# Lab-8

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

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

struct Node* createNode(int data) {
        struct Node* newNode = (struct Node*)
        malloc(sizeof(~~newNode~~ struct Node));
        newNode->data = data;
        newNode->left = newNode->right = NULL;
        return newNode;
};

struct Node* insert(struct Node* root, int dat
{
        if(root == NULL) {
                root = createNode(data);
        }
        else if (data <= root->data) {
                root->left = insert(root->left, data);
        }
        else
                root->right = insert(root->right, data);
```

15/2/24

```
}
void inOrder (struct Node * root){
        if(root != NULL){
            inOrder(root -> left);
            printf("%d \t", root->data);
            inOrder(root -> right);
        }
}

void preOrder (struct Node * root){
        if(root != NULL){
            printf("%d \t", root->data);
            preOrder(root->left);
            preOrder (root->right);
        }
}

void postOrder (struct Node * root){
        if(root!=NULL){
            postOrder (root->left);
            postOrder (root->right);
            printf ("%d \t", root->data);
        }
}
```

```
void display (struct Node *root){
        printf ("BST inorder traversal \n");
        inorder (root);
        printf ("BST preOrder traversal \n");
        preOrder (root);
        printf ("BST postorder traversal ");
        postOrder(root);
}


int main(){
        struct Node* root=NULL;

        root =insert(root,10);
        root=insert(root,5);
        root= insert(root,15);
        root=insert (root,7);
        root=insert(root,12);
```

~~root~~

```
        display (root);
        return 0;
}
```

Output                                        Values inserted->
                                                        10,5,15,7,12,3

Inorder traversal

~~5  10  12  20  32~~ 55

3  5  7  10  12  15

~~Preorder traversal~~

~~5  20  10  12  55  32~~

PreOrder traversal

10 5 3 7 15 12

PostOrder traversal

3 7 5 12 15 10

LeetCode

```
struct ListNode* rotateRight(struct ListNode* head,
  int k)
  if(head == NULL)
      return head;
  int length=1;
  struct ListNode *temp=head ,*temp2;

  while (temp->next!=NULL){
        temp= temp->next;
        length++;
  }
  int rotate = length -(k%length);
  if(rotate ==0)
        return head;
  rotate = rotate-1;
  temp->next= head;
  temp=head;
  while (rotate!=0){
        rotate--;
        temp= temp->next;
  }
  temp2 = temp->next;
  head = temp2;
  temp->next=NULL;
  return head;
}
```

13/12/24