

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

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

```
typedef struct node *Node
```

```
Node get Node() {
```

```
    Node x;
```

```
    x = (Node) malloc(sizeof(struct node));
    if (x == NULL) {
```

```
        printf("Memory full\n");
```

```
        exit(0);
```

```
    }
```

```
    return x;
```

```
}
```

```

Node insert (int data ; Node root) {
    Node temp, cur, prev;
    char direction[10];
    int i;
    temp = getNode();
    temp->data = data;
    temp->left = NULL;
    temp->right = NULL;
    if (root == NULL) {
        return temp;
    }
    printf (" give direction: \n");
    scanf ("%s", direction);
    prev = NULL;
    cur = root;
    for (i=0; i < strlen(direction) && cur != NULL; i++) {
        prev = cur;
        if (direction[i] == 'l')
            cur = cur->left;
        else
            cur = cur->right;
    }
    if (cur != NULL) {
        printf ("error\n");
        free(temp);
        return root;
    }
    if (cur == NULL) {
        if (direction[i-1] == 'l')
            prev->left = temp;
        else
            prev->right = temp;
    }
    return root;
}

```

```

void pre_order (Node root) {
    if (root != NULL) {
        printf("The item is %d\n", root->data);
        pre_order (root->left);
        pre_order (root->right);
    }
}

```

```

}
void inorder (Node root) {
    if (root != NULL) {
        printf("The
        inorder (root->left);
        printf("The item is %d\n", root->data);
        inpostorder (root->right);
    }
}

```

```

}
void postorder (Node root) {
    if (root != NULL) {
        postorder (root->left);
        postorder (root->right);
        printf("The item is %d\n", root->data);
    }
}

```

```

void display (Node root, int i) {
    int j;
    if (root != NULL) {
        display (root->right, i+1);
        for (j=1; j<=i; j++) {
            printf (" ");
        }
        Pf ("%d\n", root->data);
        display (root->left, i+1);
    }
}

```

```

void main () {
    int choice, item, flag = 1, key;
    Node root = NULL;
    while (flag == 1) {
        Pf ("1. insert\n 2. preorder\n 3. inorder\n 4. postorder\n 5. display\n");
        printf ("Enter the choice : ");
        scanf ("%d", &choice);
    }
}

```

PTO



switch (choice) {

case 1: Pf("Enter the item :");  
scanf ("%d", &item);  
root = insert (root, item);  
break

case 2: if (root == NULL) {  
printf ("Tree is empty");  
} else {  
printf ("Giuntree : \n");  
display (root, 1);  
Pf ("Pre order : \n");  
pre order (root);  
}  
break;

case 3: if (root == NULL) {  
printf ("Tree is empty \n");  
} else {  
Pf ("giuntree : \n");  
display (root, 1);  
Pf ("in order : \n");  
~~pre~~<sup>in</sup> order (root);  
}  
break;

```
case 4: if (root == NULL) {  
        printf ("Tree is empty");
```

```
    } else {  
        Pf ("Given tree : \n");  
        display (root, 1);  
        printf ("Postorder : \n");  
        postorder (root);  
    }
```

```
    break;
```

```
case 5: display (root, 1);
```

```
        break;
```

```
default: exit(0)
```