

```

1  #include<stdio.h>
2  #include<conio.h>
3  #include<stdlib.h>
4  #include<string.h>
5  struct node
6  {
7      int info;
8      struct node*llink;
9      struct node*rlink;
10 };
11 typedef struct node*NODE;
12 NODE getnode()
13 {
14     NODE x;
15     x=(NODE)malloc(sizeof(struct node));
16     if(x==NULL)
17     {
18         printf("memory not available");
19         exit(0);
20     }
21     return x;
22 }
23 void freenode(NODE x)
24 {
25     free(x);
26 }
27 NODE insert(int item,NODE root)
28 {
29     NODE temp,cur,prev;
30     char direction[10];
31     int i;

```

```

31     int i;
32     temp=getnode();
33     temp->info=item;
34     temp->llink=NULL;
35     temp->rlink=NULL;
36     if(root==NULL)
37         return temp;
38     printf("give direction to insert\n");
39     scanf("%s",direction);
40     prev=NULL;
41     cur=root;
42     for(i=0;i<strlen(direction)&&cur!=NULL;i++)
43     {
44         prev=cur;
45         if(direction[i]=='l')
46             cur=cur->llink;
47         else
48             cur=cur->rlink;
49     }
50     if(cur!=NULL||i!=strlen(direction))
51     {
52         printf("insertion not possible\n");
53         freenode(temp);
54         return(root);
55     }
56     if(cur==NULL)
57     {
58         if(direction[i-1]=='l')
59             prev->llink=temp;
60         else
61             prev->rlink=temp;
62     }

```

```

61 prev->rlink=temp;
62 }
63 return(root);
64 }
65 void preorder(NODE root)
66 {
67 if(root!=NULL)
68 {
69 printf("the item is %d\n",root->info);
70 preorder(root->llink);
71 preorder(root->rlink);
72 }
73 }
74 void inorder(NODE root)
75 {
76 if(root!=NULL)
77 {
78 inorder(root->llink);
79 printf("the item is %d\n",root->info);
80 inorder(root->rlink);
81 }
82 }
83 void postorder(NODE root)
84 {
85 if (root!=NULL)
86 {
87 postorder(root->llink);
88 postorder(root->rlink);
89 printf("the item is %d\n",root->info);
90 }
91 }
92 void display(NODE root,int i)

```

```

92 void display(NODE root,int i)
93 {
94 int j;
95 if(root!=NULL)
96 {
97 display(root->rlink,i+1);
98 for (j=1;j<=i;j++)
99 printf(" ");
100 printf("%d\n",root->info);
101 display(root->llink,i+1);
102 }
103 }
104
105 void main()
106 {
107 NODE root=NULL;
108 int choice,i,item;
109 //clrscr();
110 for(;;)
111 {
112 printf("1.insert\n2.preorder\n3.inorder\n4.postorder\n5.display\n");
113 printf("enter the choice\n");
114 scanf("%d",&choice);
115 switch(choice)
116 {
117 case 1: printf("enter the item\n");
118         scanf("%d",&item);
119         root=insert(item,root);
120         break;
121 case 2: if(root==NULL)
122         {
123         printf("tree is empty");

```

```

124     }
125     else
126     {
127         printf("given tree is \n");
128         display(root,1);
129         printf("the preorder traversal is \n");
130         preorder(root);
131     }
132     break;
133 case 3:if(root==NULL)
134     {
135         printf("tree is empty");
136     }
137     else
138     {
139         printf("given tree is \n");
140         display(root,1);
141         printf("the inorder traversal is \n");
142         inorder(root);
143     }
144     break;
145 case 4:if (root==NULL)
146     {
147         printf("tree is empty");
148     }
149     else
150     {
151         printf("given tree is \n");
152         display(root,1);
153         printf("the postorder traversal is \n");
154         postorder(root);

```

```

155     }
156     break;
157 case 5:display(root,1);
158     break;
159 default:exit(0);
160 }
161 }
162 }
163 }

```

```

1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
1
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
2
give direction to insert
1
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
3
give direction to insert
r

```

```

give direction to insert
r
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
4
give direction to insert
lr
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
5
give direction to insert
rl
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1

```

```

1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
2
given tree is
    3
   7
  5
 1
  4
   6
  2
the preorder traversal is
the item is 1
the item is 2
the item is 4
the item is 6
the item is 3
the item is 5
the item is 7
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
3
given tree is

```

```

1
enter the item
6
give direction to insert
lrl
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
7
give direction to insert
rlr
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
5
    3
      7
        5
          1
            4
              6
                2
1.insert

```

```

given tree is
    3
      7
        5
          1
            4
              6
                2
the inorder traversal is
the item is2
the item is6
the item is4
the item is1
the item is5
the item is7
the item is3
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
4
given tree is
    3
      7
        5
          1
            4
              6
                2

```

```
4.postorder
5.display
enter the choice
4
given tree is
    3
      7
    5
  1
    4
      6
    2
the postorder traversal is
the item is6
the item is4
the item is2
the item is7
the item is5
the item is3
the item is1
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
6

...Program finished with exit code 0
Press ENTER to exit console.
```