Find the Least Common Anscetor of two nodes in a given BST

```
#include <stdio.h>
#include <stdlib.h>

struct node
{
	int data;
	struct node *left;
	struct node *right;
};

struct node *newNode(int data)
{
	struct node *temp = (struct node *)malloc(sizeof(struct node));
	temp->data = data;
	temp->left = temp->right = NULL;
	return temp;
}


struct node *RecursiveLCA(struct node *root, int p, int q)
{
	if(!root) return NULL;
	if(root->data > p && root->data > q)
		return RecursiveLCA(root->left, p, q);
	if(root->data < p && root->data < q)
		return RecursiveLCA(root->right, p, q);
	return root;

}

/* Iterative solution to find least common anscetor of given two nodes */
struct node *IterativeLCA(struct node *root, int p, int q)
{
	while(root)
	{
		if(root->data > p && root->data > q)
			root = root->left;
		else if (root->data < p && root->data < q)
			root = root->right;
		else break;
	}
	return root;
}


int main()
{
	struct node *root, *lca;
	root = newNode(25);
	root->left = newNode(10);
```

```
    root->right = newNode(30);
    root->left->left = newNode(5);
    root->left->right = newNode(15);
    root->left->right->left = newNode(12);
    lca = RecursiveLCA(root, 5, 12);
    printf("%d\n", lca ? lca->data: -1); // Time is O(h) & space is O(logn)
    lca = IterativeLCA(root, 5, 12);
    printf("%d\n", lca ? lca->data: -1); // Time is O(h) & space is O(1)
    return 0;
}
```

Recursion:
Time complexity: O(logn)
Space Complexity: O(logn)

Iteration:
Time complexity: O(logn)
Space Complexity: O(1)