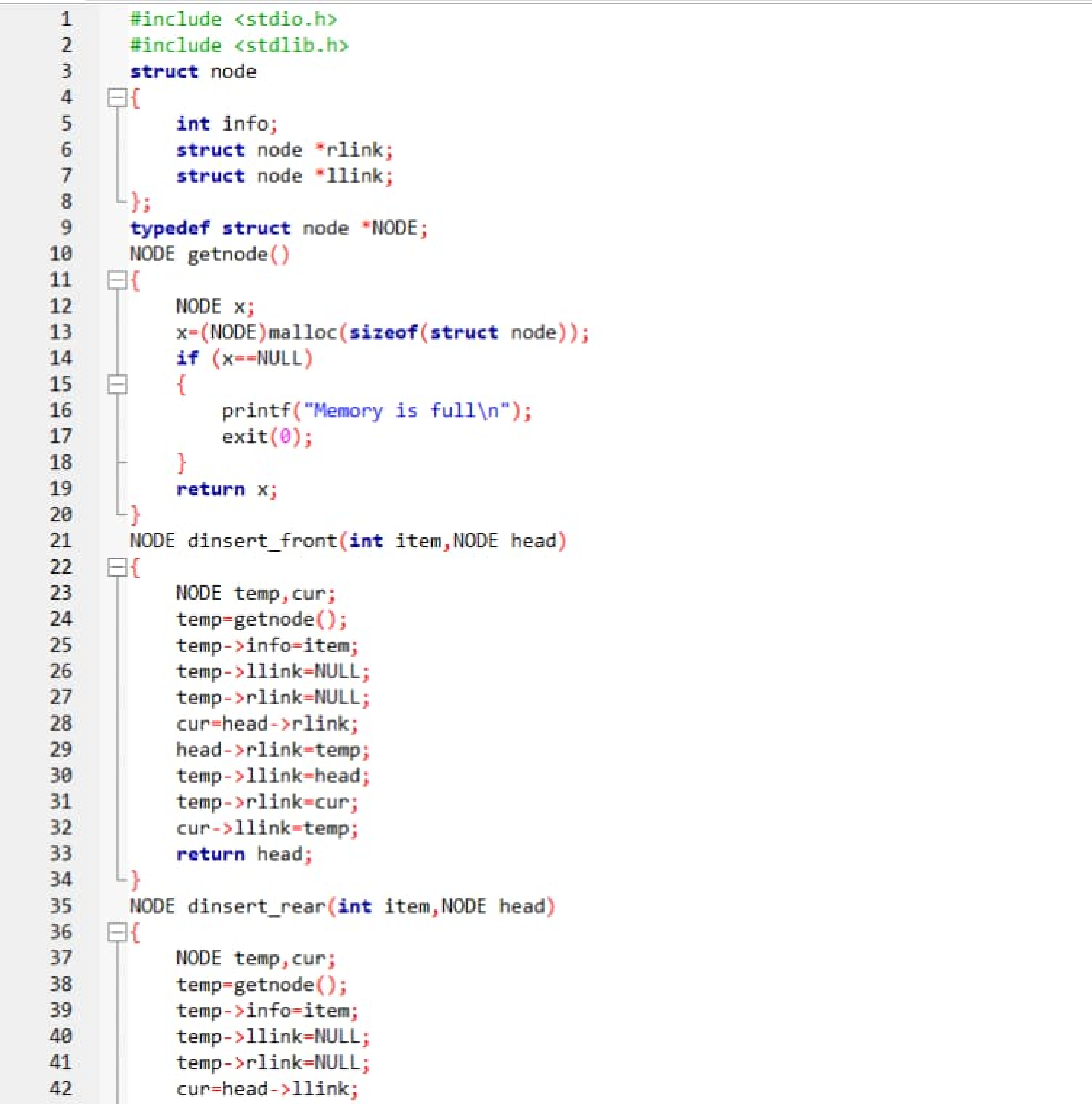
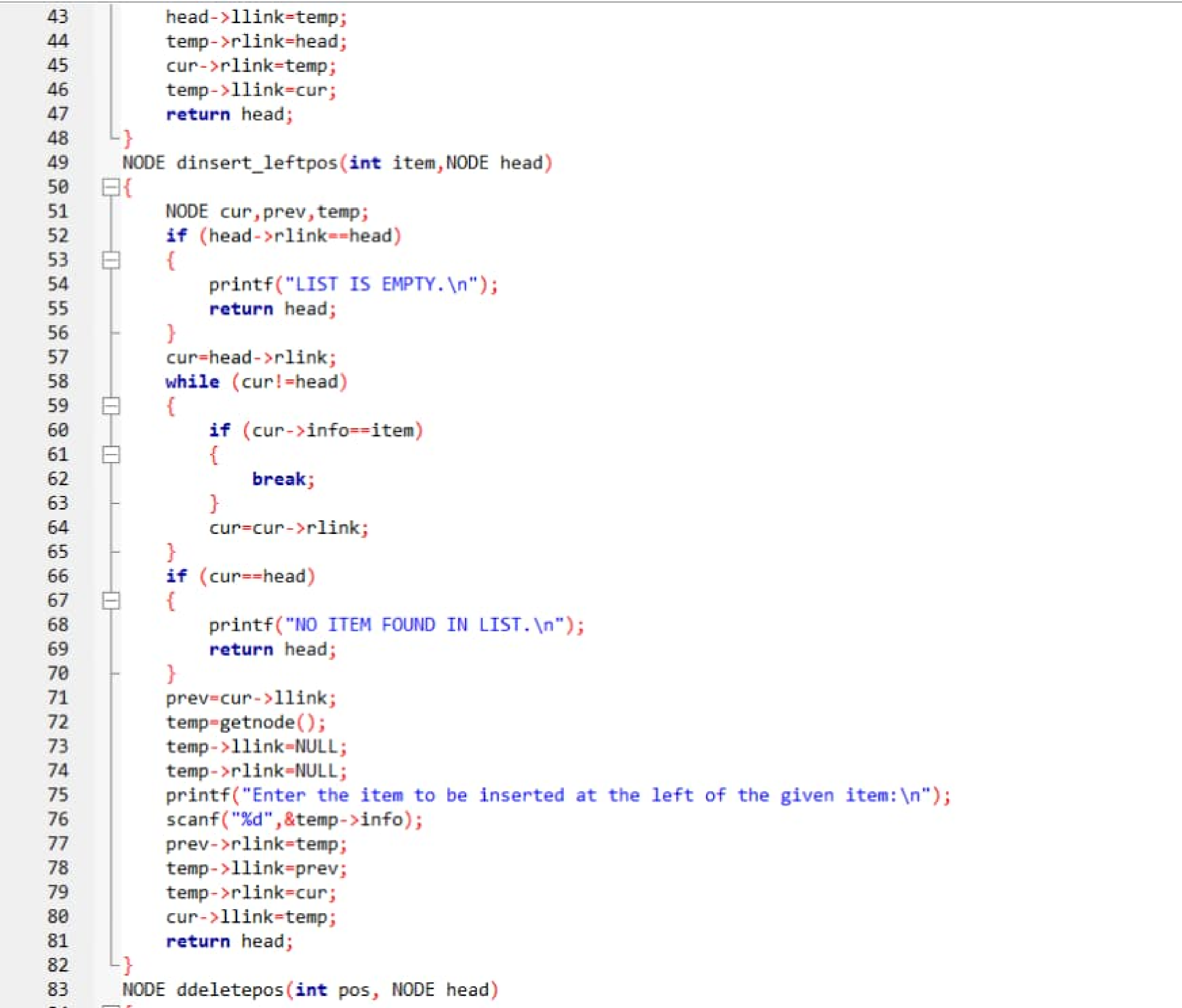
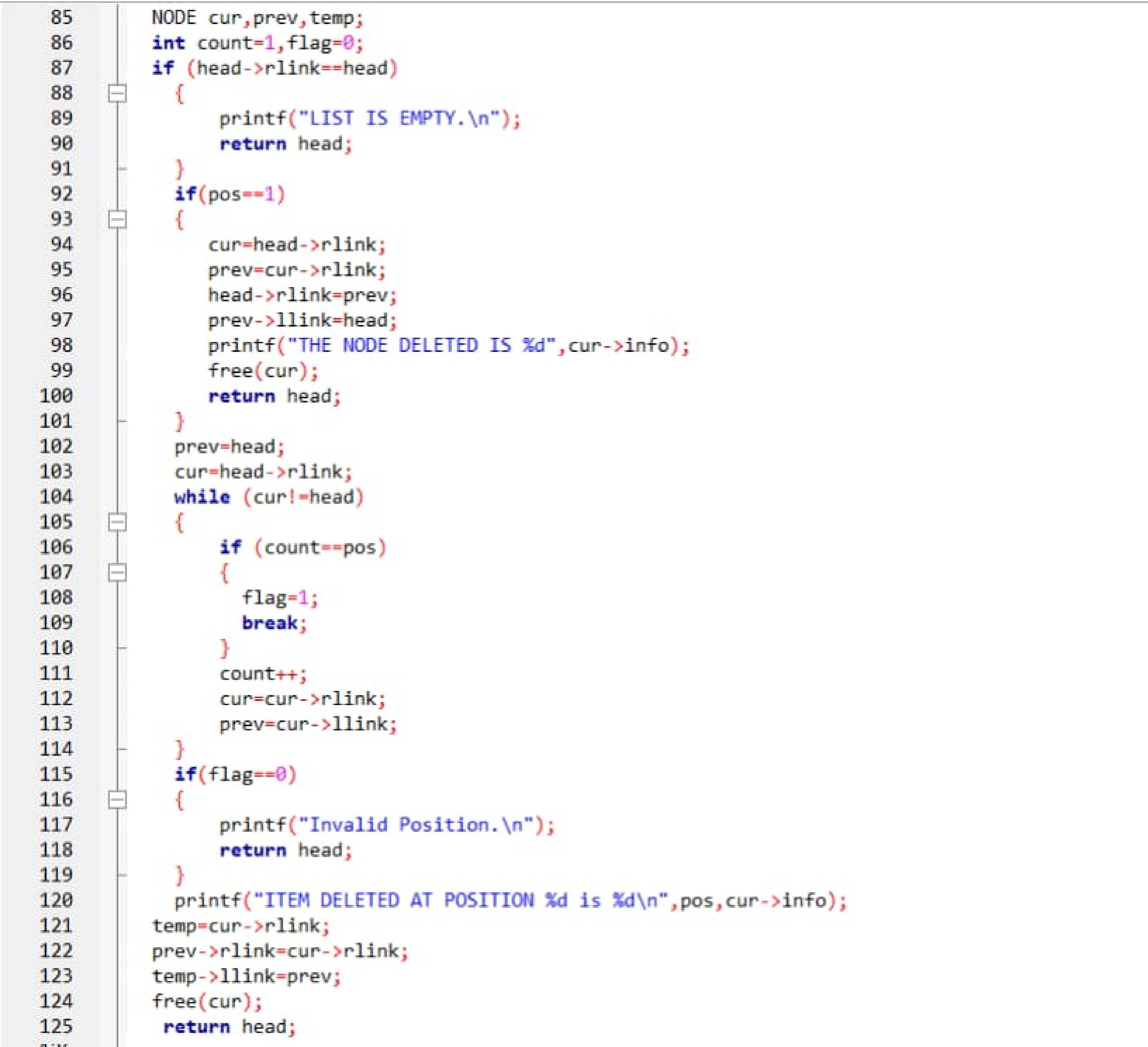


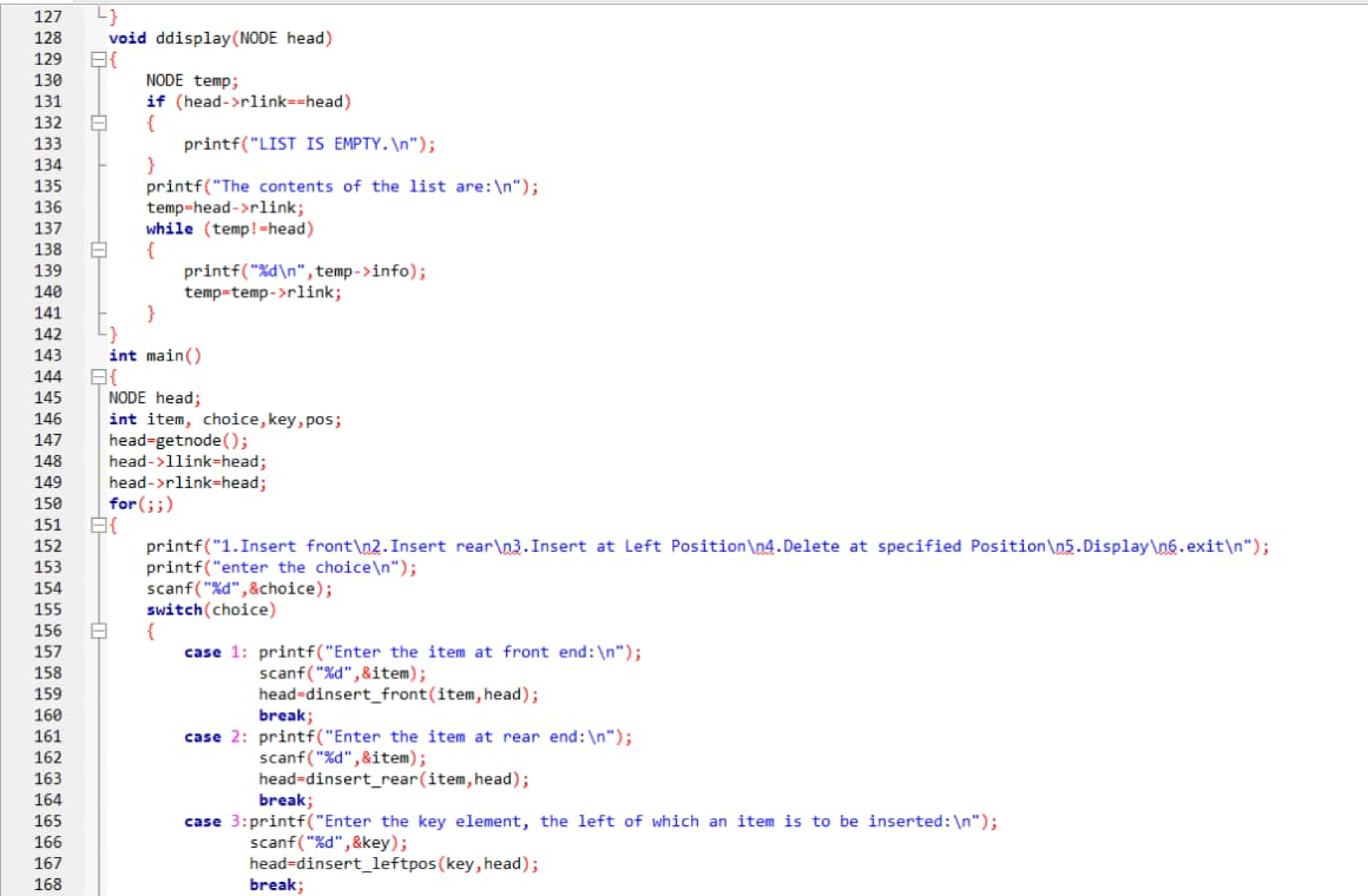
```c
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int info;
    struct node *rlink;
    struct node *llink;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if (x==NULL)
    {
        printf("Memory is full\n");
        exit(0);
    }
    return x;
}
NODE dinsert_front(int item,NODE head)
{
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->llink=NULL;
    temp->rlink=NULL;
    cur=head->rlink;
    head->rlink=temp;
    temp->llink=head;
    temp->rlink=cur;
    cur->llink=temp;
    return head;
}
NODE dinsert_rear(int item,NODE head)
{
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->llink=NULL;
    temp->rlink=NULL;
    cur=head->llink;
```



```
    head->llink=temp;
    temp->rlink=head;
    cur->rlink=temp;
    temp->llink=cur;
    return head;
}
NODE dinsert_leftpos(int item,NODE head)
{
    NODE cur,prev,temp;
    if (head->rlink==head)
    {
        printf("LIST IS EMPTY.\n");
        return head;
    }
    cur=head->rlink;
    while (cur!=head)
    {
        if (cur->info==item)
        {
            break;
        }
        cur=cur->rlink;
    }
    if (cur==head)
    {
        printf("NO ITEM FOUND IN LIST.\n");
        return head;
    }
    prev=cur->llink;
    temp=getnode();
    temp->llink=NULL;
    temp->rlink=NULL;
    printf("Enter the item to be inserted at the left of the given item:\n");
    scanf("%d",&temp->info);
    prev->rlink=temp;
    temp->llink=prev;
    temp->rlink=cur;
    cur->llink=temp;
    return head;
}
NODE ddeletepos(int pos, NODE head)
```

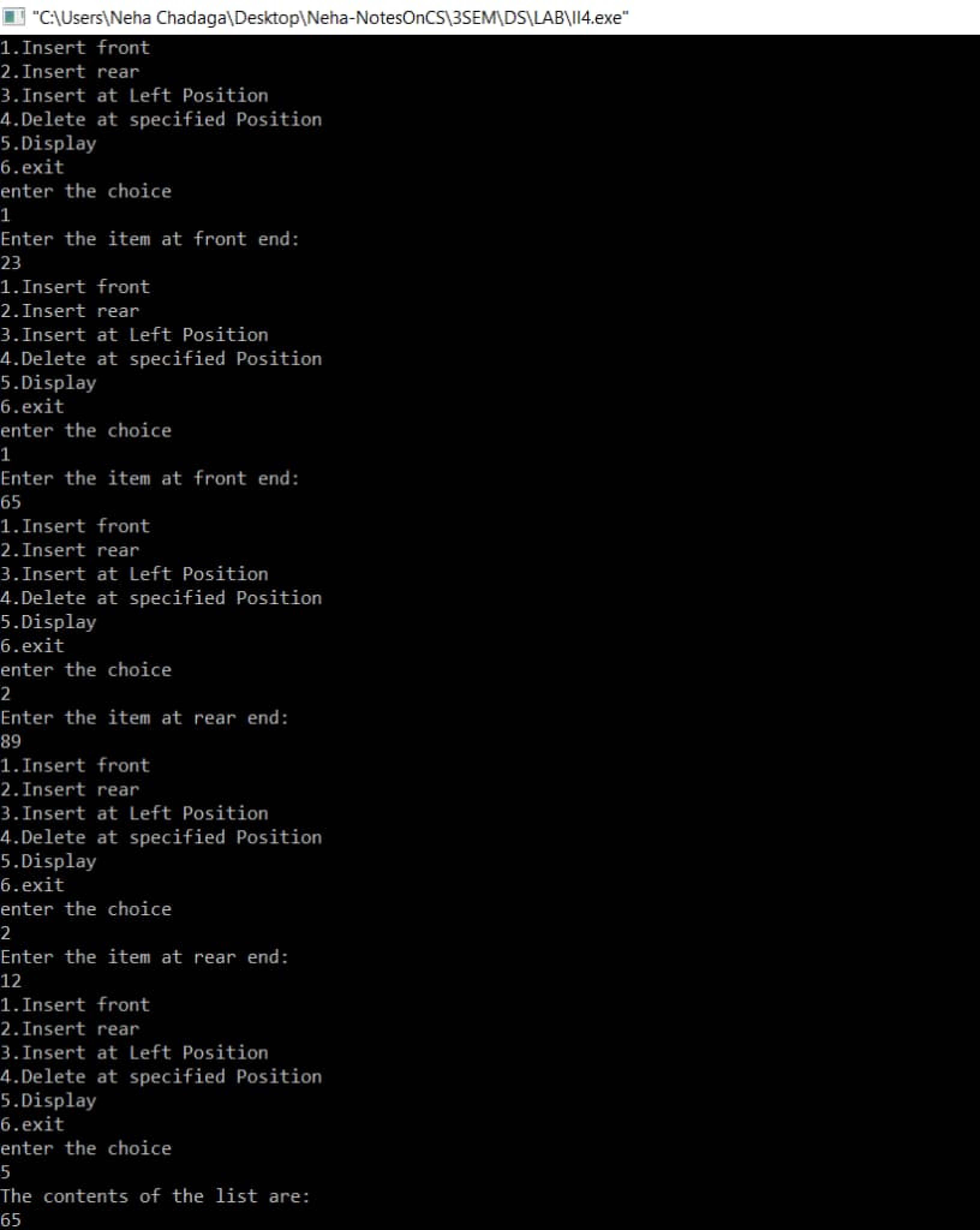


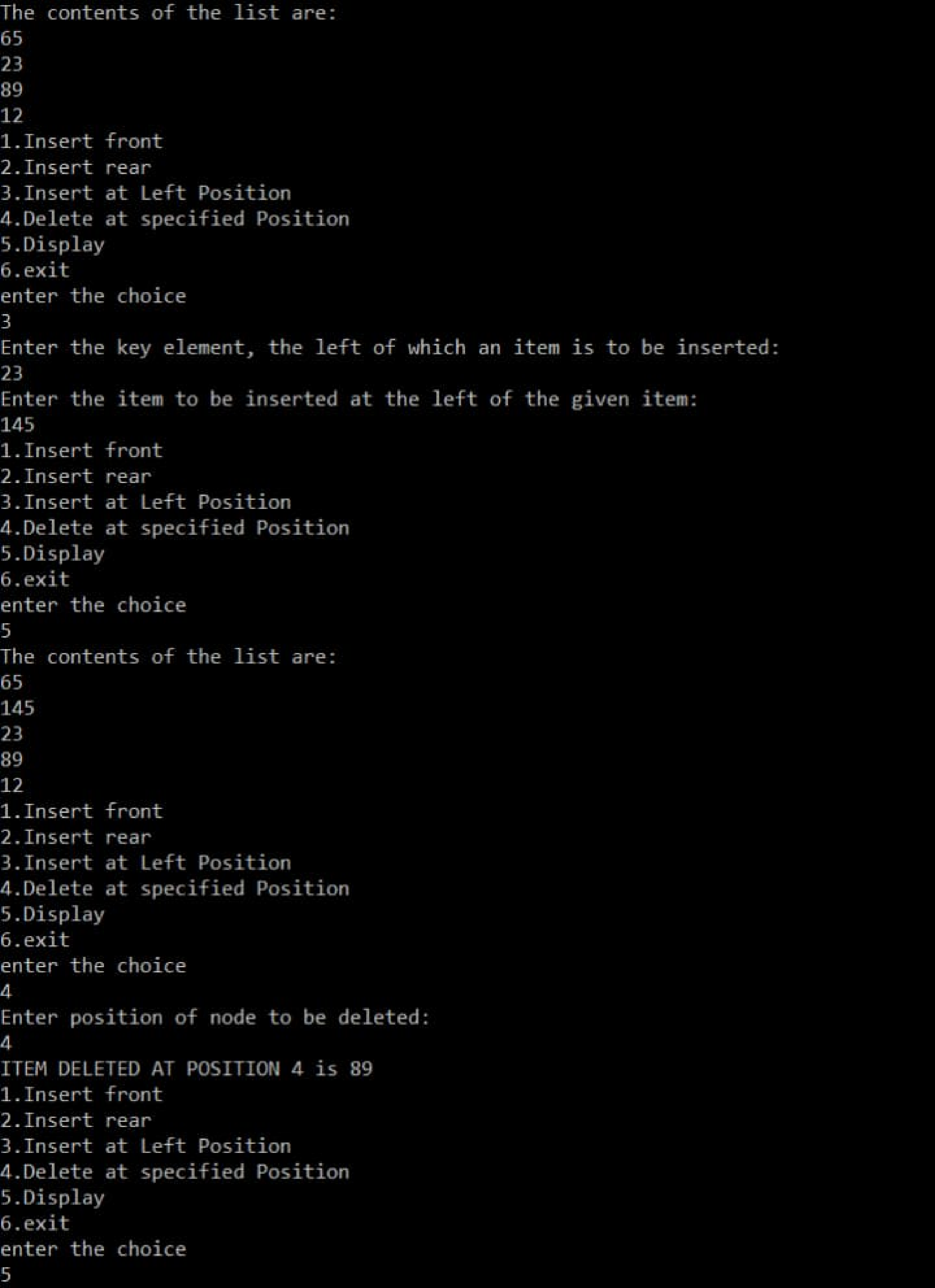
```c
NODE cur,prev,temp;
int count=1,flag=0;
if (head->rlink==head)
  {
      printf("LIST IS EMPTY.\n");
      return head;
  }
  if(pos==1)
  {
     cur=head->rlink;
     prev=cur->rlink;
     head->rlink=prev;
     prev->llink=head;
     printf("THE NODE DELETED IS %d",cur->info);
     free(cur);
     return head;
  }
  prev=head;
  cur=head->rlink;
  while (cur!=head)
  {
      if (count==pos)
      {
        flag=1;
        break;
      }
      count++;
      cur=cur->rlink;
      prev=cur->llink;
  }
  if(flag==0)
  {
      printf("Invalid Position.\n");
      return head;
  }
  printf("ITEM DELETED AT POSITION %d is %d\n",pos,cur->info);
temp=cur->rlink;
prev->rlink=cur->rlink;
temp->llink=prev;
free(cur);
 return head;
```



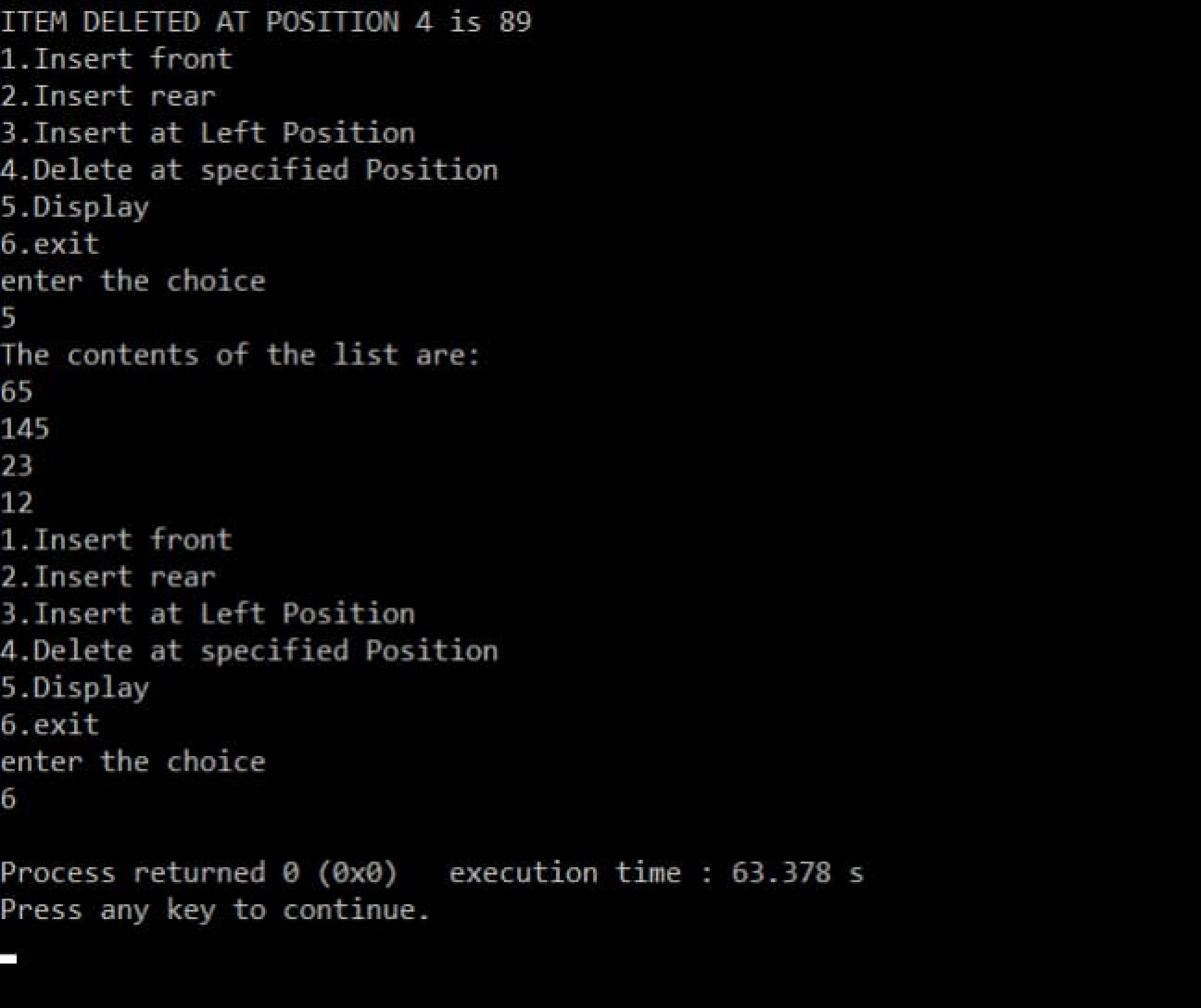
```c
}
void ddisplay(NODE head)
{
    NODE temp;
    if (head->rlink==head)
    {
        printf("LIST IS EMPTY.\n");
    }
    printf("The contents of the list are:\n");
    temp=head->rlink;
    while (temp!=head)
    {
        printf("%d\n",temp->info);
        temp=temp->rlink;
    }
}
int main()
{
NODE head;
int item, choice,key,pos;
head=getnode();
head->llink=head;
head->rlink=head;
for(;;)
{
    printf("1.Insert front\n2.Insert rear\n3.Insert at Left Position\n4.Delete at specified Position\n5.Display\n6.exit\n");
    printf("enter the choice\n");
    scanf("%d",&choice);
    switch(choice)
    {
        case 1: printf("Enter the item at front end:\n");
                scanf("%d",&item);
                head=dinsert_front(item,head);
                break;
        case 2: printf("Enter the item at rear end:\n");
                scanf("%d",&item);
                head=dinsert_rear(item,head);
                break;
        case 3:printf("Enter the key element, the left of which an item is to be inserted:\n");
               scanf("%d",&key);
               head=dinsert_leftpos(key,head);
               break;
```

```
                break;
        case 3:printf("Enter the key element, the left of which an item is to be inserted:\n");
               scanf("%d",&key);
               head=dinsert_leftpos(key,head);
               break;
        case 4:printf("Enter position of node to be deleted:\n");
               scanf("%d",&pos);
               head=ddeletepos(pos,head);
               break;
        case 5:ddisplay(head);
               break;
        default:exit(0);
        }
    }
    return 0;
}
```


```
"C:\Users\Neha Chadaga\Desktop\Neha-NotesOnCS\3SEM\DS\LAB\ll4.exe"
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
1
Enter the item at front end:
23
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
1
Enter the item at front end:
65
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
2
Enter the item at rear end:
89
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
2
Enter the item at rear end:
12
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
5
The contents of the list are:
65
```



```
The contents of the list are:
65
23
89
12
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
3
Enter the key element, the left of which an item is to be inserted:
23
Enter the item to be inserted at the left of the given item:
145
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
5
The contents of the list are:
65
145
23
89
12
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
4
Enter position of node to be deleted:
4
ITEM DELETED AT POSITION 4 is 89
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
5
```



```
ITEM DELETED AT POSITION 4 is 89
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
5
The contents of the list are:
65
145
23
12
1.Insert front
2.Insert rear
3.Insert at Left Position
4.Delete at specified Position
5.Display
6.exit
enter the choice
6

Process returned 0 (0x0)   execution time : 63.378 s
Press any key to continue.
```