Program 3
100 points

Program 3a: Write a program that asks the user to enter the width and length of a rectangle, and then computes and displays the rectangle's area. The program should contain at least four functions according to the following specification:

1. `get_length`: This function should prompt the user to enter the rectangle's length, and then return that value as a `double`. This function should force the user to enter a positive value for the length and continue to prompt the user until they do so.

2. `get_width`: This function should prompt the user to enter the rectangle's width, and then return that value as a `double`. This function should force the user to enter a positive value for the width and continue to prompt the user until they do so.

3. `get_area`: This function should accept the rectangle's length and width as arguments, and return the rectangle's area. The area is calculated by multiplying the length by the width.

4. `display_data`: This function should accept the rectangle's length, width, and area as arguments, and display them in an appropriate message on the screen. The return value for this function is not used, so you can have this function use void as the return type.

All functions must use the call-by-value parameter passing mechanism; do not attempt to mimic call-by-reference parameters in your implementation. Name your source code file `program3a.c`.

Program 3b: The gravitational attractive force between two bodies with masses m1 and m2 separated by a distance d is given by:

$$F = \frac{\mathrm{G}m_1 m_2}{d^2}$$

where G is the universal gravitational constant:

$$G = 6.673 \times 10^{-8} \left(\frac{cm^3}{g \times \mathrm{sec}^2} \right)$$

write a function definition that takes arguments for the masses of two bodies and the distance between them and that returns the gravitational force. Since you will use the preceding formula, the gravitational force will be in dynes. One dyne equals

$$\left(\frac{g \times cm}{\sec^2}\right)$$

you should use a globally defined constant for the universal gravitational constant named UNIVERSAL_GRAVITATIONAL_CONSTANT (don't be afraid to use descriptive names; G is not a good name). Embed your function definition in a complete program that computes the gravitational force between two objects given suitable inputs. Your program should allow the user to repeat this calculation as often as the user wishes.

Please note that the user will have to enter the masses in grams and the distance in centimeters unless you provide some conversion for the input. If you do decide to allow the user to enter the information in some other units then please include at least one option to enter mass as grams and distance in centimeters as that is likely what the graders will use to grade.

Also note: The mass of the earth is approximately 5.972×10^{24} kilograms and the mass of the moon is $7.34767309 \times 10^{22}$ kilograms. The moon is about 384,400 kilometers from the earth. **In your comments section list the gravitational attractive force in dynes of the moon on the earth given by your program using the notation for exponents that C uses for constants (like 6.02e23).**

Name your source code file `program3b.c`.

Rubric: In addition, you must include a comments section at the beginning of each of your files that provides information about the file and its intended purpose. For example,

```
/*
  Author:  Mortimer Sneed
  Course:  91.101.xxx, Computing I
  Date:    //date
  Time spent:
  Description:  This file implements the
                functionality required by
                Program 3b.
   Attractive force between the
```

```
   moon and the earth = ??? dynes
*/
```

The following criteria will be used to grade your submission:

- Does the code compile?
- Does the code function according to the problem specification?
- Is there an appropriate comments section at the beginning of each file (similar to the one shown above)?
- Is the code readable and well-formatted? Is it well-documented and clear?

The available points are distributed according to the following weights:

Correctness:	85%	*Supports course outcome 5*
Comments:	5%	
Organization:	10%	

A program that does not compile or link will not be graded.