Shreehari Kulkarni
USN: 1BM19CS153

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct node
{
   int info
   struct node *llink;
   struct node *rlink;
};
typedef struct node *NODE
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if(x==NULL)
    {
       printf("Memory not available");
       exit(0);
    }
```

```
return x;
}

NODE insert(int item, NODE root)
{
    NODE temp, cur, prev;
    char direction[10];
    int i;
    temp = getnode();
    temp->info = item;
    temp->llink = NULL;
    temp->rlink = NULL;
    if(root == NULL)
        return temp
    printf("Give direction to insert\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->llink;
        else
```

```
cur = cur -> rlink
}
if ( cur != NULL || i != strlen(direction)
    printf("insertion not possible");
    free(temp);
    return(root);
{
if (cur == NULL)
    if (direction[i-1] == 'l')
        prev -> llink = temp
    else
        prev -> rlink = temp
;
return(root);
}
```

```
void preorder(NODE root)
{
  if (root != NULL)
  {
    printf("Item is %d\n", root->info);
    preorder (root -> llink);
    preorder (root -> rlink);
  }
}

void inorder (NODE root)
{
  if (root != NULL)
  {
    inorder (root -> llink);
    printf ("Item is %d\n", root->info);
    inorder (root -> rlink);
  }
}
```

```
void postorder(NODE root)
{
    if(root != NULL)
    {
        postorder(root->llink);
        postorder(root->rlink);
        printf("Items: %d\n", root
                                ->info);
    }
}

void display(NODE root, int i)
{
    int j;
    if(root != NULL)
    {
        display(root->rlink, i+1);
        for(j=1; j<=i; j++)
            printf("  ");
        printf("%d\n", root->info);
        display(root->llink, i+1);
    }
}
```

```
void main()
{
    NODE root = NULL
    int choice, i, item
    for(;;)
    {
        printf("1: insert\n");
        printf("2: preorder\n");
        printf("3: inorder\n");
        printf("4: postorder\n");
        printf("5: Display\n");
        Scanf("%d", &choice);
    switch (choice)
    {
        case1:  printf("Enter item\n")
                Scanf("%d", &item);
                root = insert(item, root);
                break;

        case 2: if (root == NULL)
                {
                    printf("tree empty\n");
                }
```

```
else
{
    printf("Given tree is");
    display(root, 1);
    printf("Preorder traversal\n");
    preorder(root);
}

break;

case 3: if(root==NULL)
{
    printf("Tree empty\n");
}
else
{
    printf("Given Tree is");
    display(root, 1);
    printf("in order traversal\n");
    inorder(root);
}
break;
```

```
case 4: if (root == NULL)
        {
            printf("\nTree empty\n");
        }
        else
        {
            printf("\nInorder is ");
            display(root, 1);
            printf("\n postorder \n");
            postorder(root);
        }
        break;

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

default: exit(0);
    }
}
}
```