

**Debugging**

**LAB # 4**

**SECTION # Tuesdays at 2:10 PM**

**Thriambak Giriprakash**

**3/2/21**

**DATE: 3/2/21**

## Problem

This section of the lab was centered around fixing some syntax errors in code that was given to us.

## Analysis

The GCC compile function in the Cygwin terminal gives error messages when a syntax error is found which gives a line number and a potential fix to the solution. I used these error messages to fix the errors in the code.

## Design

No design just fixing errors

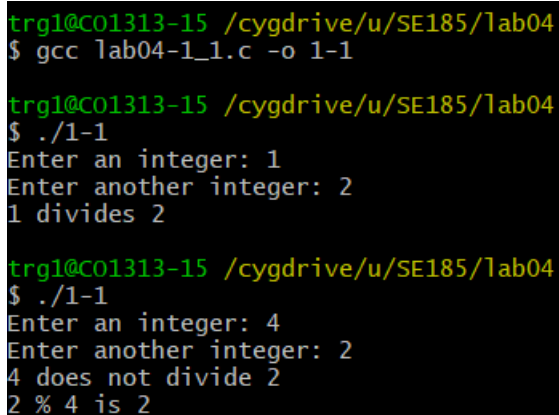
## Testing

Running the code on the Cygwin terminal after fixing the errors to see if the errors were fixed.

## Comments

None.

## Screen Shots



```
trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-1_1.c -o 1-1

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-1
Enter an integer: 1
Enter another integer: 2
1 divides 2

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-1
Enter an integer: 4
Enter another integer: 2
4 does not divide 2
2 % 4 is 2
```

1-1:

Code: /\*-----  
-----

```

-                                     SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-
-                                     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 missing semicolon
    scanf("%d", &i);

    //printf("Enter another integer: ");
    printf("Enter another integer: "); //added missing quote mark
    //scanf("%d", &j)
    scanf("%d", &j); //added missing semicolongcc

    if (j % i == 0)
    {
        printf("%d divides %d\n", i, j);
    }
    else

```

```

    {    //added missing bracket

        //printf("%d does not divide %d\n", i, j);
        printf("%d does not divide %d\n", i, j); //fixed typo in printf -
>printf
        printf("%d %% %d is %d\n", j, i, (j % i));
    }

    return 0;
}

```

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-1_2.c -o 1-2

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-2
Enter an acceleration in m/s^2: 9.81
Enter the mass of the object in kg: 10

You entered 9.810000 m/s^2.
You entered 10.000000 kg.

The force is approximately 98.10 Newtons.

```

1-2:

```

Code: /*-----
-----

-                                     SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----
-                                     Includes
-
-----
*/
#include <stdio.h>

/*-----
-----
-                                     Prototypes
-
-----
*/
//void force(int mass, int acceleration);

```

```

void force(double mass, double acceleration); //changed types taken by force
function to match the function code

/*-----
-
-                               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 mass, acceleration; //initialized acceleration variable

    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);
}

```

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-1_3.c -o 1-3

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 1
Have a nice day! :)

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 2
:(

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 3
Meh :\

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 5
Have a nice day! :)

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 4
:(

```

1-3:

```

Code: /*-----
-----

-                               SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----
-                               Includes
-
-----
*/
#include <time.h>
#include <stdio.h> //included stdio
#include <stdlib.h> //included stdlib
/*-----
-----
-                               Prototypes
-

```

```

-----
*/
void hoo();
void print_face(int selection); //declared 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");
    }
}

```

```
/**
 * Prints an owl face.
 */
```

1-4: approximately 0000.0000000000000000496611421 joules of energy.

```

-                                     SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-

```



```

/*-----
-
-                               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 exclamation
    //double wave-length;
    double wave_length; //changed - to _
    //double ~length_in_meters;
    double length_in_meters; //removed tildy sign
    //double plank const;
    double plank_const ; //placed underscore to make it a single name
    //double 0energy;
    double energy; //removed 0 before energy

    //plank const = 6.62606957 * pow(10, -34); // Planck's constant
    plank_const = 6.62606957 * pow(10, -34); //changed to match variable name
    //speed_of_light! = 2.99792458 * pow(10, 8); // Constant for the speed of
light
    speed_of_light = 2.99792458 * pow(10, 8); //chanegd to match variable
name
    //wave-length = 0
    wave_length = 0; //changed to match variable name
    //~length_in_meters = 0;
    length_in_meters = 0; //changed to match variable name
    //0energy = 0;
    energy = 0; //changed to match variable name

    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); //changed to match variable names

    if (wave_length > 0.0) //changed to match variable name
    {
        //~length_in_meters = wave-length / pow(10, 9); // Converting nano-
meters to meters
        length_in_meters = wave_length / pow(10, 9); //changed to match
variable names

```

```

        //0energy = (plank_const * speed_of_light!) / ~length_in_meters; //
Calculating the energy of 1 photon
        energy = (plank_const * speed_of_light) / length_in_meters; //changed
to match variable names
        //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); //changed to match variable names
    } else
    {
        printf("Sorry, you put in an invalid number.");
        printf("Please rerun the program and try again.");
    }

    return 0;
}

```

```

1-5: trgl@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-1_5.c -o 1-5

trgl@C01313-15 /cygdrive/u/SE185/lab04
$ ./1-5
Please input a number from to sum up to: 4
The sum of 1 to 4 is 10

```

```

Code: /*-----
-----

-                               SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----
-                               Includes
-
-----
*/
#include <stdio.h>

/*-----
-

```

```

-                                     Prototypes
-
-----
*/
int sum_function(int number);

//int main(); //removed couple main declaration

/*-----
-
-                                     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[]) //removed double main declaration
//{
//    printf("Sum is 32!\n");
//}

/**
 * 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;
}

```

## Problem

This section of the lab was centered around fixing some logic errors in code that was given to us.

## Analysis

The Cygwin terminal does not help with logic errors so I had to go in the code file and read the code to find the errors

## Design

No design just fixing errors

## Testing

Running the code on the Cygwin terminal after fixing the errors to see if the errors were fixed.

## Comments

None.

## Screen Shots

```
2-1: /*-----
-----

-                               SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----

-                               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; //syntax error

    printf("Please input an integer: ");
    scanf("%d", &input);

    if (is_odd(input) == 1) // changed "=1" to ==1
    {
        printf("%d is an odd number!\n", input);
    }

    if (is_even(input) == 1) // changed "=1" to ==1
    {
        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;
}

```

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-2_1.c -o 2-1

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-1
Please input an integer: 3
3 is an odd number!

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-1
Please input an integer: 6
6 is an even number!

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-1
Please input an integer: 5001
5001 is an odd number!

```

Code:

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-2_2.c -o 2-2

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-2
Please input an integer from 1 up to 10000000: 5
1 digit

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-2
Please input an integer from 1 up to 10000000: 500
3 digits

```

2-2:

Code: /\*-----  
-----

```

-                                     SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-

```

```

- NetID:
-
- Date:
-
-----
*/

/*-----
-
-                               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)
    // {
    //     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");
    // }

    //removed (double) type change so int division can occur
    if ( number / 10000000 != 0)
    {
        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 / 10 != 0 )
    {
        printf("1 digit\n");
    }
}

```

```

trgl@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-2_3.c -o 2-3

trgl@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-3
Please input two integers separated by a space: 2 3

Now doing a swap using an extra variable:
Before Swap: First: 2, Second: 3
After Swap: First: 3, Second: 2

Now doing a swap using addition and subtraction:
Before Swap: First: 2, Second: 3
After Swap: First: 3, Second: 2

```

2-3:

```

Code: /*-----
-----

-                               SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----
-                               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); //chanegd the scan to read an int
    instead of a double

    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);
}

```

```

trg1@c01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-2_4.c -o 2-4

trg1@c01313-15 /cygdrive/u/SE185/lab04
$ ./2-4
selection:
1 for voltage
2 for resistance
3 for current
1
Enter floating point numbers for input...
Please enter a resistance value: 4.0
Please enter a current value: 4.0
Your voltage is: 16.000000 Volts

```

2-4:

```

Code: /*-----
-----

-                                     SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-----
-                                     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; //variables were initiated to be ints instead of doubles

    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;
}

```

2-5:

```
Code: /*-----
-----

-                               SE 185: Lab 04 - Debugging Code
-
-   Name:
-
-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-
-                               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) //changed =0 to ==0
    {
        //printf("%d is zero and ", n);
        printf("%d is zero and ", number); //changed n to number
        return 1;
    }

    printf("%d is non-zero and ", number);
    return 0;
}

```

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ gcc lab04-2_5.c -o 2-5

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./2-5
Please type a number between -10000 and 10000: -500
-500 is non-positive and -500 is negative and -500 is non-zero and -500 is non-whole number.

```

## Problem

This section of the lab was aimed at combining parts 1 and 2

## Analysis

Needed to combine the analytical tasks performed in parts 1 and 2 to fix part 3

## Design

No design just fixing errors



## Testing

Running the code on the Cygwin terminal after fixing the errors to see if the errors were fixed.

## Comments

None.

## Screen Shots

```

trg1@C01313-15 /cygdrive/u/SE185/lab04
$ ./3
Do you want to play a game? Enter 'y' to play, anything else not to play.
-> y
y
You are guessing a number. The options are 1 through 100.
What is your guess on what number I will select?
-> 38
You guessed too low. Enter another guess.
-> 50
You guessed too low. Enter another guess.
-> 90
You guessed too low. Enter another guess.
-> 100
You guessed too high. Enter another guess.
-> 98
You guessed too high. Enter another guess.
-> 97
You guessed too high. Enter another guess.
-> 96
You guessed too high. Enter another guess.
-> 94
You guessed too high. Enter another guess.
-> 93
The number was 93!
You guessed the number correctly!
Do you want to play again? ('y' for yes)
-> n
n
Thanks for playing!

```

Code:

```

/*-----
-
-                               SE 185: Lab 04 - Debugging Code
-
-   Name :
-

```

```

-   Section:
-
-   NetID:
-
-   Date:
-
-----
*/

/*-----
-
-                               Includes
-
-----

*/
#include <stdio.h>
#include <stdlib.h> //included stdlib
#include <time.h>

/*-----
-- //added missing *
-                               Prototypes
-
-----

-*/
char ask_to_play(int times_played);
void run_game(int computer_number); //added function declaration
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 missing /

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); //fixed typo playd -> 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);
        scanf(" %c", &yes_or_no); //added & symbol
    } 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 number = 0, correct = 0; //correct was not declared

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 this 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) // was previously: number =
computer_number which is not a boolean argument
    {
        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) // was previously: (number <
computer_number); removed ';'
    {
        printf("\nYou guessed too low. Enter another guess.\n -> ");
    } else
    {
        printf("\n You guessed too high. Enter another guess.\n -> ");
    }

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

```

Additional questions: the wall flag is there to show any potential errors that may be in the code you are writing for example if a scan is supposed to read an int but the variable is a character it will show up as a warning when gcc -Wall.