CS100 Fall 2015

Name ________________________________

# CPADS Assignment 4 – Due 10-28-15

"Loops"

The goal of this assignment is to introduce you to using the "for" loop construct in Python to calculate repetitive series of mathematical operations. Computer Scientists are constantly concerned with the growth rate of certain functions, as that growth rate can relate to the execution time of certain algorithms. If an algorithm's execution time grows too fast for an increasing data set size, that algorithm becomes inappropriate for use as part of a solution to a complex problem. You will study various algorithms and algorithm complexity later in your course work.

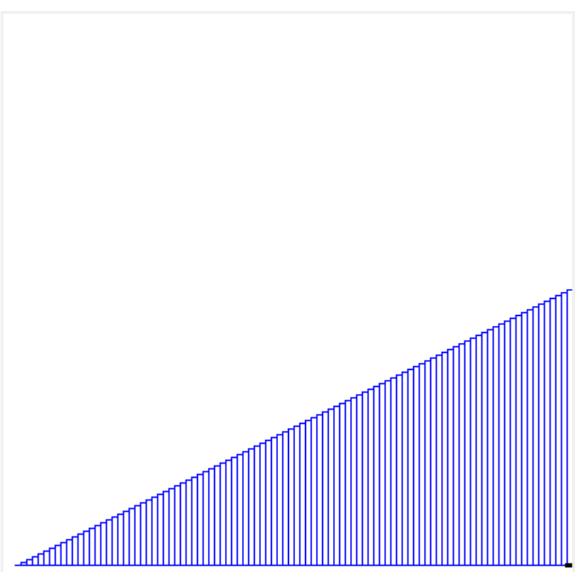
This exercise will give you a numeric, as well as visual idea of the growth rate of various mathematical functions, while you learn to use the **for** loop construct. Your task is to first implement two Python functions that will be used in the graphing portions of the loops, and then to implement the loops that will calculate and graph the requested mathematical functions.

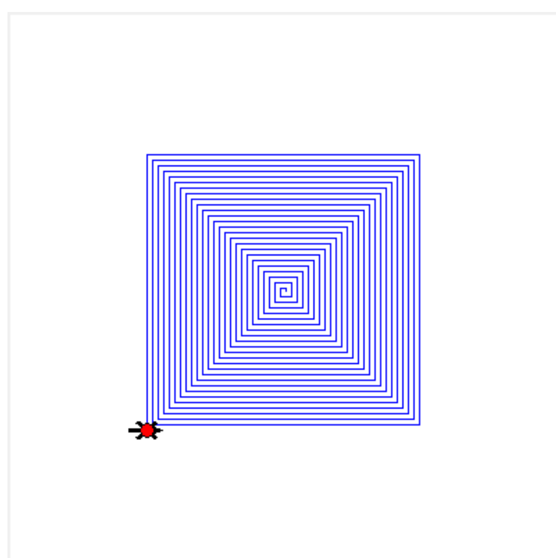
You will graph the functions in two ways: by creating a bar graph, and also by creating a spiral figure, both based on the intermediate terms of the functions as each term is produced inside a **for** loop. Print statements have been supplied that will also output the numeric values from inside each **for** loop so that you can see the growth rate numerically, as well as visually.

To start, download the **Assign4.zip** Python project from the **CS100 Assignments** page, unzip the file into your **CS100** directory, and open the directory in **PyCharm**. The file you will be working in is **loops.py**.

## 1. Linear Function (turtles t1 and t2)

Two examples have been given: a bar graph example and a spiral example of a linear function. Those functions can be used to test the **draw_bargraph** and **draw_spiral_arm** functions that you must first define. They should produce the linear bar graph and linear spiral as shown below:

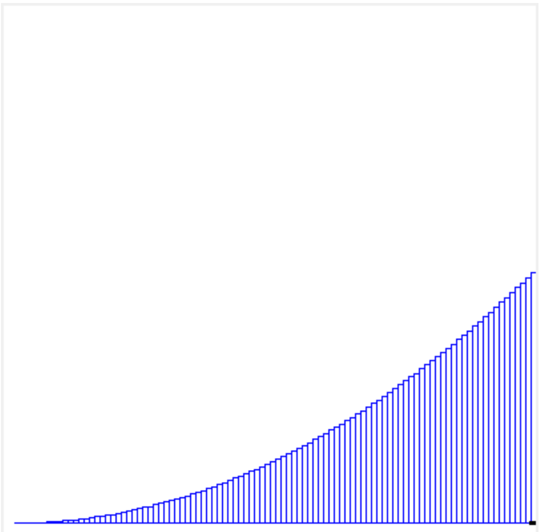


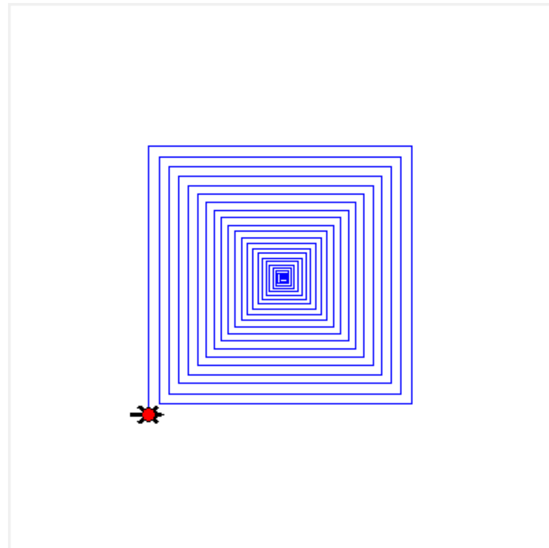


Name ___________________________________

## 2. Squared Function (turtles t3 and t4)

The next set of graphs will be based on the squared terms of the number of entries input by the user. The graphical output will look as follows:



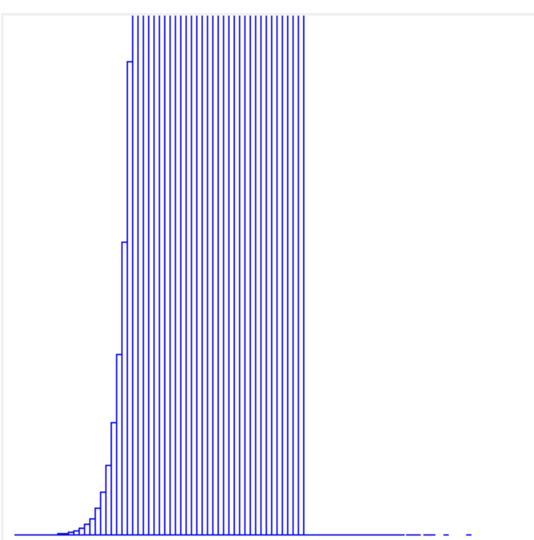


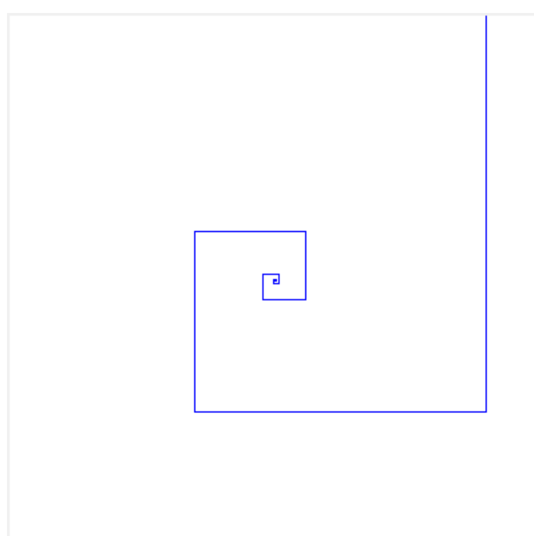
## 3. Fibonacci Series (turtles t5 and t6)

The next set of graphs to draw will be of the terms of the Fibonacci series. The Fibonacci series is renowned for its occurrence in nature in a variety of areas. The first 10 terms look like this:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55…

Each succeeding term is the sum of the two terms immediately preceding it. The first term is the base case, and the second term is considered to be the sum of 0 and the first term. Your solution has been supplied with the two initialized variables: **sum1 = 0**, **sum2 = 1.** You should use those variables inside the Fibonacci loops (along with a temporary variable) to calculate the successive Fibonacci terms. If you implement the loops properly, **sum2** will be the value you use for the bar graph and the spiral through each iteration of the loop. The Fibonacci function grows very fast, as you will see in the console output, and will quickly exceed the ability of bar graph and spiral functions to display it.





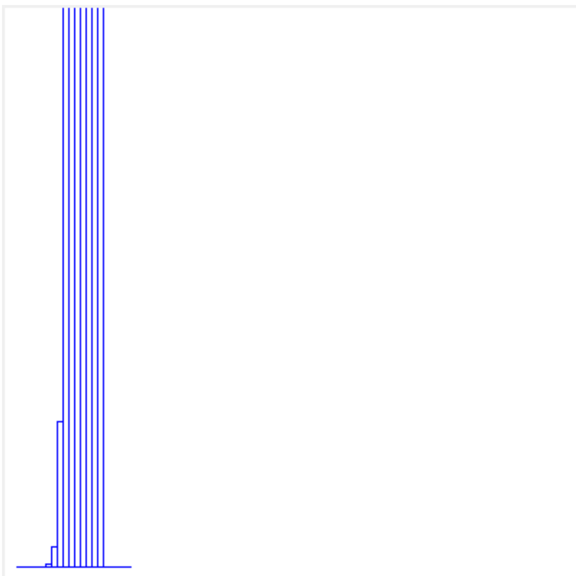
Name ________________________________

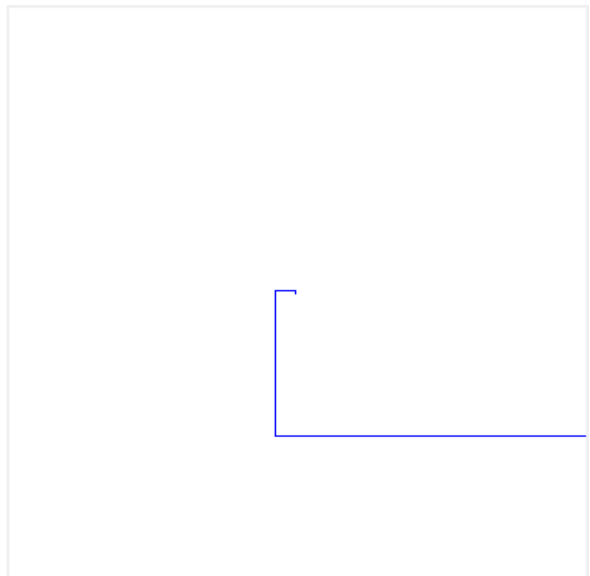
## 4. Factorial Function (turtles t7 and t8)

The last set of graphs is for the Factorial function, which is the product of the integers starting with 1 up to the number of elements entered. The Factorial function is expressed as **n!**, i.e.,

5! = 1 * 2 * 3 * 4 * 5 = 120

This function grows the fastest of all of the given functions, as you will see in the console output, and the graphs. In fact, it very quickly exceeds the capabilities of the bar graph and spiral functions to represent it. To implement this function, you will want to start the **for** loop iterator at 1, rather than its default of 0 (consider why).





## 5. Submit through Marmoset

Once you have completed the assignment, submit your source file (**loops.py**) through Marmoset. Open a web browser (**USE** Google Chrome) and enter the following URL (continue to the website if it brings up a certificate error page)

- **https://cs.ycp.edu/marmoset/**

- Enter your login information.
- Select **CS100: Computer Science Practice and Design Studio**
- Select the **submit** link under **web submission** for **assign4**
- Click **Choose File…** , navigate to your program directory and select your **loops.py** file (do not worry about the instructions for jar and zip files).
- Click **Submit project!**