bst.c

```c
#include<stdio.h>
#include<process.h>
struct node
 {
  int info;
  struct node *rlink;
  struct node *llink;
 };
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 {
  printf("Memory is Full.\n");
  exit(0);
 }
 return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert(NODE root,int item)
{
NODE temp,cur,prev;
temp=getnode();
temp->rlink=NULL;
temp->llink=NULL;
temp->info=item;
if(root==NULL)
 return temp;
prev=NULL;
```

```
35  cur=root;
36  while(cur!=NULL)
37  {
38  prev=cur;
39  cur=(item<cur->info)?cur->llink:cur->rlink;
40  }
41  if(item<prev->info)
42   prev->llink=temp;
43  else
44   prev->rlink=temp;
45  return root;
46  }
47  void display(NODE root,int i)
48  {
49  int j;
50  if(root!=NULL)
51   {
52    display(root->rlink,i+1);
53    for(j=0;j<i;j++)
54        printf("   ");
55     printf("%d\n",root->info);
56       display(root->llink,i+1);
57   }
58  }
59  NODE delete(NODE root,int item)
60  {
61  NODE cur,parent,q,suc;
62  if(root==NULL)
63  {
64  printf("Tree is Empty.\n");
65  return root;
66  }
67  parent=NULL;
68  cur=root;
```

```
69  while(cur!=NULL&&item!=cur->info)
70  {
71  parent=cur;
72  cur=(item<cur->info)?cur->llink:cur->rlink;
73  }
74  if(cur==NULL)
75  {
76   printf("Not Found.\n");
77   return root;
78  }
79  if(cur->llink==NULL)
80   q=cur->rlink;
81  else if(cur->rlink==NULL)
82   q=cur->llink;
83  else
84   {
85   suc=cur->rlink;
86   while(suc->llink!=NULL)
87    suc=suc->llink;
88   suc->llink=cur->llink;
89   q=cur->rlink;
90   }
91   if(parent==NULL)
92    return q;
93   if(cur==parent->llink)
94    parent->llink=q;
95   else
96    parent->rlink=q;
97   freenode(cur);
98   return root;
99   }
100
```

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

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

  inorder(root->llink);
  printf("%d\n",root->info);
  inorder(root->rlink);
  }
 }
void main()
{
int item,choice;
NODE root=NULL;
```

```
for(;;)
{
printf("1.Insert\n2.Display\n3.Preorder\n4.Postorder\n5.Inoreder\n6.Delete\n7.Exit\n");
printf("Enter the choice:\n");
scanf("%d",&choice);
switch(choice)
 {
  case 1:printf("Enter the item:\n");
         scanf("%d",&item);
         root=insert(root,item);
         break;
  case 2:display(root,0);
         break;
  case 3:preorder(root);
         break;
  case 4:postorder(root);
         break;
  case 5:inorder(root);
         break;
  case 6:printf("Enter the item:\n");
         scanf("%d",&item);
         root=delete(root,item);
         break;
  default:exit(0);
          break;
      }
    }
 }
```

"C:\Users\Neha Chadaga\Desktop\bst.exe"

```
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
67
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
56
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
100
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
2
  100
67
  56
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
```

```
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
78
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
23
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
90
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
Enter the item:
123
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
1
```

```
Enter the item:
45
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
2
    123
  100
      90
    78
67
  56
      45
    23
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
3
PREORDER
67
56
23
45
100
78
90
123
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
4
```

```
4
POSTORDER
45
23
56
90
78
123
100
67
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
5
INORDER
23
45
56
67
78
90
100
123
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
6
Enter the item:
56
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
2
```

```
    123
  100
      90
    78
67
    45
  23
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inoreder
6.Delete
7.Exit
Enter the choice:
7

Process returned 0 (0x0)   execution time : 40.279 s
Press any key to continue.
```