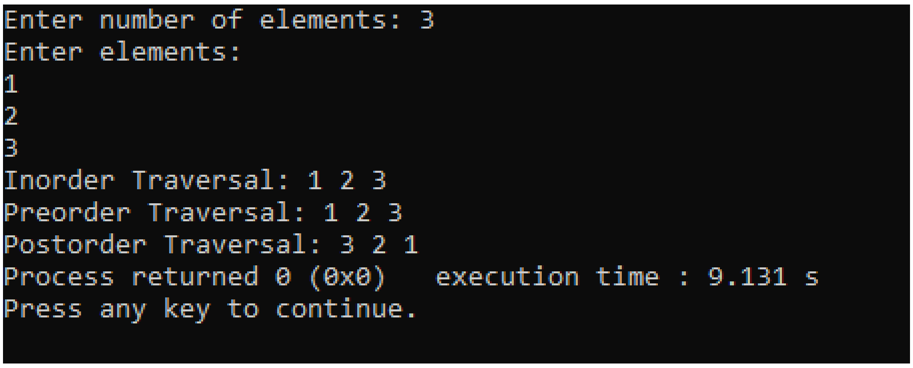
```
#include <stdio.h>
#include <stdlib.h>

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

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

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

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

    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, data, i;

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

    printf("Enter elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &data);

        root = insert(root, data);

    }


    printf("Inorder Traversal: ");

    inorder(root);


    printf("\nPreorder Traversal: ");

    preorder(root);


    printf("\nPostorder Traversal: ");

    postorder(root);


    return 0;

}
```



```
Enter number of elements: 3
Enter elements:
1
2
3
Inorder Traversal: 1 2 3
Preorder Traversal: 1 2 3
Postorder Traversal: 3 2 1
Process returned 0 (0x0)   execution time : 9.131 s
Press any key to continue.
```