	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

Title:	Binary Search tree
Problem Statement	Write a program in C to implement binary search tree with operations create, search and recursive traversal

Programmer Name: Arpan Agrawal

Batch: E6

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Define the structure for a node in the binary search tree
```

```
struct Node {
```

```
    int data;
```

```
    struct Node* left;
```

```
    struct Node* right;
```

```
};
```

```
// Function to create a new node
```

```
struct Node* createNode(int data) {
```

```
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
```

```
    newNode->data = data;
```

```
    newNode->left = NULL;
```

```
    newNode->right = NULL;
```

```
    return newNode;
```

```
}
```

```
// Function to insert a node in the binary search tree
```

```
struct Node* insert(struct Node* root, int data) {
```

```
    if (root == NULL) {
```


```
        // If the tree is empty, create a new node
```

```
        root = createNode(data);
```

```
    } else if (data < root->data) {
```

```
        // Insert in the left subtree
```

```
        root->left = insert(root->left, data);
```

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

```

    } else {
        // Insert in the right subtree
        root->right = insert(root->right, data);
    }
    return root;
}


// Function to perform inorder traversal (left, root, right)
void inorderTraversal(struct Node* root) {
    if (root != NULL) {
        inorderTraversal(root->left);
        printf("%d ", root->data);
        inorderTraversal(root->right);
    }
}

// Function to perform preorder traversal (root, left, right)
void preorderTraversal(struct Node* root) {
    if (root != NULL) {
        printf("%d ", root->data);
        preorderTraversal(root->left);
        preorderTraversal(root->right);
    }
}

// Function to perform postorder traversal (left, right, root)
void postorderTraversal(struct Node* root) {
    if (root != NULL) {
        postorderTraversal(root->left);
        postorderTraversal(root->right);
        printf("%d ", root->data);
    }
}

// Function to search for a value in the binary search tree

```

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

```

struct Node* search(struct Node* root, int key) {
    if (root == NULL || root->data == key) {
        return root;
    }
    if (key < root->data) {
        return search(root->left, key);
    }
    return search(root->right, key);
}

```

```


int main() {
    struct Node* root = NULL;
    int choice, value, key;
    struct Node* foundNode;

    while (1) {
        printf("\nBinary Search Tree Operations:\n");
        printf("1. Insert\n");
        printf("2. Search\n");
        printf("3. Inorder Traversal\n");
        printf("4. Preorder Traversal\n");
        printf("5. Postorder Traversal\n");
        printf("6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &value);
                root = insert(root, value);
                break;

            case 2:
                printf("Enter value to search: ");

```

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

```

scanf("%d", &key);
foundNode = search(root, key);
if (foundNode != NULL) {
    printf("Value %d found in the tree.\n", key);
} else {
    printf("Value %d not found in the tree.\n", key);
}
break;

```

```

case 3:
    printf("Inorder Traversal: ");
    inorderTraversal(root);
    printf("\n");
    break;

```

```

case 4:
    printf("Preorder Traversal: ");
    preorderTraversal(root);
    printf("\n");
    break;

```

```

case 5:
    printf("Postorder Traversal: ");
    postorderTraversal(root);
    printf("\n");
    break;

```

```

case 6:
    printf("Exiting...\n");
    exit(0);

```

```


default:
    printf("Invalid choice! Please try again.\n");

```

```

}
}

```

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

```

return 0;
}

```

OUTPUT

Binary Search Tree Operations:

1. Insert
2. Search
3. Inorder Traversal
4. Preorder Traversal
5. Postorder Traversal
6. Exit

Enter your choice: 1

Enter value to insert: 50

Binary Search Tree Operations:

1. Insert
2. Search
3. Inorder Traversal
4. Preorder Traversal
5. Postorder Traversal
6. Exit

Enter your choice: 1


Enter value to insert: 30

Binary Search Tree Operations:

1. Insert
2. Search
3. Inorder Traversal
4. Preorder Traversal
5. Postorder Traversal
6. Exit

Enter your choice: 3

Inorder Traversal: 30 50

	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESSMENT YEAR: 2024-2025	CLASS: SE
	SUBJECT: DATA STRUCTURES	
EXPT No:	LAB Ref: SE/2024-25/	Starting date:
	Roll No: 22203	Submission date:

Binary Search Tree Operations:

1. Insert
2. Search
3. Inorder Traversal
4. Preorder Traversal
5. Postorder Traversal
6. Exit

Enter your choice: 2

Enter value to search: 30

Value 30 found in the tree.