

```

1  #include <stdio.h>
2  #include <malloc.h>
3  #include <stdlib.h>
4  struct Node
5  {
6      int data;
7      struct Node *next;
8  };
9
10 void Insert (struct Node *l, int x);
11 void Destory (struct Node *l);
12 void Print (struct Node *l);
13 struct Node * Back(struct Node* l);
14 struct Node * NewNode()
15 {
16     struct Node *p;
17     p = (struct Node *) malloc (sizeof (struct Node));
18     if (p == NULL) {
19         printf ("Error : out of memory\n");
20         exit (-1);
21     }
22     return p;
23 }
24
25 int main ()
26 {
27     struct Node *la = NewNode();//正整数的链表
28     struct Node *lb = NewNode();//负整数的链表
29
30     la->next = NULL;
31     lb->next = NULL;
32     int x;
33     printf("请输入数字，以0结束，以空格或回车间隔");
34     scanf ("%d", &x);
35     while(x!=0){
36         if(x>0){
37             Insert(la, x);
38         }else{
39             Insert(lb, x);
40         }
41         scanf ("%d", &x);
42     }
43     Print(la);
44     printf("逆置为: ");
45     Print(Back(la));
46     Print(lb);
47     printf("逆置为: ");
48     Print(Back(lb));
49
50     Destory(la);
51     Destory(lb);
52     return 0;
53 }
54 void Insert (struct Node *l, int x)
55 {

```

```

56     struct Node *q= NewNode ();
57     q->data = x;
58     struct Node *p = l;
59     while (p->next && x > p->next ->data)
60         p = p->next;
61     q->next = p ->next ;
62     p->next = q;
63 }
64 void Destory (struct Node *l)
65 {
66     while (l)
67     {
68         struct Node *q = l->next;
69         free (l);
70         l = q;
71     }
72 }
73 void Print (struct Node *l)
74 {
75     l = l ->next;
76     if (l)
77     {
78         printf ("%d", l->data);
79         l = l->next;
80     }
81     while (l)
82     {
83         printf ("->%d", l->data);
84         l = l->next;
85     }
86     printf ("\n");
87 }
88 struct Node * Back(struct Node* l){
89     struct Node *p;
90     struct Node *q;
91     p=l->next;
92     l->next=NULL;
93     while(p){
94         q=p;
95         p=p->next;
96         q->next=l->next;
97         l->next=q;
98     }
99     return l;
100
101 }

```

```

PS D:\csjjg\程序设计综合实践> cd "d:\csjjg\程序设计综合实践\" ; if ($?) { gcc third.c -o third } ; if ($?) { .\third }
请输入数字, 以0结束,以空格或回车间隔1 2 3 4 5 32 322 -21 -2 -1 0
1->2->3->4->5->32->322
逆置为: 322->32->5->4->3->2->1
-21->-2->-1
逆置为: -1->-2->-21

```