LAB #5
Python Loops and Turtle Graphics

You can choose to work individually or in pairs at this time.

(5 pts) Continue to Practice: Loops in Python
Since we didn't get to have class this week, let's continue to practice conditionals and loops.
Write a loop (or loops) to print the following picture:

```
print("   *   ");
print("  ***  ");
print(" ***** ");
print("*******");
```

First, think about the pattern you see!

- What if you want to make the problem more generic and get the max number of odd stars to print at the bottom of the triangle?

Now, **design an algorithm that would take the odd number of max stars and print the corresponding triangle to the screen.**

Now, exchange designs with another group. Code the design and add error handling for good positive ints and making sure it is an odd number.

> **Error Handling Hint**: take the input as a string and make sure the string is a good positive integer by looking at each character. Here are a few important details about strings:
>
> ```
> my_string=”hello”
> my_string[0] #gives you first character
> len(my_string) #gives you the number of characters in the string
> ```

Stop!!! Take volunteers to show the code and talk about different designs for the algorithm. Discuss the solution given to the TA.

(2 pts) Setting Up Python Turtle Graphics
To use turtle with Python on our server, you have to create a symbolic link in your home directory and use ~/python3 to run the correct python from wherever you are. Make sure you are in your home directory, cd ~, and use ln –s to create the symbolic link.

ln –s /usr/local/apps/python/current/bin/python3 python3

Now, we are going to learn to read Python documentation to create a graphical program using the turtle library. http://docs.python.org/2/library/turtle.html

Another resource is:
http://openbookproject.net/thinkcs/python/english3e/hello_little_turtles.html

First, export the ENGR display to your machine…

In order to see things displayed on the ENGR server, you have to export your display to your local machine. Below are the instructions for Windows and Linux.

Windows: If you are using MobaXterm, then X11 forwarding and X server are already built-in and ready.

Try running xeyes or xclock in your terminal to test that it's working☺

```
% xeyes
% xclock
```

Linux/Mac:

Go to Mac support and download xquartz: https://support.apple.com/en-us/HT201341

Install the xquartz by clicking on the download and then open the package. After it installs, restart your machine.

Now, when you ssh to ENGR, use **ssh –Y username@access.engr.oregonstate.edu**, and this will setup the X11 forwarding.

Try running xeyes or xclock in your terminal to test that it's working☺

```
% xeyes
% xclock
```

Using Turtle…

Practice writing some of the examples in the documentation. You must have these following statements in your program, as a bare minimum. **DO NOT name your file turtle.py!!!**

```
import turtle          #bring in the turtle library

window = turtle.Screen()   #create a variable for the window
my_turtle = turtle.Turtle() #create a variable for your turtle

window.mainloop()     #wait for the user to close the window
```

Now play with changing the background color of the screen, the shape and color of your turtle and pen, and learn how to move the turtle around the screen.

(3 pts) More Python Turtle Graphics

Use *for* loops to make a turtle draw these regular polygons (regular means all sides the same lengths, all angles the same):

a. An equilateral triangle
b. A square
c. A hexagon (six sides)
d. An octagon (eight sides)