/* 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>

/* Definition of 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 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;
}

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

/* Pre-order Traversal */
void preorder(struct node* root) {
```

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

/* Post-order 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 n, i, value;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    printf("Enter elements:\n");
    for (i = 0; i < n; i++) {
```

```
        scanf("%d", &value);
        root = insert(root, value);
    }

    printf("\nIn-order Traversal: ");
    inorder(root);

    printf("\nPre-order Traversal: ");
    preorder(root);

    printf("\nPost-order Traversal: ");
    postorder(root);

    return 0;
}
```

OUTPUT:-