DEBUGGING CODE

LAB 04

Section 5

ARYAN RAO

Date-09/24/21

Submission Date-09/28/21

Problem 1

Compile and debug the 5 codes provided so that the program runs correctly. Comment out the changes made and put the correct code below.

Analysis

In all the codes there are small error for example a missing parenthesis or a missing semi-colon. We can run the code in the beginning to look for the compiler errors we get and then go on correcting them.

Design

- Run the code as it is.
- Look for the errors/warnings generated by the compiler.
- Debug the code.

Testing

For the given source code manually find out the output and then compare it with the corrected code you have written

Comments

Look out for small mistakes like syntax error or the correct variable declaration for example using int instead of double.

SCREENSHOTS

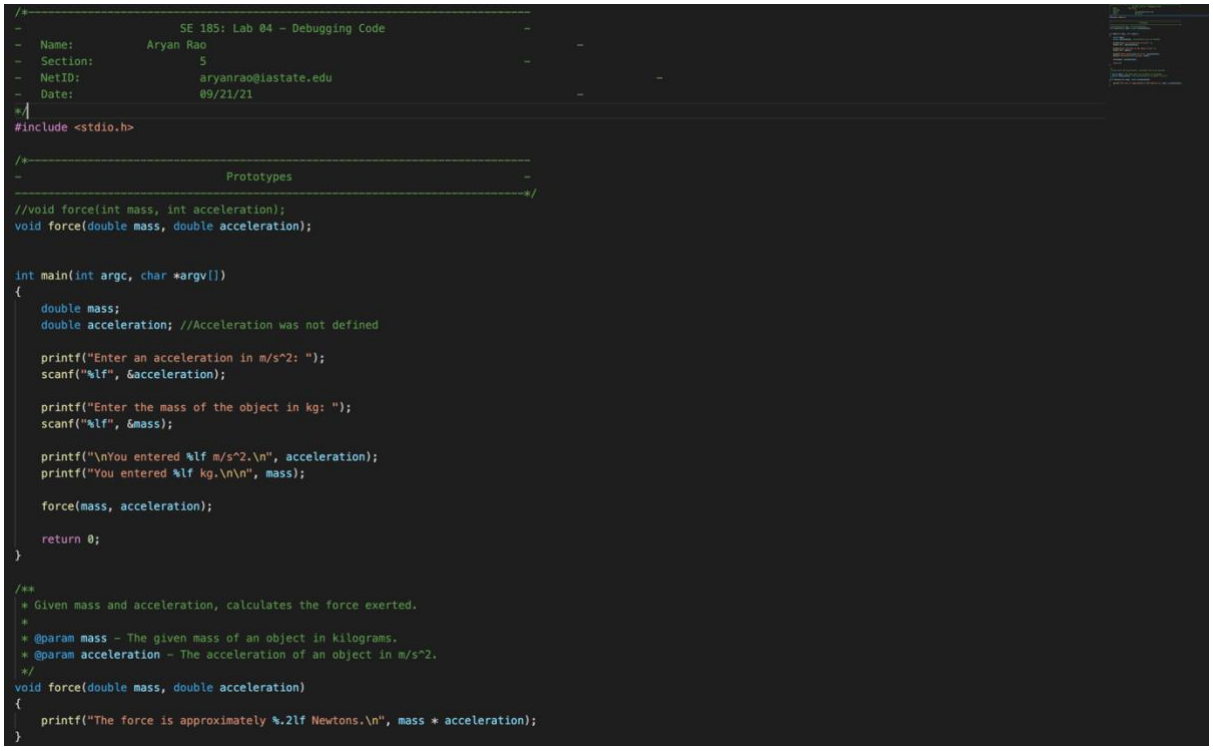
PART 1- CODE

```c
int main(int argc, char *argv[])
{
    int i, j;

    printf("Enter an integer: ");

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

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

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

      //Added a bracket
    {


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

    return 0;
}
```

PART 1-OUTPUT

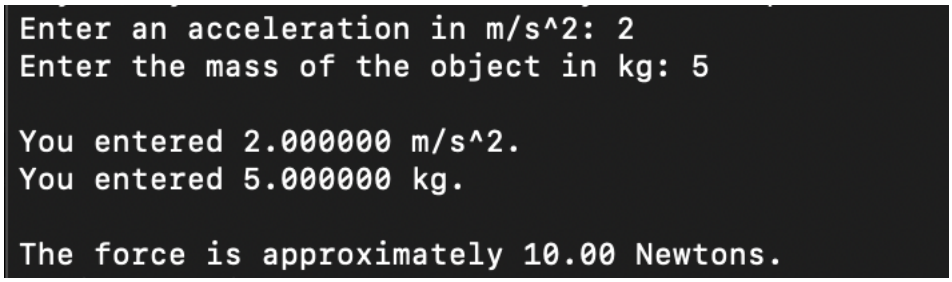
```
aryan@aryanrao11 ~ % /Users/aryan/Desktop/lab04/lab04-1_1 ; exit;
Enter an integer: 5
Enter another integer: 4
5 does not divide 4
4 % 5 is 4
```

PART 2-CODE



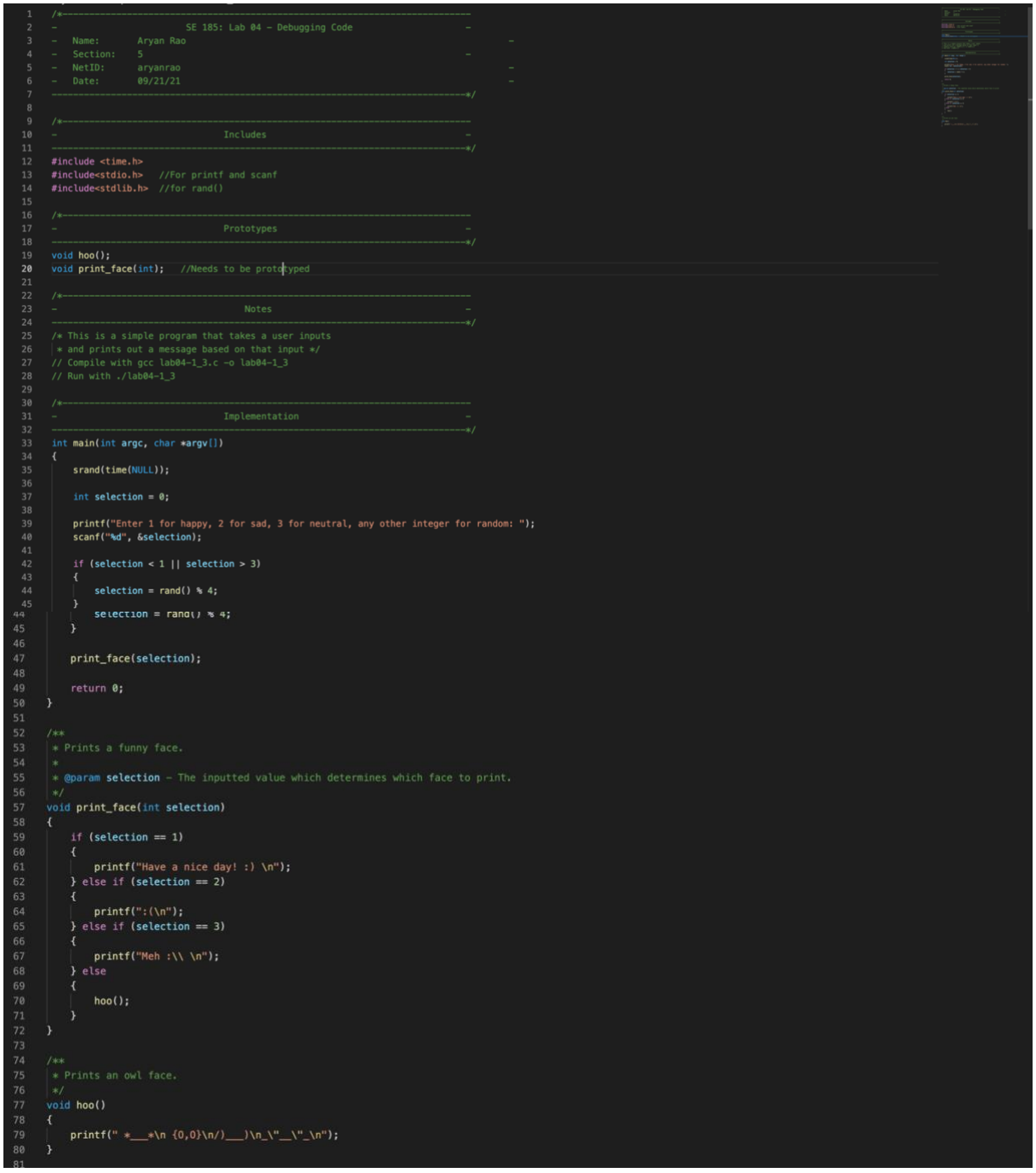
```c
/*-----------------------------------------------------------------------------
-                    SE 185: Lab 04 - Debugging Code                          -
-   Name:          Aryan Rao                                                  -
-   Section:               5                                                  -
-   NetID:                 aryanrao@iastate.edu                               -
-   Date:                  09/21/21                                           -
*/
#include <stdio.h>

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


int main(int argc, char *argv[])
{
    double mass;
    double acceleration; //Acceleration was not defined

    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);
}
```

PART 2-OUPUT



```
Enter an acceleration in m/s^2: 2
Enter the mass of the object in kg: 5

You entered 2.000000 m/s^2.
You entered 5.000000 kg.

The force is approximately 10.00 Newtons.
```

PART 3-CODE

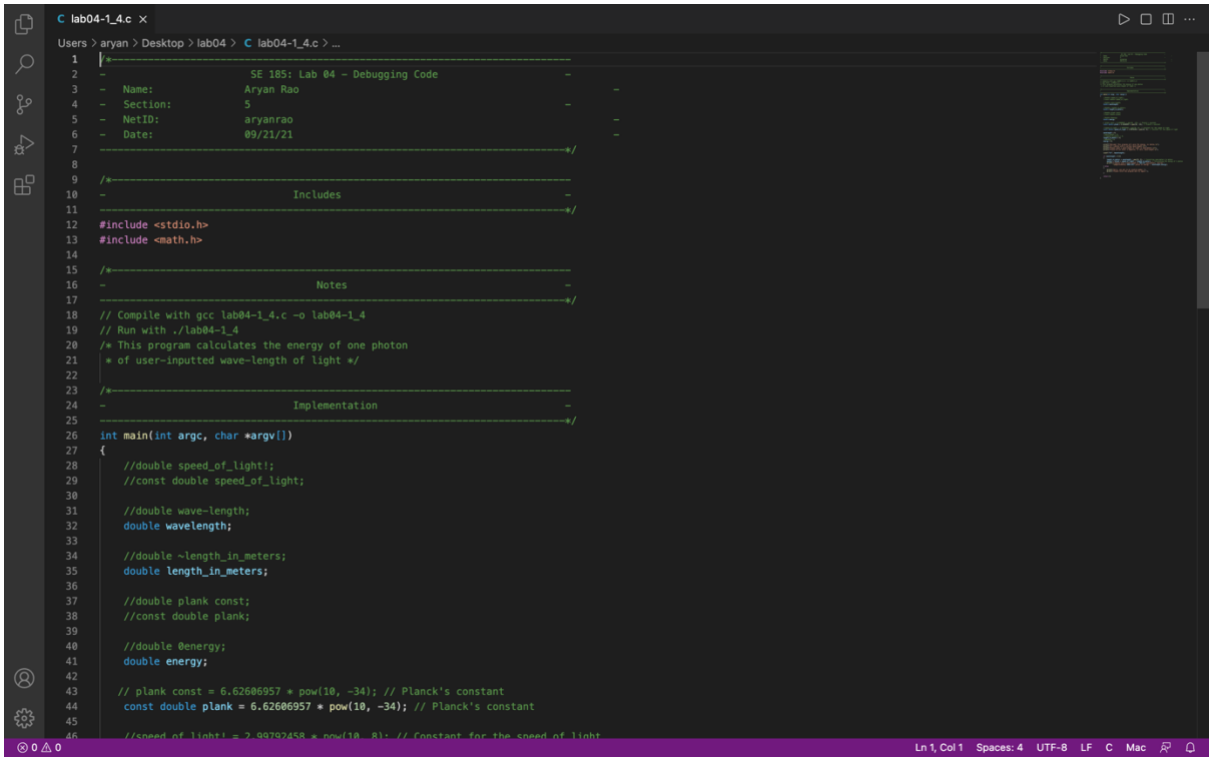


```c
/*-----------------------------------------------------------------------------
-                       SE 185: Lab 04 - Debugging Code                       -
-   Name:       Aryan Rao                                                     -
-   Section:    5                                                             -
-   NetID:      aryanrao                                                      -
-   Date:       09/21/21                                                      -
-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------
-                               Includes                                      -
-----------------------------------------------------------------------------*/
#include <time.h>
#include<stdio.h>   //For printf and scanf
#include<stdlib.h>  //for rand()

/*-----------------------------------------------------------------------------
-                               Prototypes                                    -
-----------------------------------------------------------------------------*/
void hoo();
void print_face(int);   //Needs to be prototyped

/*-----------------------------------------------------------------------------
-                                 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;
    }
        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 {0,0}\n/)___)\n_\"__\"_\n");
}
```

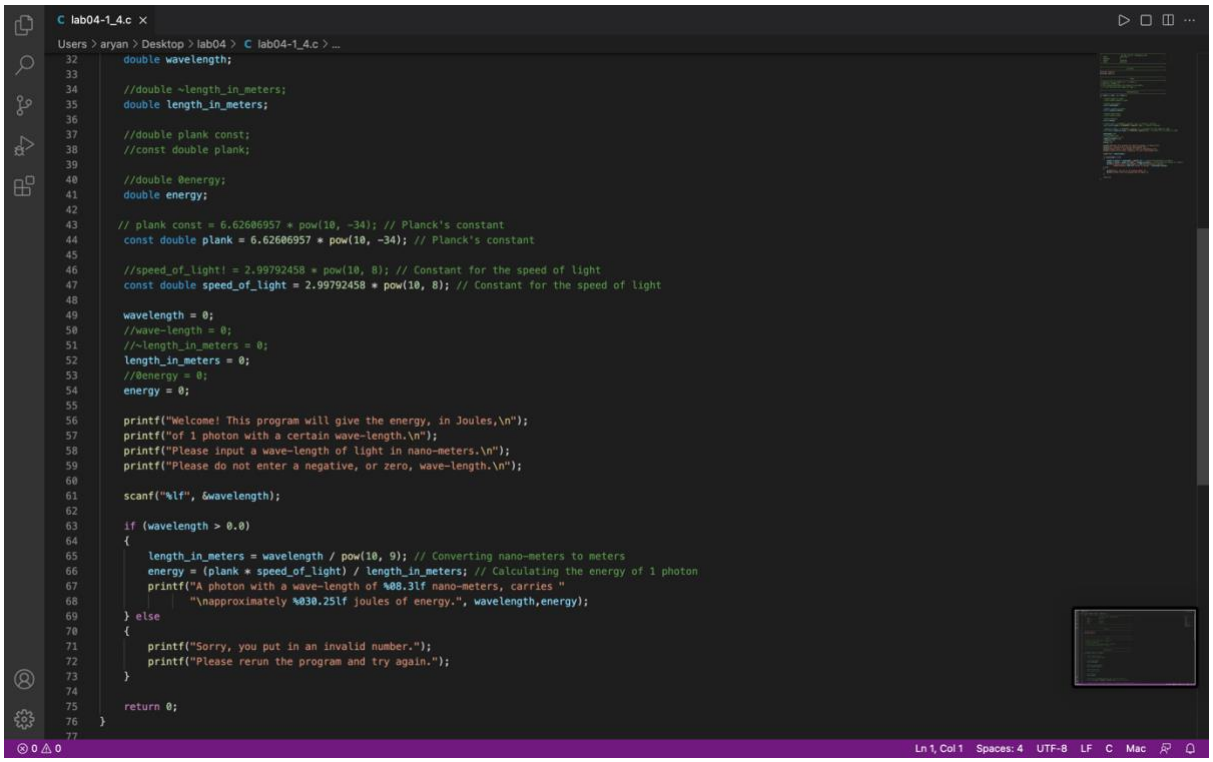
PART 3-OUPUT

```
Enter 1 for happy, 2 for sad, 3 for neutral, any other integer for random: 1
Have a nice day! :)
```

PART 4-CODE



```c
/*-------------------------------------------------------------------------
-                       SE 185: Lab 04 - Debugging Code                  -
-   Name:               Aryan Rao                                        -
-   Section:            5                                                -
-   NetID:              aryanrao                                         -
-   Date:               09/21/21                                         -
-------------------------------------------------------------------------*/

/*-------------------------------------------------------------------------
-                               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!;
    //const double speed_of_light;

    //double wave-length;
    double wavelength;

    //double ~length_in_meters;
    double length_in_meters;

    //double plank const;
    //const double plank;

    //double @energy;
    double energy;

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

    //speed_of_light! = 2.99792458 * pow(10, 8); // Constant for the speed of light
```

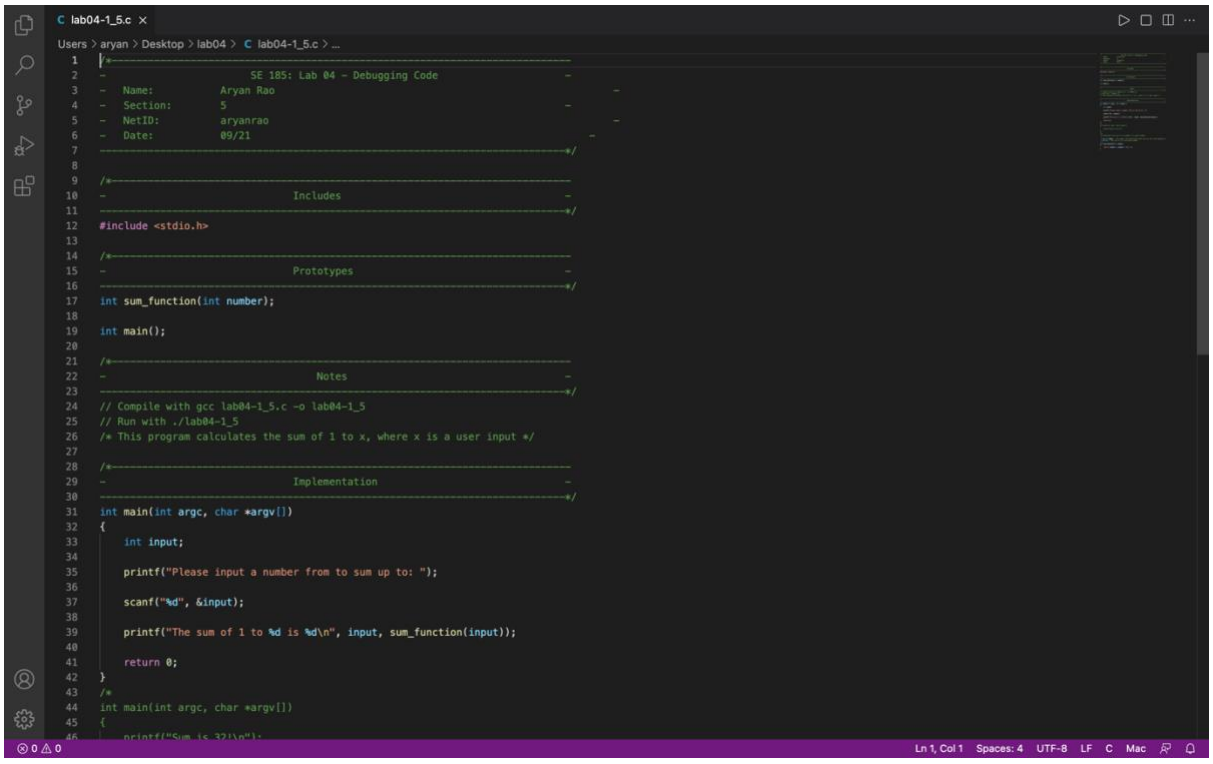


```c
    double wavelength;

    //double ~length_in_meters;
    double length_in_meters;

    //double plank const;
    //const double plank;

    //double @energy;
    double energy;

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

    //speed_of_light! = 2.99792458 * pow(10, 8); // Constant for the speed of light
    const double speed_of_light = 2.99792458 * pow(10, 8); // Constant for the speed of light

    wavelength = 0;
    //wave-length = 0;
    //~length_in_meters = 0;
    length_in_meters = 0;
    //@energy = 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", &wavelength);

    if (wavelength > 0.0)
    {
        length_in_meters = wavelength / pow(10, 9); // Converting nano-meters to meters
        energy = (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.", wavelength,energy);
    } else
    {
        printf("Sorry, you put in an invalid number.");
        printf("Please rerun the program and try again.");
    }

    return 0;
}
```

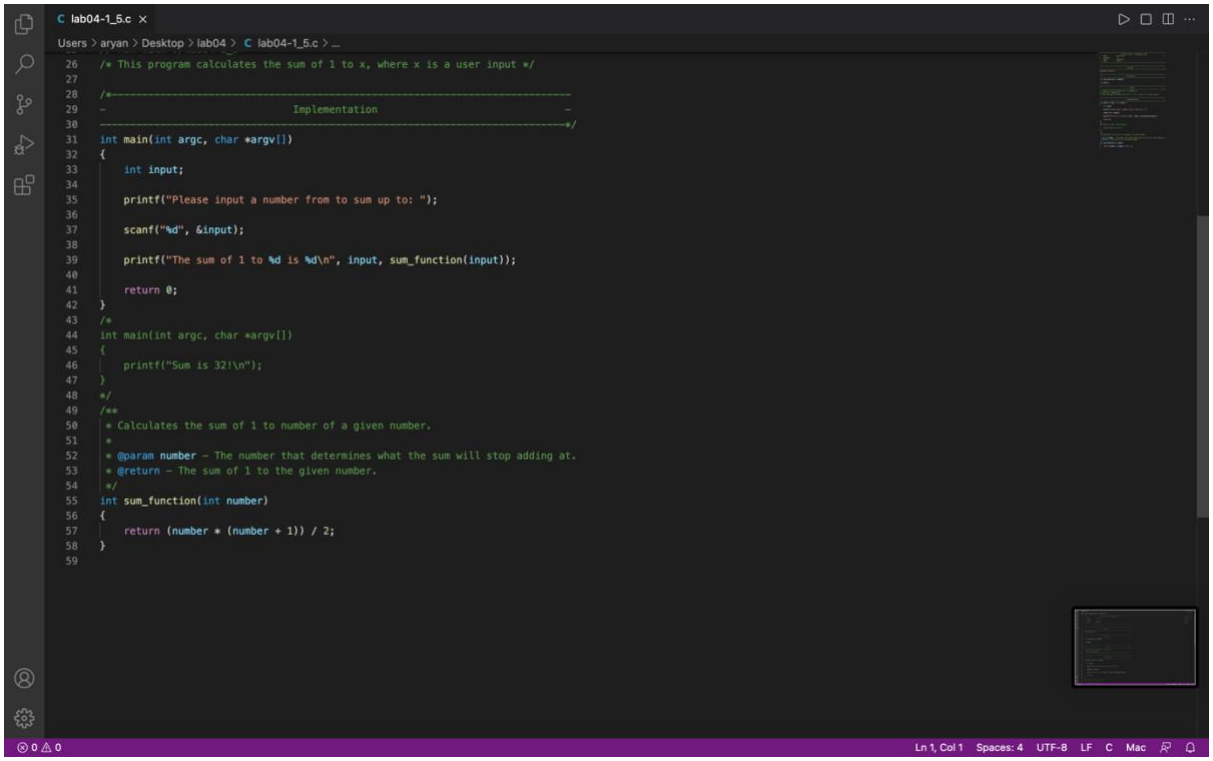
PART 4-OUPUT

```
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.
345
A photon with a wave-length of 0345.000 nano-meters, carries
approximately 0000.000000000000000005757814 joules of energy.
```

PART 5-CODE(a)



PART 5-CODE(b)



PART 5-OUTPUT

```
Please input a number from to sum up to: 10
The sum of 1 to 10 is 55
```

Problem 2

Compile and debug the 5 codes provided so that the program runs correctly. Comment out the changes made and put the correct code below. These codes might contain logical errors as well.

Analysis

In all the codes there are logical errors which change the overall meaning and the output of the code. We can run the code in the beginning to look for the compiler errors we get and then go on correcting them. Then go line by line fixing the logical errors.

Design

- Run the code as it is.
- Look for the errors/warnings generated by the compiler.
- If there are no compiler errors look why the desired output is not coming.
- Go line by line and fix the errors.

Testing

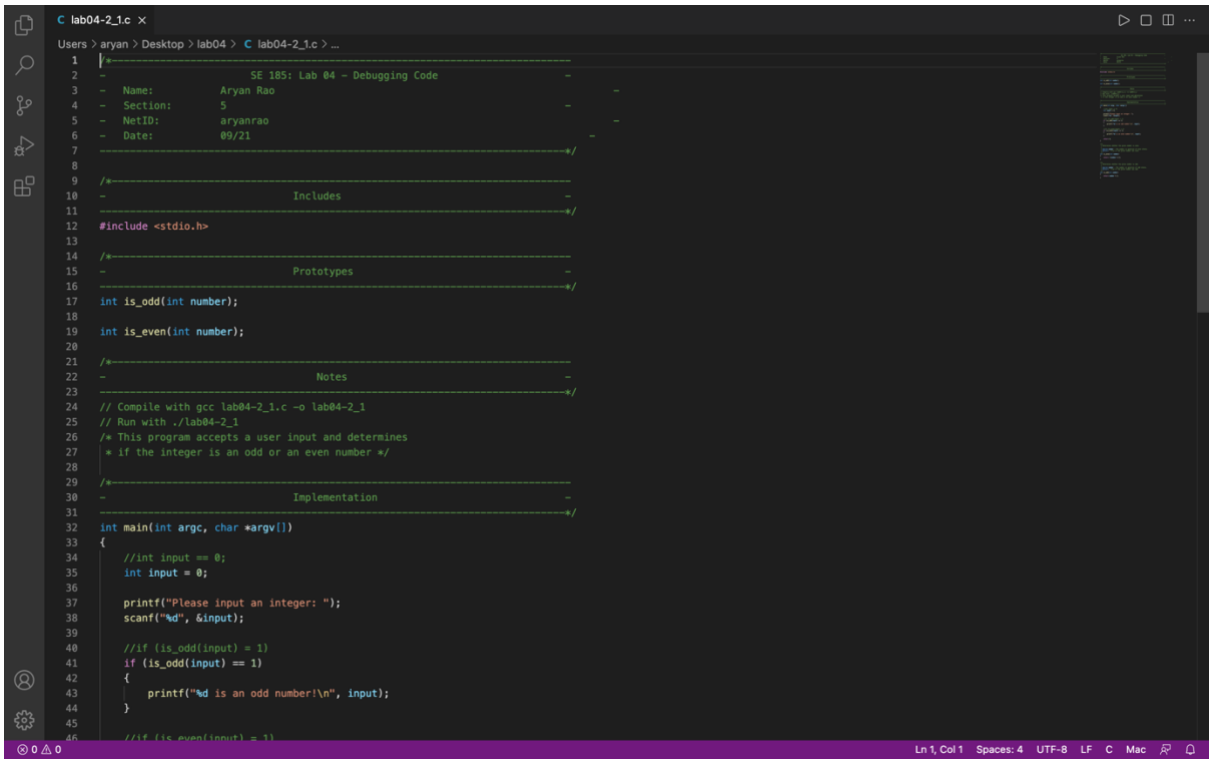
For the given source code manually find out the output and then compare it with the corrected code you have written.

Comments

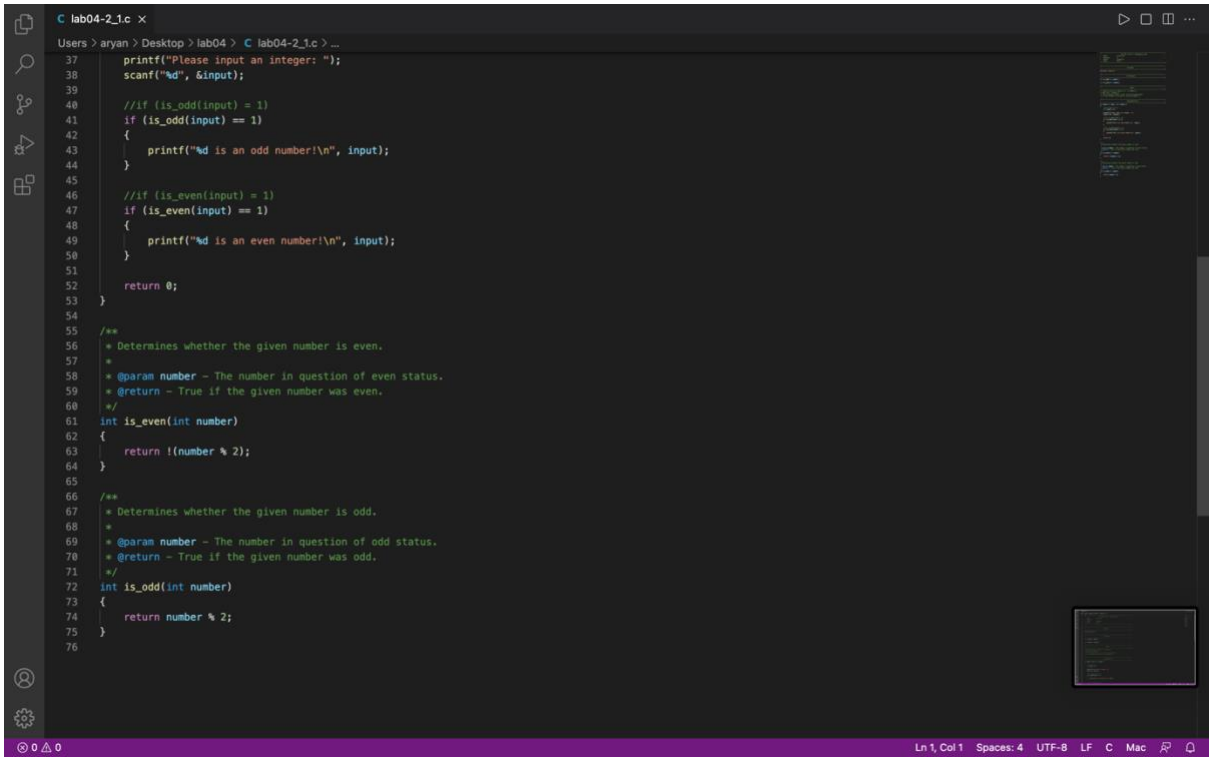
It is best to look for errors in the printf statements as most of the output comes from there and the statements in which some value is assigned to some variables.

SCREENSHOTS

PART 1-CODE



```c
/*-----------------------------------------------------------------------------
-                       SE 185: Lab 04 - Debugging Code                       -
-   Name:           Aryan Rao                                                 -
-   Section:        5                                                         -
-   NetID:          aryanrao                                                  -
-   Date:           09/21                                                     -
-----------------------------------------------------------------------------*/

/*-----------------------------------------------------------------------------
-                               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;

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

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

    //if (is_even(input) = 1)
```

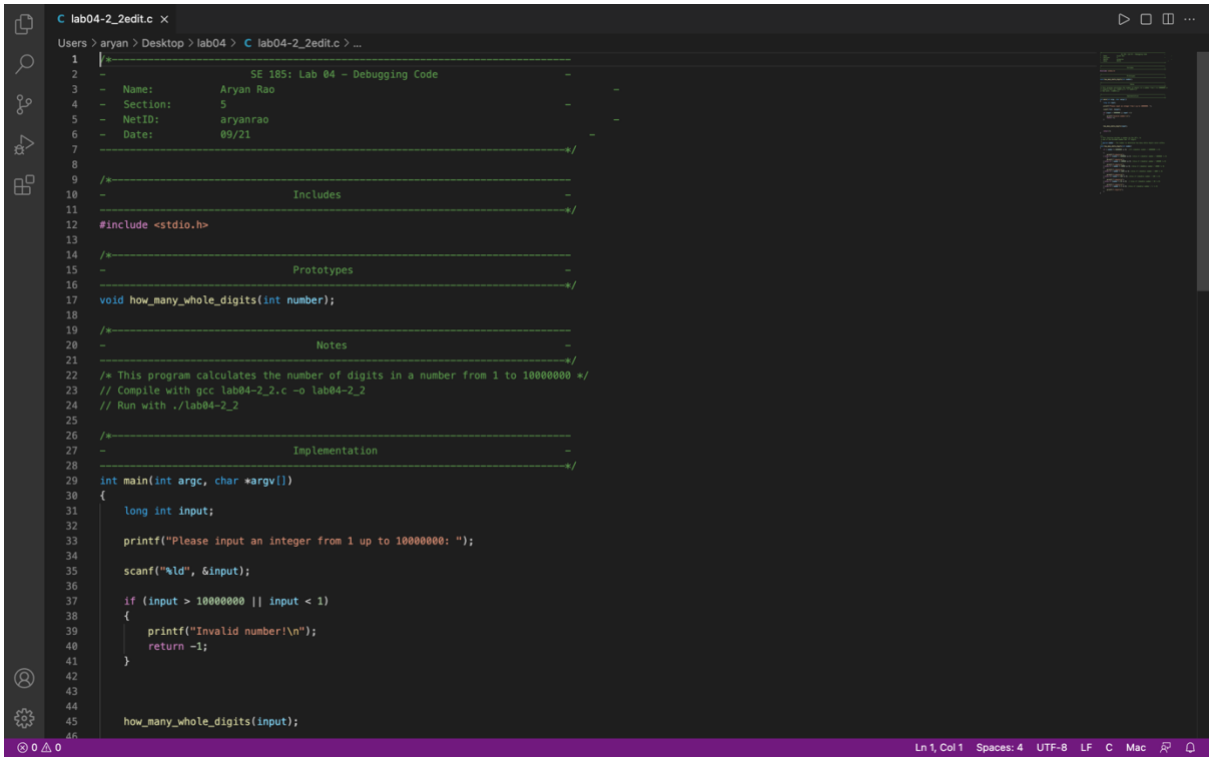


```c
    printf("Please input an integer: ");
    scanf("%d", &input);

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

    //if (is_even(input) = 1)
    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;
}
```

PART 1-OUPUT

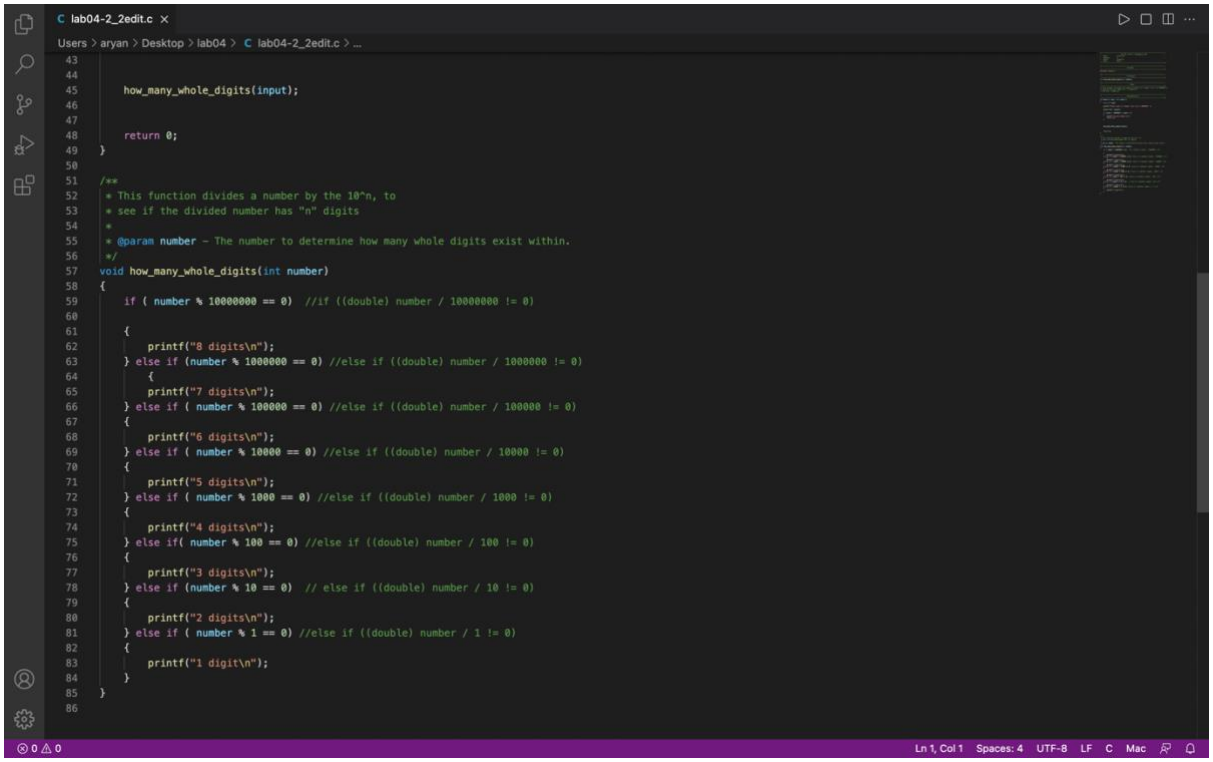
```
aryan@aryanrao11 ~ % /Users/aryan/Desktop/lab0
Please input an integer: 13
13 is an odd number!
```

PART 2-CODE(a)



```c
/*-------------------------------------------------------------------------
-                    SE 185: Lab 04 - Debugging Code                     -
-   Name:           Aryan Rao                                            -
-   Section:        5                                                    -
-   NetID:          aryanrao                                             -
-   Date:           09/21                                                -
-------------------------------------------------------------------------*/

/*-------------------------------------------------------------------------
-                             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[])
{
    long int input;

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

    scanf("%ld", &input);

    if (input > 10000000 || input < 1)
    {
        printf("Invalid number!\n");
        return -1;
    }



    how_many_whole_digits(input);
```

PART 2-CODE(b)



```c

    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)  //if ((double) number / 10000000 != 0)

    {
        printf("8 digits\n");
    } else if (number % 1000000 == 0) //else if ((double) number / 1000000 != 0)
        {
        printf("7 digits\n");
    } else if ( number % 100000 == 0) //else if ((double) number / 100000 != 0)
    {
        printf("6 digits\n");
    } else if ( number % 10000 == 0) //else if ((double) number / 10000 != 0)
    {
        printf("5 digits\n");
    } else if ( number % 1000 == 0) //else if ((double) number / 1000 != 0)
    {
        printf("4 digits\n");
    } else if( number % 100 == 0) //else if ((double) number / 100 != 0)
    {
        printf("3 digits\n");
    } else if (number % 10 == 0)  // else if ((double) number / 10 != 0)
    {
        printf("2 digits\n");
    } else if ( number % 1 == 0) //else if ((double) number / 1 != 0)
    {
        printf("1 digit\n");
    }
}

```

PART 2-OUTPUT

```
aryan@aryanrao11 ~ % /Users/aryan/Desktop/lab04/lab04-2_2edit ; exit;
Please input an integer from 1 up to 10000000: 500
3 digits
```

PART 3-CODE(a)

```c
/*-----------------------------------------------------------------------
-                       SE 185: Lab 04 - Debugging Code                 -
-   Name:           Aryan Rao                                           -
-   Section:        5                                                   -
-   NetID:          aryanrao                                            -
-   Date:           09/21                                               -
-----------------------------------------------------------------------*/

/*-----------------------------------------------------------------------
-                               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);

    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);
```

PART 3-Code(b)

```c

/**
 * 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);
}

```

PART 3-OUPUT

```
$ ./lab04-2_3.exe
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
```

PART 4-CODE

```c
/*-------------------------------------------------------------------------
-                          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;
    //float v, i, r;
    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);
```

PART 4-CODE(b)

```
59
60          printf("Your voltage is: %lf Volts\n", voltage(r, i));
61      } else if (selection == 2)
62      {
63          printf("Please enter a voltage value: ");
64          scanf("%lf", &v);
65
66          printf("Please enter a current value: ");
67          scanf("%lf", &i);
68
69          printf("Your Resistance is: %lf Ohms\n", resistance(v, i));
70
71      } else if (selection == 3)
72      {
73          printf("Please enter a resistance value: ");
74          scanf("%lf", &r);
75
76          printf("Please enter a voltage value: ");
77          scanf("%lf", &v);
78
79          printf("Your current is: %lf Amps\n", current(v, r));
80      }
81
82      return 0;
83  }
84
85  /**
86   * Given the resistance and current, calculates and returns the voltage.
87   *
88   * @param resistance - The resistance used to calculate the voltage.
89   * @param current - The current used to calculate the voltage.
90   * @return - The voltage calculated from the resistance and current.
91   */
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 }
```

PART 4-OUPUT

```
aryanrao@CO1318-03 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-2_4.exe
selection:
1 for voltage
2 for resistance
3 for current
1
Enter floating point numbers for input...
Please enter a resistance value: 2.1
Please enter a current value: 4.5
Your voltage is: 9.450000 Volts
```

PART 5-CODE

```c
/*-------------------------------------------------------------------------
-                       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;
```

```c
 * 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 ", n);
        printf("%d is zero and ", number);
        return 1;
    }

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

PART 5-OUPUT

```
aryanrao@CO1318-03 /cygdrive/u/fall2021/se185/lab04
$ gcc lab04-2_5.c -o lab04-2_5

aryanrao@CO1318-03 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-2_5.exe
Please type a number between -10000 and 10000: -20
-20 is non-positive and -20 is negative and -20 is non-zero and -20 is non-whole number.
```

Problem 3

The given code has a mixture of syntax, compiler, and logical errors. Run and compile the code and make suitable changes to get the desired output.

Analysis

Since the number of errors in the code is a lot, we must go line by line and inspect carefully what might be causing the errors.

Design

- Run the code as it is.
- Look for the errors/warnings generated by the compiler.
- If there are no compiler errors look why the desired output is not coming.
- Go line by line and fix the errors.
- Use -Wall while compiling as this gives potential issues in the code.

Testing

For the given source code manually find out the output and then compare it with the corrected code you have written.

Comments

The -wall statement while compiling may be very useful as it lists all the potential issues in our code which might not be seen while compiling but may come while running the code.

SCREENSHOTS

CODE(a)

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

/*-------------------------------------------------------------------------
-                              Includes                                   -
-------------------------------------------------------------------------*/
#include <stdio.h>
#include<stdlib.h> // For srand()
#include <time.h>

/*-------------------------------------------------------------------------- //Missing *
-                              Prototypes                                 -
--------------------------------------------------------------------------*/
char ask_to_play(int times_played);

int select_random_number();

void run_game(int);

/*-------------------------------------------------------------------------
-                                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                             -
-------------------------------------------------------------------------*/ //Missed /
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);
    }

    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;
```

CODE(b)

```
        scanf(" %c", &yes_or_no);
        return yes_or_no;
    }

    else
    {
        //scanf(" %c", yes_or_no);
        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=0; // initializing 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("%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)
        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;
        }

        //else if (number < computer_number);
        else if (number < computer_number)
            printf("\nYou guessed too low. Enter another guess.\n  -> ");

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


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

OUPUT

```
aryanrao@CO1318-03 /cygdrive/u/fall2021/se185/lab04
$ gcc lab04-3.c -o lab04-3
aryanrao@CO1318-03 /cygdrive/u/fall2021/se185/lab04
$ ./lab04-3.exe
Do you want to play a game? Enter 'y' to play, anything else not to play. :(
  -> y


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

What is your guess on what number I will select?
  -> 54

You guessed too low. Enter another guess.
  -> 98

 You guessed too high. Enter another guess.
  -> 66

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

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

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

 You guessed too high. Enter another guess.
  -> 87

 You guessed too high. Enter another guess.
  -> 84

 You guessed too high. Enter another guess.
  -> 82

 You guessed too high. Enter another guess.
  -> 81

The number was 81!

You guessed the number correctly!

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