Recursive Turtle
Lab/Project for 5001

In class we explored the concept of recursion and the stack. For recitation I'd like you to practice with recursive functions by incorporating them into work you've done with Python's Turtle.

All images should be drawn with recursion and you should use no while/for loops.
All tasks should be separated into individual method calls.
Organize your code logically and see if you can call previous created functions from other files you've generated in previous labs.

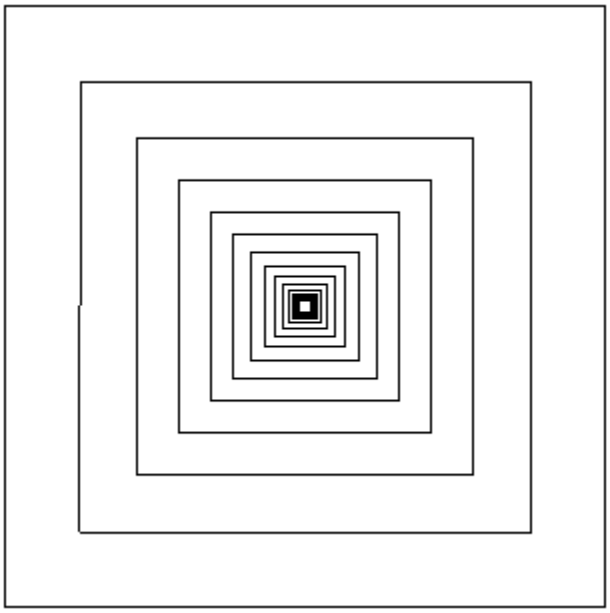
Tip: Just draw two iterations first, and then figure out how to use recursion.

Task 1: Psychedelic boxes

Create a pattern of boxes one inside of the other. Your function should accept an x and y position, an initial size, a scaleDown factor, and a minimum size.

Your scaledown factor is how much you want to reduce each square. For example, a factor of .5 would reduce each square size by half and .25 by 1/4. Make your task easier by using functions you've already created like goto(x,y) and box(x,y,size).

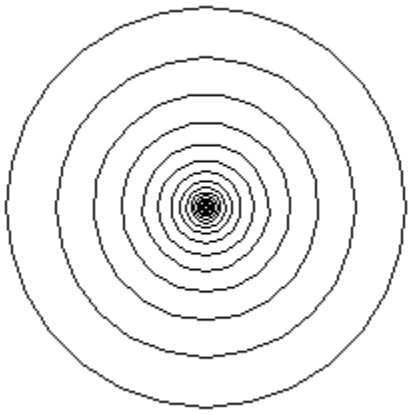
```
def boxes(x, y, size, scaleDown, min_size):
```



Task 2: Bull's Eye

Create a pattern of circles within circles. Your function should accept an x and y position, an initial radius, a minimum size(when to stop), and a scaleDown factor similar to task 1.

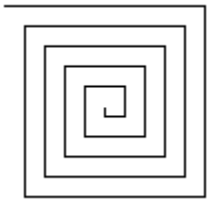
```
def circles(x, y, radius, scaleDown, min_radius):
```



Task 3 : Inward Turning Spiral

Draw an inward twisting spiral that decreases by 5 each turn until line length is less than or equal to 0. This method should take an x, y value and a starting line length.

```
def drawSpiral(x, y, lineLen):
```

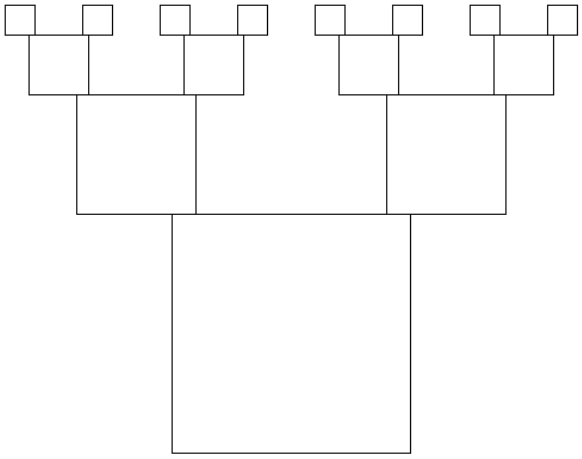


Task 4 : Box Tree

This one will require a little more work as the recursion goes in two different directions. The easiest way to get started is to focus on drawing the left half of the tree and then add the right half.

Each box has another box offset from the right and left corners. I shifted to the left by .4 and to the right by .9 to get the appearance below. Your tree doesn't have to look exactly like this, but I expect something similar.

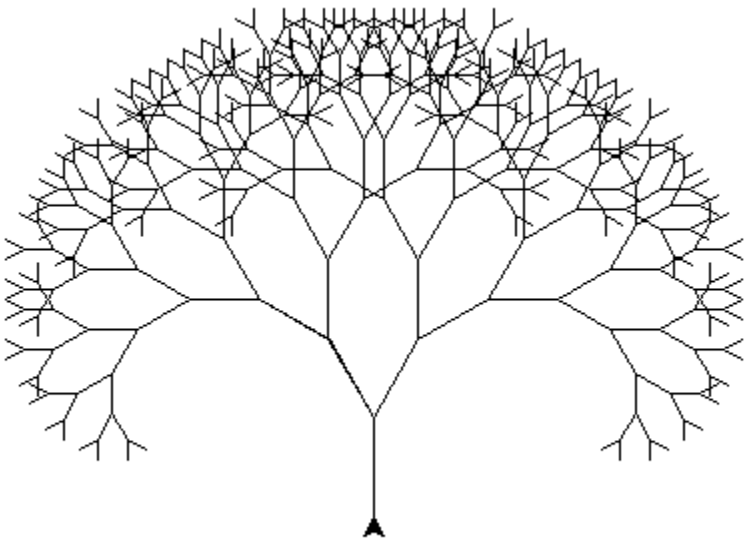
```
def boxTree(x, y, size, scaleDown, min_size):
```



Task 5 : Realistic tree

For this one I want you to do some research and implement a recursive tree in python. I've included an image below of the one I just made. There are so many examples of this online that I have complete faith you'll find the solution. However, I do want you to make it your own somewhat. Add a creative element to your tree. For example, add leaves by drawing filled rectangles or circles. Have some fun with it.

Since you'll likely find the code online for this, make sure to cite where you found it and add extra comments explaining where you found the solution.



Remember you can get up to 26/30 points for finishing the lab, but for 30 points you have to go above and beyond. You don't have to use any of the below extensions. They are just examples. Explore on your own and come up with something fun.

Extension ideas:

1. Add some color to make your shapes more interesting
2. Create a recursive shape of your own design
3. Put your shapes all together in one place and make an artistic scene
4. Add some randomization to your shape creation

Report:

Reflection:
- What was the easiest and hardest part of this assignment?
- What did you learn?
- What grade would you give yourself?

Shapes:
- Include a screenshot of each shape.

Extension:
- What extension(s) did you add to the assignment?
- Include a screenshot of each extension added if possible.

Rubric:

Total Points: 30

Task 1 - 3 points
Task 2 - 3 points
Task 3 - 3 points
Task 4 - 5 points
Task 5 - 5 points
- Tree created successfully - 1 points
- Code commented well - 2 point
- Extra element added (different shape, coloring, leaves) - 2 point

Extensions - 4 points
Code Quality - 4 points
Report - 3 points