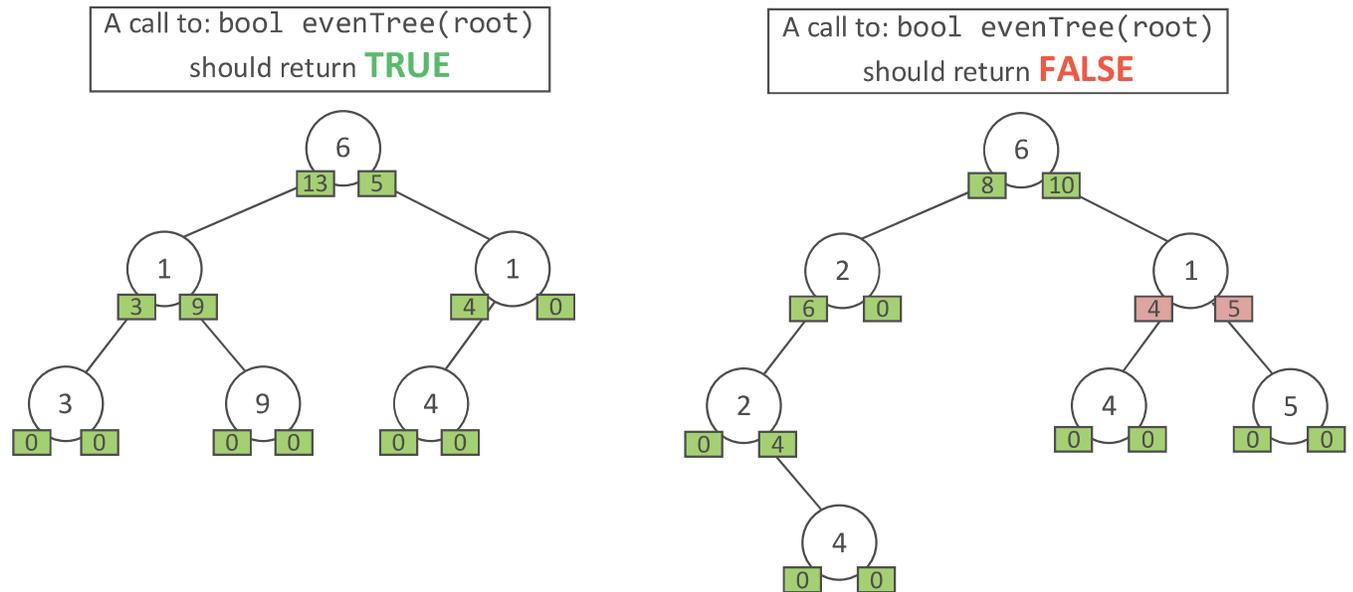


4. (10 pts.) **Trees and Recursion.** Given a binary tree of **non-negative integers**, write a **recursive** function that returns TRUE if the sum of all nodes in the left and right subtree of **EACH** node is an **even number**. If the sum of nodes in the left and right subtree of even 1 node is odd, return false. Non-existent nodes have an implicit value of 0. Also, an empty tree should return `true`. No loops are allowed so you must use recursion.



Complete your code in the linked [eventree.cpp](#) and submit that file. A portion is reproduced below.

```
// You may NOT change this struct definition
struct Node {
    int val;
    Node *left, *right;
};

// You may NOT change this function prototype
bool evenTree(Node* root);

// You may add any prototypes of helper functions here

// Now implement the evenTree function and any necessary helpers
// ----- YOUR CODE HERE -----
```

```
//----- Solution -----
```

```
bool evenTree(Node* root)
```

```
{  
    bool res = true;  
    evenTreeHelper(root, res);  
    return res;  
}
```

```
int evenTreeHelper(Node* root, bool& result)
```

```
{  
    if(root == nullptr) return 0;  
    int sum = evenTreeHelper(root->left, result) +  
              evenTreeHelper(root->right, result);  
    if( sum % 2 == 1)  
    {  
        result = false;  
    }  
    return sum + root->val;  
}
```