Data structure tree Assignment 2

Question 1:

Given part of the binary tree ADT as follows:

- Each tree node itself is a data structure called TNode which supports the following method:
  
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object element()</td>
<td>Returns the object stored in the node</td>
</tr>
</tbody>
</table>

Part of the primitive operations a tree ADT provides are listed as follows:

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isLeaf(TNode n)</td>
<td>Returns true if node n is a leaf of the tree, otherwise, returns false.</td>
</tr>
<tr>
<td>boolean isRoot(TNode n)</td>
<td>Returns true if node n is the root of the tree, otherwise, returns false.</td>
</tr>
<tr>
<td>TNode leftChild(TNode n)</td>
<td>Returns the left child of node n.</td>
</tr>
<tr>
<td>TNode rightChild(TNode n)</td>
<td>Returns the right child of node n.</td>
</tr>
</tbody>
</table>

Using the above binary tree ADT, write an algorithm in pseudocode that returns the largest value in a binary tree in which the value stored at each node stores is an integer.

Question 2:

Draw an arithmetic expression tree that has four external nodes, storing the numbers 1, 5, 6, and 7 (with each number stored in a distinct external node, but not necessarily in this order), and has three internal nodes, each storing an operator from the set {+, -, x, /}, so that the value of the root is 21. The operators may return and act on fractions, and an operator may be used more than once.
Question 3:

Draw a (single) binary tree $T$ such that

- Each internal node of $T$ stores a single character
- A preorder traversal of $T$ yields EXAMFUN
- An inorder traversal of $T$ yields MAFXUEN.

Question 4:

Let $T$ be the binary tree of Figure 7.11.

a. Give the output of `toStringPostorder(T, T.root())` (Code Fragment 7.12).

b. Give the output of `printExpression(T, T.root())` (Code Fragment 7.27).
Code Fragment 7.12: Method `toStringPostorder(T, v)` that performs a postorder printing of the elements in the subtree of node `v` of `T`. The method implicitly calls `toString` on elements, when they are involved in a string concatenation operation.

```java
public static <E> String toStringPostorder(Tree<E> T, Position<E> v) {
    String s = "";
    for (Position<E> w : T.children(v))
        s += toStringPostorder(T, w) + " ";
    s += v.element(); // main visit action
    return s;
}
```

Code Fragment 7.27: An algorithm for printing the arithmetic expression associated with the subtree of an arithmetic expression tree `T` rooted at `v`.

**Algorithm** `printExpression(T, v)`:  
if `T.isInternal(v)` then  
    print "("
if `T.hasLeft(v)` then  
    printExpression(`T`, `T.left(v)`)
if `T.isInternal(v)` then  
    print the operator stored at `v`
else  
    print the value stored at `v`
if `T.hasRight(v)` then  
    printExpression(`T`, `T.right(v)`)
if `T.isInternal(v)` then  
    print ")"

---

**Question 5:**

What is the number of internal node in a proper binary tree with a total of 31 nodes?

**Question 6:**

Describe in pseudo code a non recursion method for performing an in order traversal of a binary tree in linear time (Hint: Use Stack).