

Q8.

For this problem, you need to construct a **binary search tree** using a **linked list**. You can assume that the value of each element will not repeat.

You must use the template to do this lab.

You should implement the following functions by yourself within the class:

Operations	Description
A. void insert(TreeNode* node)	Insert the node to the binary search tree.
B. void preorder(TreeNode* root)	Use the preorder method to traverse the binary search tree.
C. void inorder(TreeNode* root)	Use the inorder method to traverse the binary search tree.
D. void postorder(TreeNode* root)	Use the postorder method to traverse the binary search tree

Hint:

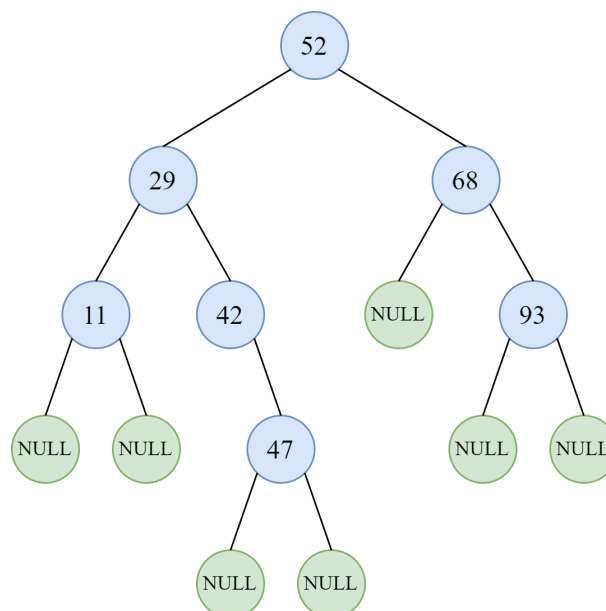
The following is an example of the binary search tree.

The order of inserting numbers is: 52, 29, 68, 11, 42, 47, 93

Preorder: 52, 29, 11, 42, 47, 68, 93

Inorder: 11, 29, 42, 47, 52, 68, 93

Postorder: 11, 47, 42, 29, 93, 68, 52



You can refer to these traversals pseudo code below to print an sorted order in a binary search tree.

```
preorder(node) {
    if node is not null {
        visit current node
        preorder(the left child of current node)
        preorder(the right child of current node)
    }
}
```

```
inorder(node) {
    if node is not null {
        inorder(the left child of current node)
        visit current node
        inorder(the right child of current node)
    }
}
```

```
postorder(node) {
    if node is not null {
        postorder(the left child of current node)
        postorder(the right child of current node)
        visit current node
    }
}
```

Input Format

The first line indicates the number of cases. Following that are lines of numbers, with each set of numbers forming a binary search tree. You need to read the numbers in each line to construct its corresponding binary search trees in the order of the numbers provided.

Output Format

You should display the binary search trees using three orders: first is preorder, second is inorder, and third is postorder.

Sample Input

```
2
2 1 3 8 9 4
52 29 68 11 42 47 93
```

Sample Output

```
preorder: 2 1 3 8 4 9
inorder: 1 2 3 4 8 9
postorder: 1 4 9 8 3 2
```

```
preorder: 52 29 11 42 47 68 93
inorder: 11 29 42 47 52 68 93
postorder: 11 47 42 29 93 68 52
```