Rajalakshmi Engineering College

Name: HARSHA SREE SK
Email: 240801113@rajalakshmi.edu.in
Roll no: 240801113
Phone: 9944045712
Branch: REC
Department: I ECE FB
Batch: 2028
Degree: B.E - ECE

Scan to verify results

NeoColab_REC_CS23231_DATA STRUCTURES

REC_DS using C_Week 5_COD_Question 3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are required to implement basic operations on a Binary Search Tree (BST), like insertion and searching.

Insertion: Given a list of integers, construct a Binary Search Tree by repeatedly inserting each integer into the tree according to the rules of a BST.

Searching: Given an integer, search for its presence in the constructed Binary Search Tree. Print whether the integer is found or not.

Write a program to calculate this efficiently.

Input Format

The first line of input consists of an integer n, representing the number of nodes

in the binary search tree.

The second line consists of the values of the nodes, separated by space as integers.

The third line consists of an integer representing, the value that is to be searched.

Output Format

The output prints, "Value <value> is found in the tree." if the given value is present, otherwise it prints: "Value <value> is not found in the tree."

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 7
8 3 10 1 6 14 23
6
Output: Value 6 is found in the tree.

Answer

```
// You are using GCC
#include<stdio.h>
#include<stdlib.h>
struct TreeNode{
    int data;
    struct TreeNode* left,*right;
};
typedef struct TreeNode node;
node* create(int data){
    node* nn=(node*)malloc(sizeof(node));
    nn->data=data;
    nn->left=nn->right=NULL;
    return nn;
}
node* insert(node*root,int data){
    if(root==NULL)
     return create(data);
    if(data<root->data)
     root->left=insert(root->left,data);
```

```
    else if(data>root->data)
     root->right=insert(root->right,data);
    return root;
}
int findValue(node *root,int value){
    if(root==NULL)
     return 0;
    else if(value==root->data)
     return 1;
    else if(value<root->data)
     return findValue(root->left,value);
    else if(value>root->data)
     return findValue(root->right,value);
}
 int main(){
     int value,n;
     scanf("%d ",&n);
     node* root=NULL;
     int arr[n];
     for(int i=0;i<n;i++){
         scanf("%d ",&arr[i]);
         root=insert(root,arr[i]);
     }
     scanf("%d",&value);
     if(findValue(root,value))
      printf("Value %d is found in the tree.\n",value);
     else
      printf("Value %d is not found in the tree.\n",value);
      return 0;
 }
```

Status : Correct **Marks : 10/10**