

Homework no. 1
Due Thursday, April 8

Write a program to read in this tree (represented in the Newick representation):

```
(Mouse:1.00909,(((BarbMacaq:0.20451,((Jpn_Macaq:0.05090,
Rhesus_Mac:0.05090):0.07494,Crab-E.Mac:0.12583):0.07868):0.31871,
((Orang:0.26371,((Chimp:0.13116,Human:0.13116):0.05105,
Gorilla:0.18221):0.08150):0.10382,
Gibbon:0.36754):0.15569):0.09691,Squir_Monk:0.62014):0.24988,
((Bovine:0.57749,Lemur:0.57749):0.13539, Tarsier:0.71288):0.15714):0.13907);
```

This tree will be found downloadable from a link near this homework:

<http://evolution.gs.washington.edu/gs541/2010/homework1.tre>

It can be saved from your browser into a file using the "Save As" menu item in the browser's File menu.

Set up the tree inside your computer (that is, construct some representation inside your system, perhaps with pointers and objects. Then the user is asked to specify which species (choosing them by giving the number of the species, when they are in left-to-right order) to use to reroot the tree.

Reroot the tree on the middle of the branch to that species. That means that if the branch length was 0.25, the branch is divided into two branches, each of length 0.125. Correspondingly, the branches connecting to the old root become one branch, and those lengths are summed to get the length of that branch. Thus, of the old root connected to a node which connected to two other branches, whose lengths were 0.1 and 0.23, when the node is removed and those two branches are merged into one, it would have length 0.33. Print out the tree resulting from the rerooting in Newick format, with branch lengths.

I know this assignment is going to be difficult and tiresome, but the software you write will then be used in two later assignments. Get started early (not the night before), decide what internal representation you will use in your program, and see if you can parse the string of characters that comes as the Newick format tree. For each open parenthesis, start setting up an internal fork in the tree. For the material between the parentheses, start a branch descended from that fork, and when a comma is encountered, complete that branch and start a new one. Lots of tiresome bookkeeping, error-prone, no matter how you do it.

E-mail me (joe (at) gs.washington.edu) with the results, including the input tree and the output tree for a couple of runs of the program, and also attach your source code (so I can help find any problems).