

## WEEK12

### Binary tree:

#### CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.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\n");
        exit(0);
    }
    return x;
}
void freenode(NODE x)
{
    free(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\n");
    freenode(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("%d\n", root->info);
        preorder(root->llink);
        preorder(root->rlink);
    }
}

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

```

```

    }
}
void postorder(NODE root)
{
    if (root != NULL)
    {
        postorder(root->llink);
        postorder(root->rlink);
        printf("%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\n2.Preorder\n3.Inorder\n4.Postorder\n5.Display\n6.Exit\n");
        printf("Enter the choice:\n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("Enter the item:\n");
                scanf("%d", &item);
                root = insert(item, root);
                break;
            case 2:
                if (root == NULL)

```

```

    {
        printf("Tree is empty!\n");
    }
    else
    {
        printf("Given tree is\n");
        display(root, 1);
        printf("The preorder traversal is \n");
        preorder(root);
    }
    break;
case 3:
    if (root == NULL)
    {
        printf("Tree is empty\n");
    }
    else
    {
        printf("Given tree is\n");
        display(root, 1);
        printf("The inorder traversal is \n");
        inorder(root);
    }
    break;
case 4:
    if (root == NULL)
    {
        printf("Tree is empty\n");
    }
    else
    {
        printf("Given tree is\n");
        display(root, 1);
        printf("The postorder traversal is \n");
        postorder(root);
    }
    break;
case 5:
    printf("The given tree is:\n-----\n");
    display(root, 1);
    break;
case 6:
    exit(0);

```

```

        default:
            printf("Invalid input!\nTRY AGAIN\n");
        }
    }
}

```

## OUTPUT:

```

C:\Users\misaf\Desktop\DS LAB\week12>btc
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
100
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
20
Give direction to insert
1
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:

```

```
Enter the choice:
1
Enter the item:
200
Give direction to insert
r
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
10
Give direction to insert
l
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
30
Give direction to insert
lr
```

```
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
150
Give direction to insert
rl
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
1
Enter the item:
300
Give direction to insert
rr
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
```

Enter the choice:

5

The given tree is:

-----

300

200

150

100

30

20

10

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Display

6.Exit

Enter the choice:

2

Given tree is

300

200

150

100

30

20

10

The preorder traversal is

100

20



The preorder traversal is

100

20

10

30

200

150

300

1.Insert

2.Preorder

3.Inorder

4.Postorder

5.Display

6.Exit

Enter the choice:

3

Given tree is

300

200

150

100

30

20

10

The inorder traversal is

10

20

30

100

150

```
The inorder traversal is
10
20
30
100
150
200
300
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
4
Given tree is
      300
     200
    150
   100
  30
 20
 10
The postorder traversal is
10
30
20
150
300
```

```
The postorder traversal is
10
30
20
150
300
200
100
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice:
6
C:\Users\misaf\Desktop\DS LAB\week12>
```

