



软件技术基础实验报告

文献阅读：软件技术基础教程（周肆清）

姓名：	你的名字
学院：	信息科学与工程学院
专业：	20 级电子信息类二班
学号：	你的校园卡号
老师：	你的老师

2021 年 4 月 27 日

A 题目 1（学时 2）：

创建一个由 6 个结点的单向链表，实现增加、删除、查找、移动、显示结点的基本功能。

Listing 1: 题目 1

```
1 #pragma warning(disable:4996)
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <malloc.h>
5 #define NULL 0
6 typedef struct ListNode {
7     int data;
8     struct ListNode* next;
9 } NODE;
10 /* 用尾插法建立带头结点的单链表*/
11 NODE* CreateList(int n) {
12     NODE* head, * p, * q;
13     int num;
14     int i = 1;
15     head = (NODE*)malloc(sizeof(NODE));
16     head->next = NULL;
17     p = head;
18     while (i <= n) {
19         printf("please input data:\n");
20         scanf("%d", &num);
21         q = (NODE*)malloc(sizeof(NODE));
22         q->data = num;
23         p->next = q; p = q;
24         i++;
25     }
26     p->next = NULL;
27     return (head);
28 }
29 /* 输出链表*/
30 void PrintList(NODE* head) {
31     NODE* p;
32     p = head->next;
33     printf("Linear List: ");
34     while (p != NULL) { printf("%d ", p->data); p = p->next; }
```

```
35     printf("\n");
36 }
37 /* 单链表查找节点*/
38 int FoundList(NODE* head, int x) {
39     NODE* p;
40     int pos = 1;
41     p = head->next;
42     while ((p != NULL) && (p->data != x)) { p = p->next; pos++; }
43     if (p != NULL) return (pos); else return (0);
44 }
45 /* 插入新结点（第i 个位置上插入） */
46 void InsertList(NODE* head, int x, int i)
47 {
48     NODE* p, * s;
49     int j = 0;
50     p = head;
51     while ((p != NULL) && (j < i - 1))
52     {
53         p = p->next;
54         j++;
55     }
56     if ((p == NULL) || (j > i - 1))
57         printf("\nPosition Error\n");
58     else
59     {
60         s = (NODE*)malloc(sizeof(NODE));
61         s->data = x;
62         s->next = NULL;
63         s->next = p->next;
64         p->next = s;
65         PrintList(head);
66     }
67 }
68 /* 删除第i 个结点*/
69 void DeleteList(NODE* head, int i)
70 {
71     NODE* p, * s;
72     int j = 0;
73     p = head;
74     while ((p->next != NULL) && (j < i - 1))
75     {
76         p = p->next;
```

```
77     j++;
78 }
79 if ((p->next == NULL) || (j > i - 1))
80     printf("\nPostion Error\n");
81 else
82 {
83     s = p->next;
84     p->next = s->next;
85     free(s);
86 }
87 PrintList(head);
88 }
89 /* 求线性链表长度*/
90 int LengthList(NODE* head) {
91     int n;
92     NODE* p;
93     n = 0; p = head->next;
94     while (p != NULL) {
95         p = p->next;
96         n++;
97     }
98     return (n);
99 }
100 void main() {
101     NODE* head;
102     int i, x, l, n;
103     printf("Enter node number for creating: ");
104     scanf("%d", &n); /* n=6*/
105     head = CreateList(n);
106     PrintList(head);
107     printf("Enter data for found: ");
108     scanf("%d", &x);
109     i = FoundList(head, x);
110     if (i == 0) printf("No found node!\n");
111     else printf("Found node at position: %d\n", i);
112     printf("Please input insert location\n");
113     scanf("%d", &i);
114     printf("Please input insert data\n");
115     scanf("%d", &x);
116     InsertList(head, x, i);
117     printf("Please input delete location\n");
118     scanf("%d", &i);
```

```
119 DeleteList(head, i);  
120 l = LengthList(head);  
121 printf("length=%d\n", l);  
122 }
```

B 运行结果

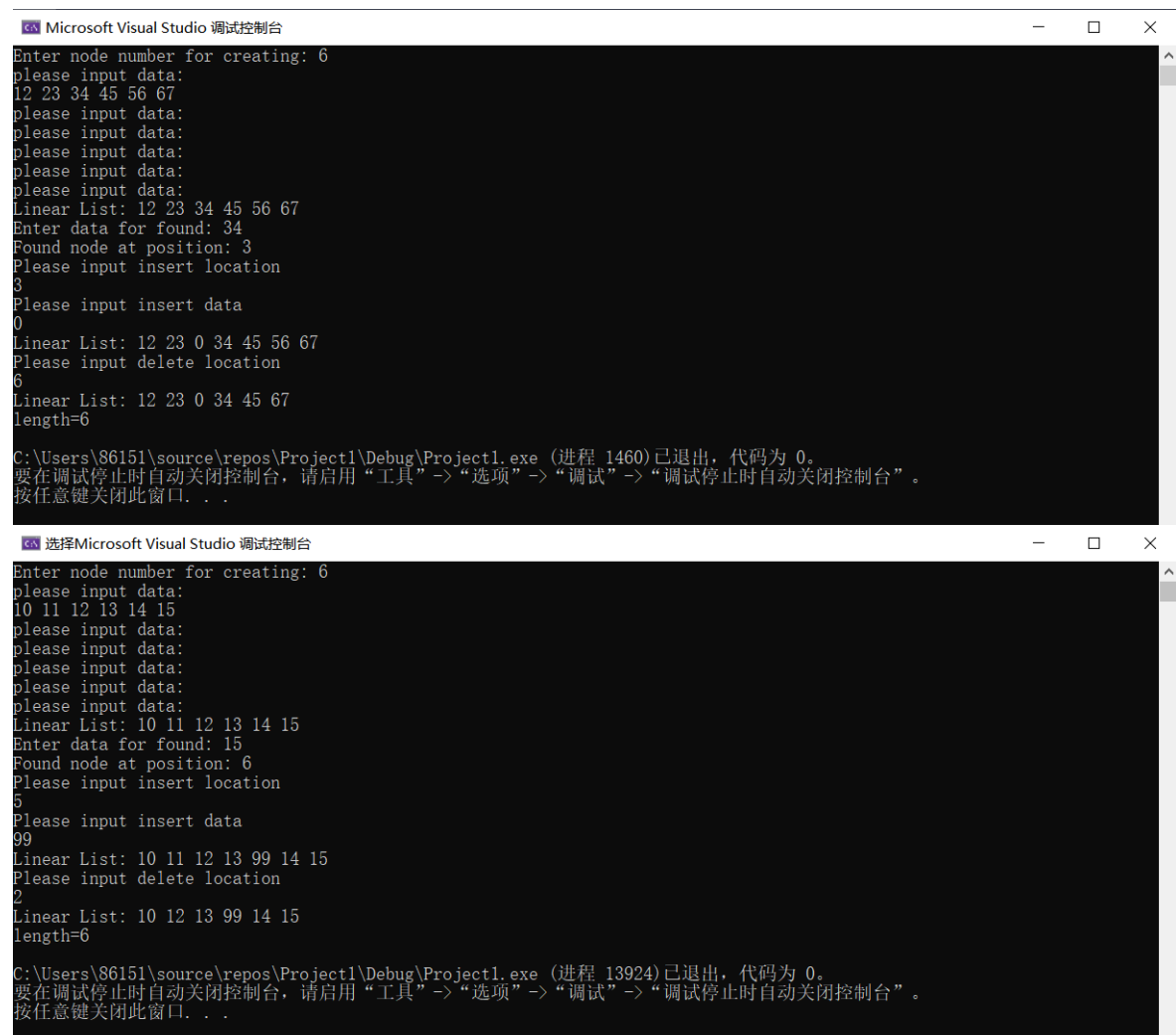


图 1: 题目 1 运行结果

C 题目 2：字符逆转（学时：2）

从键盘读入一个字符串，把它存入一个链表（每个结点存储 1 个字符），并按相反的次序将字符串输出到显示屏。

Listing 2: 题目 2：字符逆转（学时：2）

```
1 #pragma warning(disable:4996)
2 #include <stdio.h>
3 #include <stdlib.h>
4 struct node
5 {
6     struct node* prev;
7     char c;
8     struct node* next;
9 };
10 struct node* input(struct node* top);
11 int main(void)
12 {
13     struct node T, * top = &T, * bottom = &T, * p = NULL;
14     T.prev = NULL;
15     T.next = NULL;
16     T.c = '\\0';
17     bottom = input(top);
18     p = bottom->prev;
19     while (p != NULL)
20     {
21         printf("%c", p->c);
22         p = p->prev;
23     }
24     return 0;
25 }
26 // 逆序//
27 struct node* input(struct node* top)
28 {
29     struct node* t;
30     char x;
31     while ((x = getchar()) != '\\n')
32     {
33         top->c = x;
34         t = (struct node*)malloc(sizeof(struct node));
35         top->next = t;
```

```
36     t->prev = top;
37     t->next = NULL;
38     t->c = '\0';
39     top = top->next;
40 }
41 return top;
42 }
```

D 题目 2 运行结果

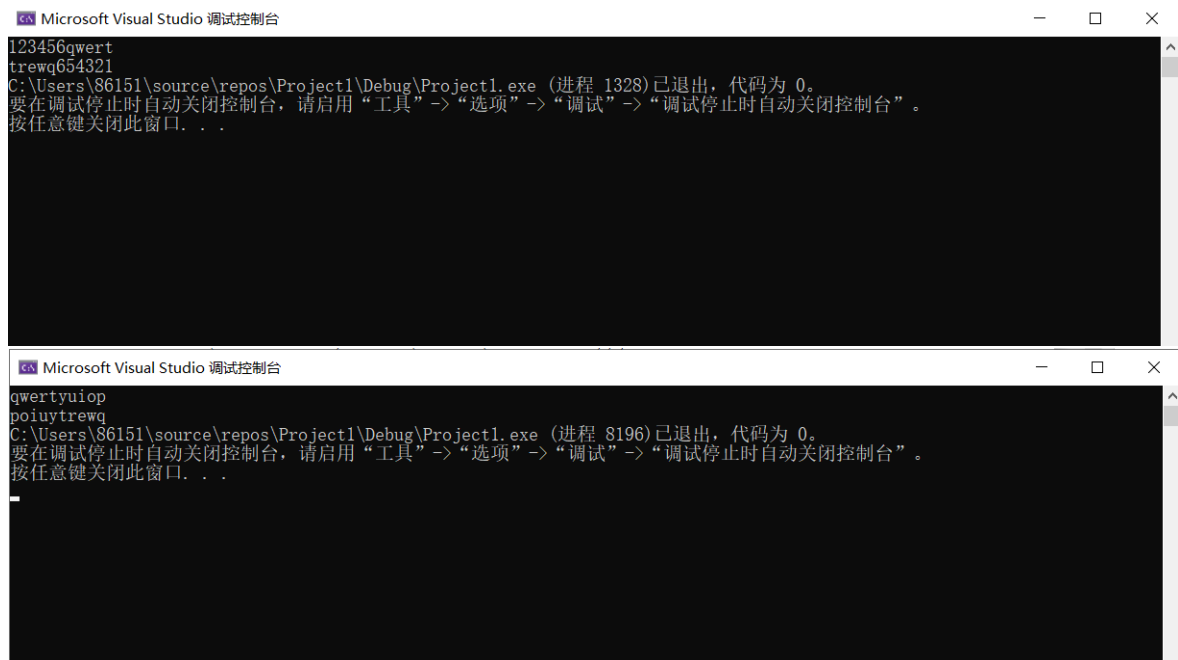


图 2: 题目 2 运行结果