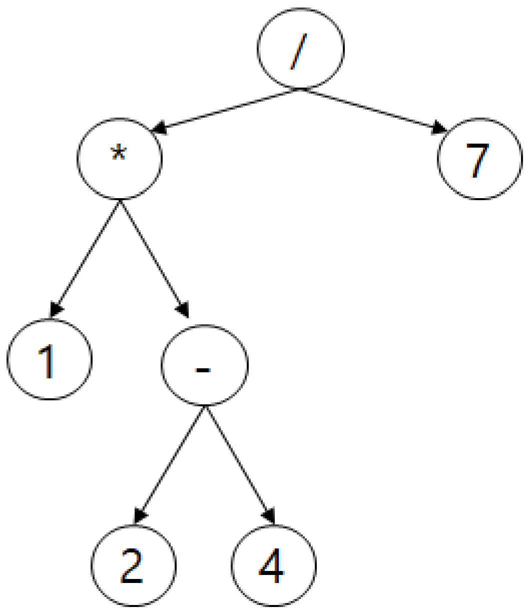
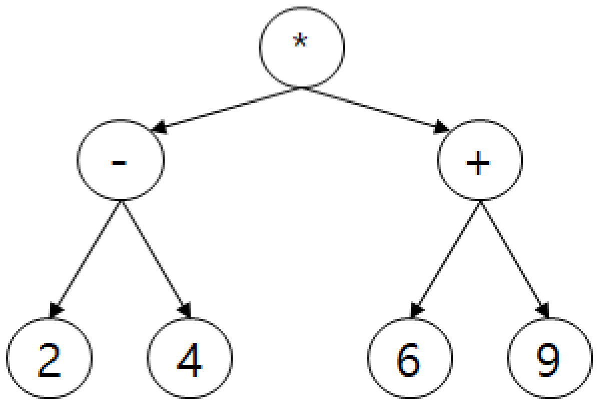
Homework 6 – Due Nov. 5th 23:59 KST

Instructions: Complete the implementation and turn it in before the due date. Any deviations from the instructed deliverable format will result in a deduction of grade. DO NOT COPY OTHER'S WORKS!

In this assignment, you will again work with mathematical expressions similar to the ones you dealt with in homework 4. The task is to convert the given infix expression into a ***parse tree***. A parse tree is a binary tree whose internal nodes are the operators and whose leaves correspond to operands. The expressions in this assignment will consist of the four basic operators (+-*/) and positive integers as operands. In addition to these symbols, you will also have to deal with parentheses. Two sample parse trees and their corresponding expressions are shown below:



1*(2-4)/7=-2/7



(2-4)*(6+9)=-30

You are to implement the following five main methods:

- buildTree(String): Build the parse tree that represents the given expression string. The input string is an infix notation consisting of the four operators, parentheses, and numerical operands. Provide a recursive solution, possibly by using helper methods.
- eval(): Perform the evaluation of the expression. That is, do the computation described by the expression.
- iterativeEval(): Same as above, except in an iterative manner.
- toString(): Convert the tree representation back to the original infix string.
- toPostfixString(): Same as above, except return a postfix notation.

In addition to these methods, you have to complete the Node class, as well as the constructor of the main class. See the comments in HW6.java.

Rubric: Grading will be based on, but not limited to, the following criteria.

- Documentation (20 points): For each of the five main methods, you should provide extensive descriptions in the header comments. In particular, the descriptions should include the outline of your algorithm in a paragraph. Be sure to mention the base and recursive cases.
- Correctness (80 points): Your implementation should behave as specified above in an error-free manner.
- Miscellaneous: Do not change the method and class names. Iterative methods should not use recursions. Recursive methods MUST use recursion as their main component.

Deliverable: A single HW6.java file not part of any package structures.