







**Code:**

#include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node\* left;

struct node\* right;

};

struct node\* createNode(int value) {

struct node\* newNode = malloc(sizeof(struct node));

newNode->data = value;

newNode->left = NULL;

newNode->right = NULL;

return newNode;

}

struct node\* insert(struct node\* root, int value) {

if (root == NULL) {

return createNode(value);

}

if (value < root->data) {

root->left = insert(root->left, value);

}

else {

root->right = insert(root->right, value);

}

return root;

}

// search op

struct node\* search(struct node\* root, int value) {

if (root == NULL || root->data == value) {

return root;

}

if (value < root->data) {

return search(root->left, value);

}

else {

return search(root->right, value);

}

}

// Inorder

void inorderTraversal(struct node\* root) {

if (root == NULL) {

return;

}

inorderTraversal(root->left);

printf("%d ", root->data);

inorderTraversal(root->right);

}

// Preorder

void preorderTraversal(struct node\* root) {

if (root == NULL) {

return;

}

printf("%d ", root->data);

preorderTraversal(root->left);

preorderTraversal(root->right);

}

// Postorder

void postorderTraversal(struct node\* root) {

if (root == NULL) {

return;

}

postorderTraversal(root->left);

postorderTraversal(root->right);

printf("%d ", root->data);

}

int main() {

struct node\* root = NULL;

int n, i, x, s;

printf("Enter the number of nodes: ");

scanf("%d", &n);

printf("Enter %d values: ", n);

for (i = 0; i < n; i++) {

scanf("%d", &x);

root = insert(root, x);

}

printf("Inorder traversal is: ");

inorderTraversal(root);

printf("\n");

printf("Preorder traversal is: ");

preorderTraversal(root);

printf("\n");

printf("Postorder traversal is: ");

postorderTraversal(root);

printf("\n");

printf("Enter the value you want to search in Tree: ");

scanf("%d", &s);

struct node\* result = search(root, s);

if (result == NULL) {

printf("%d is not found.\n", s);

}

else {

printf("%d is found.\n", s);

}

return 0;

}





