Week 8 Assignment

Write a program that implements a tree class and a node or an edge class.

- In the tree class have a function that reads in a file with a newick string for a tree and then identifies all the nodes (internal and external) and stores them in instances of the node or edge class
  - Internal nodes can be identified by counting open and closed parentheses from the start of the string, or substring
  - External node labels begin with letter of number characters and they always begin immediately after an open parentheses or a comma
- Have your tree class implement an overloading of str(), that is a __str__ method for the tree class that prints out basic information for each node as a table.
- The information about a node should include the substring from the newick string, corresponding to that node
- Include branch length information if you can but not required

- Run your program on the newick string generated using phyml on your data for last week's assignment (can also use drosophilatree.nwk).
  - Print to a file the information about the tree (that uses the overloading of str())

Turn in your program, the newick file you used, and the output file you generated.

Most (partial) credit for getting all nodes or edges represented as strings in a tree class.

Full credit for a tree with properly connected up and down edges, with labels that facilitate the connections between up and down edges.

Bonus points

1. Have your tree class implement a subtree prune and regraft (SBR) method that makes a random change in the tree by randomly selecting an edge (and all its descendants) and moves them to a different part of the tree).
    a. Have a loop that makes a series of 10 random SBR operations, one after the other, and writes the new newick string to a file after each step to the output file (turn in this file with your other files)
2. Use branch lengths, read in branch lengths and do bonus part 1 using branch lengths (branch lengths will be conserved throughout a branch/regraft operation).