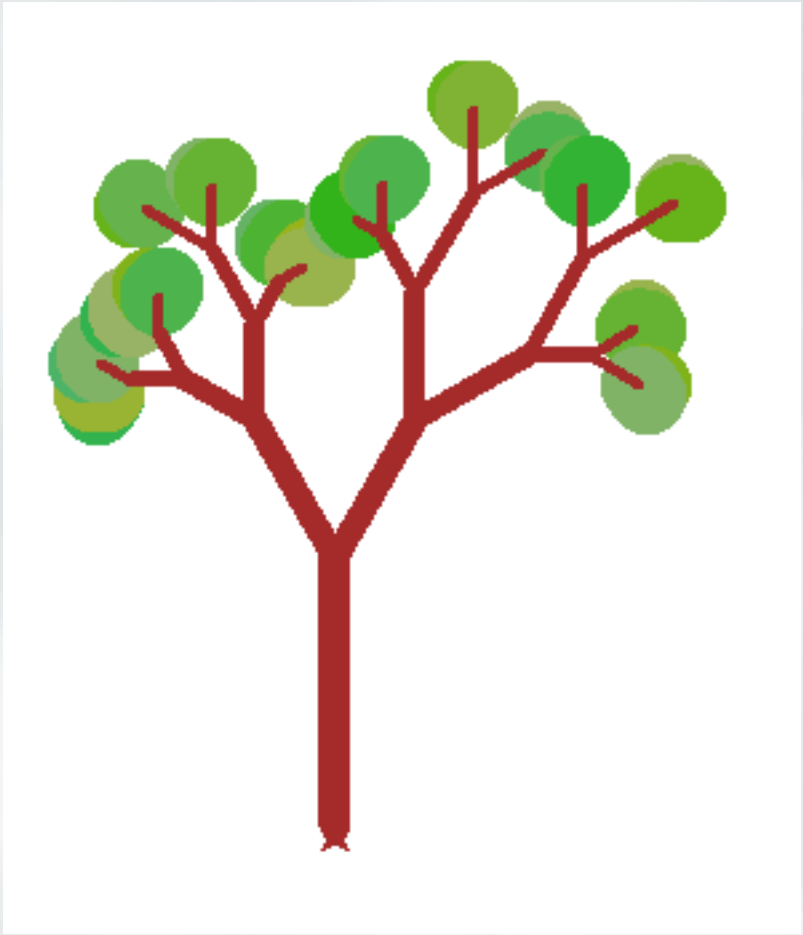
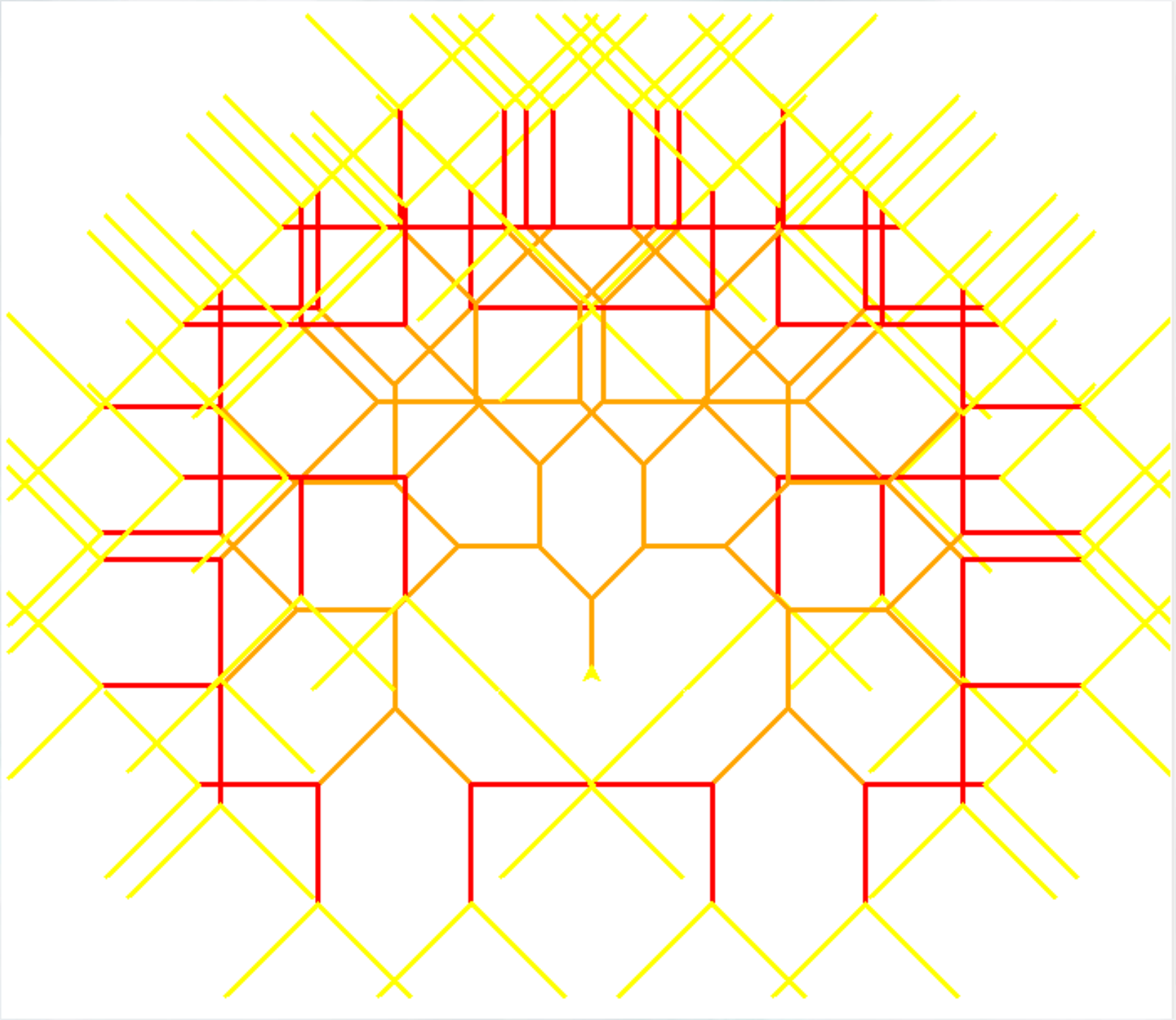
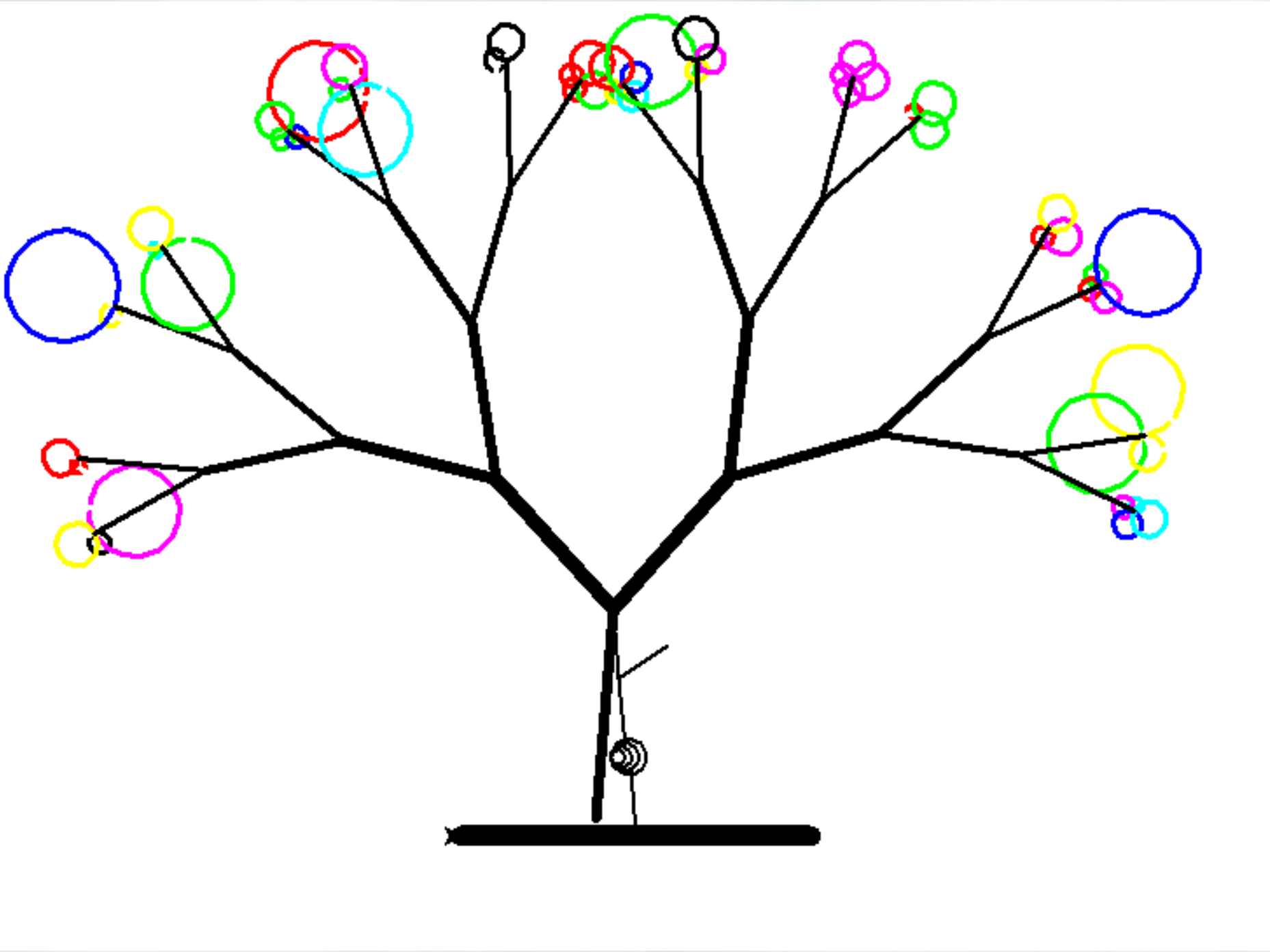
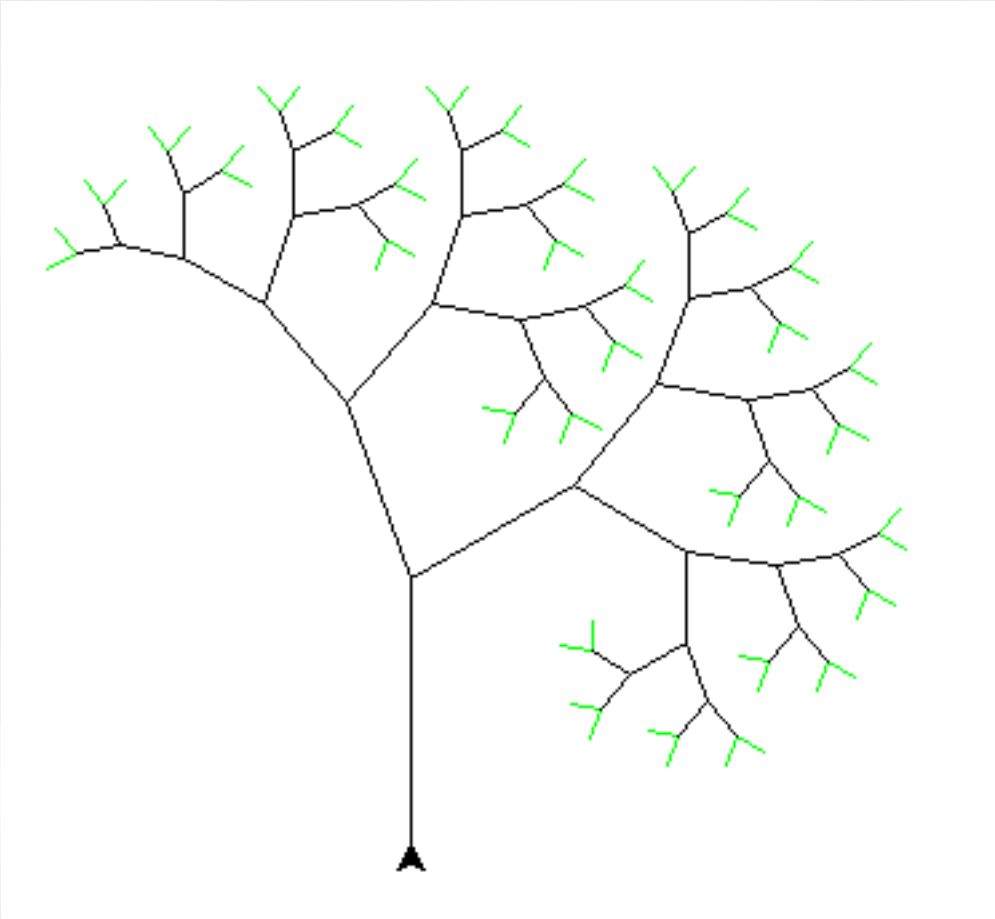
CIS 122

Throwing you for a loop

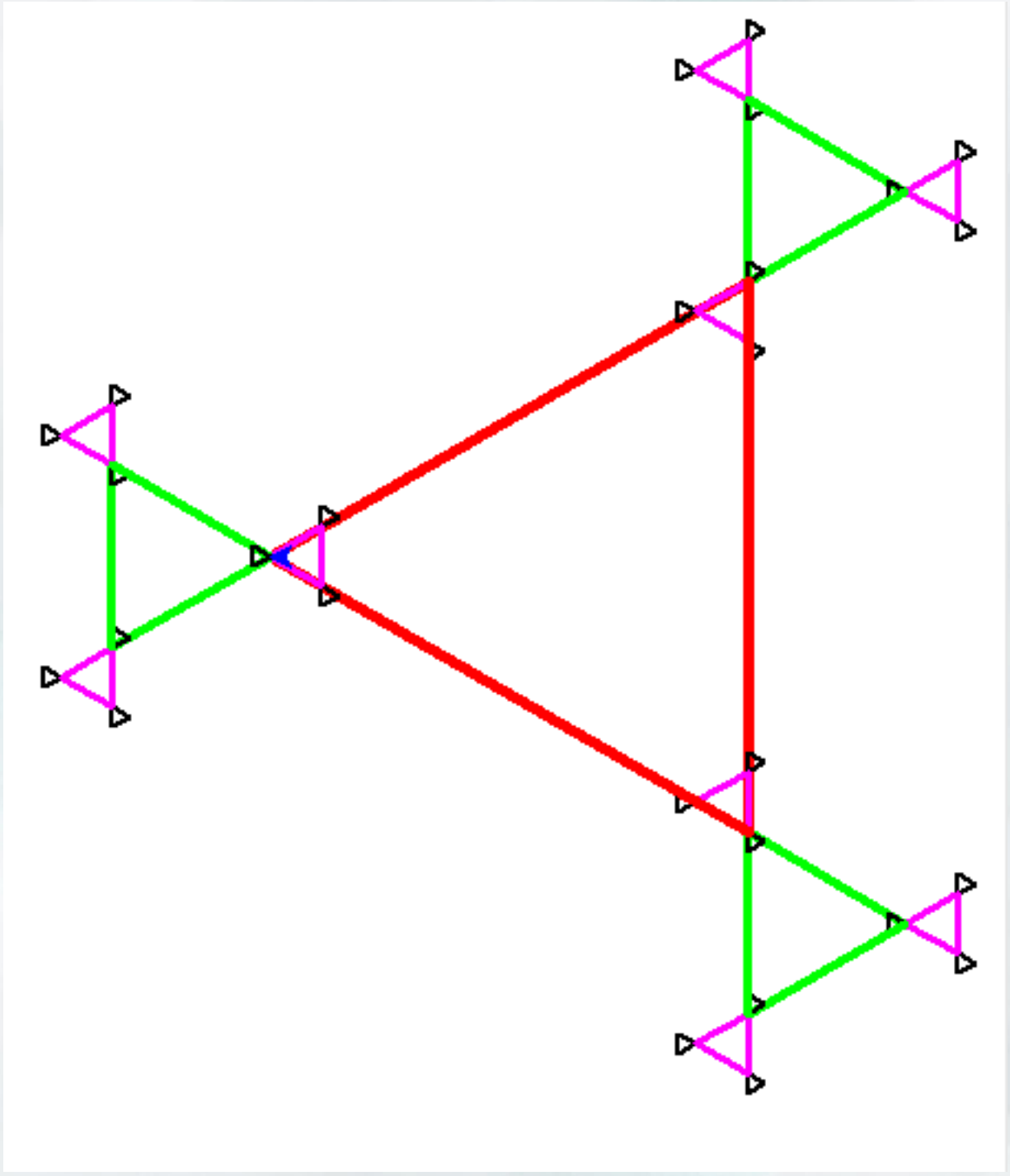


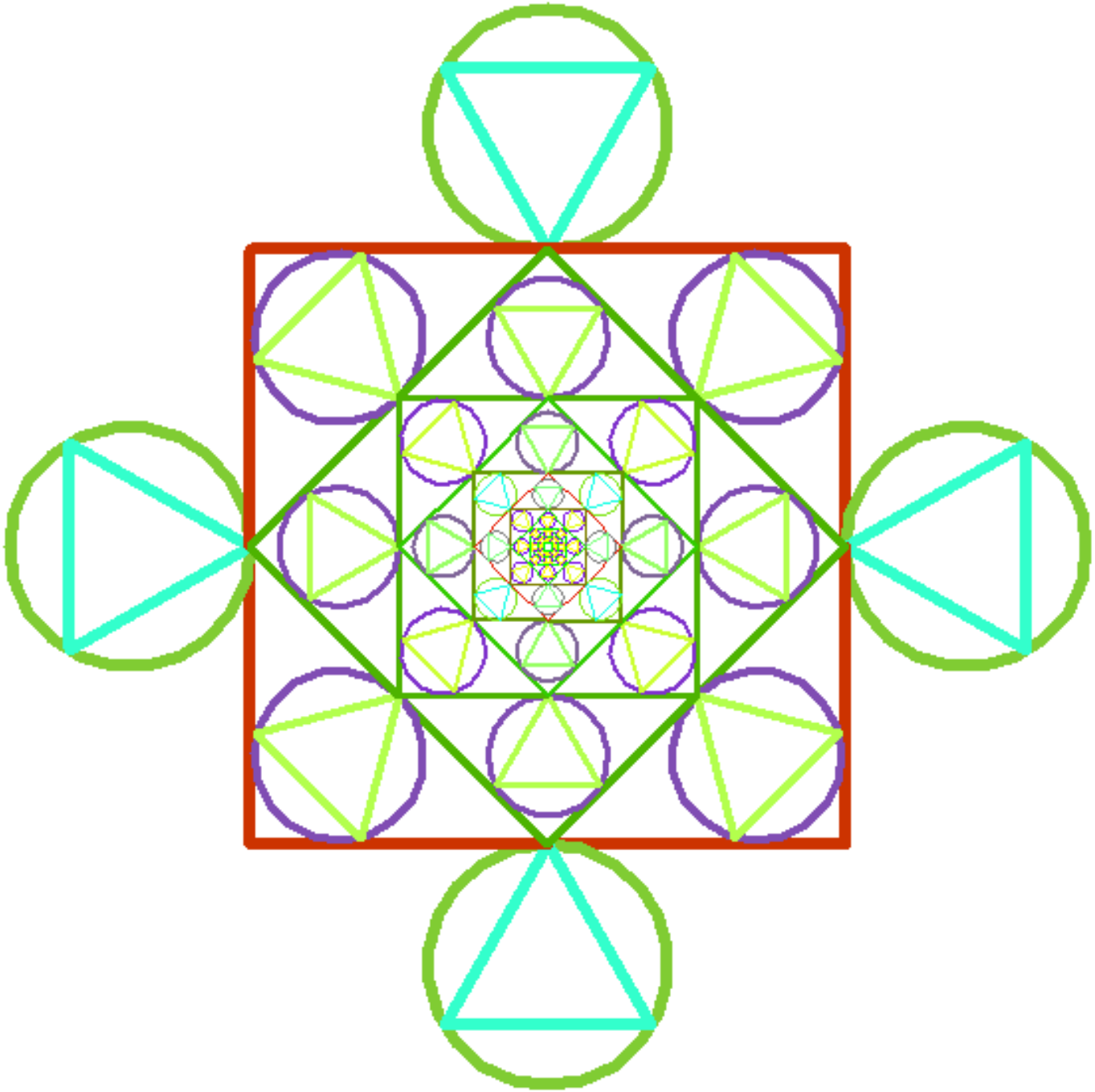


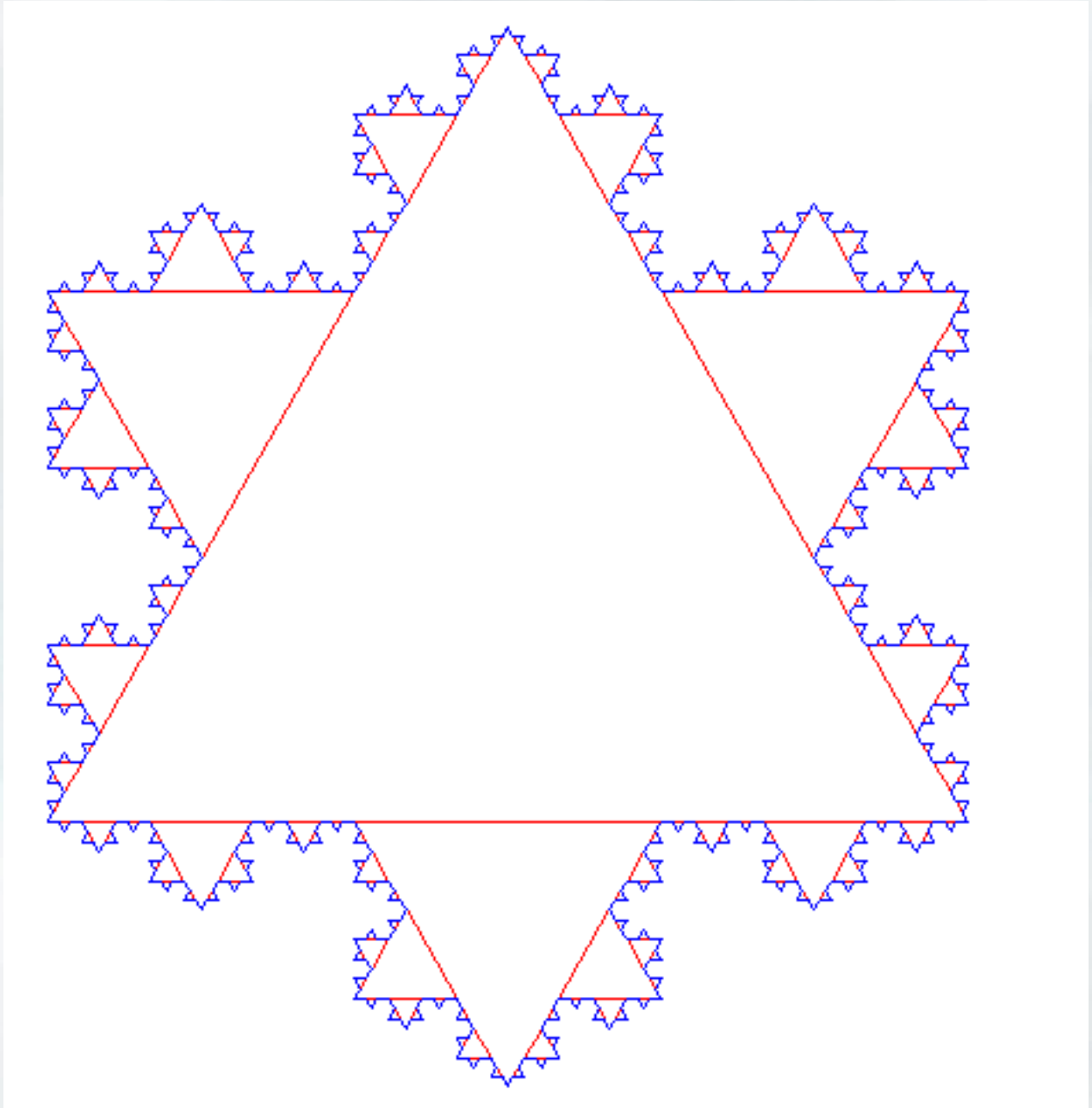


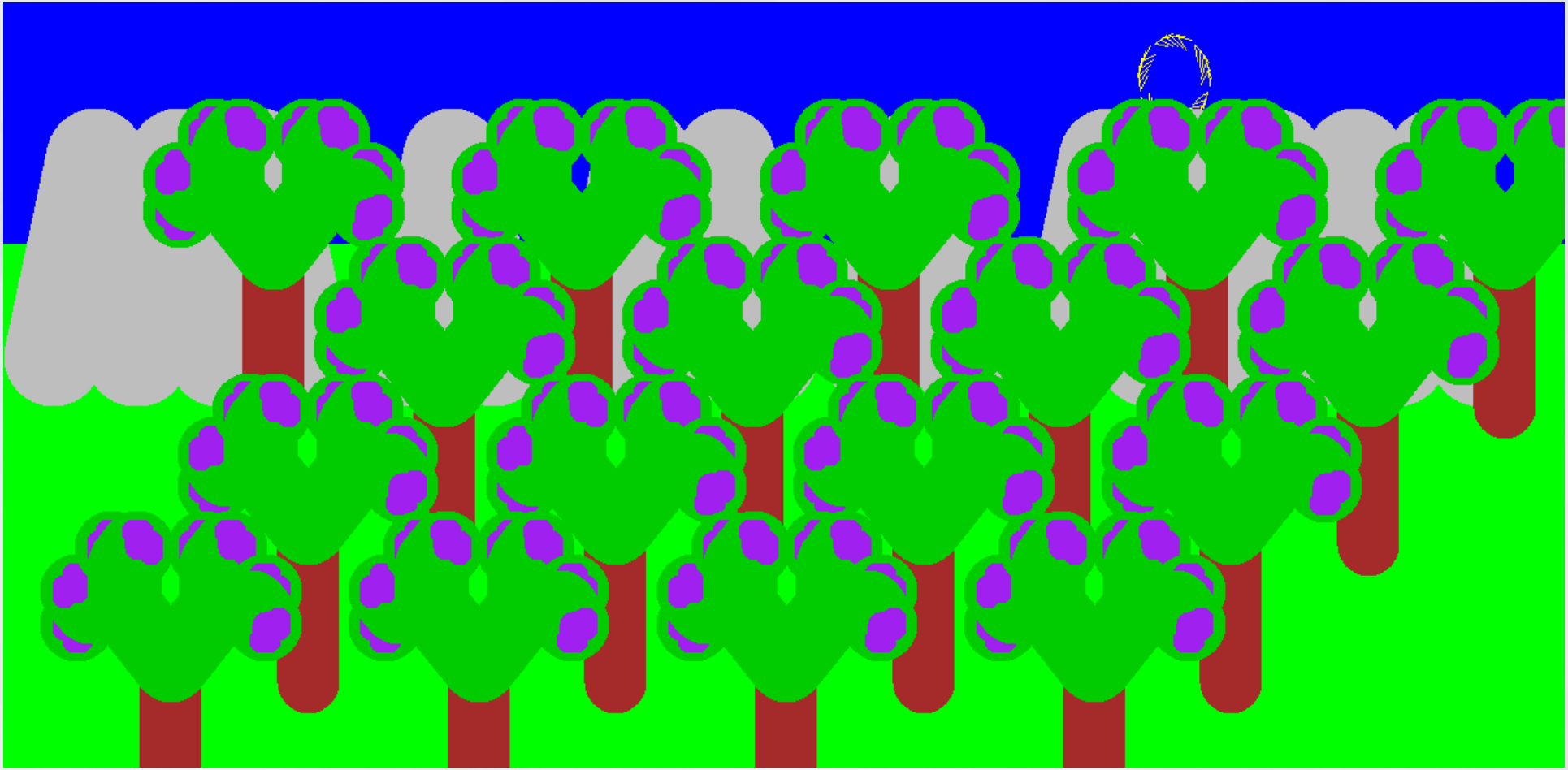


fractalTree(depth, trunkLength, angle1=60, angle2=20)









Put it in reverse

- Yesterday, we tried to reverse a string
  - Instantiate a new string
  - Loop through each character in the old string
  - Add each one to the new string

```
def reverse(string):
    rev = ''
    for char in string:
        rev = rev + char
    return rev
```

- This just gives us our old string back...

Put it in reverse

- Yesterday, we tried to reverse a string
  - Instantiate a new string
  - Loop through each character in the old string
  - Add each one to the new string

```
def reverse(string):
    rev = ''
    for char in string:
        rev = char + rev
    return rev
```

- That's better!

Put it in reverse

- Accumulator Pattern
  - Initialize a variable
  - Loop through a sequence, modifying that variable
  - When we're done, we've got some useful value

```
def reverse(string):
    rev = ''
    for char in string:
        rev = char + rev
    return rev
```

Put it in reverse

- Accumulator Pattern
  - Initialize a variable
  - Loop through a sequence, modifying that variable
  - When we're done, we've got some useful value

- What happens if we initialize our variable inside our loop?

```
def reverse(string):
    for char in string:
        rev = ''
        rev = char + rev
    return rev
```

Back to max

- Let's write a max function
  - Given a list of numbers, return the largest
  - Generalization from assignment 1
- How would we approach this problem?

Back to max

- Let's write a max function
  - Given a list of numbers, return the largest
  - Generalization from assignment 1
- How would we approach this problem?
  - Instantiate a max variable
  - Loop through elements in list, updating max when we can
  - When we're done, we must have the largest value
- Give it a shot
  - Work with the students in your row

While Loops

- Avoid using break statements when you can
  - Tend to make code less clear
  - A good loop condition is far more readable
- If you use break statements, comment them well

```
x = 0
while x < 10:
    print x
    x = x + 1
```

```
x = 0
while True:
    print x
    x = x + 1
    if x == 10:
        break
```

While Loop Practice

even x → x/2
odd x → 3*x+1

- Implement collatz(x) using a while loop
  - How many times do we need to perform HOTPO on x before it reaches 1?

- How could we use a while loop to solve this problem?

While Loop Practice

even x → x/2
odd x → 3*x+1

- Implement collatz(x) using a while loop
  - How many times do we need to perform HOTPO on x before it reaches 1?
- How could we use a while loop to solve this problem?
  - Initialize a counter to 0
  - While x hasn't reached 1...
    - Apply HOTPO to x
    - Increment counter
- Go do it!
  - Work with the students in your row