

## 8. Write a program

a) To construct a binary Search tree.

b) To traverse the tree using all the methods i.e., in order, preorder and post order

c) To display the elements in the tree.

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

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

```
/* Structure of a BST node */
```

```
struct node {
```

```
    int data;
```

```
    struct node *left;
```

```
    struct node *right;
```

```
};
```

```
/* Create a new node */
```

```
struct node* createNode(int value) {
```

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

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

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

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

```
    return newNode;
```

```
}
```

```
/* Insert a node into the BST */
```

```
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 if (value > root->data) {
```

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

```
    }
```

```
    return root;
```

```
}
```

```
/* Inorder Traversal */
```

```
void inorder(struct node* root) {
```

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

```
        inorder(root->left);
```

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

```
        inorder(root->right);
```

```
    }
```

```
}
```

```
/* Preorder Traversal */
```

```

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

```

```

/* Postorder Traversal */
void postorder(struct node* root) {
    if (root != NULL) {
        postorder(root->left);
        postorder(root->right);
        printf("%d ", root->data);
    }
}

```

```

int main() {
    struct node* root = NULL;
    int choice, value;

    do {
        printf("\n--- Binary Search Tree Menu ---\n");
        printf("1. Insert element\n");
        printf("2. Inorder Traversal\n");
        printf("3. Preorder Traversal\n");
        printf("4. Postorder Traversal\n");
        printf("5. 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("Inorder Traversal: ");
                inorder(root);
                printf("\n");
                break;

            case 3:
                printf("Preorder Traversal: ");
                preorder(root);

```

```

        printf("\n");
        break;

    case 4:
        printf("Postorder Traversal: ");
        postorder(root);
        printf("\n");
        break;

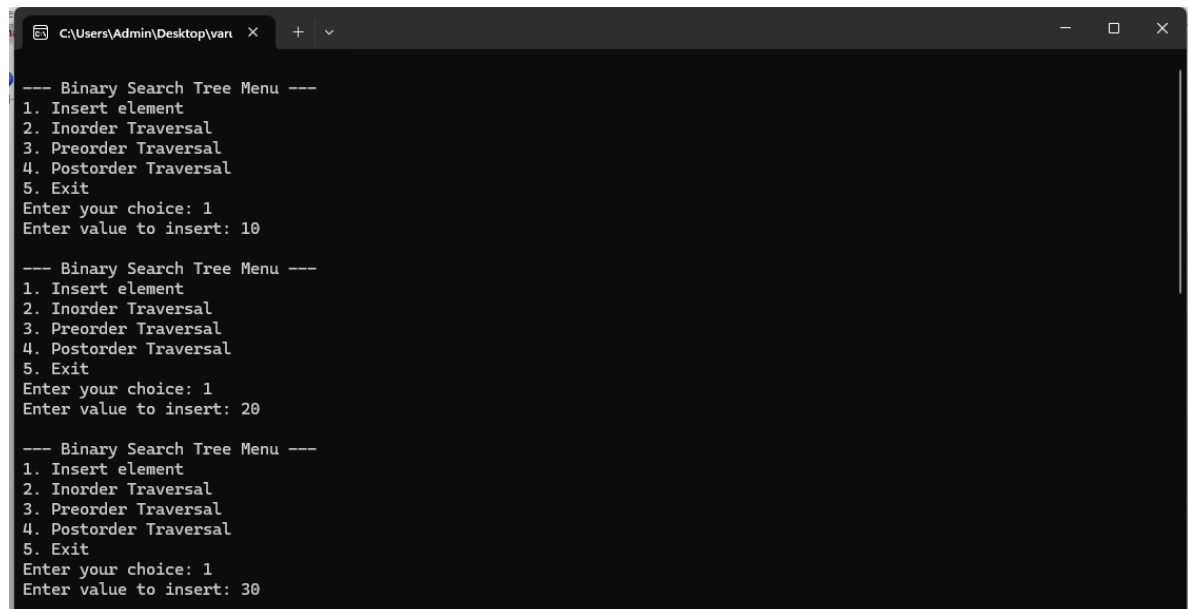
    case 5:
        printf("Exiting...\n");
        break;

    default:
        printf("Invalid choice!\n");
    }
} while (choice != 5);

return 0;
}

```

## Output:



```

C:\Users\Admin\Desktop\var >
--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 1
Enter value to insert: 10

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 1
Enter value to insert: 20

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 1
Enter value to insert: 30

```

```
C:\Users\Admin\Desktop\var  x + v

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 1
Enter value to insert: 40

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 2
Inorder Traversal: 10 20 30 40

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 3
Preorder Traversal: 10 20 30 40
```

```
--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 4
Postorder Traversal: 40 30 20 10

--- Binary Search Tree Menu ---
1. Insert element
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Exit
Enter your choice: 5
Exiting...

Process returned 0 (0x0)   execution time : 53.435 s
Press any key to continue.
```