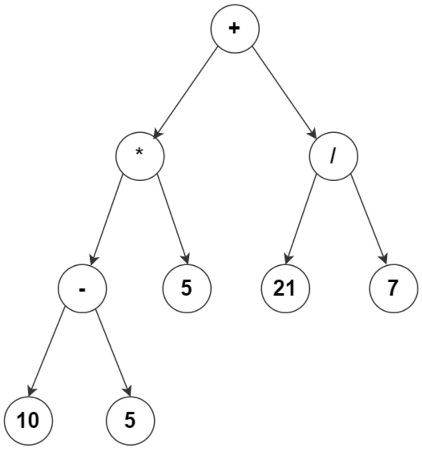
Algorithms Laboratory (CS29203)
Assignment 6: Binary Trees
Department of CSE, IIT Kharagpur

3rd October 2024

Question-1

(50 points)
Consider a full binary tree (i.e. each node has zero or two children) which represents an algebraic expression. The internal nodes of the tree stores operators, and the leaf nodes store constant values. The following five operators are allowed in the tree: addition (+), subtraction (−), multiplication (∗), division (÷), and exponentiation (ˆ). For example, consider the following full binary tree:



The evaluation of the tree will be $((10 - 5) * 5) + (21/7) = 28$. We can evaluate such a tree by applying the operator at the root to values obtained by recursively evaluating left and right subtrees. This can be done by traversing the tree using *postorder* traversal.

Given a tree in the above form, your task is to evaluate the expression and find the value. You may build the tree without taking user input (i.e. directly in the code).

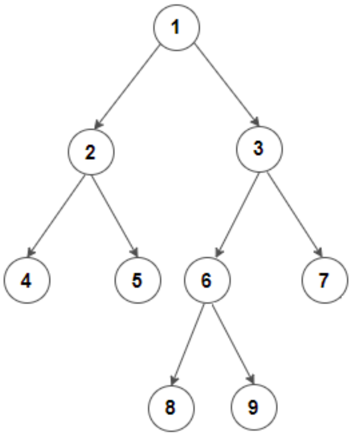
Example:

```
Building the tree:
                Node *root = newNode("+");
                root->left = newNode("*");
                root->right = newNode("/");
                root->left->left = newNode("-");
                root->left->right = newNode("5");
                root->right->left = newNode("21");
                root->right->right = newNode("7");
                root->left->left->left = newNode("10");
                root->left->left->right = newNode("5");


(Output) The value of the expression is: 28
```

Question-2

(50 points)
Consider a full binary tree (i.e. every node has 0 or 2 children) having n nodes. You are given a preorder traversal sequence of the tree. You are also supplied with an information of whether a given node is a leaf node or not. This is represented as a boolean array of size n, which determines whether the node at the corresponding index in the preorder sequence is a leaf node or not (1 means it is a leaf node, 0 means it is not a leaf node). For example, consider the following preorder traversal array `pre = [1, 2, 4, 5, 3, 6, 8, 9, 7]` and the boolean array `L = [0, 0, 1, 1, 0, 0, 1, 1, 1]`. It means that the nodes having values $4, 5, 8, 9, 7$ are leaf nodes. The full binary tree representing the preorder sequence and the array `L [ ]` is the following:



Given the arrays `pre[ ]` and `L[ ]`, your task is to construct the full binary tree. *Hint: The idea is to first construct the full binary tree's root node using the first key in the preorder sequence, and then using the given boolean array, check if the root node is an internal node or a leaf node. If the root node is an internal node, recursively construct its left and right subtrees. To construct the complete full binary tree, recursively repeat the above steps with subsequent keys in the preorder sequence.*

After reconstructing the tree, print the inorder, preorder, and postorder traversal of the tree.

Example:

```
(Input) pre = [1, 2, 4, 5, 3, 6, 8, 9, 7]
        L = [0, 0, 1, 1, 0, 0, 1, 1, 1]

(we will verify the output tree by printing their inorder, preorder, and postorder traversals)

(Output) The reconstructed tree's traversals are as follows:
         Inorder: 4, 2, 5, 1, 8, 6, 9, 3, 7
         Preorder: 1, 2, 4, 5, 3, 6, 8, 9, 7
         Postorder: 4, 5, 2, 8, 9, 6, 7, 3, 1
```