

```

#include<stdio.h>
#include<stdlib.h>
#include "btree.h"
int main()
{
int ch,n,i,value,cnt;
struct bst *newnode,*root,*temp;
root=NULL;
while(1)
{
printf("\n---Binary Search Tree---\n");
printf("1.Insert\n");
printf("2.Search\n");
printf("3.Count Total Nodes\n");
printf("4.Count Leaf Nodes\n");
printf("5.Count Non Leaf Nodes\n");
printf("6.Inorder Traversal (Recursive)\n");
printf("7.Postorder Traversal (Recursive)\n");
printf("8.Preorder Traversal (Recursive)\n");
printf("9.Inorder Traversal (Non Recursive)\n");
printf("10.Exit\n");
printf("Enter your choice:");

scanf("%d",&ch);
switch(ch)
{
case 1: printf("\nHow many nodes to create:");
scanf("%d",&n);
for(i=0;i<n;i++)
{
newnode=create();
printf("\nEnter the node data:");
scanf("%d",&newnode->data);
if(root==NULL)
root=newnode;
else
insert(root,newnode);
}
break;
case 2: printf("\nEnter the node value to be searched:");
scanf("%d",&value);
temp=search(root,value);
if(temp==NULL)
printf("\nNode Not Found\n");
else

```

```
printf("\nNode Found\n");
break;
case 3: cnt=count(root);
printf("\nTotal Nodes=%d\n",cnt);
break;
case 4: cnt=countleaf(root);
printf("\nTotal Leaf Nodes=%d\n",cnt);

break;
case 5: cnt=countnleaf(root);
printf("\nTotal Non Leaf Nodes=%d\n",cnt);
break;
case 6: printf("\nInorder Traversal=");
inorder(root);
break;
case 7: printf("\nPostorder Traversal=");
postorder(root);
break;
case 8: printf("\nPreorder Traversal=");
preorder(root);
break;
case 9: printf("\nInorder Traversal=");
inorder_n(root);
break;
case 10: exit(0);
default: printf("\nInvalid Choice\n");
}
}
}
```