C:\Users\91966\Desktop\DSLAB\BST.c - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

BST.c

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.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("mem 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;
cur=root;
```

```
41  }
42  if(item<prev->info)
43   prev->llink=temp;
44  else
45   prev->rlink=temp;
46  return root;
47  }
48  void display(NODE root,int i)
49  {
50  int j;
51  if(root!=NULL)
52   {
53    display(root->rlink,i+1);
54    for(j=0;j<i;j++)
55      printf("   ");
56     printf("%d\n",root->info);
57     display(root->llink,i+1);
58   }
59  }
60
61  void preorder(NODE root)
62  {
63  if(root!=NULL)
64   {
65    printf("%d\t",root->info);
66    preorder(root->llink);
67    preorder(root->rlink);
68    }
69   }
70  void postorder(NODE root)
71  {
72  if(root!=NULL)
```

BST.c

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

   inorder(root->llink);
   printf("%d\t",root->info);
   inorder(root->rlink);
   }
 }
void main()
{
int item,choice;
NODE root=NULL;
for(;;)
{
printf("\n1.insert\n2.display\n3.pre\n4.post\n5.in\n6.exit\n");
printf("enter the choice:");
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;
```

```
107         case 3:preorder(root);
108             break;
109         case 4:postorder(root);
110             break;
111         case 5:inorder(root);
112             break;
113         default:exit(0);
114             break;
115         }
116     }
117 }
```