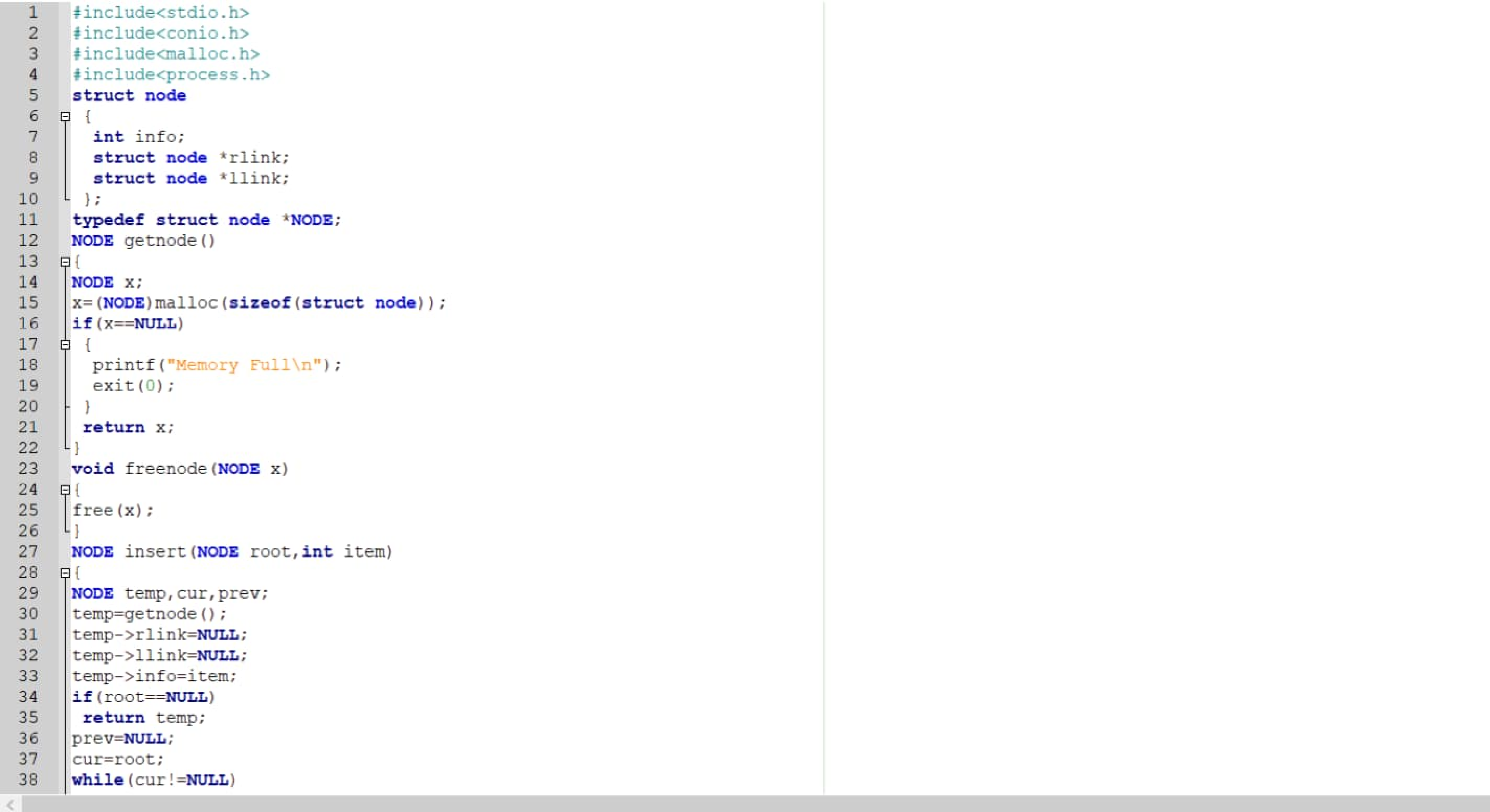
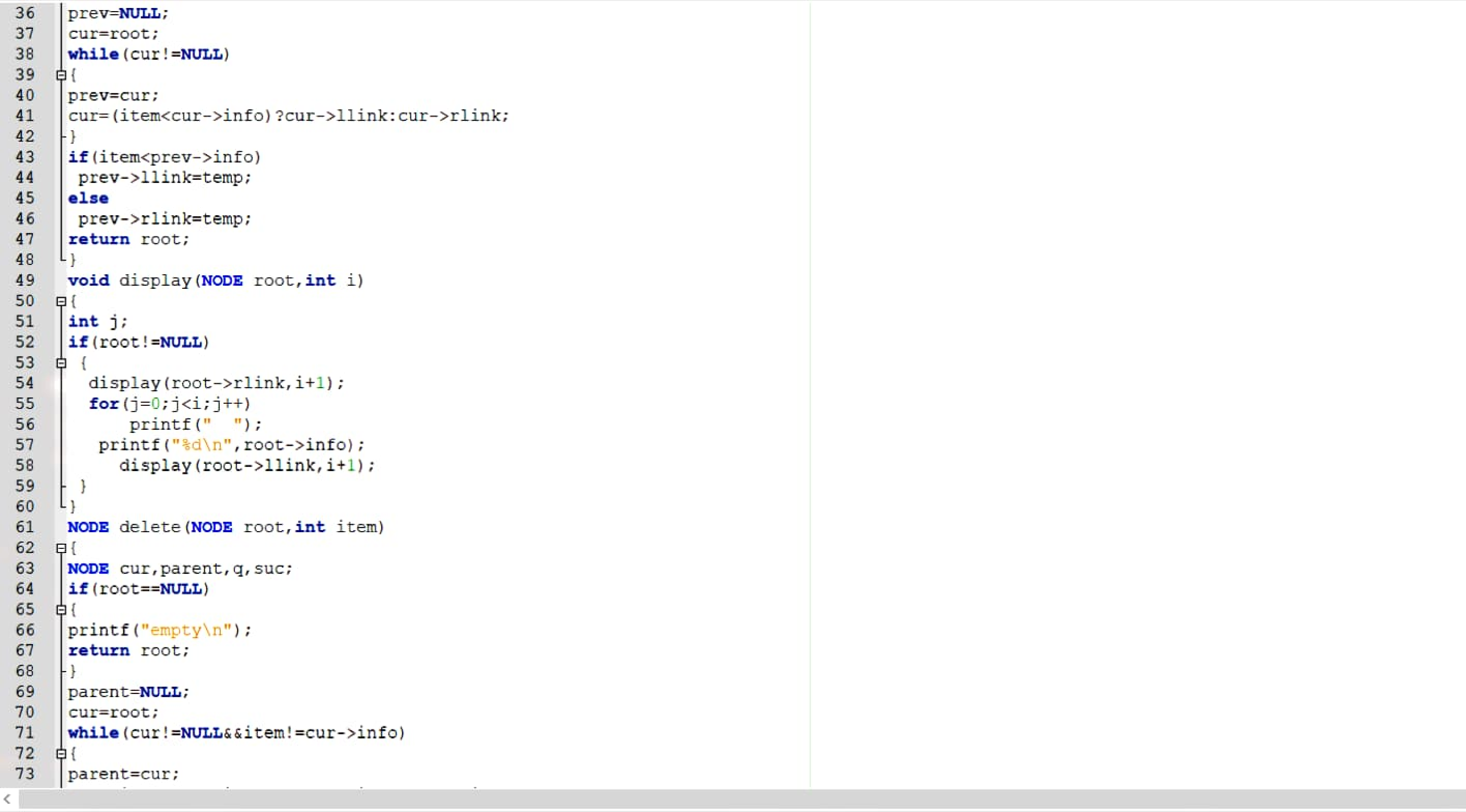
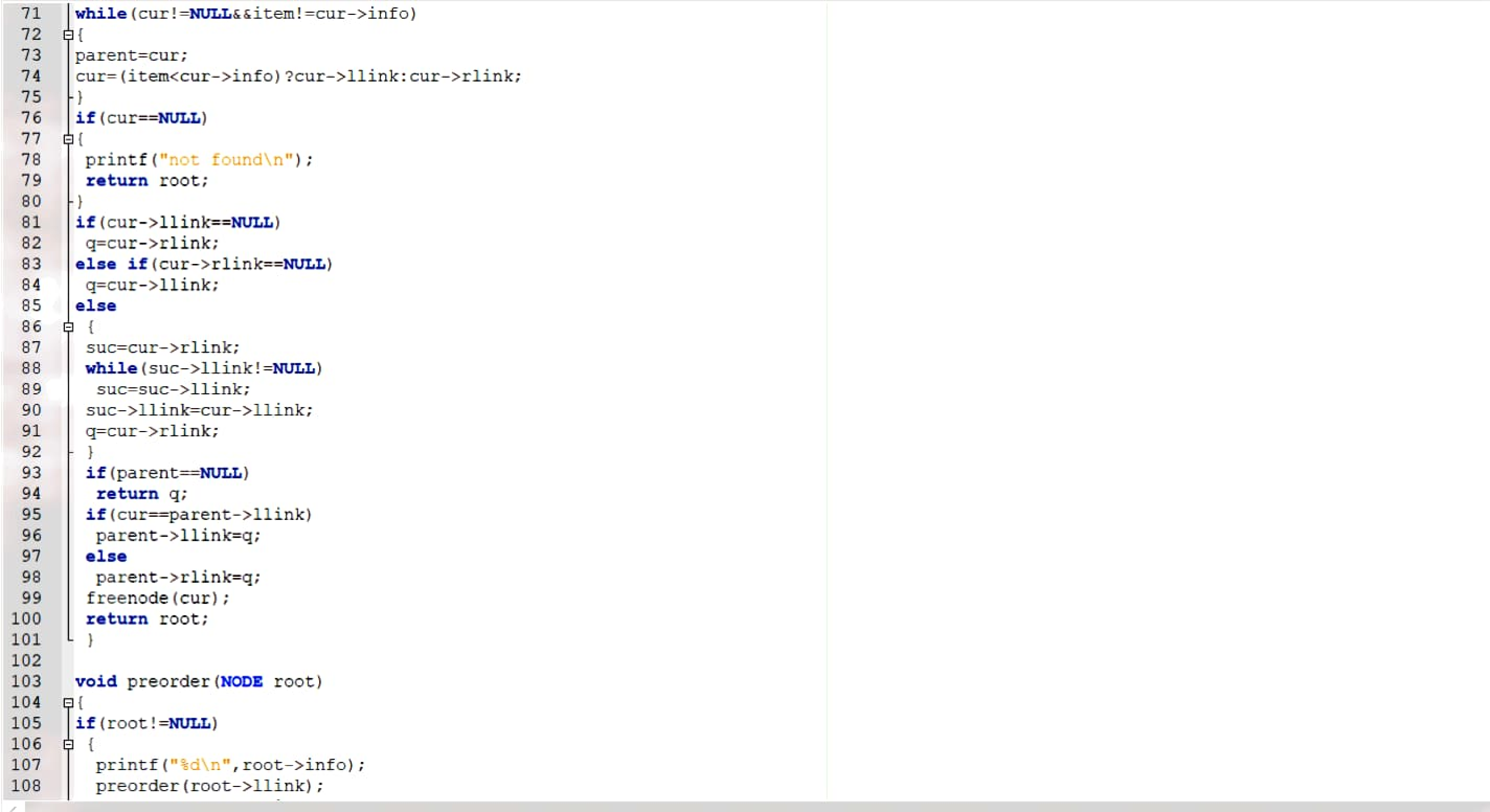


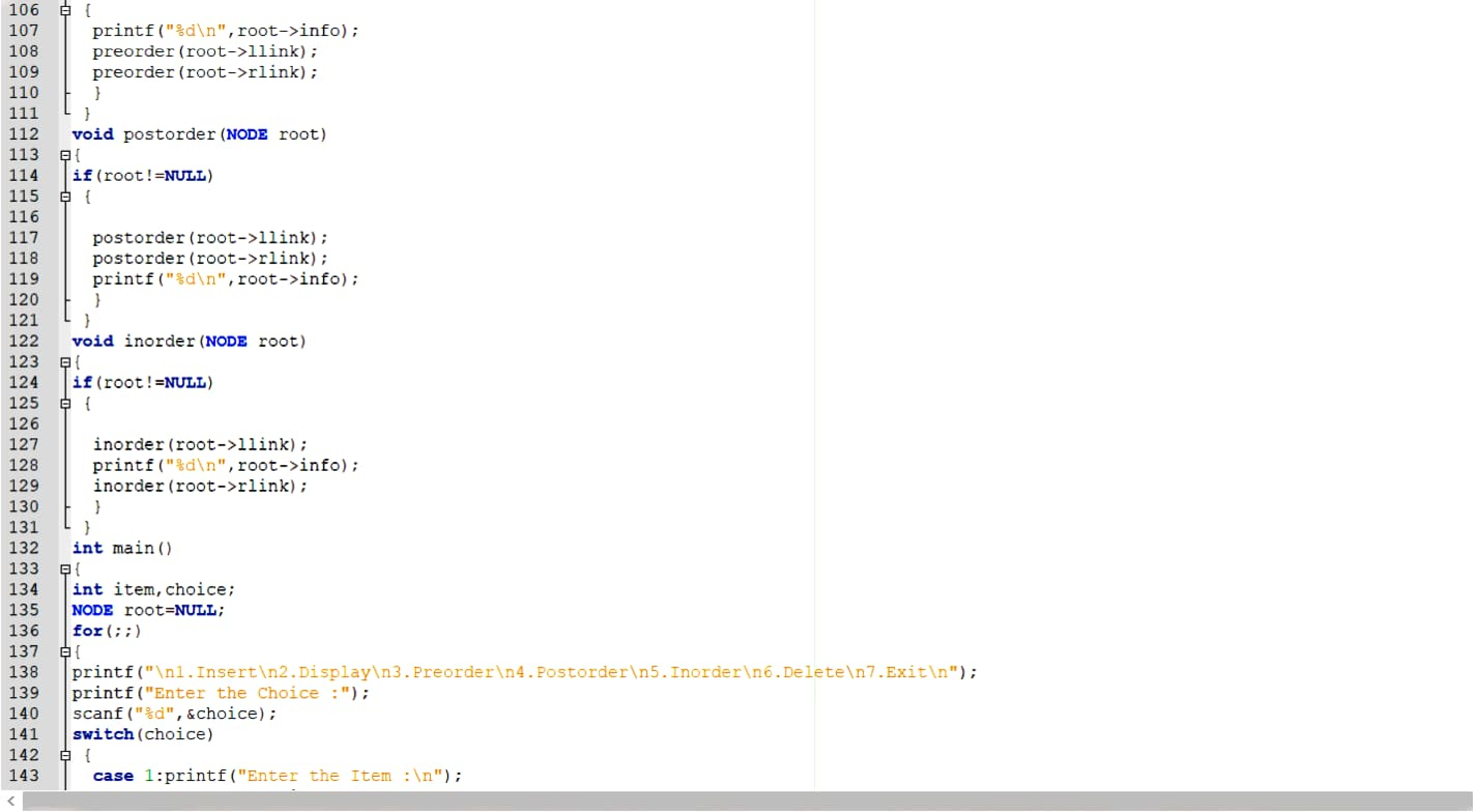
```c
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<process.h>
struct node
 {
  int info;
  struct node *rlink;
  struct node *llink;
 };
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 {
  printf("Memory Full\n");
  exit(0);
 }
 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)
```



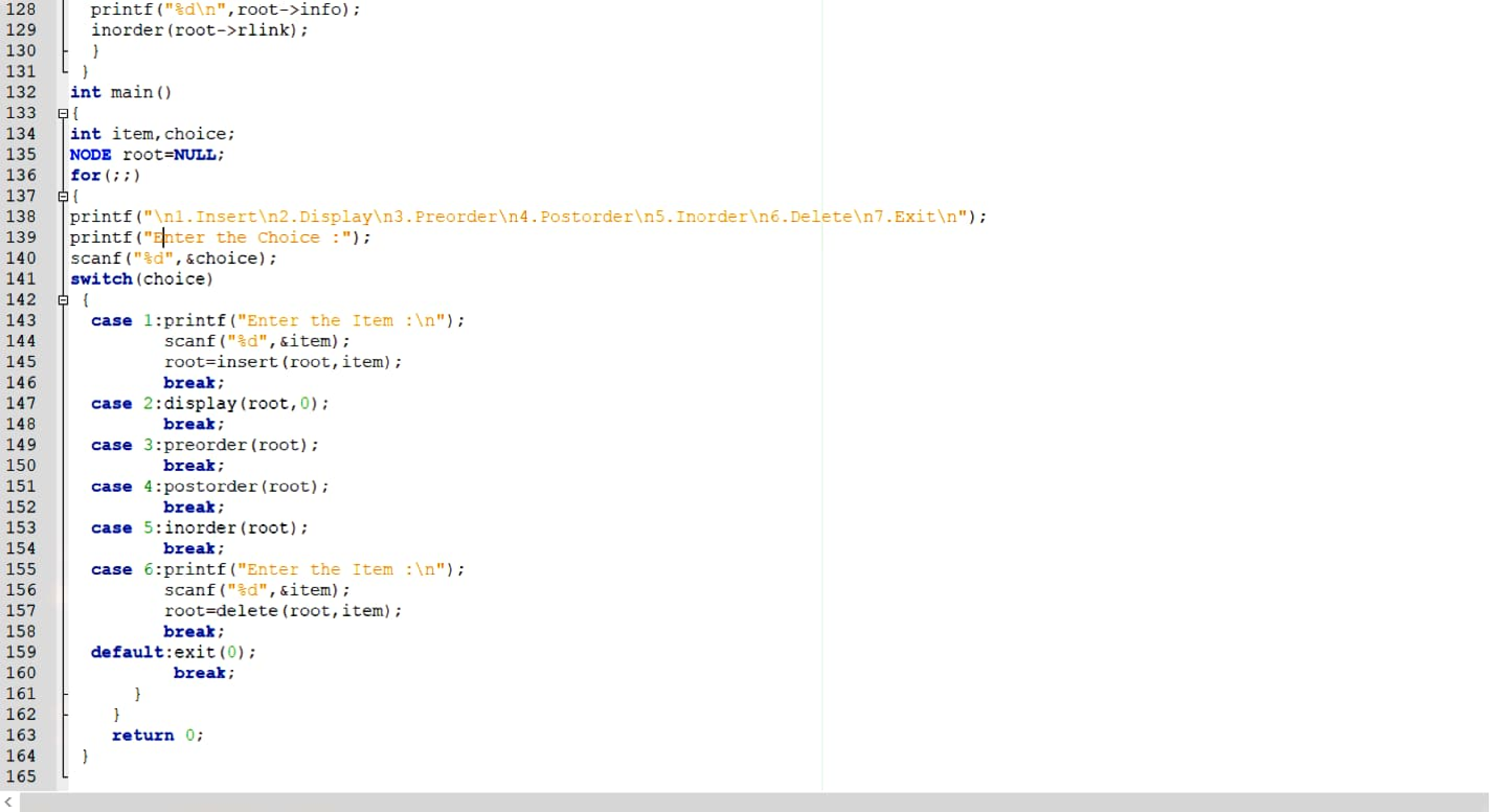
```
prev=NULL;
cur=root;
while(cur!=NULL)
{
prev=cur;
cur=(item<cur->info)?cur->llink:cur->rlink;
}
if(item<prev->info)
 prev->llink=temp;
else
 prev->rlink=temp;
return root;
}
void display(NODE root,int i)
{
int j;
if(root!=NULL)
 {
  display(root->rlink,i+1);
  for(j=0;j<i;j++)
      printf("  ");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



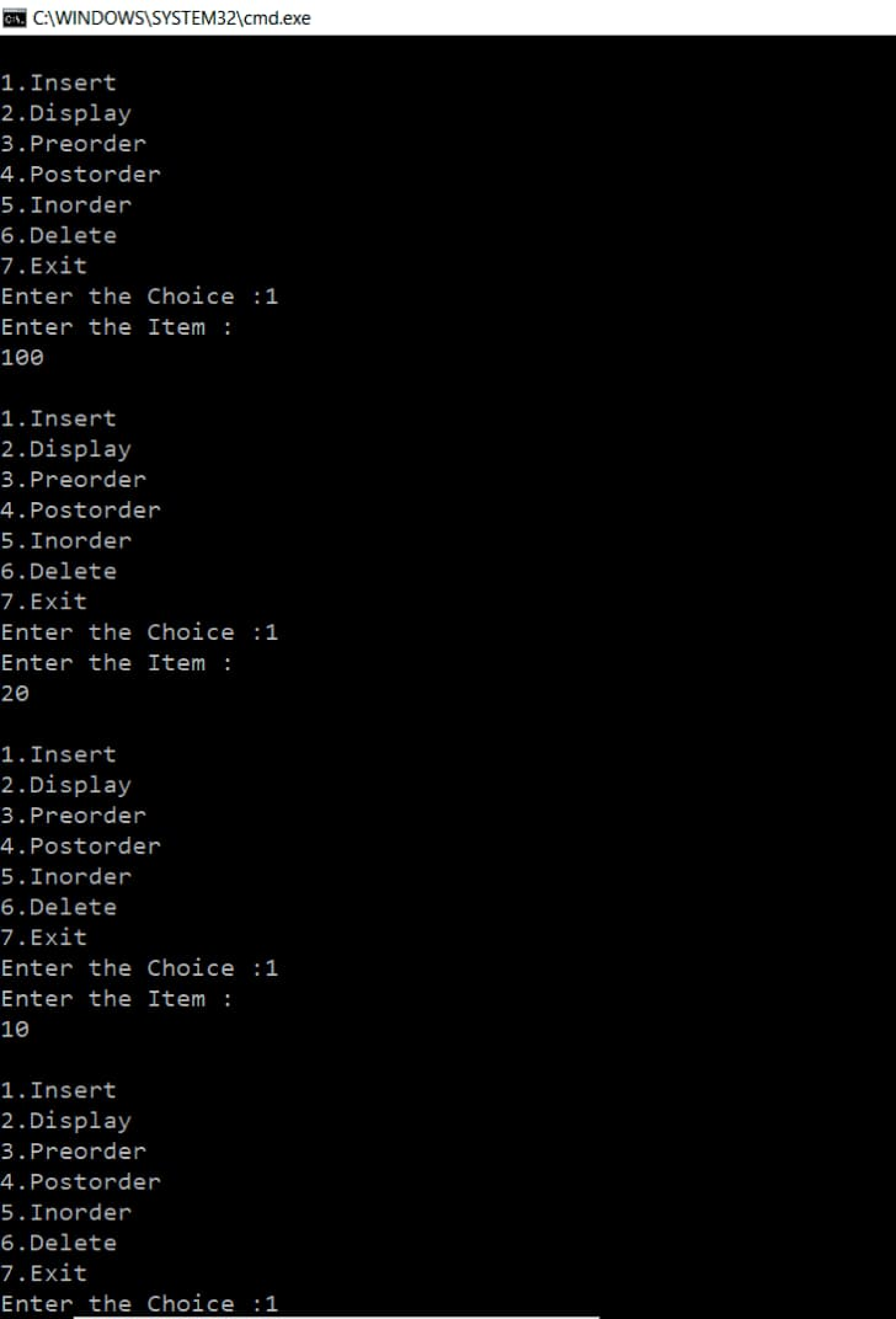
```
while(cur!=NULL&&item!=cur->info)
{
parent=cur;
cur=(item<cur->info)?cur->llink:cur->rlink;
}
if(cur==NULL)
{
 printf("not found\n");
 return root;
}
if(cur->llink==NULL)
 q=cur->rlink;
else if(cur->rlink==NULL)
 q=cur->llink;
else
 {
 suc=cur->rlink;
 while(suc->llink!=NULL)
  suc=suc->llink;
 suc->llink=cur->llink;
 q=cur->rlink;
 }
 if(parent==NULL)
  return q;
 if(cur==parent->llink)
  parent->llink=q;
 else
  parent->rlink=q;
 freenode(cur);
 return root;
 }

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



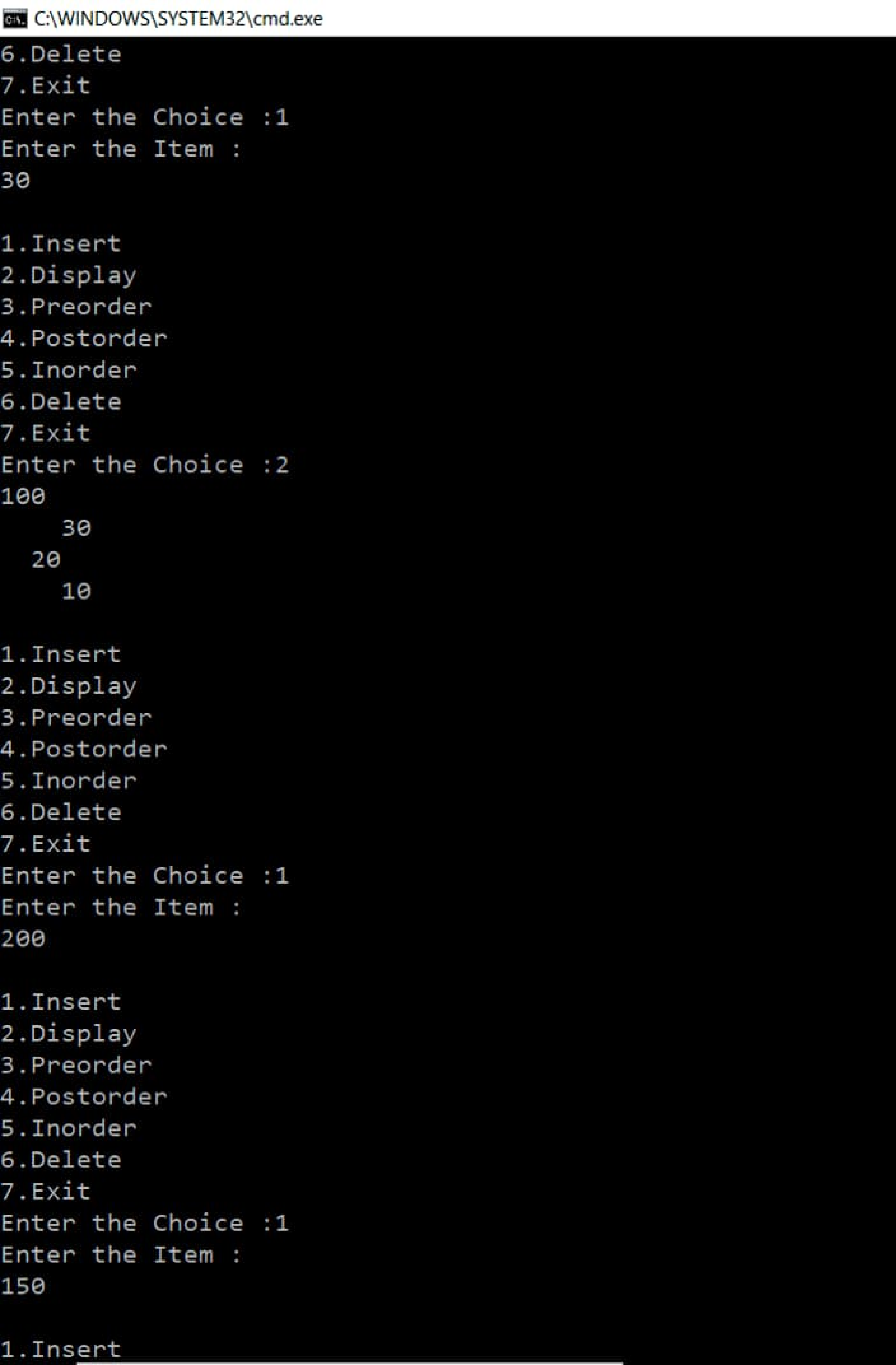
```
 {
  printf("%d\n",root->info);
  preorder(root->llink);
  preorder(root->rlink);
  }
 }
void postorder(NODE root)
{
if(root!=NULL)
 {

  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.Preorder\n4.Postorder\n5.Inorder\n6.Delete\n7.Exit\n");
printf("Enter the Choice :");
scanf("%d",&choice);
switch(choice)
 {
  case 1:printf("Enter the Item :\n");
```



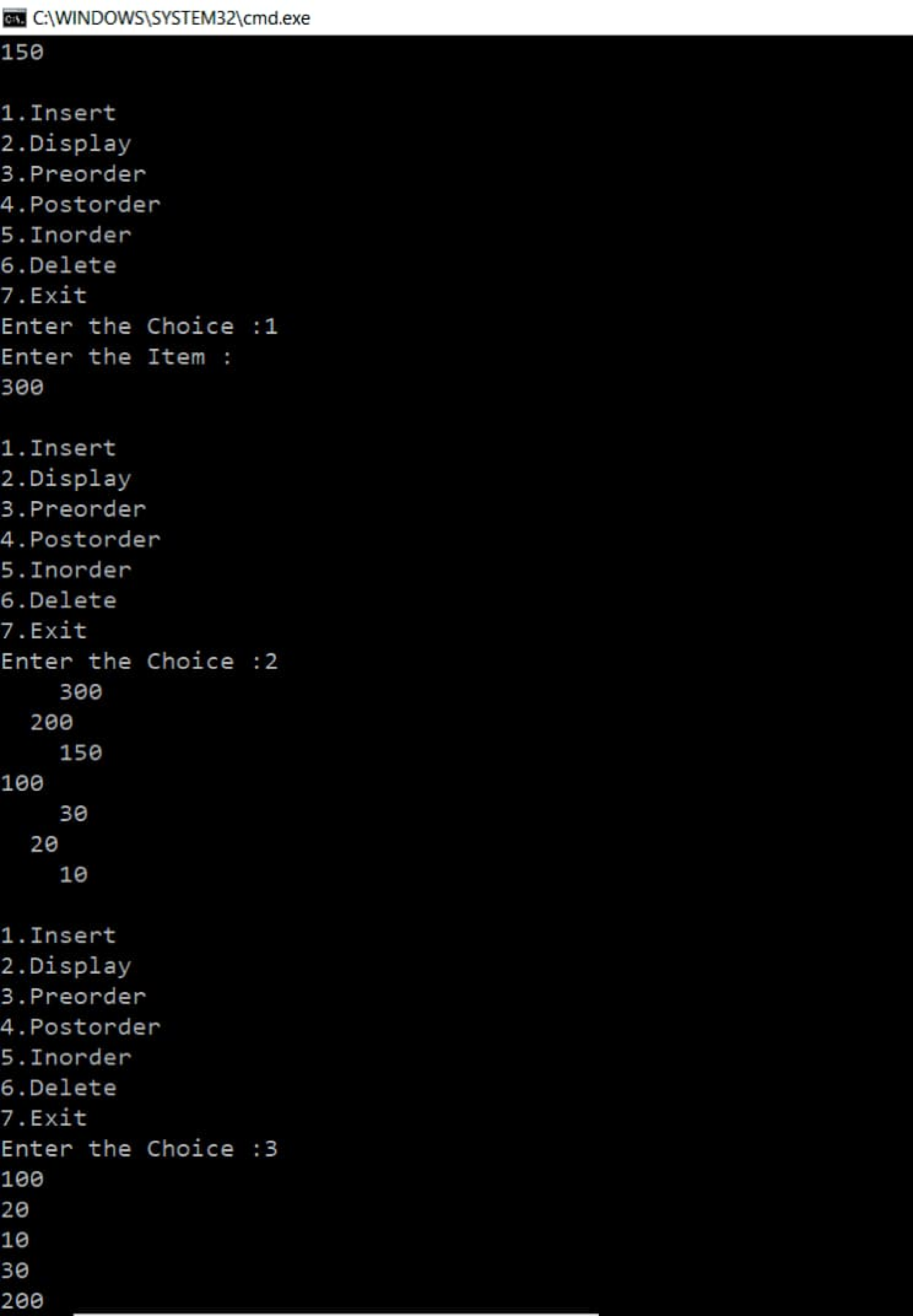
```
  printf("%d\n",root->info);
  inorder(root->rlink);
  }
 }
int main()
{
int item,choice;
NODE root=NULL;
for(;;)
{
printf("\n1.Insert\n2.Display\n3.Preorder\n4.Postorder\n5.Inorder\n6.Delete\n7.Exit\n");
printf("Enter the Choice :");
scanf("%d",&choice);
switch(choice)
 {
  case 1:printf("Enter 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:printf("Enter the Item :\n");
         scanf("%d",&item);
         root=delete(root,item);
         break;
  default:exit(0);
          break;
      }
    }
    return 0;
 }
```



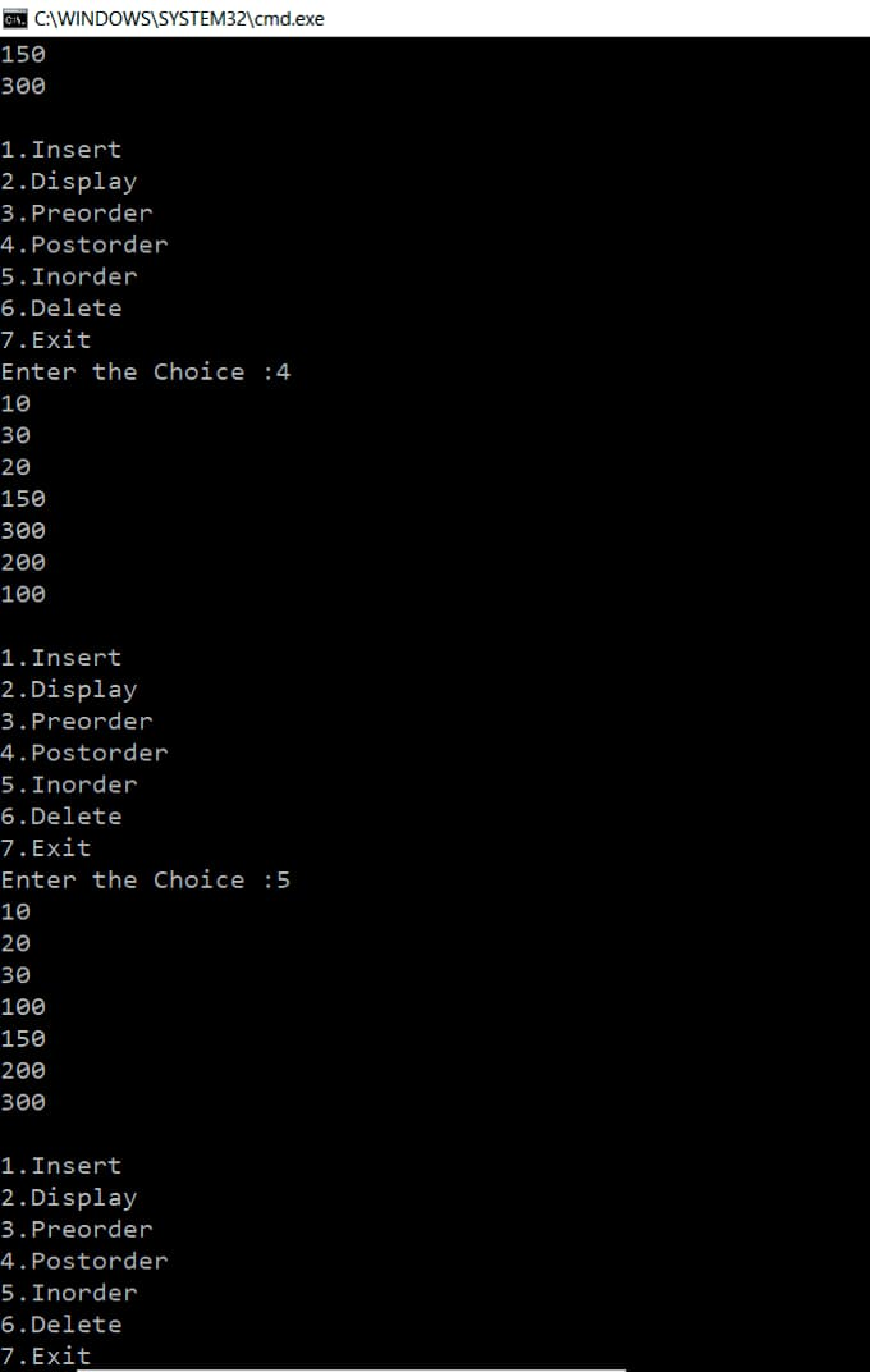
```
1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
100

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
20

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
10

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
```



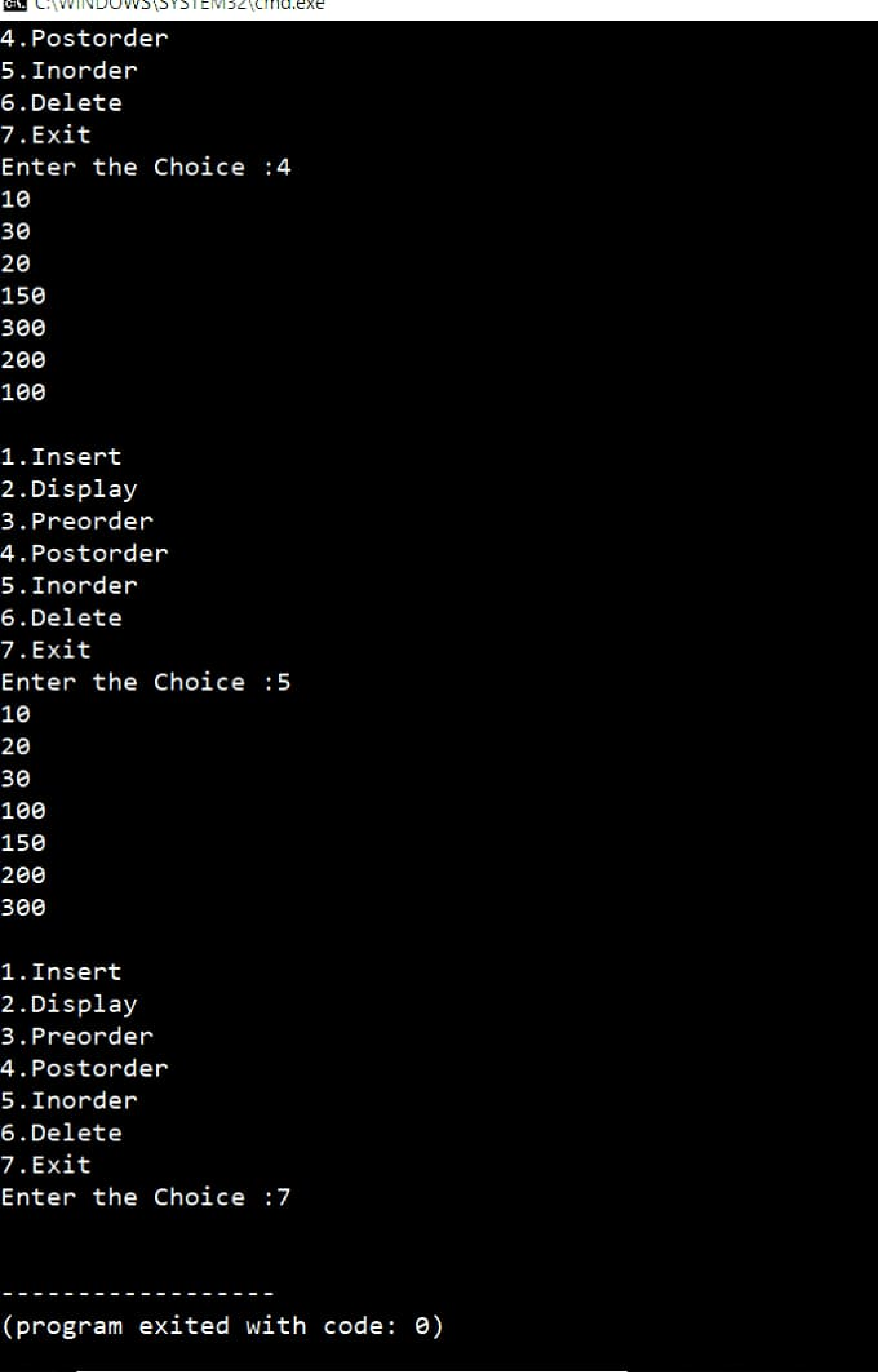
```
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
30

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :2
100
    30
  20
    10

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
200

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
150

1.Insert
```



```
150

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :1
Enter the Item :
300

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :2
    300
  200
    150
100
    30
  20
    10

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :3
100
20
10
30
200
```



```
150
300

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :4
10
30
20
150
300
200
100

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :5
10
20
30
100
150
200
300

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
```



```
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :4
10
30
20
150
300
200
100

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :5
10
20
30
100
150
200
300

1.Insert
2.Display
3.Preorder
4.Postorder
5.Inorder
6.Delete
7.Exit
Enter the Choice :7



------------------
(program exited with code: 0)
```