(Lab-10) Binary Search tree

```
#include<stdio.h>
#include<conio.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("mem full\n");
 exit(0);
}
 return x;
}
```

```
NODE insert(NODE root, int item)
{ NODE temp, prev, cur;
  temp = getnode();
  temp->rlink == NULL;
  temp->llink = NULL;
  temp->info = item;
  if(root == NULL)
  return temp;
 prev = NULL;
 cur = root;
 while (cur != NULL)
{ prev = cur;
  cur = (item < cur->info)? cur->llink : cur->rlink;
}
 if (item < prev->info)
  prev = cur;
  prev->llink = temp;
  else
  prev->rlink = temp;
  return root;
}
 void display (NODE root, int i)
{ int j;
 if(root != NULL)
 { display (root->rlink, i+1);
   for(j=0; j<i; j++)
    printf("   ");
   printf("%d\n", root->info);
   display (root->llink, i+1);
 }}
```

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

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

void main()
{ int item, choice;
 NODE root = NULL;
 for (;;)
{ printf("\n 1: Insert \n 2: Display\n 3: pre-order\n
      4: Inorder\n 5: Postorder
```

```
scanf("%d\n", &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: Inorder (root);
  break;
case 5: postorder(root);
    break;
default : exit(0);
    break;
}
}
```