```c
//Binary tree
#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");
        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;
```

```
3178        temp->info=item;
3179        temp->llink=NULL;
3180        temp->rlink=NULL;
3181        if(root==NULL)
3182            return temp;
3183        printf("Give direction to insert\n");
3184        scanf("%s",direction);
3185        prev=NULL;
3186        cur=root;
3187        for(i=0;i<strlen(direction)&&cur!=NULL;i++)
3188        {
3189            prev=cur;
3190            if(direction[i]=='l'||direction[i]=='L')
3191                cur=cur->llink;
3192            else
3193                cur=cur->rlink;
3194        }
3195        if(cur!=NULL||i!=strlen(direction))
3196        {
3197            printf("Insertion not possible\n");
3198            freenode(temp);
3199            return(root);
3200        }
3201        if(cur==NULL)
3202        {
3203            if(direction[i-1]=='l'||direction[i-1]=='L')
3204                prev->llink=temp;
3205            else
3206                prev->rlink=temp;
3207        }
3208        return(root);
3209    }
3210    void preorder(NODE root)
3211    {
```

```
3211 {
3212     if(root!=NULL)
3213     {
3214         printf(" %d -> ",root->info);
3215         preorder(root->llink);
3216         preorder(root->rlink);
3217     }
3218 }
3219 void inorder(NODE root)
3220 {
3221     if(root!=NULL)
3222     {
3223         inorder(root->llink);
3224         printf(" %d -> ",root->info);
3225         inorder(root->rlink);
3226     }
3227 }
3228 void postorder(NODE root)
3229 {
3230     if (root!=NULL)
3231     {
3232         postorder(root->llink);
3233         postorder(root->rlink);
3234         printf(" %d -> ",root->info);
3235     }
3236 }
3237 void display(NODE root,int i)
3238 {
3239     int j;
3240     if(root!=NULL)
3241     {
3242         display(root->rlink,i+1);
3243         for (j=1;j<=i;j++)
3244             printf("   ");
```

```
3244                printf("   ");
3245            printf("%d\n",root->info);
3246            display(root->llink,i+1);
3247        }
3248    }
3249    int main()
3250    {
3251        NODE root=NULL;
3252        int choice,item;
3253        for(;;)
3254        {
3255            printf("1.Insert\n2.Preorder\n3.Inorder\n4.Postorder\n5.Display\n6.Exit\n");
3256            printf("Enter the choice\n");
3257            scanf("%d",&choice);
3258            switch(choice)
3259            {
3260                case 1: printf("Enter the item  : ");
3261                    scanf("%d",&item);
3262                    root=insert(item,root);
3263                    break;
3264                case 2: if(root==NULL)
3265                        {
3266                         printf("Tree is empty");
3267                        }
3268                        else
3269                        {
3270                         printf("Preorder traversal is \n");
3271                         preorder(root);
3272                        }
3273                        break;
3274                case 3:if(root==NULL){
3275                         printf("Tree is empty");
3276                       }
3277                       else
```

```
3277                    else
3278                    {
3279                     printf("The inorder traversal is \n");
3280                     inorder(root);
3281                    }
3282                    break;
3283            case 4:if (root==NULL){
3284                     printf("Tree is empty");
3285                    }
3286                    else
3287                    {
3288                     printf("The postorder traversal is \n");
3289                     postorder(root);
3290                    }
3291                    break;
3292            case 5:display(root,1);
3293                    break;
3294            case 6:exit(0);
3295            default:printf("Enter proper instructions!!\n");
3296                    break;
3297        }
3298    }
3299    return 0;
3300 }
3301
```

```
"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 10

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 20
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
R

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 21
Give direction to insert
LL

1.Insert
2.Preorder
3.Inorder
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 22
Give direction to insert
LR

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 31
Give direction to insert
RL

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
1
Enter the item  : 32
Give direction to insert
RR

1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
5
      32
    30
      31
  10
      22
    20
      21
```

"C:\Users\Shreshtha Aggarwal\Desktop\1stpro\bin\Debug\1stpro.exe"

```
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
2
Preorder traversal is
 10 ->  20 ->  21 ->  22 ->  30 ->  31 ->  32 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
3
The inorder traversal is
 21 ->  20 ->  22 ->  10 ->  31 ->  30 ->  32 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
4
The postorder traversal is
 21 ->  22 ->  20 ->  31 ->  32 ->  30 ->  10 ->
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
6.Exit
Enter the choice
6

Process returned 0 (0x0)   execution time : 49.412 s
Press any key to continue.
```