

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

```
struct Node {
    int data;
    struct Node *left;
    struct Node *right;
};
```

```
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;
}
```

```
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;
}
```

```
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 n, value, i;

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

    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &value);
        root = insert(root, value);
    }

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

    printf("\nPreorder Traversal: ");
    preorder(root);

    printf("\nPostorder Traversal: ");
    postorder(root);

    return 0;
}

```

```

Enter number of elements: 6
Enter the elements:
6
9
5
5
7
4

In-order Traversal: 4 5 6 7 9
Preorder Traversal: 6 5 4 9 7
Postorder Traversal: 4 5 7 9 6
Process returned 0 (0x0)   execution time : 13.480 s
Press any key to continue.

```