Problem

Problem 1: I am given 5 different C programs that I have to check and then fix the compilation errors for

Problem 2: I am given 5 different C programs and need to fix them to produce to expected outputs

Problem 3: I have to do a combination of fixing compilation errors and fixing logic errors in order for a given C program to work as intended

Analysis

Problem 1: I need to run the code for each of the programs then using the information that the errors from the compiler give me, I need to fix each program

Problem 2: using the information from any compiler errors, as well as inspecting the output once it compiles correctly, I need to deduce why the output isn't what is expected

Problem 3: First I have to fix all the compilation issues with the help of the -Wall flag, and then I have to very carefully inspect the program's code as well as its output to find any possible logic errors

Design

Problem 1: I do just that, I run all the programs then carefully one by one fix all the errors that I am told exist by the compiler

Problem 2: The first program was just a compiler error so it was pretty easy. The second program I noticed that the if statements were checking if a double being divided by a number will be 0 as its logic, and it never will be so I fixed that. The third program just I noticed that it was scanning doubles instead of integers so that was a quick fix. The fourth program the only issue was that the variables were declared as the wrong type, it should've been double instead of int. The last problem just had the if statements written incorrectly as number = 0 instead of == which causes it to not function correctly.

Problem 3: Firstly, I fixed several issues with the compilation such as functions or variables not being declared or a scanf scanning the wrong type of variable. Next I fixed any logic errors due to small stuff such as not enough = signs or comments not being started/ended correctly

Testing

Problem 1: after repeated compiling and fixing of the codes, I got each problem to run correctly with an expected output and no errors

Problem 2: Everything pretty much ran as expected once I figured out what was causing the unexpected output

Problem 3: Everything ran as expected after fixing all the compiler errors and then noticing where the unexpected ouputs were coming from

Comments

Problem 1: For the first program the errors were in what is is now line 28 missing semicolon, line 33 missing “,line 37 missing semicolon, line 44 missing bracket, line 48 typo. The second program’s error was only in what is now line 37 in that there was a missing declaration of a variable. The Third programs only issue was missing a prototype function definition on line 22 and not including stdio.h and stdiolib.h on what is now line 15-16. The fourth program has a ton of issues that are mostly variable names not being defined with correct syntax. These errors occur on what is now line 28, 31,34,37,49,44,47,50,53,56,65,69,73,76 and 80. The fifth programs only error is a second definition of the main function in lines 47-49.

Problem 2: The first program had an error in line 34 of an extra equals sign during variable declaration and line 41 and 50 in lacking an equals sign in an if statement. The second program has an issue in lines 56 -80 in that it casts the variable number as a double which messes up the logic and causes it not to work as intended. The third problem’s only issue is using lf instead of d in a scanf on line 37. The fourth function just defines variables as the wrong type in line 37. The fifth program only has two errors, needed an = sign in line 110 for the if statement to work properly and in line 114 where n needs to be written as number

Problem 3: The wall flag helps you catch even more issues than just the ones that the basic compiler gives you. You do not have to fix all the issues it says there are since depending on what it is you can still compile the program, however it is a good idea to do so

Screenshots

Screenshot 1 for Problem 1:

```c
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 "
    printf("Enter another integer: ");
    //missing semicolon
    //scanf("%d", &j)
    scanf("%d", &j);

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

    } //else
        //missing bracket
    else {

        //pritf("%d does not divide %d\n", i, j);
        //spelled printf wrong
        printf("%d does not divide %d\n", i, j);
        printf("%d %% %d is %d\n", j, i, (j % i));
    }

    return 0;
}
```

Screenshot 2 for Problem 1:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-1_1
Enter an integer: 4
Enter another integer: 3
4 does not divide 3
3 % 4 is 3
```

Screenshot 3 For Problem 1:

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

/*-------------------------------------------------------------------------------
-                                Prototypes                                     -
-------------------------------------------------------------------------------*/
//void force(int mass, int acceleration);
//is not declared the same way below in the function
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;
    //acceleration was not a declared variable
    double 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);
}
```

Screenshot 4 for Problem 1:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-1_2
Enter an acceleration in m/s^2: 4
Enter the mass of the object in kg: 3

You entered 4.000000 m/s^2.
You entered 3.000000 kg.

The force is approximately 12.00 Newtons.
```

Screenshot 5 for Problem 1:

```c
#include <time.h>
//stdio h and stdlib are not included
#include<stdio.h>
#include<stdlib.h>
/*------------------------------------------------------------------------------
-                                Prototypes                                    -
------------------------------------------------------------------------------*/

void hoo();
//print_face is not defined as a prototype function
void print_face(int selection);
/*------------------------------------------------------------------------------
-                                   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();
    }
}
```

Screenshot 6 for Problem 1:

```
72          hoo();
73      }
74  }
75
76  /**
77   * Prints an owl face.
78   */
79  void hoo()
80  {
81      printf(" *___*\n {O,O}\n/)___)\n_\"__\"_\n");
82  }
83
```

Screenshot 7 for Problem 1:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-1_3
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 3
Meh :\
```

Screenshot 8 for Problem 1:

```c
int main(int argc, char *argv[])
{
    //double speed_of_light!;
    //shouldnt have exclamation point
    double speed_of_light;
    //double wave-length;
    //shouldnt have a hyphen
    double wave_length;
    //double ~length_in_meters;
    //shouldnt have a tilda
    double length_in_meters;
    //double plank const;
    //const should be before plank not after, also should have the definition of what it is
    double const plank = 6.62606957 * pow(10, -34);  // Planck's constant
    //double 0energy;
    //variable declaration shouldnt begin with a number
    double zeroEnergy;

    //plank const = 6.62606957 * pow(10, -34);
    //cant redefine a constant

    //speed_of_light! = 2.99792458 * pow(10, 8);
    //again, variable shouldnt have an exclamation point
    speed_of_light = 2.99792458 * pow(10, 8);
    //wave-length = 0;
    //shouldnt have a hyphen
    wave_length = 0;
    //~length_in_meters = 0;
    //shouldnt have a tilda
    length_in_meters = 0;
    //0energy = 0;
    //variable name shouldnt begin with a number
    zeroEnergy = 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);
    //wave-length should be wave_length
    scanf("%lf", &wave_length);

    //if (wave-length > 0.0)
    //wave-length should be wave_length
    if (wave_length > 0.0)
    {
        //~length_in_meters = wave-length / pow(10, 9);
        //uneccesary tildaa and hyphen in variable names
        length_in_meters = wave_length / pow(10, 9); // Converting nano-meters to meters
        //0energy = (plank const * speed_of_light!) / ~length_in_meters;
        //wrong variable names as well as uneccesary use of const
        zeroEnergy = (plank * 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);
        //wrong variable names
                "\napproximately %030.25lf joules of energy.", wave_length, zeroEnergy);
    } else
```

Screenshot 9 for Problem 1:

```c
    } else
    {
        printf("Sorry, you put in an invalid number.");
        printf("Please rerun the program and try again.");
    }

    return 0;
}
```

Screenshot 10 for Problem 1:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-1_4
Welcome! This program will give the energy, in Joules,
of 1 photon with a certain wave-length.
Please input a wave-length of light in nano-meters.
Please do not enter a negative, or zero, wave-length.
409
A photon with a wave-length of 0409.000 nano-meters, carries
approximately 0000.000000000000000004856835 joules of energy.
```

Screenshot 11 for Problem 1:

```c
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: ");
    //should have from 1 to x to sum up to just for nicer readability
    printf("Please input a number from 1 to x 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");
}
*/
//second definition of main
/**
 * 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;
}
```

Screenshot 12 for Problem 1:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-1_5
Please input a number from 1 to x to sum up to: 4
The sum of 1 to 4 is 10
```

Screenshot 1 for Problem 2:

```c
int main(int argc, char *argv[])
{
    //int input == 0;
    //extra equals sign
    int input = 0;

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

    //if (is_odd(input) = 1)
    //should have another equal sign
    if (is_odd(input) == 1)
    {
        printf("%d is an odd number!\n", input);
    }

    //if (is_even(input) = 1)
    //should have another equal sign
    if (is_even(input) == 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;
}
```

Screenshot 2 for Problem 2:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-2_1
Please input an integer: 4
4 is an even number!
```

Screenshot 3 for Problem 2:

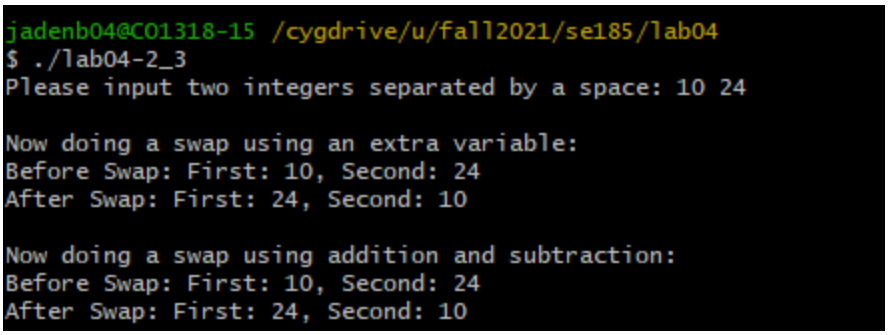
```
54  void how_many_whole_digits(int number)
55  {/*
56      if ((double) number / 10000000 != 0)
57      {
58          printf("8 digits\n");
59      } else if ((double) number / 1000000 != 0)
60      {
61          printf("7 digits\n");
62      } else if ((double) number / 100000 != 0)
63      {
64          printf("6 digits\n");
65      } else if ((double) number / 10000 != 0)
66      {
67          printf("5 digits\n");
68      } else if ((double) number / 1000 != 0)
69      {
70          printf("4 digits\n");
71      } else if ((double) number / 100 != 0)
72      {
73          printf("3 digits\n");
74      } else if ((double) number / 10 != 0)
75      {
76          printf("2 digits\n");
77      } else if ((double) number / 1 != 0)
78      {
79          printf("1 digit\n");
80      }
81      */
82      //get rid of the double casting, it makes it so anything will not be 0
83      if ( number / 10000000 != 0)
84      {
85          printf("8 digits\n");
86      } else if (number / 1000000 != 0)
87      {
88          printf("7 digits\n");
89      } else if (number / 100000 != 0)
90      {
91          printf("6 digits\n");
92      } else if ( number / 10000 != 0)
93      {
94          printf("5 digits\n");
95      } else if (number / 1000 != 0)
96      {
97          printf("4 digits\n");
98      } else if (number / 100 != 0)
99      {
100         printf("3 digits\n");
101     } else if (number / 10 != 0)
102     {
103         printf("2 digits\n");
104     } else if (number / 1 != 0)
105     {
106         printf("1 digit\n");
107     }
108 }
109
```

Screenshot 4 of Problem 2:

Screenshot 5 for Problem 2:

```c
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);
    //scanning doubles instead of integers
    scanf("%d %d", &first, &second);

    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);
}
```

Screenshot 6 for Problem 2:



Screenshot 7 for Problem 2:

```c
int main(int argc, char *argv[])
{
    int selection = 0;
    //int v, i, r;
    //variables should be doubles
    double v, i, r;

    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.
 */
```

Screenshot 8 for Problem 2:

```
 92  double voltage(double resistance, double current)
 93  {
 94      return resistance * current;
 95  }
 96
 97  /**
 98   * Given the voltage and current, calculates and returns the resistance.
 99   *
100   * @param voltage - The voltage used to calculate the resistance.
101   * @param current - The resistance used to calculate the resistance.
102   * @return - The resistance calculated from the voltage and current.
103   */
104  double resistance(double voltage, double current)
105  {
106      return voltage / current;
107  }
108
109  /**
110   * Given the voltage and resistance, calculates and returns the current.
111   *
112   * @param voltage - The voltage used to calculate the current.
113   * @param resistance - The resistance used to calculate the current.
114   * @return - The current calculated from the voltage and resistance.
115   */
116  double current(double voltage, double resistance)
117  {
118      return voltage / resistance;
119  }
120
```

Screenshot 9 for Problem 2:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./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: 20.9
Please enter a current value: 29.4
Your voltage is: 614.460000 Volts
```

Screenshot 10 for Problem 2:

```c
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;
}
```

Screenshot 11 for Problem 2:

```c
/**
 * 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)
    //should be ==
    if (number == 0)
    {
        //printf("%d is zero and ", n);
        //n should be number
        printf("%d is zero and ", number);
        return 1;
    }

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

Screenshot 12 for Problem 2:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-2_5
Please type a number between -10000 and 10000: 9
9 is positive and 9 is non-negative and 9 is non-zero and 9 is a whole number.
```

Screenshot 1 for Problem 3:

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

/*------------------------------------------------------------------------------
-                                Includes                                      -
------------------------------------------------------------------------------*/
#include <stdio.h>
#include <stdlib.h>
//stdlib was not included
#include <time.h>

/*------------------------------------------------------------------------------
-                                Prototypes                                    -
------------------------------------------------------------------------------*/
//comment was not declared properly
char ask_to_play(int times_played);
void run_game(int computer_number);
//run game was not declared about the main function before it was called
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                                -
------------------------------------------------------------------------------*/
//comment is not ended correctly
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);
        //played not spelled right
        prompt = ask_to_play(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.
 */
```

Screenshot 2 for Problem 3:

```
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);
        //missing & symbol in the function
        scanf(" %c", &yes_or_no);
    } 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;
    //correct is not declared
    int correct = 0;
    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);
    //should be %d
    scanf("%d", &number);

    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)
            //should be ==
        else if (number == computer_number)
        {
            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;
```

Screenshot 3 for Problem 3:

```
124             correct = 1;
125         } //else if (number < computer_number);
126         //semi colon shouldnt be there
127         else if (number < computer_number)
128         {
129             printf("\nYou guessed too low. Enter another guess.\n  -> ");
130         } else
131         {
132             printf("\n You guessed too high. Enter another guess.\n  -> ");
133         }
134 
135         scanf("%d", &number);
136     }
137 }
138 
```

Screenshot 4 for Problem 3:

```
jadenb04@CO1318-15 /cygdrive/u/fall2021/se185/lab04
$ ./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?
  -> 40

You guessed too low. Enter another guess.
  -> 70

You guessed too low. Enter another guess.
  -> 90

You guessed too low. Enter another guess.
  -> 99

The number was 99!

You guessed the number correctly!

Do you want to play again? ('y' for yes)
  -> y
y

You are guessing a number. The options are 1 through 100.

What is your guess on what number I will select?
  -> 40

You guessed too low. Enter another guess.
  -> 80

You guessed too low. Enter another guess.
  -> 90

You guessed too low. Enter another guess.
  -> 99

 You guessed too high. Enter another guess.
  -> 97

 You guessed too high. Enter another guess.
  -> 95

You guessed too low. Enter another guess.
  -> 96

The number was 96!

You guessed the number correctly!
```