Binary search

```c
#include<stdio.h>

struct node
{
int data;
struct node *left,*right;
}*root=NULL,*newnode;
 struct node* insert(struct node*,int);
void inorder(struct node *);
void preorder(struct node *);
void postorder(st # ruct node *);
 struct node* insert(struct node  *root,int ele)
{
if(root==NULL) #
{
newnode=(struct node*)malloc(sizeof(struct node));
newnode->data=ele;
newnode->left= NULL;
newnode->right=NULL;
return (newnode);
}
else if(ele>root->data)
root->right=insert(root->right,ele);
else if(ele<root->data)
root->left=insert(root->left,ele);
return (root);
}
 void inorder(struct node *root)
{
if(root!=NULL)
```

```c
{
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);
}


}
 int main()
{
int ele,op,n;
```

```c
do
{
printf("\n***Main Menu*\n");
printf("1.Insert\n2.Display-Inorder\n3.Display-Postorder\n4.Display-Preorder\n5.Exit");
printf("\nEnter your option");
scanf("%d",&op);
switch(op)
{
case 1:  printf("\nEnter element to insert");
scanf("%d",&ele);
root=insert(root,ele);
break;

case 2:  inorder(root);break;
case 3:postorder(root);break;
case 4:preorder(root); break;
case 5: exit(0);
default: printf(" Please enter option between 1 to 5");
}
}while(op>0 && op<=5);

}
```

Output:

```
3.Display-Postorder
4.Display-Preorder
5.Exit
Enter your option1

Enter element to insert3

***Main Menu*
1.Insert
2.Display-Inorder
3.Display-Postorder
4.Display-Preorder
5.Exit
Enter your option4
14       3
***Main Menu*
1.Insert
2.Display-Inorder
3.Display-Postorder
4.Display-Preorder
5.Exit
Enter your option3
3        14
***Main Menu*
1.Insert
2.Display-Inorder
3.Display-Postorder
4.Display-Preorder
5.Exit
Enter your option
```