WEEK-8

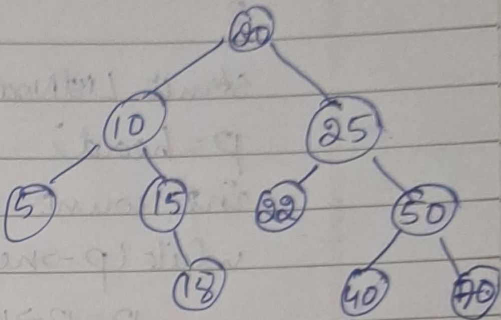
6/12/24.

```
//Binary tree -
#include <stdio.h>
#include <stdlib.h>
struct node{
    int data;
    struct node *left;
    struct node *right;
}*root=NULL;
struct node *create(struct node *t, int ele){
    if (t==NULL){
        struct node *temp=(struct node*) malloc(sizeof(struct
                                                        node));
        temp->data=ele;
        temp->left =temp->right =NULL;
        return temp;
    }.
    else{
        if (ele <t->data){
            t->left=create(t->left,ele);
        }
        else{
            t->right=create(t->right,ele);
        }
    }.
    return t;
}

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

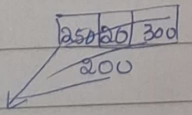
```
void in(struct node *root){
    struct node *r;
    r=root;
    if(r!=NULL){
      in(r->left);
      printf("%d\t",r->data);
      in(r->right);
    }
  }

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

void main(){
    int n,ele;
   printf("enter the no of elements:");
   scanf("%d",&n);
for(int i=0;i<n;i++){
     printf("enter the element %d:",i+1);
     scanf("%d",&ele);
     root=create(root,ele);
  }
printf("display the elements in preorder traversal");
pre(root);
printf("\n display the elements in inorder traversal\n");
in(root);
printf("\n display the elements in postorder traversal:");
post(root);
}
```

O/P

```
enter the no of elements:10
enter the element 1:20
enter the element 2: 10
enter the element3: 5
enter the element4: 15
enter the element 5: 25
enter the element 6 : 22
enter the element 7: 50
enter the element 8: 18
enter the element 9:40
enter the element 10:70
display the elements in preorder traversal:20 10 5 15
18 25 22 50 40 70
display the elements in inorder traversal: 5 10 15 18
20 22 25 40 50 70
display the elements in postorder traversal: 5 18 15 10
22 40 70 50 25 20
```



~~post order~~



15/2/2024

15/2/24

Leetcode Rotate List

Question Given the head of Linked list, rotate the list to the right by k pla

```
struct ListNode* rotateRight (struct ListNode* head, int k){
    if (head==NULL || head->next==NULL || k==0){
        return head;
    }
    struct ListNode *p, *q;
    p=head;
    int count=1;
    while (p->next!=NULL){
        p=p->next;
        count++;
    }
    k=k%count;
    if (k==0){
        return head;
    }
    p->next=head;
    p=head;
    for (int i=0; i<count-k-1; i++){
        p=p->next;
    }
    q=p->next;
    p->next=NULL;
    return q;
}.
```

Case 1
O/P head = [1,2,3,4,5]
k = 2
O/P = [4,5,1,2,3]

N 15/2/24

Case 2 head = [0,1,2]
k = 4
O/P = [2,0,1]