

TLab 2: Solving Simple Problems in C

LAB 2

SECTION C

James Mechikoff

SUBMISSION DATE: 9/4/2018

Date

9/4/2018

Problem

The purpose of the lab today was all about learning basic inputs, outputs, arithmetic, and the applications of them. The program is taking out inputs and processing them to produce a useable output through arithmetic operations.

Analysis

The problem in this lab starts off with needing new programs made by the user to be created. After that, it starts off with the most basic of input and output. It involved editing code provided from the area of a rectangle to the volume of a rectangular prism. Then the problem gives another bit of code that needs debugging. After that it requires simple arithmetic to be input, formatted properly and output. Then more advanced arithmetic is added and finally it asks to correct the mistakes with the outputs in the comments. The final bit was applying math knowledge and what was learned about the `math.h` import statement to perform the Pythagorean theorem.

Design

All steps of the problem were to take basic inputs, run them through a form of basic math or algebraic function then output it back to the user. The basic steps involved were as follows:

1. Get the input.
2. Run the input through the desired function.
3. Display the output in a desired manner

All inputs were determined by what kind of function was needed or what was output. Examples being division requiring floats to show decimal places or any function involving a power or square root as they only accept doubles.

Testing

The solutions were all verified by doing out the problems by hand with multiple inputs before and after running them through the functions and matching the results. Then multiple numbers were tested in the functions to ensure a wide range that works. Problems were initially noticed with the power functions in problem 5, however with thorough testing it was revealed to be an output error, as it uses long floats instead of decimals as an output. After that testing on all problems were resumed until satisfied, outputs recorded and saved and then work continued to the next part.

Comments

I really learned first things first, always check your inputs match your outputs as most of my errors were caused because I had a mismatch in those two. Another one is to make sure order of

operations is done properly and parenthesis are used when they are needed to ensure it is done in the right order. Don't be afraid to use more than you might need as it ensures everything is accurate.

Source Code

2.1

```
/*-----  
--  
-                               SE 185 Lab 02  
-       Developed for 185-Rursch by T.Tran and K.Wang  
-       Name: James Mechikoff  
-       Section: C  
-       NetID: 726219551  
-       Date: 9/4/2018  
-----  
*/  
  
/*-----  
--  
-                               Includes  
-----  
*/  
#include <stdio.h>  
#include <math.h>  
  
/*-----  
--  
-                               Defines  
-----  
*/  
  
/*-----  
--  
-                               Prototypes  
-----  
*/  
  
/*-----  
--  
-                               Implementation  
-----  
*/  
int main()  
{  
  
    /* Put your code after this line */
```

```

    return 0;
}

```

2.2.1

```

/*-----
--
-                               SE 185 Lab 02
-       Developed for 185-Rursch by T.Tran and K.Wang
-       Name: James Mechikoff
-       Section: C
-       NetID: 726219551
-       Date: 9/4/2018
-----
*/

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

/*-----
--
-                               Defines
-----
*/

/*-----
--
-                               Prototypes
-----
*/

/*-----
--
-                               Implementation
-----
*/
int main()
{
    /* Put your code after this line */
    int x, y;

```



```

/* Put your code after this line */
int x, y, z;
printf("Enter a width: ");
scanf("%d", &x);
printf("Enter a height: ");
scanf("%d", &y);
printf("Enter a length: ");
scanf("%d", &z);
printf("A %d by %d by %d cube's volume is %d\n", x, y, z, x * y * z);

return 0;
}

```

2.3

```

/*-----
--
--                               SE 185 Lab 02
--       Developed for 185-Rursch by T.Tran and K.Wang
--       Name: James Mechikoff
--       Section: C
--       NetID: 726219551
--       Date: 9/4/2018
-----
*/

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

/*-----
--
--                               Defines
-----
*/

/*-----
--
--                               Prototypes
-----
*/

```



```

/*-----
--
-                                     Prototypes
-----
*/

/*-----
--
-                                     Implementation
-----
*/
int main()
{

    /* Put your code after this line */

    double tempF = 72;
    double circ = 23.567;

    const double feetToMeters = .3048;

    int a = (6427 + 1725);
    int b = ((6971 * 3925)-95);
    double c = (79 + 12 / 5);
    double d = (3640.0 / 107.9);
    int e = ((22 / 3) * 3);
    int f = (22 / (3 * 3));
    double g = (22 / (3 * 3));
    double h = (22 / 3 * 3);
    double i = ((22.0 / 3) * 3.0);
    int j = (22.0 / (3 * 3.0));
    double k = (22.0 / 3.0 * 3.0);
    double l = (pow((circ), 2) / (4*M_PI));

    /*I know that the formula for Area normally is  $A = \pi r^2$ ,
    therefore I broke it back down with  $C = 2 \pi r$  to what it was.
    For the actual coding formula, I was trying to recall what the
    power and pi functions were. Thankfully c is pretty well documented
    and a simple search using resources online gave me how both functions
work
    and helped show me a little bit more just what is included in the
import statement
#include <math.h>*/

    double m = (14 * feetToMeters);

    /* I had to learn this conversion ratio for my high school physics
course.
    Using what I know about the conversion, I just wrote a constant and
plugged in 14 times that constant.
    */

    double n = ((tempF - 32) / 1.8);

```



```

    int answer = 1 + 2;

    printf("The output for these problems is: \n");
    printf("a = 6427 + 1725 = %d\n", a); // This line is okay.
    printf("b = (6971 * 3925)-95 = %d\n", b); // This line is okay.
    printf("c = 79 + 12 / 5 = %.2lf\n", c); //This one is wrong because I
made it display only 2 digits. It should display 3.
    printf("d = 3640.0 / 107.9 = %.2lf\n", d); //Just like the last one,
this ones issue is that ir rounds up to two digits because the question
requested it.
    printf("e = (22 / 3) * 3 = %d \n", e); //This one is correct
    printf("f = 22 / (3 * 3) = %d \n", f); //This one should be outputted
as a float.
    printf("g = 22 / (3 * 3) = %.2lf\n", g); //This one should have been
displayed out past 2 decimals.
    printf("h = 22 / 3 * 3 = %.2lf\n", h); // This one is a ( ) problem, it
is doing order of operations wrong.
    printf("i = (22.0 / 3) * 3.0 = %.2lf\n", i); //This one is correct, but
does not need a float output.
    printf("j = 22.0 / (3 * 3.0) = %d\n", j); //This one should have been a
float.
    printf("k = 22.0 / 3.0 * 3.0 = %.2lf\n", k); //Again this one is right,
but does not need a float.
    printf("The area of circle l is %lf. \n", l);
    printf("There are %lf meters in 14 feet.\n", m);
    printf("72 degrees Fahrenheit is %.2lf degrees centigrade.\n", n);

    printf("Answer Output: %d", answer);

    return 0;
}

```

2.5

```

`/*-----
---
-                               SE 185 Lab 02
-       Developed for 185-Rursch by T.Tran and K.Wang
-       Name: James Mechikoff
-       Section:      C
-       NetID:        726219551
-       Date: 9/8/2018
-----
*/

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

```

```

/*-----
--
-                                     Defines
-----*/


/*-----
--
-                                     Prototypes
-----*/


/*-----
--
-                                     Implementation
-----*/
*/
int main()
{
    double a, b, c;
    double filler;
    /* Put your code after this line */

    printf("Please input A: ");
    scanf("%lf", &a);

    printf("Please input B: ");
    scanf("%lf", &b);

    c = pow(a,2.0) + pow(b,2.0);

    printf("%lf Squared + %lf Squared = %lf Squared \n", a,b,c);

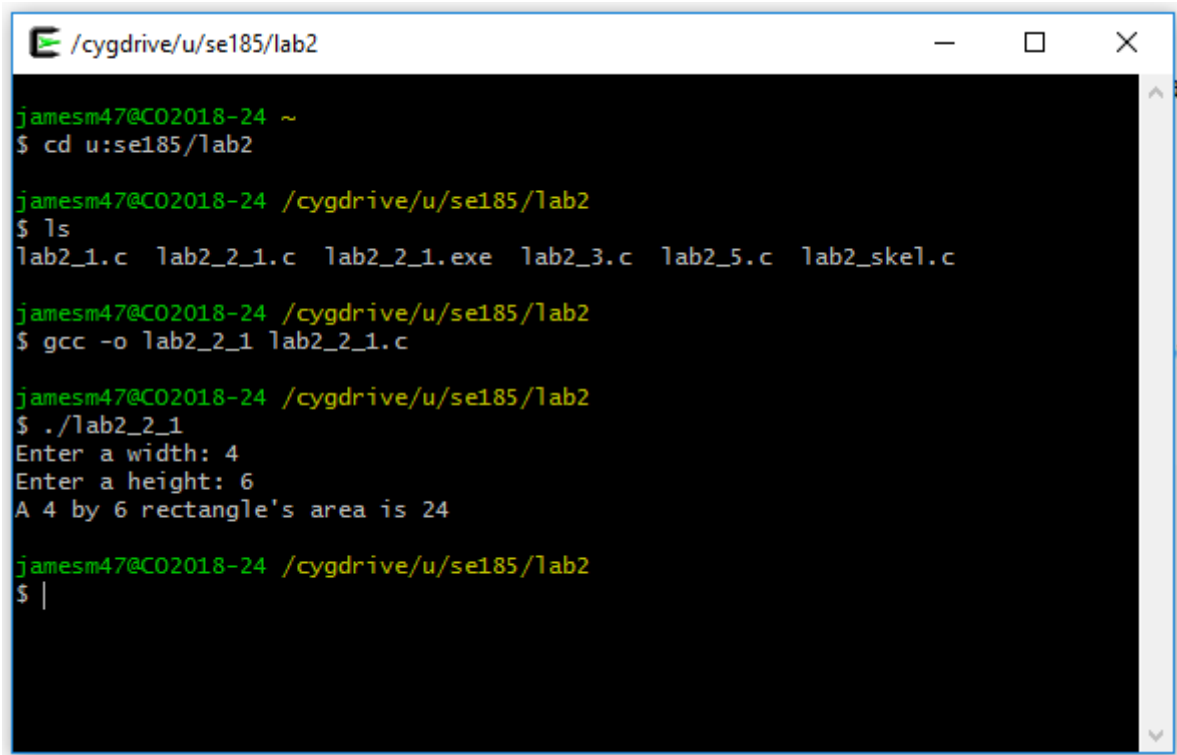
    /* This next line will calculate the square root of whatever value is
       inside the parenthesis and assigns it to the variable filler. */
    filler = sqrt(c);

    printf("Square root of c is %lf", filler);

    return 0;
}

```

Screen Shots



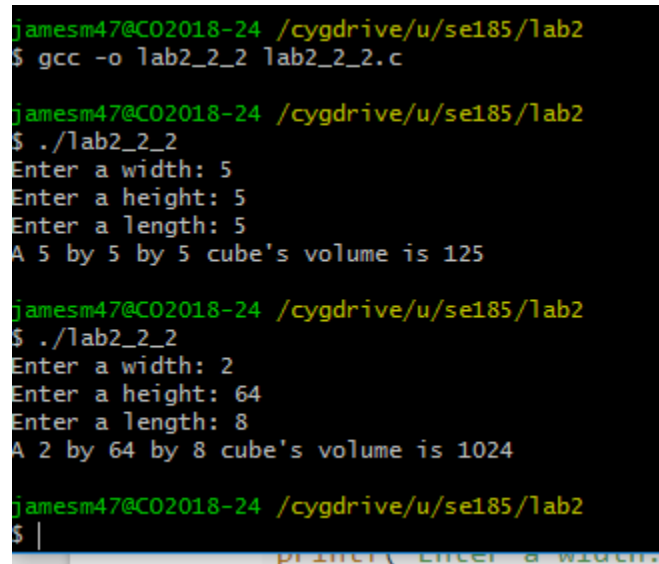
```
/cygdrive/u/se185/lab2
jamesm47@C02018-24 ~
$ cd u:se185/lab2

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ls
lab2_1.c  lab2_2_1.c  lab2_2_1.exe  lab2_3.c  lab2_5.c  lab2_skel.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_2_1 lab2_2_1.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_2_1
Enter a width: 4
Enter a height: 6
A 4 by 6 rectangle's area is 24

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ |
```



```
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_2_2 lab2_2_2.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_2_2
Enter a width: 5
Enter a height: 5
Enter a length: 5
A 5 by 5 by 5 cube's volume is 125

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_2_2
Enter a width: 2
Enter a height: 64
Enter a length: 8
A 2 by 64 by 8 cube's volume is 1024

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ |
```

```
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_3 lab2_3.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_3
The value of 77/5 is 0.000000, using integer math
The value of 2+3 is 0
The value 1.0/22.0 is 1952257862
```

```
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_3 lab2_3.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_3
The value of 77/5 is 15, using integer math
The value of 2+3 is 5
The value 1.0/22.0 is 0.045455
```

```
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_4 lab2_4.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_4
The output for these problems is: a = 8152, b = 27361080, c = 81.000000, d = 33.
734940, e = &d, f = 21, g = 0.000000, h = 2.000000, i = 21.000000, j = 0, k = 0.
000000.
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ |
```

```
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ gcc -o lab2_4 lab2_4.c

jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ ./lab2_4
The output for these problems is: a = 6427 + 1725 = 8152

b = (6971 * 3925)-95 = 27361080
c = 79 + 12 / 5 = 81.00
d = 3640.0 / 107.9 = 33.73
e = (22 / 3) * 3 = &d
f = 22 / (3 * 3) = 2
g = 22 / (3 * 3) = 2.00
h = 22 / 3* 3 = 21.00
i = (22.0 / 3) * 3.0 = 22.00
j = 22.0 / (3 * 3.0) = 2
k = 22.0 / 3.0 * 3.0 = 22.00
The area of circle l is 44.197605.
There are 4.267200 meters in 14 feet.
72 degrees Fahrenheit is 22.22 degrees centigrade.
Answer Output: 3
jamesm47@C02018-24 /cygdrive/u/se185/lab2
$ |
```

```
James@DESKTOP-PL645G3 /cygdrive/c/Users/James/Documents/lab2
$ gcc -o lab2_5 lab2_5.c

James@DESKTOP-PL645G3 /cygdrive/c/Users/James/Documents/lab2
$ ./lab2_5
Please input A: 5
Please input B: 9
5.000000 Squared + 9.000000 Squared = 106.000000 Squared
Square root of c is 10.295630
James@DESKTOP-PL645G3 /cygdrive/c/Users/James/Documents/lab2
$
```