

# LAB #4

## Conditionals and Loops in Python 3

### (2 pts) Characters and ASCII Values

Now, look up the ASCII character set in a web browser. Determine the numbers that correspond to the letters in your full name. This includes a capital letter for your first and last name with a capital middle initial (if you have one) followed by a period, i.e. Jennifer Parham-Mocello or Jennifer R. Parham. Write down the numbers for both you and your partner.

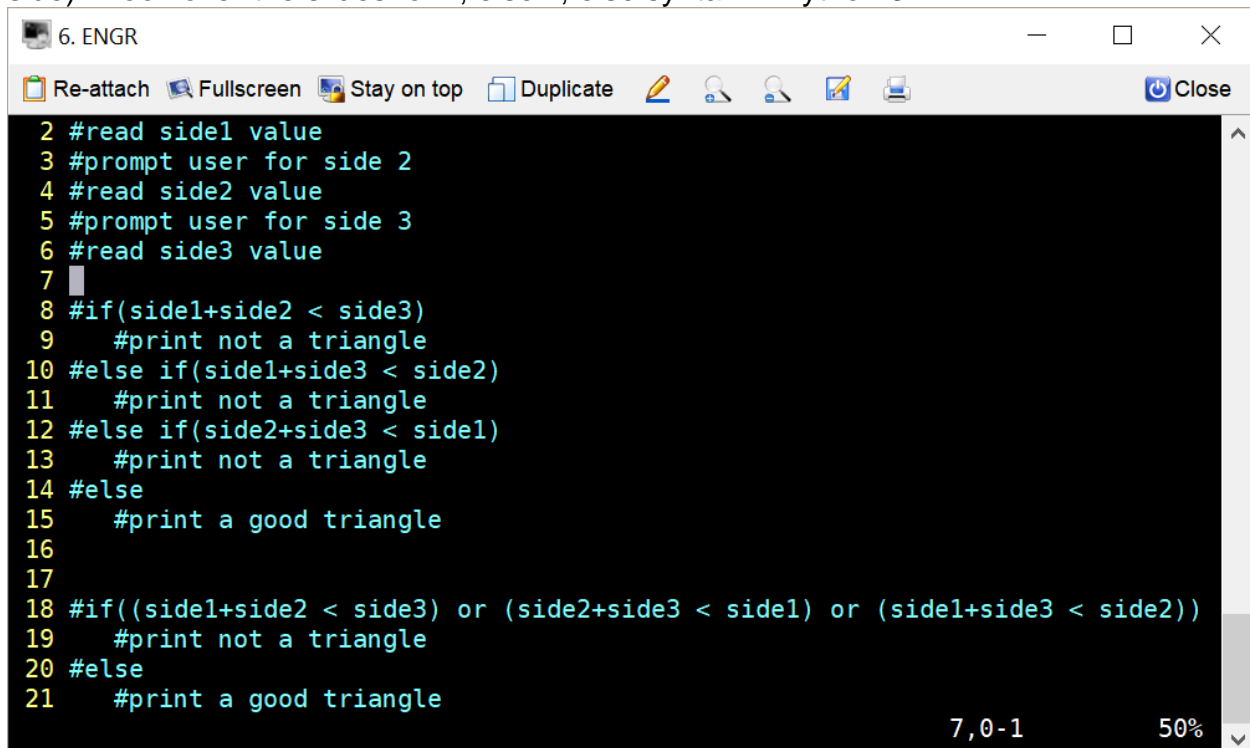
Now, write a python program that uses the ordinal values from the ASCII character set that you found for your full name and prints the characters corresponding to the values. Your name should come out with the letters beside each other and spaces between your first, middle initial, and last name. In other words, you do not want the print function to automatically insert a newline at the end.

Example:

```
print(chr(79), end="")
```

### (4 pts) Program Our Design

Write a python program that will tell a user whether they have entered a valid triangle using the triangle inequality property (any sum of 2 sides cannot be less than the third side). Look over the slides for if, else if, else syntax in Python 3.



```
2 #read side1 value
3 #prompt user for side 2
4 #read side2 value
5 #prompt user for side 3
6 #read side3 value
7
8 #if(side1+side2 < side3)
9     #print not a triangle
10 #else if(side1+side3 < side2)
11     #print not a triangle
12 #else if(side2+side3 < side1)
13     #print not a triangle
14 #else
15     #print a good triangle
16
17
18 #if((side1+side2 < side3) or (side2+side3 < side1) or (side1+side3 < side2))
19     #print not a triangle
20 #else
21     #print a good triangle
```

Modify your program to accommodate the two ways you could allow a user to determine any number of valid triangles. Make sure you use two different looping techniques to accommodate this. Hint: Look at class note slides for the two approaches and syntax.

### (2 pts) Will it sink or will it float?

Now, let's **design another program** that takes the weight (in pounds) and radius (in feet) of a sphere as input and outputs the buoyant force. Use  $\gamma = 62.4 \text{ lb/ft}^3$  as the specific weight of water, and the volume of the sphere is  $(4/3)\pi r^3$ . The buoyant force can be computed by

$$F_b = V \times \gamma$$

where  $F_b$  is the buoyant force,  $V$  is the volume of the submerged object, and  $\gamma$  is the specific weight of the fluid.

Now, **write the python program, buoyancy.py**, that uses your design.

Modify this so that it will continue to ask the user if they want to find out if another object sinks or floats until they do not want to do this task any longer.

### (2 pts) Finding the square root

Design a Python program that takes a **positive whole number**  $n$  as input and **outputs the square root of  $n$**  using the Babylonian algorithm. The Babylonian algorithm computes the square root of a positive number,  $n$ , as follows:

1. Make a guess at the answer (you can pick  $n/2$  as your initial guess).
2. Compute  $r = n / \text{guess}$
3. Set  $\text{guess} = (\text{guess} + r) / 2$
4. Go back to step 2 for as many iterations as necessary. The more steps 2 and 3 are repeated, the closer guess will become to the square root of  $n$  without using `math.sqrt()`.
5. Compare your calculated square root with the `math.sqrt()` result to see which uses a better approximation algorithm.

### Extended Learning:

Write an algorithm to determine if input for your assignment #4 is bad without using exceptions or built-in functions, i.e. it would work in any language!!! 😊