These first two problems should be relatively simple, but should put us into the mindset of what we are going to do next week, which is look at object-oriented programming. The main part of the homework should be having fun with drawing a tree (and in doing so, hopefully becoming comfortable enough with the idea of recursion).

1. Suppose that a Cartesian point is described by a list containing two numbers, the $x$ and $y$ coordinates of the point. Write a function called, $EuclideanDistance(point1, point2)$, that receives two cartesian points, and calculates and returns the Euclidean distance between the two points. Therefore, each of the parameters to the function should be lists containing two numbers. For example, this is what a call to your function should look like:

$$
>>> p1 = [1,1]
>>> p2 = [2,4]
>>> EuclideanDistance(p1, p2)
>>> 3.16227.
$$

Optional: Make the same Euclidean function work for points of any dimension. Such that it also works for, for example, three dimensional points: $p1 = [1,1,1]$ and $p2 = [1,4,3]$.

2. Now suppose that a rectangle is represented by a list containing two Cartesian points. Write a function, $area(rect)$, that receives a rectangle and returns its area. For example, this is what a call to your function should look like:

$$
>>> p1 = [1,1]
>>> p2 = [2,4]
>>> rect = [p1, p2]
>>> area(rect)
>>> 3.0
$$

Tip: Remember that to access the elements of a list inside a list, you can use multiple indexing. For example, you can access the different elements in the variables $x = [[1,2],[4,5]]$ by saying $x[i][j]$, where $i$ refers to the element in the outer list and $j$ refers to the element in the inner list. That is, $x[0]$ gives you the first element of the list in $x$, which is $[1,2]$. Then, $x[0][0]$ gives you the first element of the first element, which is 1.

3. **Recursion.** Go crazy with the $drawTree()$ program that we started in class on Monday and submit the coolest trees your program can draw (along with your code). Where necessary, explain in your code the things that are nontrivial.