```
//By Ritwik Chandra Pandey
//On 1st Sep 2021
//BST Operations - Search and post order traversal

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


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

typedef struct node *BSTNODE;

BSTNODE newNodeInBST(int item) {
   BSTNODE temp = (BSTNODE)malloc(sizeof(struct node));
   temp->data = item;
   temp->left = temp->right = NULL;
return temp;}

void postorderInBST(BSTNODE root) {
    if(root==NULL)
    return;

   postorderInBST(root->left);
   postorderInBST(root->right);
 printf("%d ", root->data);}

BSTNODE insertNodeInBST(BSTNODE node, int ele) {
   if (node == NULL) return newNodeInBST(ele);
   if (ele < node->data)
       node->left = insertNodeInBST(node->left,ele);
   else if (ele > node->data)
       node->right = insertNodeInBST(node->right,ele);
 return node;}

BSTNODE searchNodeInBST(BSTNODE root, int ele) {
   if (root==NULL || root->data == ele){
       return root;
```

```
  }
  if(ele > root->data ){
    return searchNodeInBST(root->right,ele);}
  else
 return searchNodeInBST(root->left,ele);}

void main() {
  int x, op;
  BSTNODE root = NULL;
  while(1)
  {
    printf("1.Insert 2.Search 3.Postorder Traversal 4.Exit\n");
    printf("Enter your option : ");
    scanf("%d", &op);
    switch(op) {
      case 1: printf("Enter an element to be inserted : ");
          scanf("%d", &x);
          root = insertNodeInBST(root,x);
          break;
      case 2:
          printf("Enter an element to be searched : ");
          scanf("%d",&x);
          if( searchNodeInBST(root,x) == NULL)
            printf("Element not found in the binary search tree.\n");
          else
            printf("Element found in the binary search tree.\n");
          break;
      case 3:
          if(root == NULL) {
            printf("Binary Search Tree is empty.\n");
          }
          else {
            printf("Elements of the BST (post-order traversal): ");
            postorderInBST(root);
            printf("\n");
          }
          break;
    case 4: exit(0);}
  }
}
```