Write a program

a) To construct a binary Search tree.

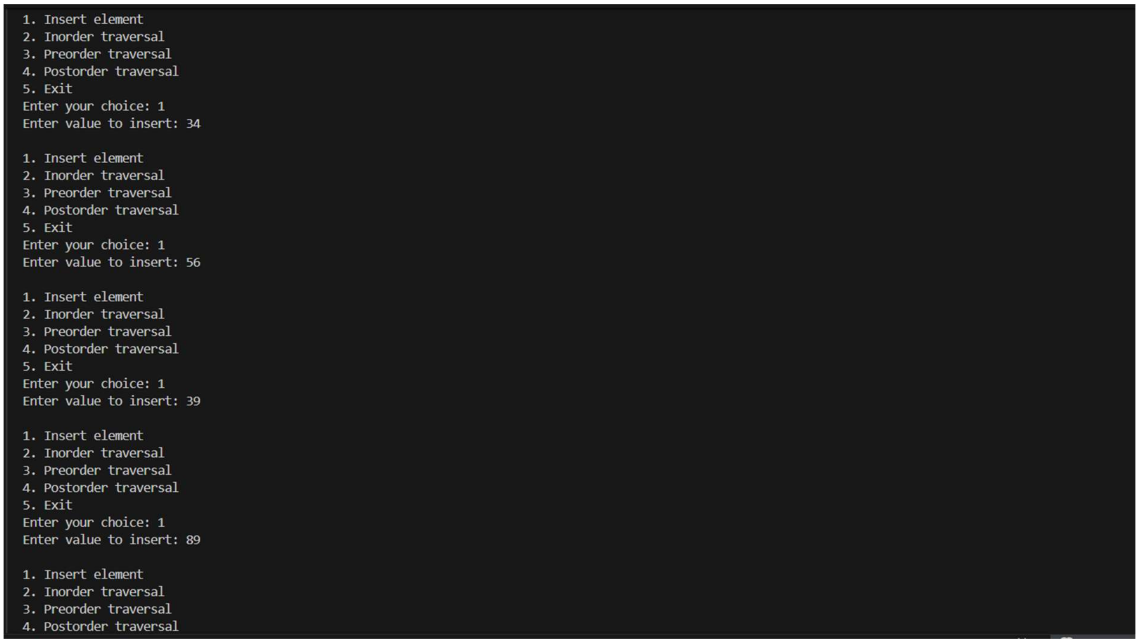
b) To traverse the tree using all the methods i.e., in-order, preorder and post order

c) To display the elements in the tree.

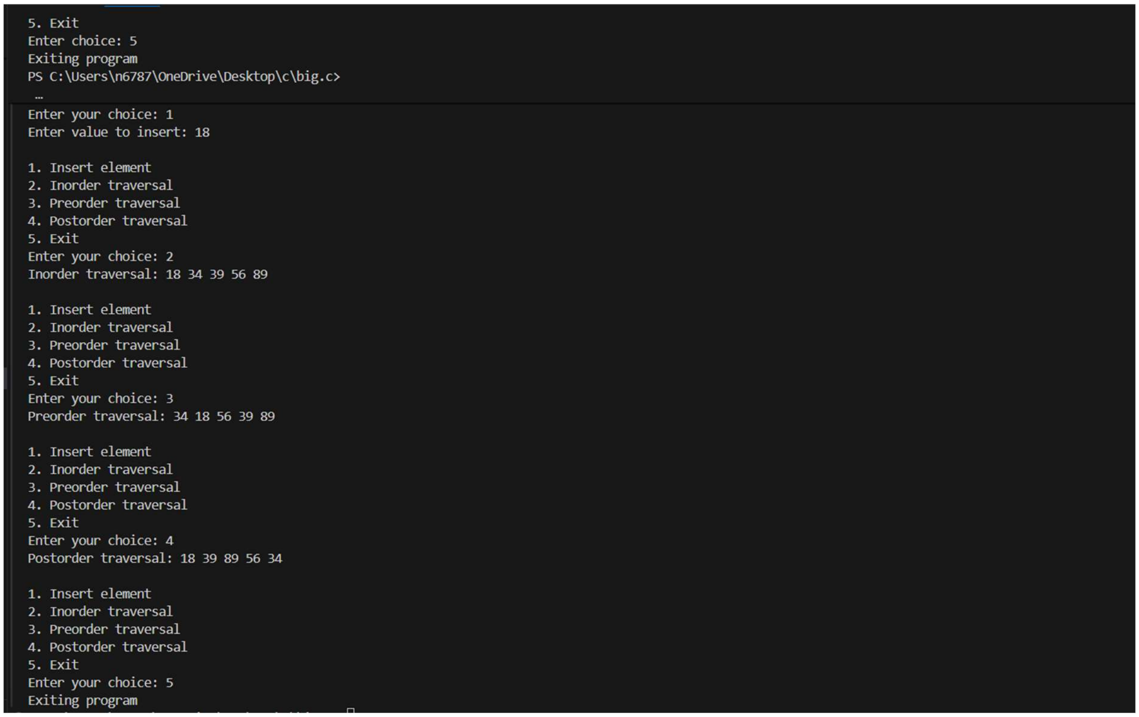
```
#include <stdio.h>
#include <stdlib.h>

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

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

struct node* insert(struct node* root, int item) {
    if (root == NULL)
        return createNode(item);

    if (item < root->data)
        root->left = insert(root->left, item);
    else if (item > root->data)
        root->right = insert(root->right, item);

    return root;
}

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

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

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, val;

    while (1) {
        printf("\n1. Insert element");
        printf("\n2. Inorder traversal");
        printf("\n3. Preorder traversal");
        printf("\n4. Postorder traversal");
        printf("\n5. Exit");
        printf("\nEnter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to insert: ");
                scanf("%d", &val);
                root = insert(root, val);
                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 program\n");
                return 0;

            default:
                printf("Invalid choice\n");
        }
    }
}
```

Output:



```
1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 1
Enter value to insert: 34

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 1
Enter value to insert: 56

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 1
Enter value to insert: 39

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 1
Enter value to insert: 89

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
```



```
5. Exit
Enter choice: 5
Exiting program
PS C:\Users\n6787\OneDrive\Desktop\c\big.c>
 …
Enter your choice: 1
Enter value to insert: 18

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 2
Inorder traversal: 18 34 39 56 89

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 3
Preorder traversal: 34 18 56 39 89

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 4
Postorder traversal: 18 39 89 56 34

1. Insert element
2. Inorder traversal
3. Preorder traversal
4. Postorder traversal
5. Exit
Enter your choice: 5
Exiting program
```