```
/***** Binary Search Tree ****/
#include <stdio.h>
#include <stdlib.h>

struct node
{
 int data;
 struct node *left, *right;
};

struct node *newnode(int data)
{
 struct node *temp = (struct node *)malloc(sizeof(struct node));
 temp->data = data;
 temp->left = temp->right = NULL;
 return temp;
}

struct node *insert(struct node *node, int data)
{
 if (node == NULL)
    return newnode(data);
 if (data < node->data)
   node->left = insert(node->left, data);
 else
   node->right = insert(node->right, data);
 return node;
}

struct node *minvalue(struct node *node)
{
 struct node *temp = node;
 while (temp && temp->left != NULL)
   temp = temp->left;
 return temp;
}

struct node *deleteNode(struct node *root, int data)
{
 struct node *temp = minvalue(root->right);
 if (root == NULL)
    return root;
 if (data < root->data)
   root->left = deleteNode(root->left, data);
 else if (data > root->data)
   root->right = deleteNode(root->right, data);
 else
 {
```

```
     if (root->left == NULL)
     {
            struct node *temp = root->right;
            free(root);
            return temp;
    }
    else if (root->right == NULL)
    {
            struct node *temp = root->left;
            free(root);
            return temp;
    }
   root->data = temp->data;
   root->right = deleteNode(root->right, temp->data);
  }
  return root;
}
int search(struct node *root,int data)
{
   while(root!=NULL && root->data!=data)
   {
           if(data<root->data)
              root=root->left;
           else
             root=root->right;
   }
   if (root!=NULL)
      return (1);
   else
     return (0);
}

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

void main()
{
  struct node *root = NULL;
  int opt,data;
  clrscr();
  while (1)
```

```
 {
 printf("\n1.Insert\n2.Delete\n3.Search\n4.Display\n5.Exit\n");
 printf("Enter your option : ");
 scanf("%d",&opt);
 switch(opt)
 {
  case 1:
    printf("Enter Data : ");
    scanf("%d",&data);
    root = insert(root, data);
    break;
  case 2:
    printf("Enter Data : ");
    scanf("%d",&data);
    if (search(root,data))
          root = deleteNode(root, data);
    else
         printf("%d is not found\n",data);
    break;
 case 3:
    printf("Enter data : ");
    scanf("%d",&data);
    if (search(root,data))
          printf("Found\n");
    else
          printf("Not Found\n");
    break;
 case 4:
   display(root);
   break;
 case 5:
   exit(0);
 default:
   printf("Invalid Option\n");
 }
}
}
```