Programming Assignment 1

Roll no: 2023115008

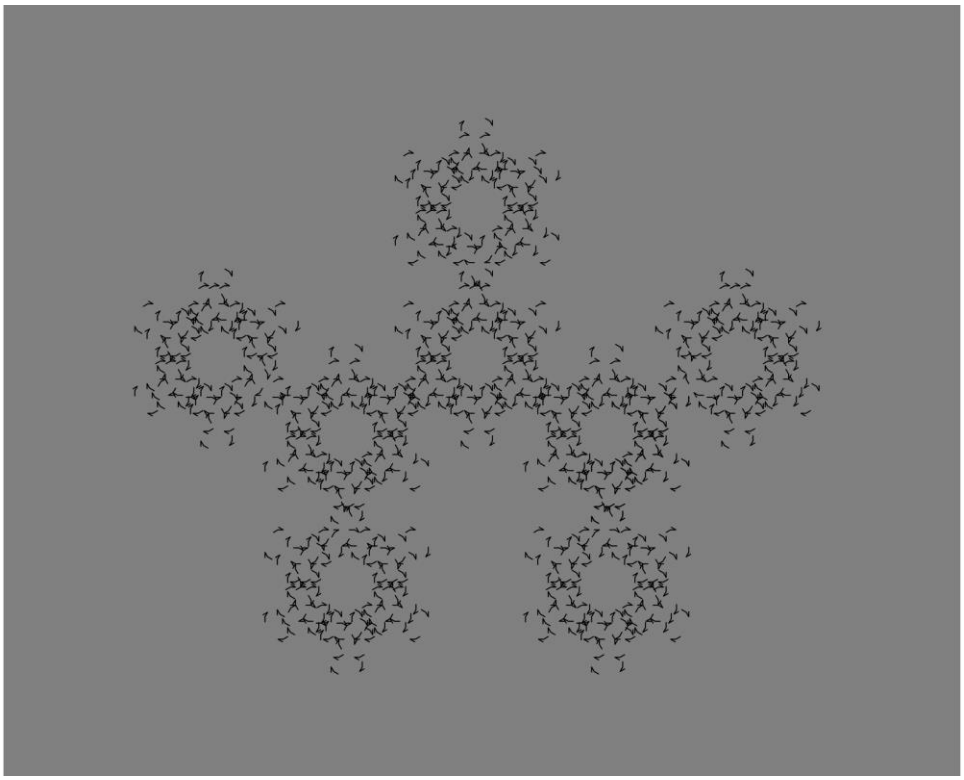
Q1) In this question we have only one alphabet, first inputted all transitions using a list and we can consider a variable to show the present state , and update that variable to move states while adding in the frequency list but it would O(X) complexity, we know that there would be a cycle if X>N(N is no of states, X is size of input) so we would be going through the same cycle multiple times so instead with some book keeping we can identify the cycle check how many times the cycle would have run originally using modulo and int division functions and assign them , this we are only moving through the cycle twice , since the max length of the cycle can be N the time complexity is only O(N).

Q2) For the second question I used JavaScript, for 1,2 and 3 parts of the question I used the “p5” drawing library,

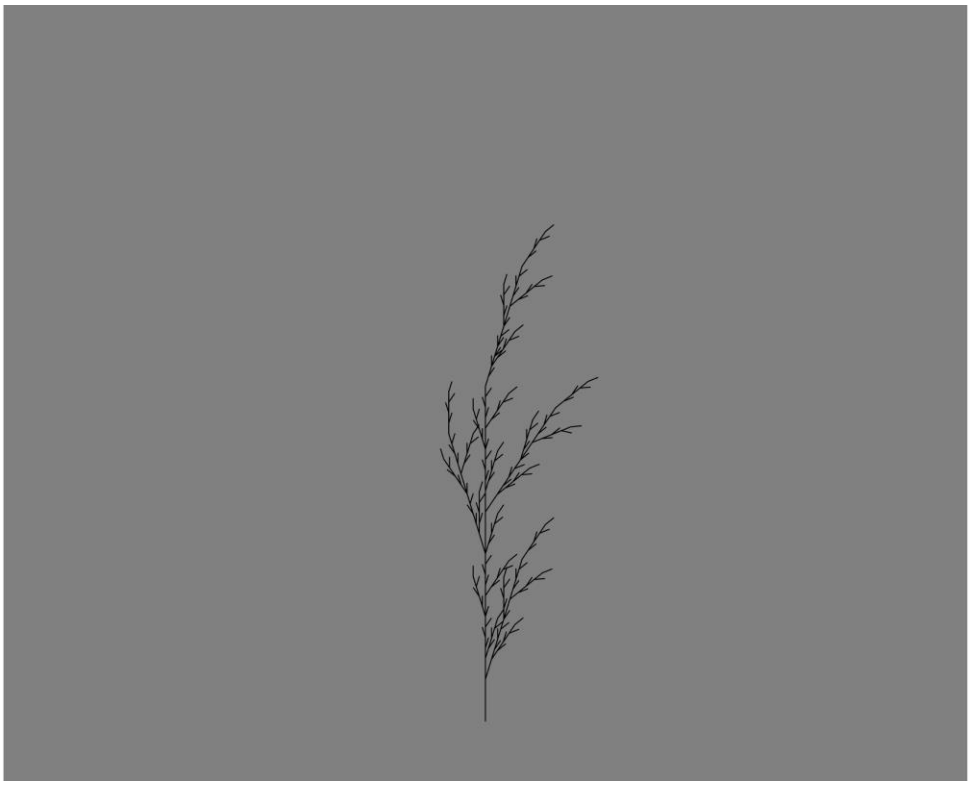
I made a rules library with that variable as a key and its output and the value, I wrote a generate function that goes through all the characters in a string and appends that variable if it is terminal and appends the value and the function outputs the new final string and I iterate this generating function multiple times to get the final string, I wrote a set of functions for F,G,+,-,[,] using the p5 library, then i made the canvas and drew the final string using the set of functions.
For the part 3 I also wrote a random function that takes a random number typecast it to int and modulo 2 or 3 append it to X or Y to get a random output for a variable,

For part 4 I just used the Lindenmayer library, modified the axiom production and finals.

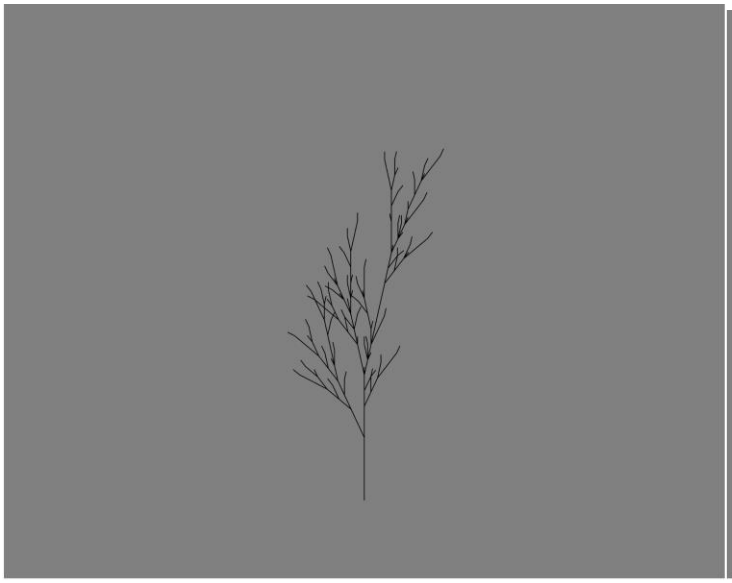
Mirrorball (11 iterations)

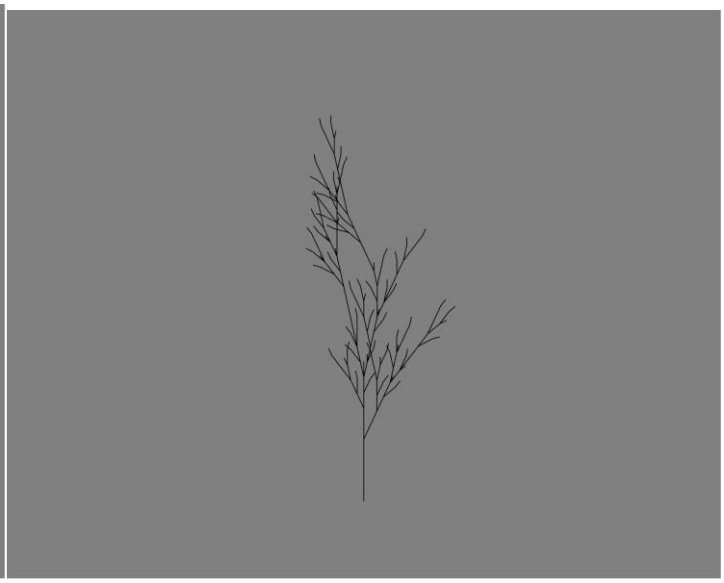


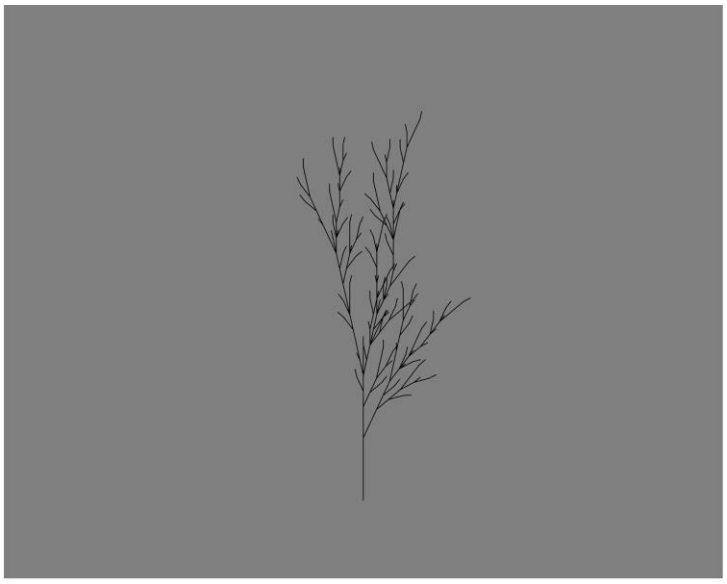
Tree(4 iterations)

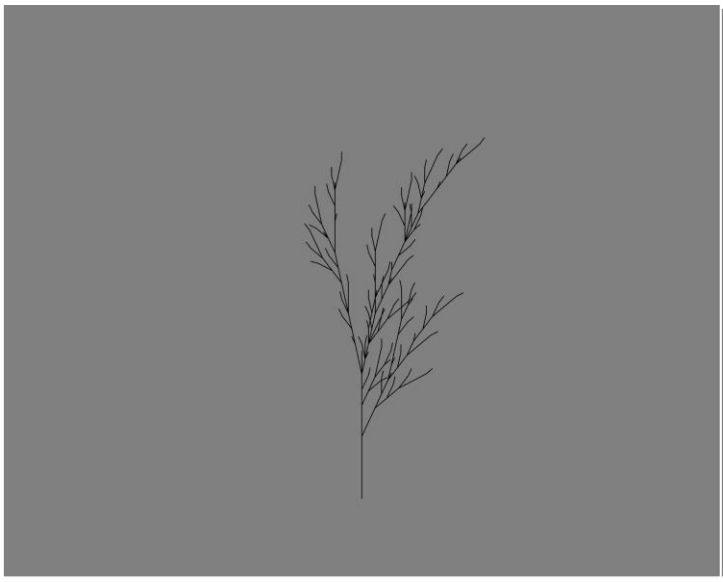


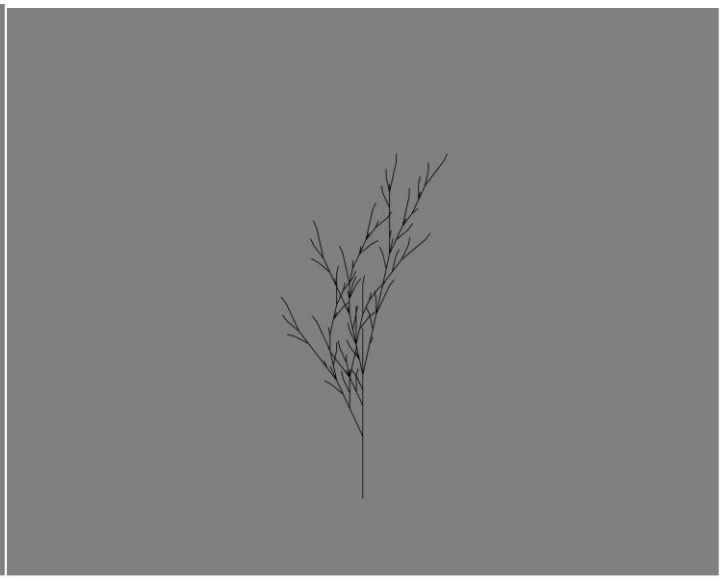
Anything(4 iterations)

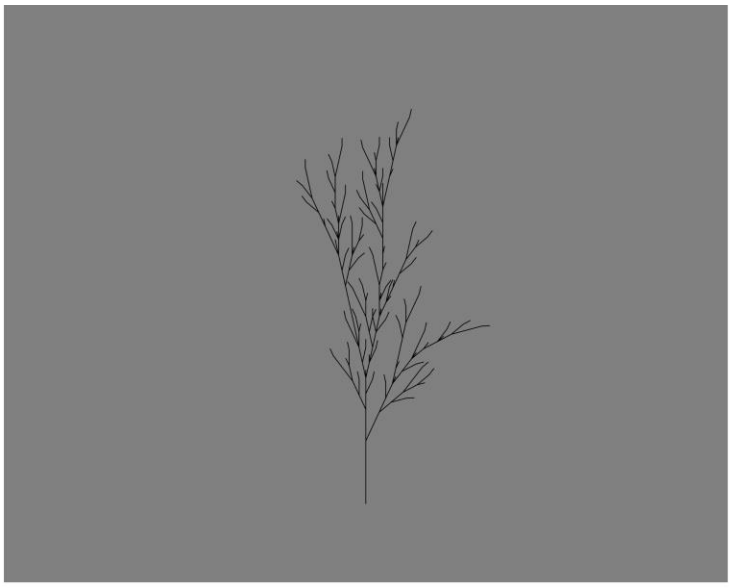




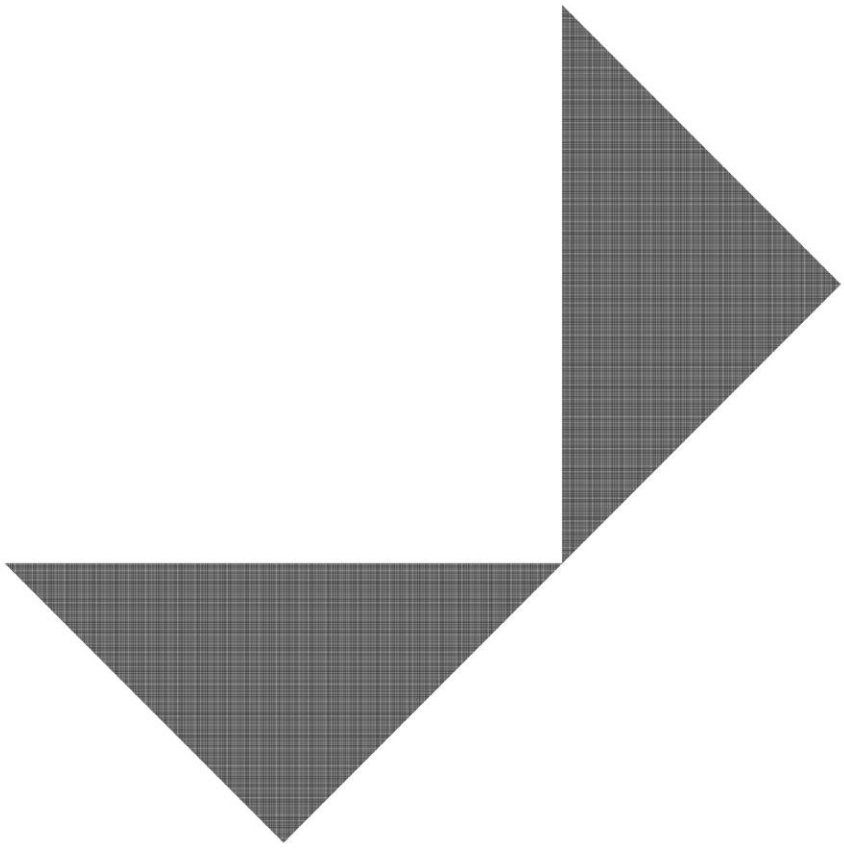








Noise (9 iterations)



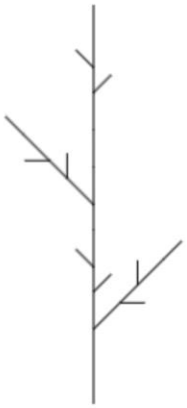
Q3) For the first part “stick plant” you can see that the right branch, center line, left branch, and above branch are all equivalent in both the iteration, and in 3rd iteration every branch equivalent to the entire plant in 2nd iteration, so in 2nd iteration every branch must be equal to entire plant of 1st iteration

And 1st iteration is straight line turn right straight line come back straight line turn left straight line come back straight line, but here the difference is the first straight line is not the same as others it doesn’t transform so it is another variable G for just a straight line when I did this I noticed that in 2nd and 3rd iteration the top branch is a little closer than in image so I added another G there but in 3rd iteration they were getting closer so I made KG instead of G and K transforms in KG increasing G by 1 in each iteration.

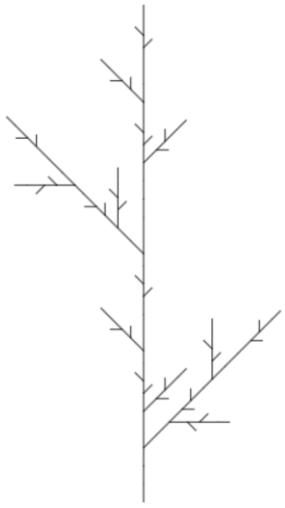
For the second part “Santa", it is a famous l system Koch Snowflake, rotated it in the axiom and set the angle to make it look closer to the provided image

Stick plant

2 iterations



3 iterations



Santa

4 iterations

5 iterations