Luis Renteria

ID 88740232

CS 2302

Dr. Fuentes

Lab Report 1

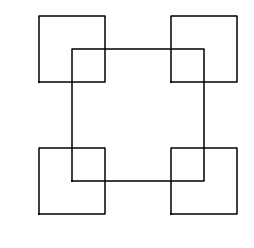
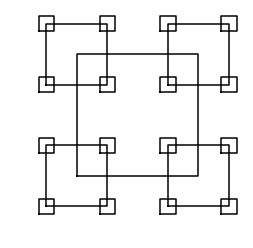
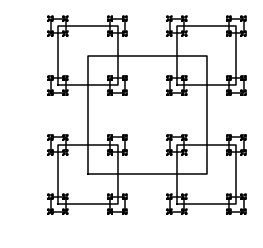
**Introduction**

Our task for Lab 1 was to create different patterns, figures, and shapes using two methods: draw\_squares and draw\_circles. These two methods received a certain input and plotted a shape based on the input. The purpose of the lab was to rework these methods to create shapes that matched the desired outcomes shown in the Lab file.

**Problem 1**

For the first problem, we had to recursively print squares with 4 smaller squares on each corner of the previously printed square(s). By looking at the provided result pictures, I thought that each of the smaller squares appeared to be at least ¼ the size of the larger square that they were attached to.

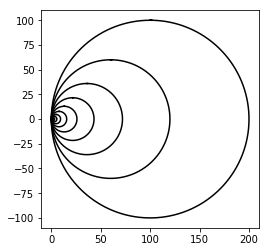
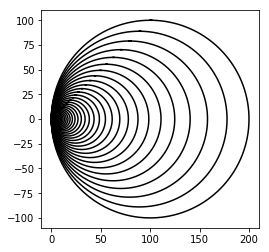
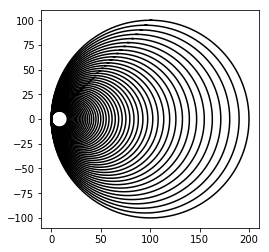
The draw\_squares method receives 5 points and draws a line between these points that creates a square. Using this, I first created the base square of dimensions of 10 by 10, and from there, based the size of the attached corner squares this size. By using a value of .25, I multiplied the original size value by this number, and subtracted it from the values of each point of the original larger square. I stored these points in 4 different arrays, one for each corner square, and recursively called the method 4 times. Although my first result was successful, my next two were slightly off, with the squares being smaller than desired. My results can be seen below:



**Problem 2**

For the second problem, we had to recursively print circles with a smaller radius than the previously printed circle, while also ensuring that the newly printed circles all remained on the right side so that they all intersected at one point. Using the pictures of the desired outcomes, I was able to deduce that the way this is accomplished is by continually moving the center further right while also reducing the width of the radius.

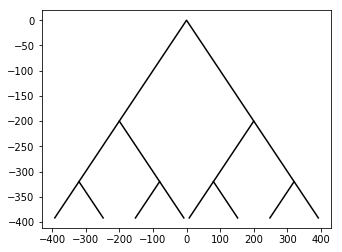
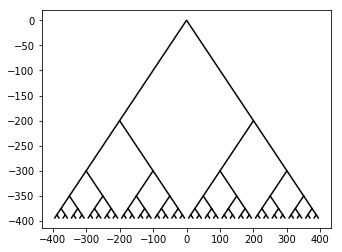
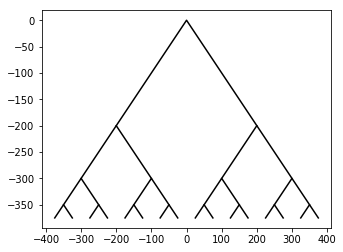
Using the draw\_circles method that was given to us, this task could be accomplished. The draw\_circles method receives, the plot, the number of desired recursions, the center of a circle, and the radius of a circle. Using the desired center and radius, the method plots a circle and recursively call itself to draw another circle. Using this method, I reduced the radius of the previously drawn circle by a certain fraction; for example, in Problem 1 a, the radius is multiplied by 3/5. Using the same radius multiplied by the same fraction (radius\*3/5), I set that value as the x-coordinate of the center, while keeping the y-value the same. This resulted in circles that progressively got smaller while moving to the right on the x axis. My results of parts a-c care shown below:



**Problem 3**

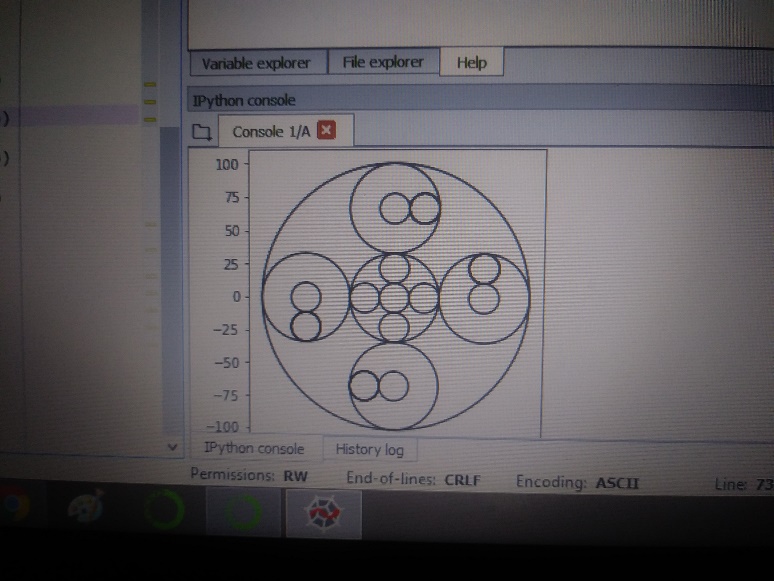
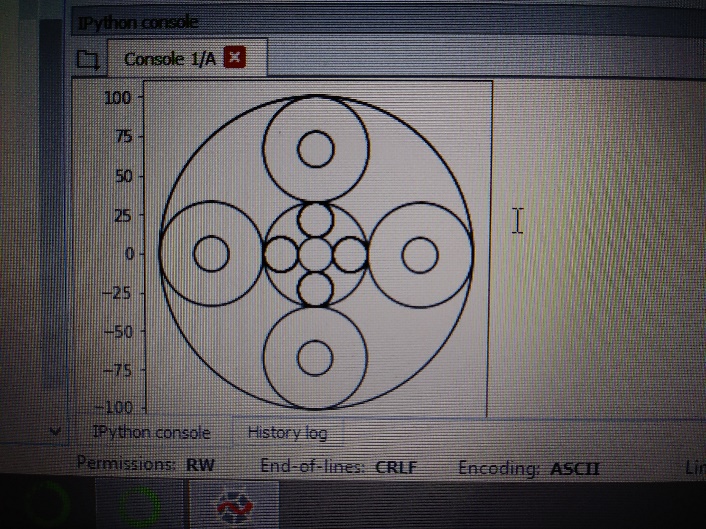
For our third problem, we were instructed to recursively build a tree that continued to split into two branched the further down it progressed. After brainstorming and drawing a few diagrams, I decided that I would create these trees by subtracting values from the coordinates of each point on the tree. Although this resulted in trees that looked different from the tress shown in the lab, they are the same in terms of the number of branches.

Using the draw\_squares method, I reduced the number of points in the array to three. One for the left branch, one for the top origin point, and one for the right branch. I also added a new variable, mod, as a modifier to consistently reduce the size of each branch. For new left and right branch, I created 2 arrays and modified the coordinates of the previously received array, p, so that the left and right branches would grow smaller as the tree progressed, while also changing the point of origin to the location of the previous left or right branch. My results are shown below:



**Problem 4**

For our final problem, we had to recursively print 5 smaller circles within a larger circle in the pattern of 1 in the center, top, bottom, left, and right. After drawing a few diagrams, I determined that the radius of each of the smaller circles was 1/3 of the radius of the larger circle. With this in mind I attempted to plot each of the 5 circles using varied values based off of the radius of the circle that they are within, while reducing the size of the radius with each call to ensure that the circles continued to get smaller and were able to fit within the circle before it. Unfortunately, in the end, I was not able to complete this problem. The first call of the method was successful, but the subsequent calls failed to plot circles within each of the five. I did attempt to correct the program, but was forced to move on to the other problems due to poor time management. Below are two plots that I obtained while attempting to fix the problem.



**Conclusion:**

From this lab, I was able to become familiar with the matplotlib library, as well as how to access coordinates within arrays. Although I am still unfamiliar with this language, I hope that future labs will help me to learn how to use Python to its fullest potential.

**Appendix**

draw\_squares

import numpy as n

import matplotlib.pyplot as plt

def draw\_squares(ax,n,p,w):

if n>0:

i1 = [1,2,3,0,1]

q = p\*w + p[i1]\*(1-w)

ax.plot(p[:,0],p[:,1],color='k')

draw\_squares(ax,n-1,q,w)

plt.close("all")

orig\_size = 800

p = np.array([[0,0],[0,orig\_size],[orig\_size,orig\_size],[orig\_size,0],[0,0]])

fig, ax = plt.subplots()

draw\_squares(ax,15,p,.8)

ax.set\_aspect(1.0)

ax.axis('off')

plt.show()

fig.savefig('squares.png')

draw\_circles

import matplotlib.pyplot as plt

import numpy as np

import math

def circle(center,rad):

n = int(4\*rad\*math.pi)

t = np.linspace(0,6.3,n)

x = center[0]+rad\*np.sin(t)

y = center[1]+rad\*np.cos(t)

return x,y

def draw\_circles(ax,n,center,radius,w):

if n>0:

x,y = circle(center,radius)

ax.plot(x,y,color='k')

draw\_circles(ax,n-1,center,radius\*w,w)

plt.close("all")

fig, ax = plt.subplots()

draw\_circles(ax, 50, [100,0], 100,.9)

ax.set\_aspect(1.0)

ax.axis('off')

plt.show()

fig.savefig('circles.png')