Ericka Najera

11 February 2019

CS2302 MW 10:30

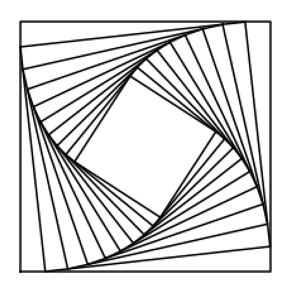
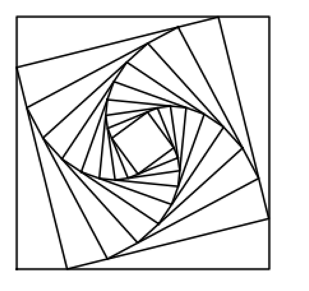
Lab 1

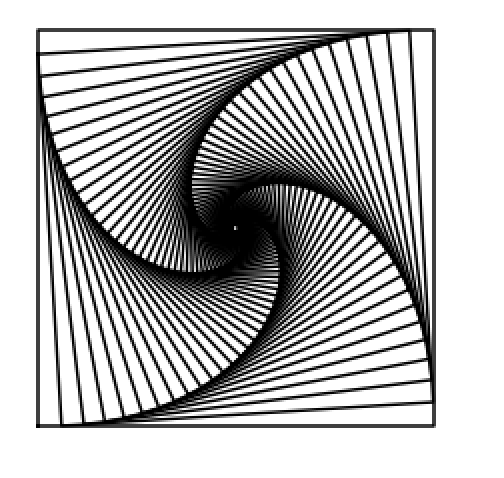
The implementation of recursion and matplot.lib was the basis of this lab. The idea was to recreate figures provided using the modules the professor provided as well as creating new recursion methods to create new figures. Not only did we have to find how to draw the figures but know the exact number of repetitions the recursion has to be called in order to produce the figure exactly how it is shown.

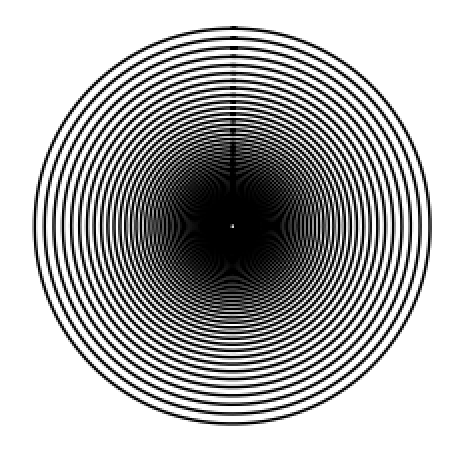
First I tried to understand the method Dr.Fuentes provided and played around with the inputs and saw what the modules did exactly. For the modules I had to create I had to know the exact inputs of the origins of each corner of the squares or the radius and center of each circle. I decided to draw the square figures in a graph sheet. I plotted what the first points as well as the next points that should be called when using the recursion. From there I realized that all points had either a quarter minus or plus of the size of the square on each point. I came up with that by calculating the difference between the origin and new points. From there I decided to make 4 different calls for each point of the square and alter all the points I was sending.

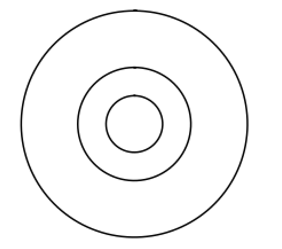
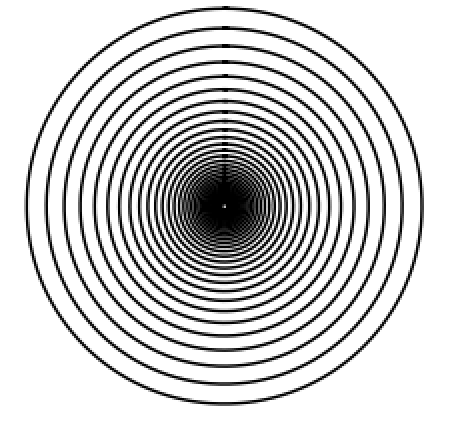
For the circles inside circles I also drew them in graph sheet and measured how big was the radius and how short it had to be in the next call which was the radius split is three, because of the three circles on each axis. There I had to measure where to put my center of each square and divide the radius all using 5 calls.

As for the circles with the same origin I had to send the same origin to all the circles but change it according to the width of each circle I am sending. When it came to the binary tree I also drew by hand in a graph sheet. I realized that I had to draw two lines to each extent then keep that point as the origin for the next two lines. Considering how much height the binary tree would have I would divided the size of the y axis by the height and x axis will always divide by two on each call.

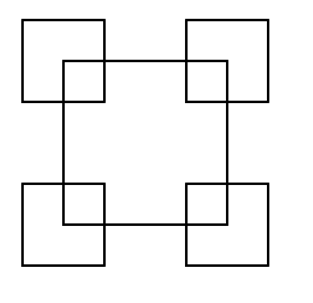
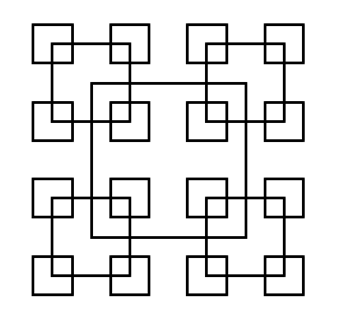
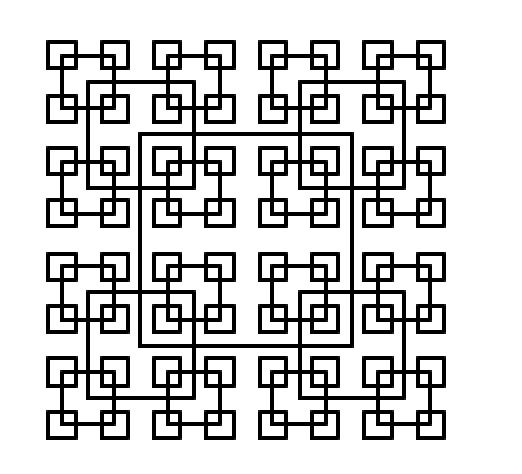
The first two modules in which we had to play around in order to get the figures, I just started to change the input by the repetitions and the width of the how apart each call is going to. I played around until I got the following figures.

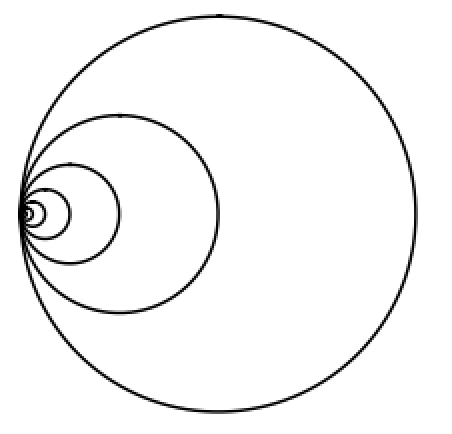
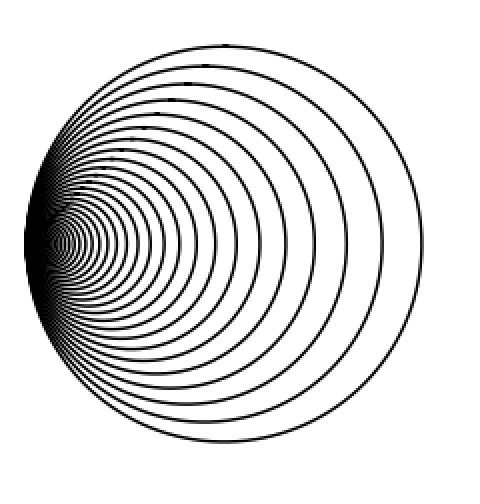
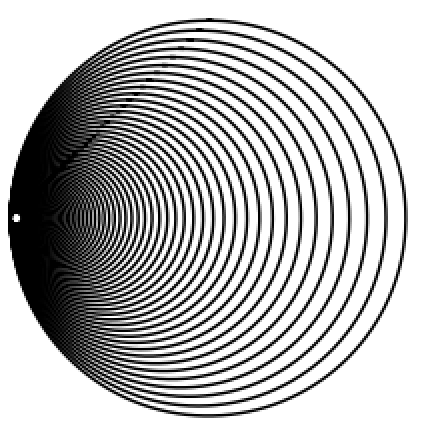






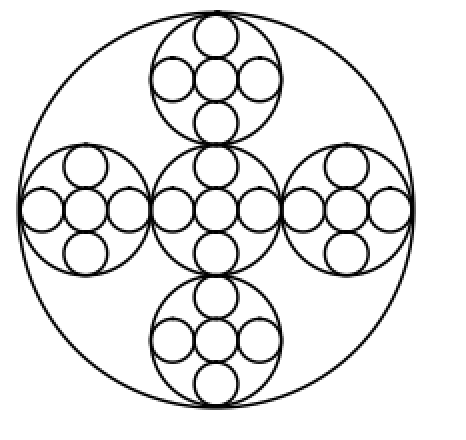
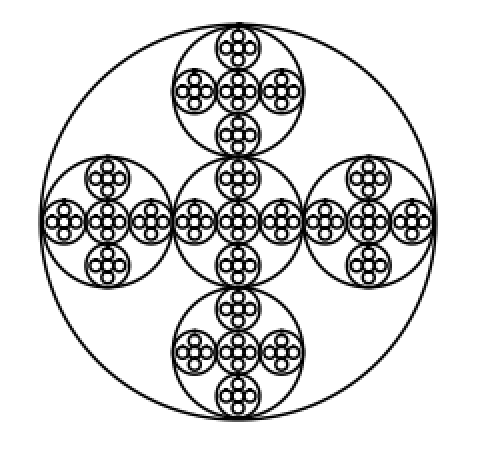
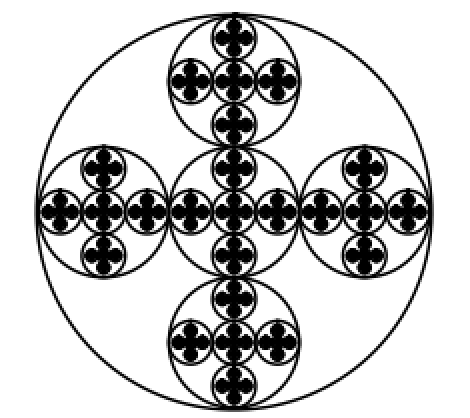
The squares separating into four squares module was one module I took a long time debugging. I remember I managed to print the 5 squares from the begging but then when I made another call it wouldn’t print or they would print in the wrong place. Until I finally realized that I wasn’t decreasing the size of the size of the square I was just leaving it the same. So each time I decreased by two. Finally I managed to get the figures and just increasing the number of repetitions it should make.



Creating the module of the circles with the same origin was fairly easy for me. One problem I encountered was shifting the radius just a little bit to make the circles align. To fix that error I took the x or the y of the origin and multiplied the radius and the width on every call. The x and y was depending on which circle it was since two circles were on each axis. All was left was alter the number of repetitions and the width in order to get the following figures.

I wrote the binary tree module by hand assuming that if I used the plt.plot it would mark the lines but when I typed it and run it, it didn’t work. No matter how I plotted the lines nothing would work. even though I had all of the correct measurements I just couldn’t make the lines plot where I wanted them to.



The last module I had to create was the recursive circle inside each circle. I remember I encountered the problem of the right circle being too to the middle. From there I realized I had to divide the radius into three but then add double the new radius or subtract to the x or y of the origin, depending on which axis it is in. After I got the first call I managed to get the keep adding repetitions until I got the desired figure.

From this lab I learned that for me it is better to draw or sketch it by hand and go step by step by creating a flow chart and brainstorm list. I learned how implement a little of matplot.lib since I am completely new to it. I learned that you can draw figures by drawing each circle or square but instead you can make a recursive module to do the work for you with just one input.

I Ericka Najera certify that this project is entirely my own work. I wrote, debugged, and teste the code being presented, performed the experiments, and wrote the report. I also certify that I did not share my code or report or provided inappropriate assistance to any student in the class.