1BM19CS053
Diganshu Thanwar

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

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

void insert(node **root, int d)
{
    if (*root==NULL)
    {
        (*root)=(node*)malloc(sizeof(node));
        (*root)->left = NULL;
        (*root)->data = d;
        (*root)->right = NULL;
    }
    else
    {
        if (d < (*root->data))
            insert (&((*root->left), d);
        else
            insert(&((*root->right), d);
    }
}
```

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

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

bool search(node *root, int key)
{
    if(root==NULL)
        return false;
    if(root->data=key)
        return true;
    else
    {
        if(key<root->data)
        return search(root->left,key);
        else
        return search(root->right,key);
    }
}
```

```
int main() {

    node *root = NULL;
    int choice;
    int d;
    printf("1. Insert in BST \n 2. PreOrder \n 3. InOrder \n4. PostOrder \n5. Search \n6.Exit \n");
    printf("Your choice : ");
    scanf("%d", &choice);

    while (choice != 6)
    {
        switch (choice)
        {

        case 1: printf("Enter element to be inserted : ");
                scanf("%d", &d);
                insert(&root, d);
                printf("%d inserted in the tree \n", d);
                break;

        case 2: printf("PreOrder traversal is : \n");
                preorder(root);
                printf("\n");
                break;

        case 3: printf("Inorder traversal is \n");
                inorder(root);
                printf("\n");
                break;

        case 4: printf("PostOrder traversal is \n");
                postorder(root);
                printf("\n");
                break;

        case 5: printf("Enter element to be searched ");
                scanf("%d", &d);
                if(search(root, d)) {
                    printf("Element found! \n");
                }
                else
```

```
        {
            printf("Element not found\n");
        }
    }
    printf("Your next choice : ");
    scanf("%d", &choice);
}
return 0;
}
```