SOURCE CODE: BINARY SEARCH TREE:

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
struct node
{
        int data;
        struct node *left;
        struct node *right;
};
struct node *tree=NULL;
struct node *insertElement(struct node *,int);
void preorderTraversal(struct node *);
void inorderTraversal(struct node *);
void postorderTraversal(struct node *);
int main()
{
        int option,val;
        struct node *ptr;
        clrscr();
        do
        {
                printf("\n ***MAIN MENU***\n 1:Insert element\n 2:Preorder traversal\n 3:Inorder
traversal\n 4:Postorder traversal\n 5:Exit\n*****n Enter your option:");
                scanf("%d",&option);
                switch(option)
                {
                        case 1:
                                printf("\n Enter the value of the new node:");
```

```
                    scanf("%d",&val);
                    tree=insertElement(tree,val);
                    break;
                case 2:
                    printf("\n The elements of the tree are:\n");
                    preorderTraversal(tree);
                    break;
                case 3:
                    printf("\n The elements of the tree are :\n");
                    inorderTraversal(tree);
                    break;
                case 4:
                    printf("\n The elements of the tree are:\n");
                    postorderTraversal(tree);
                    break;
            }
        }while(option!=5);
        getch();
        return 0;
}
struct node *insertElement(struct node *tree,int val)
{
        struct node *ptr,*nodeptr,*parentptr;
        ptr=(struct node*)malloc(sizeof(struct node));
        ptr->data=val;
        ptr->left=NULL;
        ptr->right=NULL;
        if(tree==NULL)
        {
```

```
                tree=ptr;
                tree->left=NULL;
                tree->right=NULL;
        }
        else
        {
        parentptr=NULL;
        nodeptr=tree;
        while(nodeptr!=NULL)
        {
                parentptr=nodeptr;
                if(val<nodeptr->data)
                        nodeptr=nodeptr->left;
                else
                        nodeptr=nodeptr->right;
        }
        if(val<parentptr->data)
                parentptr->left=ptr;
        else
                parentptr->right=ptr;
        }
        return tree;
}
void preorderTraversal(struct node *tree)
{
        if(tree!=NULL)
        {
                printf("%d\t",tree->data);
                preorderTraversal(tree->left);
```

```
                preorderTraversal(tree->right);
        }
}
void inorderTraversal(struct node *tree)
{
        if(tree!=NULL)
        {

                inorderTraversal(tree->left);
                printf("%d\t",tree->data);
                inorderTraversal(tree->right);
        }
}
void postorderTraversal(struct node *tree)
{
        if(tree!=NULL)
        {

                postorderTraversal(tree->left);
                postorderTraversal(tree->right);
                printf("%d\t",tree->data);
        }
}
```

OUTPUT:

```
***MAIN MENU***
1:Insert element
2:Preorder traversal
3:Inorder traversal
4:Postorder traversal
5:Exit

*****
Enter your option:1

Enter the value of the new node:10

***MAIN MENU***
1:Insert element
2:Preorder traversal
3:Inorder traversal
4:Postorder traversal
5:Exit

*****
Enter your option:1

Enter the value of the new node:20

***MAIN MENU***
1:Insert element
2:Preorder traversal
3:Inorder traversal
4:Postorder traversal
5:Exit

*****
Enter your option:1

Enter the value of the new node:30

***MAIN MENU***
1:Insert element
2:Preorder traversal
3:Inorder traversal
4:Postorder traversal
5:Exit
```

```
*****
 Enter your option:2

 The elements of the tree are:
10      20      30
 ***MAIN MENU***
 1:Insert element
 2:Preorder traversal
 3:Inorder traversal
 4:Postorder traversal
 5:Exit

*****
 Enter your option:3

 The elements of the tree are :
10      20      30
 ***MAIN MENU***
 1:Insert element
 2:Preorder traversal
 3:Inorder traversal
 4:Postorder traversal
 5:Exit

*****
 Enter your option:4

 The elements of the tree are:
30      20      10
 ***MAIN MENU***
 1:Insert element
 2:Preorder traversal
 3:Inorder traversal
 4:Postorder traversal
 5:Exit

*****
 Enter your option:5
```