**Problem**

Problem 2: we must write a C program to compute the area of a rectangular prism after being given code that computes a rectangles area

Problem 3: We must find and fix errors in a given C program

Problem 4: We must first copy simple arithmetic equations into a C program. After that we must calculate the area of a circle given circumference, convert feet to meters, and finally convert Fahrenheit to Celsius

Problem 5: We must write a C program that takes user inputs in order to do the Pythagorean theorem using said inputs

**Analysis**

Problem 2: the formula for a rectangular prism is just width \* height \* length. I just need to add an input and variable for length and change the formula inside the result print function to also multiply by length

Problem 3: I basically just need to carefully inspect each line of code to look for any minor errors that are causing the code to not give the expected output.

Problem 4: A-k is pretty simple in that its just copying what is given. For question L you must take the given circumference, then using the formula for circumference which is 2 \* pi \* r, you divide the given circumference by 2 \* pi. After that you use the area formula for a circle which is pi \* r^2 to find the area. For question m and N they are basically the same, you just convert the given units into a different type of unit using the given formula

Problem 5: The Pythagorean theorem is a^2 + b^2 = c^2 So I know that I need the math library for both sqrt and pow. I also need to scan twice for user input to assign to variables a and b. After that I just have to run it through a^2 + b^2 = c^2 to get my result

**Design**

Problem 2: I just did as I mentioned above in the analysis part by adding another int variable z, asking for user input for its value, then in the final result print function I changed x\*y to x\*y\*z

Problem 3: I read through every line carefully and noticed the first and last printf functions had the wrong syntax for putting a variable value into the printed text. The first printf had the syntax to scan a double when the variable was an int, and the second printf had the syntax to scan and int when the variable was a double. This was an easy fix that was just switching the syntaxes. The middle printf was completely missing the variable that should have been there so I added it.

Problem 4: A-K is once again just basically copy pasting. For question L I used the functions of the math library, M\_PI and pow to do the calculations that I gave above in the analysis. For questions M and N I just created two variables, one in the given units and one in the units that we need. I then set the first variable to the given number, and the second variable to the needed units after the conversion. The Question M I just multiplied 14 \* .3048 and for question N I just replaced the TF with 76 in the unit conversion equation.

Problem 5: I first take input from the user for values of a and b. Then I use the pow function from the math library to do a^2 + b^2 to find c^2. After that I use the sqrt function to find c then print it.

**Testing**

Problem 2: Ran the code after the adjustments and it worked as intended

Problem 3: Ran the code after the adjustments and it worked as intended

Problem 4: Ran the code after writing it and it worked as intended

Problem 5: Ran the code and did not get the expected value, however after inspecting the code I figured out that I forgot to put an & before my variables in the scanf function. This is shown in screenshots 1 and 2 for problem 5.

**Comments**

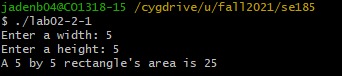
Problem 2: N/A

Problem 3: N/A

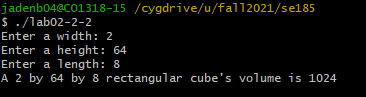
Problem 4: A – D output the values you would expect putting it through a regular calculator, however after that it starts being different. Problem e is not the expected value due to integers always rounding down within its calculations so it comes out with 7 \* 3 instead of 7.3333 \* 3. F is not what you expect for the same reason, integers can only be whole numbers, and it always rounds down. G and H are both not what you expect due to never making any of the defined numbers a double and only having integer numbers. This means that it essentially is an integer and once again will be rounding down. I give the correct value. J once again is integer rounding down. K is correct.

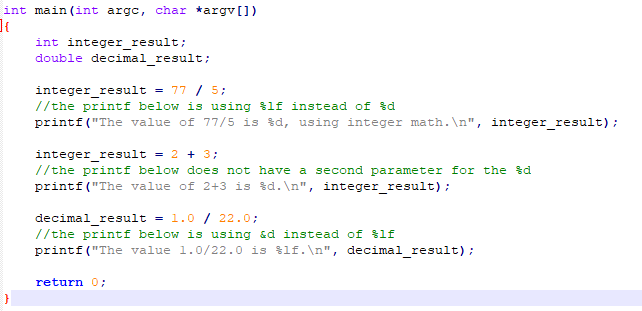
**Screenshots**

Screenshot 1 for Problem 2:

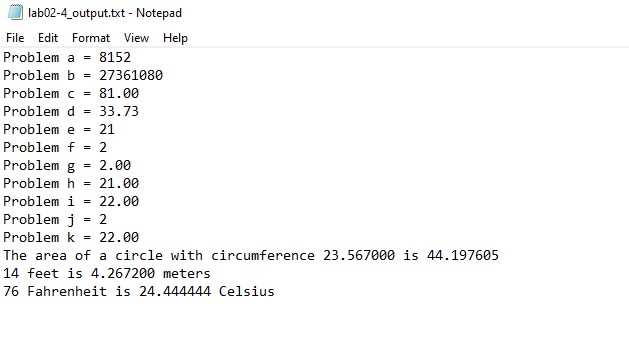


Screenshot 2 for Problem 2:

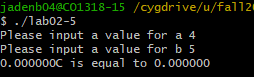


Screenshot 1 for Problem 3:  


Screenshot for problem 4:



Screenshot 1 for Problem 5:



Screenshot 2 for Problem 5:

