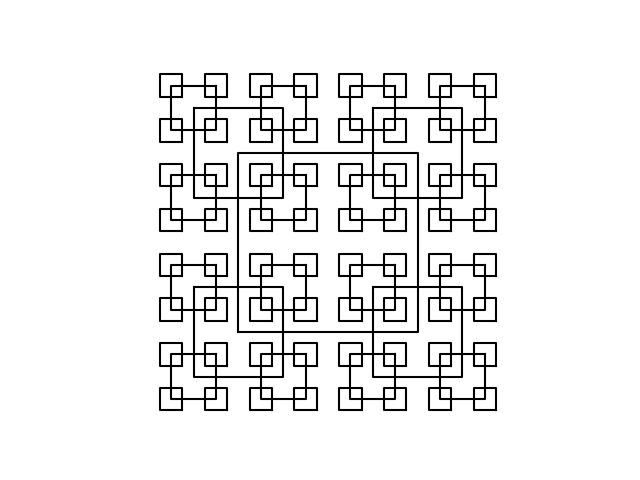
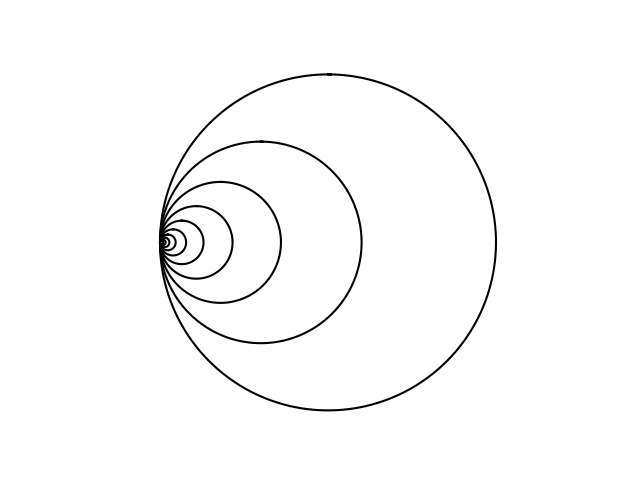
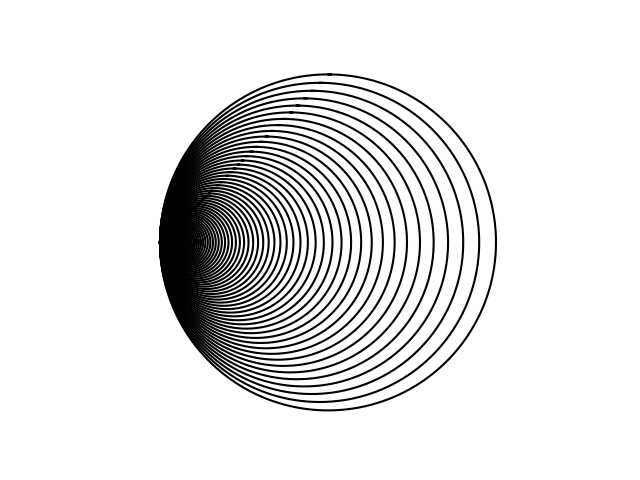
Lab #1

In the lab, we are tasked with replicating the production of shapes using recursion. These shapes must follow a specified pattern of the set example cases. We create our base shape by plotting out the points on a graph, then replicate it, with minor variations, to create our desired product.

For these problems, I started off by experimenting with how the shape was made, using the example shapes that Professor Fuentes gave us through his website. I learned from there how I could edit the code to distort/move the shapes I was dealing with and what the different parts of the code does for both squares and circles. From there I learned how I could translate/move the shape into different positions in regards to both x and y. This I used for Question 1, in which it created a square, and with four recursive calls it makes smaller squares at the corners of the previously made square. With this I was able to completely replicate the example squares.

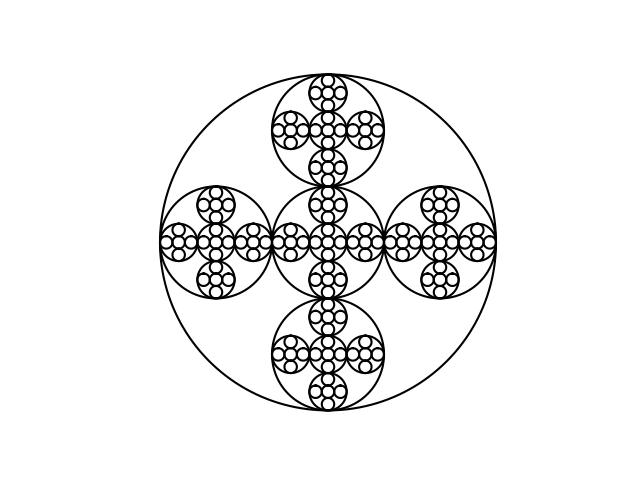


For question 2, it was similar to the squares, except the code for the circles was different. Instead of translating the coordinates of the square, I only had to move the origin of the circle along the x-axis, as well as change the radius of the circle. In order to make it look like the circles converged on one point, I had to change the size of the radius and the movement of the origin marginal to each other, and I was able to do that by creating a percentage that the value shortened/grew by each time. Set it up that changed value to recall and you finish question two. (While experimenting, I learned you can actually use only a percentage of the circle created, as well as change it to a more elliptic shape. Just thought it was interesting.)

Question 3, I was not able to solve. To create the triangular shape, I created three points, that I’ll call X-, X, and X+. The problem I faced when creating this was that when trying to create new branches, instead of the new shape starting at the end of each branch, it instead stayed with a certain point of the base shape. In addition, I was not able to get the shape to get thinner. When replicating the shapes, it produced an exact replica of the base shape, with the same shape and angle. One side would be completely overlapped, and the other would brand out from within the base shape, although not as large. After enough error, I restarted with the idea of creating one base point from X, and from there create a new branch by changing the y-value by a base amount, and adding/subtracting from X to get the points X- and X+ to create the shape. However, I was not able to create the shape, due to errors in trying to create the graph. This is where I left it.

Question 4 I met with better success. Similar to question 1 actually, you had to create a base circle, and from there place five smaller circles within the main one. One in the middle, and the others to the left, right, top and bottom of said middle circle. Dependent on how many times specified, each circle created could become a base, thus creating five circles within itself, and so on. I found that the radius of the circles was one-third of the base one, as three circles make up the diameter of the base. The next part, in translating the circle properly, and getting each circle to hold their own respective circles, you had to add/subtract the radius-times 1-**w**, (which was 1/3) from the x/y values to get the circles around the middle. Set up those specifications into 5 different recursive calls and the method is complete.



Overall, from this project I improved on writing python, and through it I learned how to graph, create circles, edit those shapes repeatedly, and overall how to follow through with recursive code, and how it consistently repeats upon each recursive call within it. I learned how you can edit the graph to draw edit the points to manipulate those lines.

**Source Code:**

<https://github.com/Penguinhedgehog/CS2302-PatrickBrannan/tree/master/CS2302/Lab1>