LAB-10 program

```
#include <stdio.h>
#include <conio.h>
#include <process.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;
  }
```

```
NODE insert(NODE root, int item)
    {
            NODE prev,cur,temp;

              temp=getnode();
    temp->rlink=NULL;
    temp->llink=NULL;
     temp->info=item;
      if (root==NULL)
            return temp;
      prev=NULL;
      cur=root;
      if (cur!=NULL)
          {
            prev=cur;

          cur=(item<cur->info)->
                              cur->llink:cur->
                                          rlink;
          }

      if (item<prev->info)
            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  postorder ( NODE root )
         {   if  (root != NULL)
              {
                ~~printf ("%d\n~~", postorder (root->llin
                             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);
      }
   }
int main()
{
 int item, choice;
    NODE root=NULL;
 for(;;)
 {
 printf(""\n1-Insert \n 2-display \n 3-pre\n
            4-post \n 5-in \n 6-Exit\n");
   printf("Enter choice\n");
     scanf("%d", &choice);
     switch (choice)
      {
       case1: printf("Enter item\n");
              scanf("%d", &item);
                root=insert(root,item);
                    break;
       Case 2:  display(root,0);
                break;
      case3:
              preorder(root);
               break;
    Case4:
            postorder(root);
              break;
    case 5:
            inorder(root);
              break;
    default:
            exit(0);
            break;
          }
        }
      }
```