

# **Lab 02 Report**

## **LAB #2**

### **SECTION #1**

**Jesus Horacio Soto Gonzalez**

**SUBMISSION DATE:**

**9/16/2022**

## Problem

### Problem 1:

Creating your own program.

## Analysis

Simple problem where I just added some printf statements that include the information asked in the problem.

## Design

```
printf("Jesus Horacio Soto Gonzalez\n");  
  
printf("SE185\n");  
  
printf("September, 9th 2022\n");
```

I made sure that each statement was printed in a different line.

## Testing

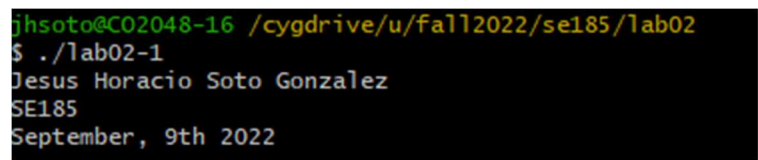
During testing I noticed that I forgot to put a semi-colon after the second printf statement that included the course title, so I fixed the code and recompile it. After, it worked appropriately.

## Comments

Simple problem to remember the basics.

## Screen Shots

### SS #1



```
jhsoto@C02048-16 /cygdrive/u/fall2022/se185/lab02  
$ ./lab02-1  
Jesus Horacio Soto Gonzalez  
SE185  
September, 9th 2022
```

## Problem

### Problem 2:

A simple program with input.

## Analysis

This program required to calculate the area of a rectangle with the given inputs. Then the problem asked us to modify the code to calculate the volume of a rectangular cube.

## Design

The first part of the design was mostly provided, and we just had to modify it to calculate the volume of the rectangular cube.

```
int x, y, z;
```

```
printf("Enter a lenght: ");
```

```
scanf("%d", &x);
```

```
printf("Enter a height: ");
```

```
scanf("%d", &y);
```

```
printf("Enter a width: ");
```

```
scanf("%d", &z);
```

```
printf(" A %d by %d by %d rectangular cube area is %d\n", x, y, z, x*y*z);
```

For this problem I had to add another integer and another printf statement for the width of the rectangular cube, also I modified the calculation to provide a correct answer.

## Testing

Testing was successful on the first try.

## Comments

Interesting problem to start thinking about the value of programing.

## Screen Shots

### SS #1

```
jhsoto@C02048-16 /cygdrive/u/fall2022/se185/lab02
$ ./lab02-2_1
Enter a width: 2
Enter a height: 2
A 2 by 2 rectangle's area is 4
```

### SS #2

```
jhsoto@C02048-16 /cygdrive/u/fall2022/se185/lab02
$ ./lab02-2_2
Enter a lenght: 2
Enter a height: 68
Enter a width: 8
A 2 by 68 by 8 rectangular cube area is 1088
```

## Problem

### Problem 3:

Mysterious Output.

## Analysis

For this problem we had to identify and correct some of the problems within the program.

## Design

For the design, I commented the error in the program and then provided a correction in each line of code that needed to be fixed.

```
int integer_result;
```

```
double decimal_result;
```

```
integer_result = 77 / 5;
```

```
printf("The value of 77/5 is %lf, using integer math.\n", integer_result); // Wrong format specifier for  
int, the correct format specifier is %d.
```

```
printf("The value of 77/5 is %d, using integer math. \n", integer_result); // Correct program
```

```
integer_result = 2 + 3;
```

```
printf("The value of 2+3 is %d.\n"); // There is no int_result assing so the answer can not be provided.
```

```
printf("The value of 2+3 is %d.\n", integer_result); // Correct program
```

```
decimal_result = 1.0 / 22.0;
```

```
printf("The value 1.0/22.0 is %d.\n", decimal_result); // Wrong format specifier for double, the  
correct one is %lf.
```

```
printf("The value 1.0/22.0 is %lf.\n", decimal_result); // Correct program
```

## Testing

For testing I was able to identify most of the errors in the program and successfully correct them. The errors were mainly format specifier errors where a non-appropriate format specifier was used.

## Comments

A good problem to start learning how to identify errors in a program.

## Screen Shots

### SS #1

```
int main(int argc, char *argv[])
{
    int integer_result;
    double decimal_result;

    integer_result = 77 / 5;
    printf("The value of 77/5 is %lf, using integer math.\n", integer_result); // Wrong format specifier for int, the correct format specifier is %d.
    printf("The value of 77/5 is %d, using integer math. \n", integer_result); // Correct program

    integer_result = 2 + 3;
    printf("The value of 2+3 is %d.\n"); // There is no int_result assing so the answer can not be provided.
    printf("The value of 2+3 is %d.\n", integer_result); // Correct program

    decimal_result = 1.0 / 22.0;
    printf("The value 1.0/22.0 is %d.\n", decimal_result); // Wrong format specifier for double, the correct one is %lf.
    printf("The value 1.0/22.0 is %lf.\n", decimal_result); // Correct program

    return 0;
}
```

### SS #2

```
$ ./lab02-3
The value of 77/5 is 0.000000, using integer math.
The value of 77/5 is 15, using integer math.
The value of 2+3 is 0.
The value of 2+3 is 5.
The value 1.0/22.0 is 1952257862.
The value 1.0/22.0 is 0.045455.
```

## Problem

### Problem 4:

Simple Arithmetic.

## Analysis

This problem shows us how to properly use the math.h folder and why it is important to know which format specifier works with an integer or double. As well as how a program can interpret a mathematical equation differently from the known mathematical rules.

## Design

The first part of this problem's design was provided, we just had to add the printf statements with the appropriate format specifiers.

## Testing

During testing I observe how the program interprets different ways of writing a mathematical equation.

## Comments

Comparison of program result and mathematical result:

- a) The output result of this program is the same as the result obtained from doing simple math. (8152)
- b) This result is the same in the program and doing basic math. (27361080)
- c) For this calculation the program provided 81.00 instead of 81.4 which is the correct answer because the program ignored the residue of the first part of the calculation.
- d) For this calculation the result was the same in the program and by doing math. (33.73)
- e) In this calculation the program provided an answer of 21 instead of 22 which will be the answer by doing basic math. The program ignores the residue of the first division because it is an integer and just multiplies the whole number obtained by three.
- f) The program ignores the residue of the operation because it is considering it an integer.  
Program = (2) Math = (2.44)
- g) For this calculation the program ignores the residue of the calculation but adds the two decimal places since it is a double. Program = (2.00) Math = (2.44)
- h) The program ignores the residue of the first calculation and then multiplies by 3. Program = (21.00) Math = (22.00)
- i) Due to the parenthesis the program provides the correct answer. (22.00)
- j) The program ignores the residue of the calculation because it is an integer. Program = (2) Math = (2.44)
- k) For this calculation the program does consider the residue outputting the correct answer. (22.00)

## Screen Shots

### SS #1

```
$ ./lab02-4
The result value of this calculation is: 8152.
The result value of this calculation is: 27361080.
The result value of this calculation is: 81.00.
The result value of this calculation is: 33.73.
The result value of this calculation is: 21.
The result value of this calculation is: 2.
The result value of this calculation is: 2.00.
The result value of this calculation is: 21.00.
The result value of this calculation is: 22.00.
The result value of this calculation is: 2.
The result value of this calculation is: 22.00.
The area of the circle is: 44.39
14 feet is equal to 4.27 meters.
76 degrees Fahrenheit equals 24.44 Degrees Celsius.
```

### SS #2

```
#include <math.h>

/*-----
   Implementation
   -----*/

int main(int argc, char *argv[])
{
    int integerResult;
    double decimalResult;

    integerResult = 6427 + 1725;
    printf("The result value of this calculation is: %d.\n", integerResult); // a

    integerResult = (6971 * 3925) - 95;
    printf("The result value of this calculation is: %d.\n", integerResult); // b

    decimalResult = 79 + 12 / 5;
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // c

    decimalResult = 3640.0 / 107.5;
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // d

    integerResult = (22 / 3) * 3;
    printf("The result value of this calculation is: %d.\n", integerResult); // e

    integerResult = 22 / (3 * 3);
    printf("The result value of this calculation is: %d.\n", integerResult); // f

    decimalResult = 22 / (3 * 3);
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // g

    decimalResult = 22 / 3 * 3;
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // h

    decimalResult = (22.0 / 3) * 3.0;
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // i

    integerResult = 22.0 / (3 * 3.0);
    printf("The result value of this calculation is: %d.\n", integerResult); // j

    decimalResult = 22.0 / 3.0 * 3.0;
    printf("The result value of this calculation is: %.2lf.\n", decimalResult); // k

    /*-----

    double areaCircle;

    areaCircle = ((33.567 * 23.67) / (4 * M_PI));
    printf("The area of the circle is: %.2lf.\n", areaCircle);

    /*-----

    double meters;

    meters = (14 * 0.3048);
    printf("14 feet is equal to %.2lf meters.\n", meters);

    /*-----

    double degreesCelsius;

    degreesCelsius = ((76 - 32) / 1.8);
    printf("76 degrees Fahrenheit equals %.2lf Degrees Celsius.\n", degreesCelsius);
```



## Problem

### Problem 5:

Working with I/O - Pythagorean Theorem

## Analysis

This problem shows us more practical uses for programming and mathematical functions which can be applied in a real setting.

## Design

```
int main(int argc, char *argv[])
{
    double a, b, csquare;
    double filler;

    printf("Enter the value for a: \n");
    scanf("%lf", &a);
    printf("Enter the value for b: \n");
    scanf("%lf", &b);
    csquare = ((a * a) + (b * b));
    filler = sqrt(csquare);
    printf("The result value is: %lf\n", filler);
    return 0;
```

- In this program we use some user entered inputs, stored them with scanf and use them later to calculate the hypotenuse using the Pythagorean theorem.

## Testing

During testing I had some complications linking the first calculation's result of the inputs to the sqrt mathematical operation to obtain the value of the hypotenuse.

## Comments

This was a problem that required more attention and knowledge of the math.h folder.

## Screen Shots

### SS #1

```
$ ./lab02-5
Enter the value for a:
5
Enter the value for b:
9
The result value is: 10.295630
```

### SS #2

```
#include <stdio.h>
#include <math.h>    // Google this header file to learn more! :)

/*-----
-                               Implementation                               -
-----*/

int main(int argc, char *argv[])
{
    double a, b, csquare;
    double filler;

    /* Put your code after this line */

    printf("Enter the value for a: \n");
    scanf("%lf", &a);
    printf("Enter the value for b: \n");
    scanf("%lf", &b);
    csquare = ((a * a) + (b * b));

    /* This next line will calculate the square root of whatever value is
    * inside the parenthesis and assigns it to the variable filler. */
    filler = sqrt(csquare);
    printf("The result value is: %lf\n", filler);
    return 0;
}
```