Sym | lab10alt.c | lab10.c

Func
dis
fre
get
ino
ins
ma
pos
pre
Struc
noc
i
ll
r
Type
NC

```c
#include<stdio.h>
#include<conio.h>
#include<process.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;
temp->llink=NULL;
```

line: 1 / 163 col: 0 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: unknown

```
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 if(direction[i]=='r')
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)
```

```
prev->rlink=temp;
}
return(root);
}
void preorder(NODE root)
{
if(root!=NULL)
{
printf("the item is %d\n",root->info);
preorder(root->llink);
preorder(root->rlink);
}
}
void inorder(NODE root)
{
if(root!=NULL)
{
inorder(root->llink);
printf("the item is%d\n",root->info);
inorder(root->rlink);
}
}
void postorder(NODE root)
{
if (root!=NULL)
{
postorder(root->llink);
postorder(root->rlink);
printf("the item is%d\n",root->info);
}
}
void display(NODE root,int i)
{
int j;
if(root!=NULL)
```

line: 1 / 163 col: 0 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: unknown

```
{
postorder(root->llink);
postorder(root->rlink);
printf("the item is%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);
}
}

int main()
{
NODE root=NULL;
int choice,item;
for(;;)
{
printf("1.insert\n2.preorder\n3.inorder\n4.postorder\n5.display\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)
```

```
105
106 int main()
107 {
108 NODE root=NULL;
109 int choice,item;
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");
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");
```

line: 1 / 163 col: 0 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: unknown

```
          preorder(root);
         }
         break;
case 3:if(root==NULL)
       {
        printf("tree is empty");
       }
       else
       {
        printf("given tree is");
        display(root,1);
        printf("the inorder traversal is \n");
        inorder(root);
        }
       break;
case 4:if (root==NULL)
        {
        printf("tree is empty");
         }
       else
       {
        printf("given tree is");
        display(root,1);
        printf("the postorder traversal is \n");
        postorder(root);
       }
      break;
case 5:display(root,1);
       break;
default:exit(0);
}
}
}
```

```
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
11
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
21
give direction to insert
l
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
33
give direction to insert
lr
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
45
give direction to insert
```

```
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
45
give direction to insert
r
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
65
give direction to insert
lrr
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
32
give direction to insert
r
insertion not possible
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
```

```
1
enter the item
32
give direction to insert
r
insertion not possible
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
rl
give direction to insert
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
34
give direction to insert
rl
insertion not possible
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
67
give direction to insert
rll
1.insert
2.preorder
```

```
enter the choice
1
enter the item
67
give direction to insert
rll
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
5
    45
      32
        67
  11
        65
      33
    21
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
1
enter the item
67
give direction to insert
ll
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
5
    45
      32
        67
```

```
11
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
5
    45
      32
        67
  11
        65
      33
    21
      67
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
2
given tree is    45
      32
        67
  11
        65
      33
    21
      67
the preorder traversal is
the item is 11
the item is 21
the item is 67
the item is 33
the item is 65
the item is 45
the item is 32
the item is 67
1.insert
```

```
       65
      33
    21
      67
the preorder traversal is
the item is 11
the item is 21
the item is 67
the item is 33
the item is 65
the item is 45
the item is 32
the item is 67
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
3
given tree is    45
      32
        67
  11
       65
      33
    21
      67
the inorder traversal is
the item is67
the item is21
the item is33
the item is65
the item is11
the item is67
the item is32
the item is45
1.insert
2.preorder
3.inorder
4.postorder
```

```
  11
        65
      33
    21
      67
the inorder traversal is
the item is67
the item is21
the item is33
the item is65
the item is11
the item is67
the item is32
the item is45
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
4
given tree is    45
      32
        67
  11
        65
      33
    21
      67
the postorder traversal is
the item is67
the item is65
the item is33
the item is21
the item is67
the item is32
the item is45
the item is11
1.insert
2.preorder
3.inorder
```

```
      33
    21
      67
the postorder traversal is
the item is67
the item is65
the item is33
the item is21
the item is67
the item is32
the item is45
the item is11
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
5
    45
      32
        67
  11
        65
      33
    21
      67
1.insert
2.preorder
3.inorder
4.postorder
5.display
enter the choice
7


------------------
(program exited with code: 0)

Press any key to continue . . .
```