main.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.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;
```

main.c

```
37    while(cur!=NULL)
38    {
39       prev=cur;
40       cur=(item<cur->info)?cur->llink:cur->rlink;
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  void preorder(NODE root)
61  {
62     if(root!=NULL)
63     {
64         printf("%d\n",root->info);
65         preorder(root->llink);
66         preorder(root->rlink);
67     }
68   }
69  void postorder(NODE root)
70  {
71     if(root!=NULL)
72     {
73         postorder(root->llink);
74         postorder(root->rlink);
75         printf("%d\n",root->info);
```

main.c

```
        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("\n1.insert\n2.preorder\n3.postorder\n4.inorder\n5.display\n6.exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        printf("------\n");
        switch(choice)
        {
            case 1:printf("enter the item\n");
                scanf("%d",&item);
                root=insert(root,item);
                break;
            case 2:preorder(root);
                break;
            case 3:postorder(root);
                break;
            case 4:inorder(root);
                break;
            case 5:display(root,0);
```

main.c

```c
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("\n1.insert\n2.preorder\n3.postorder\n4.inorder\n5.display\n6.exit\n");
        printf("enter the choice\n");
        scanf("%d",&choice);
        printf("------\n");
        switch(choice)
        {
            case 1:printf("enter the item\n");
                   scanf("%d",&item);
                   root=insert(root,item);
                   break;
            case 2:preorder(root);
                   break;
            case 3:postorder(root);
                   break;
            case 4:inorder(root);
                   break;
            case 5:display(root,0);
                   break;
            default:exit(0);
                   break;
        }
    }
}
```

input

```
1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
100

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
20

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
200
```

```
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
10

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
30

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
150

1.insert
2.preorder
3.postorder
4.inorder
```

input

```
3.postorder
4.inorder
5.display
6.exit
enter the choice
1
------
enter the item
300

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
2
------
100
20
10
30
200
150
300

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
3
------
10
```

```
4.inorder
5.display
6.exit
enter the choice
3
------
10
30
20
150
300
200
100

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
4
------
10
20
30
100
150
200
300

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
```

```
100

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
4
------
10
20
30
100
150
200
300

1.insert
2.preorder
3.postorder
4.inorder
5.display
6.exit
enter the choice
5
------
    300
  200
    150
100
    30
  20
    10

1.insert
```