WAP

1.TO CONSTRUCT BINARY SEARCH TREE

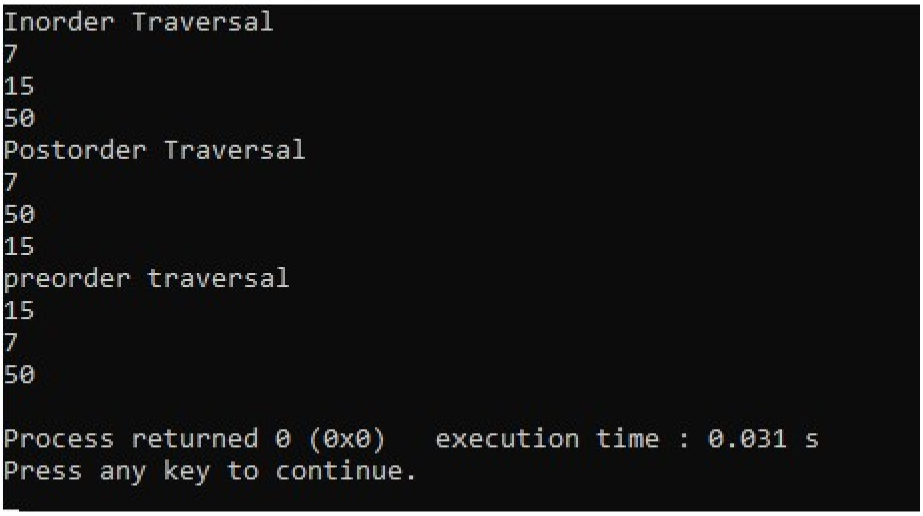
2.TO TRAVERSE THR TREE USING INORDER,POSTORDER,PREORDER

3.TO DISPLAY THE ELEMENTS IN TREE

```
#include <stdio.h>

#include <stdlib.h>

struct node{

    int data;

    struct node* right_child;

    struct node* left_child;

};

 struct node* newNode(int x){

    struct node* temp=malloc(sizeof(struct node));

    temp->data=x;

    temp->right_child=NULL;

    temp->left_child=NULL;

    return temp;

}

struct node* insert (struct node* root,int x){

    if(root==NULL){

        return newNode(x);

    }

    else if(x>root->data)

        root->right_child=insert(root->right_child,x);

    else{

        root->left_child=insert(root->left_child,x);
    }
    return root;
}
void inOrder(struct node* root){
    if(root!=NULL){
        inOrder(root->left_child);
        printf("%d\n",root->data);
        inOrder(root->right_child);
    }
}
void postOrder(struct node* root){
    if(root!=NULL){
        postOrder(root->left_child);
        postOrder(root->right_child);
        printf("%d\n",root->data);
    }
}
void preOrder(struct node* root){
    if(root!=NULL){
        printf("%d\n",root->data);
        preOrder(root->left_child);
        preOrder(root->right_child);
    }
}
void main(){
    struct node* root=NULL;
```

```
   root=insert(root,15);

   root=insert(root,7);

   root=insert(root,50);

   printf("Inorder Traversal\n");

   inOrder(root);

   printf("Postorder Traversal\n");

   postOrder(root);

   printf("preorder traversal\n");

   preOrder(root);

}
```

OUTPUT:



```
Inorder Traversal
7
15
50
Postorder Traversal
7
50
15
preorder traversal
15
7
50

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```