```
X lab_3.c X lab_10.c X
   #include<stdio.h>
   #include<stdlib.h>
   #include <comio.h>
  struct node *llink;
    struct node *rlink;
    int info;
   typedef struct node *NODE;
  □NODE getnode(){
   NODE X;
   x=(NODE) malloc(sizeof(struct node));
  if (x==NULL) [
    printf("Memory full\n");
    exit(0);
   return x;
   NODE insert (NODE root, int item)
   NODE temp, curr, prev;
   temp=getnode();
   temp->rlink=temp->llink=NULL;
   temp->info=item;
   if (root=NULL)
   return temp;
   prev=NULL;
   curr=root;
   while (curr!=NULL)
   prev=curr;
   curr=(item<curr->info)?curr->llink:curr->rlink;
   if(item<prev->info)
   prev->llink=temp;
   else
   prev->rlink=temp;
   return root;

| void preorder (NODE root) (
| if (root=NULL) return;
```

```
ere X lab_3.c X lab_10.c X
      if(root==NULL) return;
      printf("%d ",root->info);
42
     preorder(root->llink);
43
      preorder (root->rlink);
44
45
   □void postorder(NODE root) {
46
47
      if(root==NULL) return;
      postorder (root->llink);
48
      postorder (root->rlink);
49
50
      printf("%d ",root->info);
51
52
    □void inorder(NODE root) (
53
      if (root=NULL) return;
54
      inorder (root->llink);
      printf("%d ", root->info);
55
56
      inorder(root->rlink);
57
58
     void display(NODE root, int i)
   ⊟(
|int j;
59
60
     if (root!=NULL)
61
62
      display(root->rlink, i+1);
63
      for(j=0;j<i;j++)
printf(" ");</pre>
64
65
66
      printf("%d\n", root->info);
     display(root->llink, i+1);
67
68
69
70
     int main()
71
72
     int item, choice;
73
     NODE root=NULL;
74
75
76
     printf("\n1.insert\t2.display\t3.preorder\t4.postorder\t5.inorder\t6.exit\n");
     printf("enter the choice\n");
77
78
     scanf ("%d", &choice);
79
     switch (choice)
80
81
   case 1:printf("enter the item\n");
```

```
& others
Code::Blocks 🗴 🔍 Search results 🗴 📝 Cccc 🗴 🛟 Build log 🗴 🥀 Build messages 🗴
here X lab_3.c X lab_10.c X
57
       void display (NODE root, int i)
58
59
       int j;
 60
      if (root!=NULL)
61
62
        display(root->rlink,i+1);
63
64
        for (j=0; j<i; j++)
        printf(" ");
65
        printf("%d\n", root->info);
66
       display(root->llink, i+1);
67
68
69
       int main()
70
71
     ⊟(
       int item, choice;
72
73
       NODE root=NULL;
74
       for(;;)
75
       printf("\n1.insert\t2.display\t3.preorder\t4.postorder\t5.inorder\t6.exit\n");
76
77
       printf("enter the choice\n");
       scanf ("%d", &choice);
78
       switch (choice)
79
80
        case 1:printf("enter the item\n");
81
       scanf ("%d", &item);
82
83
       root=insert(root, item);
84
       break;
85
        case 2:display(root, 0);
86
       break;
87
        case 3:preorder(root);
88
       break;
89
        case 4:postorder (root);
90
       break;
91
        case 5:inorder(root);
92
        default: printf("wrong choice.THANK YOU.."); exit(1);
93
94
95
     [i]
96
```

1.insert 2.display enter the choice 1 enter the item	3.preorder	4.postorder	5.inorder	6.exit
1.insert 2.display enter the choice	3.preorder	4.postorder	5.inorder	6.exit
enter the item 31 1.insert 2.display	3.preorder	4.postorder	5.inorder	6.exit
enter the choice 2 31	J.pr.coruci	Tiposcor del	3121101 001	
1.insert 2.display enter the choice 1 enter the item 33	3.preorder	4.postorder	5.inorder	6.exit
1.insert 2.display enter the choice 2 33 31	3.preorder	4.postorder	5.inorder	6.exit
1.insert 2.display enter the choice 1 enter the item 35	3.preorder	4.po storder	5.inorder	6.exit
1.insert 2.display enter the choice 2 35 33 31 30	3.preorder	4.postorder	5.inorder	6.exit
1.insert 2.display enter the choice 3 30 31 33 35	3.preorder	4.postorder	5.inorder	6.exit

1.insert 2 enter the choice 1 enter the item 33	.display	3.preorder	4.postorder	5.inorder	6.exit	
1.insert 2 enter the choice 2 33 31	.display	3.preorder	4.postorder	5.inorder	6.exit	
1.insert 2 enter the choice 1 enter the item 35	.display	3.preorder	4.postorder	5.inorder	6.exit	
1.insert 2 enter the choice 2 35 33 31	.display	3.preorder	4.postorder	5.inorder	6.exit	
1.insert 2 enter the choice 3 30 31 33 35	.display	3.preorder	4.postorder	5.inorder	6.exit	
	.display	3.preorder	4.postorder	5.inorder	6.exit	
	.displ ay	3.preorder	4.postorder	5.inorder	6.exit	
	.display	3.preorder	4.postorder	5.inorder	6.exit	
Process returned : Press any key to	1 (0x1) exec	ution time : 310	.186 s			