

```

#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 to insert: ");
    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("\nPre-order Traversal: ");
preorder(root);

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

return 0;
}
```

```
Enter number of elements to insert: 6
Enter the elements:
8 4 6 9 2 1

In-order Traversal: 1 2 4 6 8 9
Pre-order Traversal: 8 4 2 1 6 9
Post-order Traversal: 1 2 6 4 9 8
Process returned 0 (0x0)   execution time : 8.856 s
Press any key to continue.
```