Problem 1

Show the preorder, postorder, and breadth-first order of the following tree.

![Tree Diagram]

**Answer:**

Problem 2

Discuss how you could modify the code in BreadthFirstSearch.java so that the nodes print out with an indentation that is proportional to their depth. For example, the animal taxonomy tree would print out in the form:

```
Mammal
  Rodent
  Carnivore
  Primate
    Squirrel
    Rat
    Cat
    Dog
    Skunk
    Gorilla
    Human
```

You do not have to write out all the code involved; just give a description of how it would work.

**Answer:** Instead of pushing the nodes onto the queue, push a record that shows both the node and the depth of the node. When you pop a node N off the queue, print it at indentation 3*N.depth,
and push records for each of its children noting their depths as N.depth+1.

**Problem 3**

Write a recursive function `sameShape(T,U)` which takes two binary trees and returns `true` if they are the same shape and `false` otherwise. The labels in the nodes do not matter. For instance, in the diagram below, A is the same shape as B but not as C or D.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{binary_trees.png}
\caption{Binary Trees}
\end{figure}

\textbf{Answer:}

```java
boolean sameShape(T,U) {
    if (T==null && U==null) return true;
    if (T==null || U==null) // one but not both
        return false;
    return sameShape(T.left) && sameShape(T.right);
}
```

**Honors Problem**

In a tree, the tree distance between nodes X and Y is the number of steps it takes to get from X to Y, going up and down the tree. The least common ancestor of nodes X and Y is the lowest node in the tree that is an ancestor of both X and Y. Note that if Z is the least common ancestor of X and Y, then the tree distance from X to Y is the number of steps from X to Z plus the number of steps from Y to Z.

For example, in the figure for problem 1, the tree distance from F to K is 3 (F → G → J → K). The least common ancestor of F and K is G. The least common ancestor of C and B is B itself.

Describe a method that takes as input two tree nodes X and Y and returns their least common ancestor, and runs in time proportional to the tree distance from X to Y. You may add additional fields to the nodes, and you may assumed that these are initialized to null. Hint: Climb upward from X and Y in alternating steps.

\textbf{Answer:} Mark X and Y. Climb upward from X and Y in alternating steps, marking each node as you pass it. When you reach an node that is already marked, that is the least common ancestor. If you want to be a responsible citizen, you should go back and remove the marks before quitting; otherwise, this won’t work next time you try it.
if (X==Y) return X;
U=X;
V=Y;
U.mark = true;
V.mark = true;
while (true) {
    U = U.parent;
    if (U.mark) {
        Answer=U;
        exitloop;
    }
    U.mark = true;
    V=V.parent;
    if (V.mark) {
        Answer=V;
        exitloop;
    }
    V.mark = true;
} // end while loop
while (X.mark == true) { // Cleanup
    X.mark = false;
    X = X.parent;
}
while (Y.mark == true) { // Cleanup
    Y.mark = false;
    Y = Y.parent;
}
return Answer;
}