```c
/*WAP Implement Single Link List with following operations
a) a) Sort the linked list. b) Reverse the linked list. c) Concatenation of two linked lists*/
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<process.h>
struct node
{
int info;
struct node *link;
};
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;
}
NODE insert_rear(NODE first,int item)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
cur=first;
while(cur->link!=NULL)
cur=cur->link;
cur->link=temp;
```

```
while(cur->link!=NULL)
cur=cur->link;
cur->link=temp;
return first;
}
void display(NODE first)
{
NODE temp;
if(first==NULL)
printf("list empty");
for(temp=first;temp!=NULL;temp=temp->link)
{
printf("%d\n",temp->info);
}
}
NODE concat(NODE first,NODE second)
{
NODE cur;
if(first==NULL)
return second;
if(second==NULL)
return first;
cur=first;
while(cur->link!=NULL)
cur=cur->link;
cur->link=second;
return first;
}
NODE reverse(NODE first)
{
NODE cur,temp;
cur=NULL;
while(first!=NULL)
{
temp=first;
```

```
NODE cur,temp;
cur=NULL;
while(first!=NULL)
{
temp=first;
first=first->link;
temp->link=cur;
cur=temp;
}
return cur;
}
NODE order_list(int item,NODE first)
{
NODE temp,prev,cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL) return temp;
if(item<first->info)
{
temp->link=first;
return temp;
}
prev=NULL;
cur=first;
while(cur!=NULL&&item>cur->info)
{
prev=cur;
cur=cur->link;
}
prev->link=temp;
temp->link=cur;
return first;
}
int main()
```

```
94  temp->link=cur;
95  return first;
96  }
97  int main()
98  {
99  int item,choice,i,n;
100 NODE first=NULL,a,b;
101 for(;;)
102 {
103 printf("1.insert_front\n2.concat\n3.reverse\n4.order list\n5.dislay\n6.exit\n");
104 printf("enter the choice\n");
105 scanf("%d",&choice);
106 switch(choice)
107 {
108 case 1:printf("enter the item\n");
109 scanf("%d",&item);
110 first=insert_rear(first,item);
111 break;
112 case 2:printf("enter the no of nodes in 1\n");
113 scanf("%d",&n);
114 a=NULL;
115 for(i=0;i<n;i++)
116 {
117 printf("enter the item\n");
118 scanf("%d",&item);
119 a=insert_rear(a,item);
120 }
121 printf("enter the no of nodes in 2\n");
122 scanf("%d",&n);
123 b=NULL;
124 for(i=0;i<n;i++)
125 {
126 printf("enter the item\n");
127 scanf("%d",&item);
128 b=insert_rear(b,item);
```

```
scanf("%d",&n);
a=NULL;
for(i=0;i<n;i++)
{
printf("enter the item\n");
scanf("%d",&item);
a=insert_rear(a,item);
}
printf("enter the no of nodes in 2\n");
scanf("%d",&n);
b=NULL;
for(i=0;i<n;i++)
{
printf("enter the item\n");
scanf("%d",&item);
b=insert_rear(b,item);
}
a=concat(a,b);
display(a);
break;
case 3:first=reverse(first);
display(first);
break;
case 4:printf("enter the item to be inserted in ordered_list\n");
scanf("%d",&item);
first=order_list(item,first);
break;
case 5:display(first);
break;
default:exit(0);
}
}
}
```

```
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
1
enter the item
10
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
1
enter the item
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
5
10
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
2
enter the no of nodes in 1
2
enter the item
```

```
5.dislay
6.exit
enter the choice
2
enter the no of nodes in 1
2
enter the item
11
enter the item
22
enter the no of nodes in 2
3
enter the item
33
enter the item
44
enter the item
55
11
22
33
44
55
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
5
10
20
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
1
```

```
3.reverse
4.order list
5.dislay
6.exit
enter the choice
1
enter the item
30
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
1
enter the item
40
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
5
10
20
30
40
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
3
40
30
20
```

```
5.dislay
6.exit
enter the choice
3
40
30
20
10
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
4
enter the item to be inserted in ordered_list
11
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
4
enter the item to be inserted in ordered_list
54
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
4
enter the item to be inserted in ordered_list
23
1.insert_front
2.concat
3.reverse
```

```
6.exit
enter the choice
4
enter the item to be inserted in ordered_list
23
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
4
enter the item to be inserted in ordered_list
41
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
5
11
23
40
30
20
10
41
54
1.insert_front
2.concat
3.reverse
4.order list
5.dislay
6.exit
enter the choice
5
11
23
```