Week - 6

Date : 19/03/24

8) Write a program

a) To construct Binary Search tree

b) Traverse the tree using inorder, postorder, preorder

c) Display the elements in the tree

```
Struct node {
    int data
    Struct node *left;
    Struct node *right;
}

Struct node *createnode (int data) {
    Struct node *new_node = (Struct node*) malloc (sizeof
                (struct node));
    new_node -> data = data;
    new_node -> left = new_node -> right = NULL;
    return new_node;
}

Struct node * insert (struct node *root, int data) {
    if (root == NULL)
        return createnode(data);
    if (data < root -> data)
        root -> left = insert (root -> left, data);
    else if (data > root -> data) {
        root -> right = insert (root -> right, data);
    return root;
}
```

```
void inorder (struct node *root){
    if (root != Null){
        inorder (root -> left);
        printf ("%d ", root -> data);
        inorder (root -> right);
    }
}

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

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

void display (struct node *root){
    if (root != NULL){
        printf(" Inorder ");
        inorder(root);
        printf("preorder ");
        preorder(root);
```

P.T.O

```
        printf("postorder");
        printf(preorder(root);
    }
}
int main() {
    struct node *root = NULL;
    root = insert(root, 500);
    root = insert(root, 300);
    root = insert(root, 900);
    root = insert(root, 550);
    root = insert(root, 50);
    display(root);
}
```

o/p:

Inorder : 50 300 500 550 900

preorder : 500 300 50 900 550

postorder : 50 150 300 550 900 500

1) Delete the middle node of a linkedlist → Leetcode

```
struct ListNode *deletenode(struct ListNode *Head) {
    struct ListNode *temp, *ptr, *ptr1;
    temp = head;
    ptr1 = head;
    if (head == NULL || head->next == NULL)
        return NULL;
```

```
while (temp != NULL && temp->next != NULL)
{
    temp = temp->next->next;
    ptr = ptr1;
    ptr1 = ptr1->next;
}
ptr->next = ptr1->next
return head;
}
```

Odd Even Linked List - LeetCode

```
struct ListNode *oddEvenList (struct ListNode *head){
    if (head == NULL || head->next == NULL)
        return head;
    struct ListNode *even = head;
    struct ListNode *evenhead = head->next;
    struct ListNode *odd = head->next;

    while (even->next != NULL && odd->next != NULL)
    {
        even->next = even->next->next;
        even = even->next;
        odd->next = odd->next->next;
        odd = odd->next;
    }
    even->next = evenhead;
    return head
}
```