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

struct BSTNODE
{
  struct BSTNODE *lchild;
  int data;
  struct BSTNODE *rchild;
};

typedef struct BSTNODE BNODE;

BNODE *root;
int flag;

void inorder(BNODE *root1)
{
   if(root1!=NULL)
   {
    inorder(root1->lchild);
    printf("%d\t",root1->data);
     inorder(root1->rchild);
   }
}



void preorder(BNODE *root1)
{
   if(root1!=NULL)
   {
      printf("%d\t",root1->data);
```

```
        preorder(root1->lchild);
        preorder(root1->rchild);
    }
}


void postorder(BNODE *root1)
{
    if(root1!=NULL)
    {
        postorder(root1->lchild);
        postorder(root1->rchild);
        printf("%d\t",root1->data);
    }
}

void search(BNODE *root1,int key)
{
    if(root1!=NULL)
    {
     search(root1->lchild,key);
     if(key==root1->data)
        flag=1;
     search(root1->rchild,key);
    }
}


void create(int N)
{
```

```
BNODE *temp,*cur,*prev;
int i,item;

for(i=1;i<=N;i++)
{
   temp = (BNODE*)malloc(sizeof(BNODE));
   printf("Enter the data\n");
   scanf("%d",&item);
   temp->rchild=NULL;
   temp->lchild=NULL;
   temp->data=item;

   if(root==NULL)
      root=temp;
   else
   {
      prev=NULL;
      cur=root;

      while(cur!=NULL)
      {
         prev=cur;

         if(cur->data<temp->data)
            cur=cur->rchild;
         else
            cur=cur->lchild;
      }
      if(prev->data<temp->data)
            prev->rchild=temp;
         else
```

```
                prev->lchild=temp;
        }
    }
}

void main()
{
    int choice,key,N;
    root=NULL;

    while(1)
    {

        printf("\n 1-> Create A BST\n 2-> Traversal\n 3->Search A Node\n 4->Exit\n");
        printf("Enter your Choice\n");
        scanf("%d",&choice);

        switch(choice)
        {
        case 1:printf("Enter the Number of Nodes\n");
              scanf("%d",&N);;
              create(N);;
            break;
        case 2:printf("\nInorder Traversal: \t");
              inorder(root);
              printf("\nPreorder Traversal: \t");
              preorder(root);
              printf("\nPostorder Traversal: \t");
              postorder(root);
            break;
        case 3:flag = 0;
```

```
            printf("Enter the key element to be searched\n");
             scanf("%d", &key);
             search(root,key);
             if(flag==1)
               printf("Key Found Successfully\n");
             else
               printf("key Not found\n");
            break;
        case 4:exit(0);
            break;
        default:printf("Invalid Choice\n");
        }
    }
}
```