Write a program a) To construct a binary Search tree. b) To traverse the tree using all the methods i.e., in-order, preorder and post order c) To display the elements in the tree

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

```
while(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)
 {

  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);
  }
 }
int 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\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:exit(0);
                 break;
  default: continue;
```

```
            }
        }
    }
```

```
1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
12

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
10

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
14

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
8

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
11

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
2
```

```
  14
12
    11
  10
    8

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
13

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
1

enter the item
20

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
2

    20
  14
    13
12
    11
  10
    8

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
3

12
10
8
11
14
13
20

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
4
```

```
8
11
10
13
20
14
12

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
5

8
10
11
12
13
14
20

1.insert
2.display
3.pre
4.post
5.in
6.exit
enter the choice
6
```