\_length / pow(10,9); //Removed '~' and replaced '-' with '\_'

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

energy0 = (plank \* speed\_of\_light) / length\_in\_meters; //Moved 0 to end of variable name, removed 'const', '!' and '~'

/\*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, energy0); //Replaced '-' with '\_' and moved 0 to end of variable name

} else

{

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

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

}

return 0;

}

4-1\_5:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- Includes -

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

#include <stdio.h>

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

- Prototypes -

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

int sum\_function(int number);

int main();

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

- 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");

}\*/ //Removed second main function

/\*\*

\* 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;

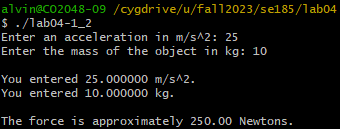
}

# Screen Shots

4-1\_1



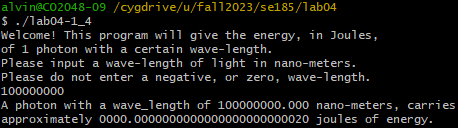
4-1\_2



4-1\_3



4-1\_4



4-1\_5



# Problem

There are 5 programs with logic errors.

# Analysis

Even though the code compiles, the output is not as expected.

# Design

I tried to figure out where the logical errors were and corrected them.

# Testing

I tried changing the code and recompiled it till the desired output was shown

# Comments

The corrected programs are given below:

4-2\_1:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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;

int input = 0; //Replaced '==' with '='

printf("Please input an integer: ");

scanf("%d", &input);

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

if (is\_odd(input) != 0) //Replaced '=' with '!=' and '1' with '0'

{

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

}

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

if (is\_even(input) == 1) //Replaced '=' with '==' and 1 with 0

{

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;

}

4-2\_2:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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 ( number / 10000000 != 0) //Removed all (double)

{

printf("8 digits\n");

} else if ( number / 1000000 != 0)

{

printf("7 digits\n");

} else if ( number / 100000 != 0)

{

printf("6 digits\n");

} else if ( number / 10000 != 0)

{

printf("5 digits\n");

} else if ( number / 1000 != 0)

{

printf("4 digits\n");

} else if ( number / 100 != 0)

{

printf("3 digits\n");

} else if ( number / 10 != 0)

{

printf("2 digits\n");

} else if ( number / 1 != 0)

{

printf("1 digit\n");

}

}

4-2\_3:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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("%lf %lf", &first, &second);

scanf("%d %d", &first, &second); //Replaced %lf with %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);

}

4-2\_4:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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;

//int v, i, r;

double v, i, r; //Replaced int with 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;

}

4-2\_5:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/19/2023 -

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

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

- 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)

if (number == 0) //Changed '==' to '='

{

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

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

return 1;

}

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

return 0;

}

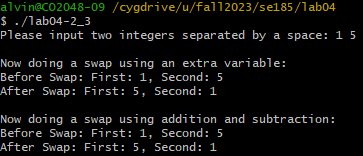
# Screen Shots

4-2\_1

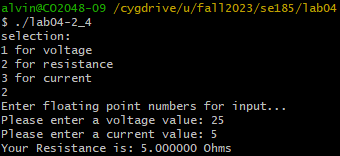


4-2\_2  

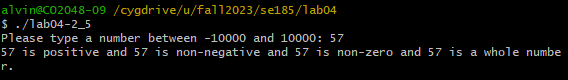

4-2\_3



4-2\_4



4-2\_5



# Problem

The program has a mixture of both compiler and logic errors.

# Analysis

I tried compiling the code and fixing the errors each time.

# Design

I used GCC compiler flag “-Wall”

# Testing

I kept on recompiling the code until the desired output was produced.ss

# Comments

The corrected program is given below:

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

- SE 185: Lab 04 - Debugging Code -

- Name: Alvin Thomas -

- Section: 2 -

- NetID: alvin -

- Date: 9/26/2023 -

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

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

- Includes -

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

#include <stdio.h>

#include <time.h>

/\*----------------------------------------------------------------------------- //Added a \*

- Prototypes -

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

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

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 a /

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(playd);

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

}

//} //Removed curly brace

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);

scanf(" %c", &yes\_or\_no); //Added a & before the variable name

} 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 run\_game(int computer\_number) //Change void to int

{

int number = 0;

int correct = 0; //Initialized variable correct

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("%c", &number);

scanf("%d", &number); //Changed %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)

} 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);

} 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);

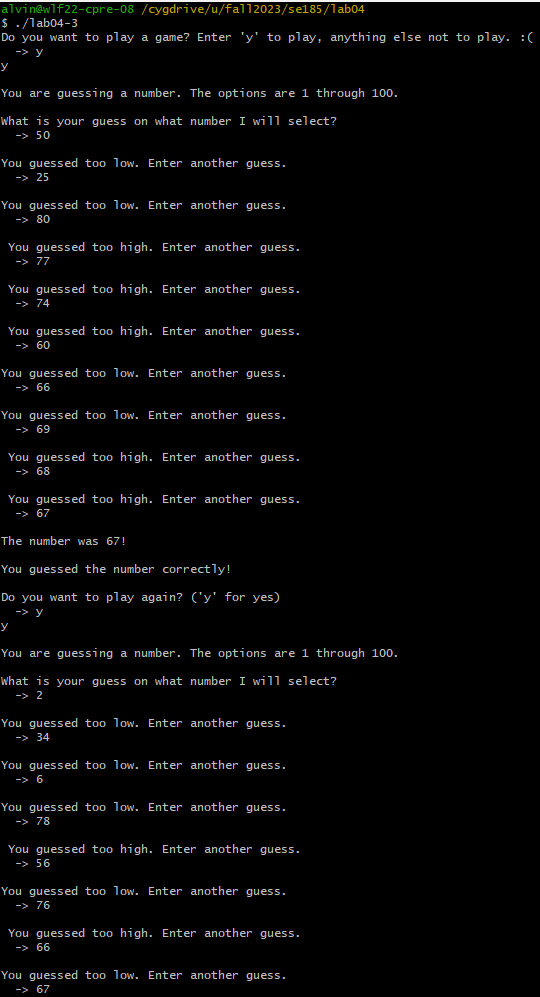
}

return 0; //Added a return 0 statement

}

# Screen Shots

4-3



Answers of the Questions:

1 and 2.

In lab04-1\_1, I added a semicolon in line 29, a quotation mark in line 33, a semicolon in line 35, a curly brace in line 42

In lab04-1\_2, I replaced int with double in line 18, declared a variable ‘acceleration’ in line 34

In lab04-1\_3, I included <stdio.h> and <stdlib.h> in lines 13 and 14

In lab04-1\_4, I removed ! in line 29, replaced – with \_ in line 31, removed ~ in line 33, moved 0 to the end of the variable name in line 36, removed ! in line 41, replaced – with \_ in line 43, removed ~ in line 45

In lab04-1\_5, I removed second main function from lines 44 to 47.

In lab04-2\_1, I replaced == with = in line 35, replaced = with == and 1 with 0 in lines 41 and 47

In lab04-2\_2, I removed “(double)” from all the if and else if statements.

In lab04-2\_3, I replaced %lf with %d in line 38.

In lab04-2\_4, I replaced int with double in line 38.

In lab04-2\_5, I changed == to = in line 111, changed n to number in line 114.

In lab04-3, I added a \* in line 16, added a / in line 32, changed playd to played in line 46, removed curly brace in line 48, added & before the variable name in line 70, changed void to int in line 98, initialized the variable correct in line 101, change %c to %d in line 106, changed = to == in line 112, removed semicolon in line 120, added a return 0; statement in line 130

3.

The -Wall flag lets the compiler show all warning messages. No, we don’t have to fix all the messages that it gives us, for eg we don’t have to fix warnings.