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：	你的班级
学号：	32020xxxxxxx
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

```
#pragma warning(disable:4996)
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#define NULL 0
typedef struct ListNode {
  int data;
  struct ListNode* next;
} NODE;
/* 用尾插法建立带头结点的单链表*/
NODE* CreateList(int n) {
  NODE* head, * p, * q;
  int num;
  int i = 1;
  head = (NODE*)malloc(sizeof(NODE));
  head->next = NULL;
  p = head;
  while (i <= n) {
    printf("please input data:\n");
    scanf("%d", &num);
    q = (NODE*)malloc(sizeof(NODE));
    q->data = num;
    p->next = q; p = q;
    i++;
  }
  p->next = NULL;
  return (head);
}
/* 输出链表*/
void PrintList(NODE* head) {
  NODE* p;
  p = head->next;
  printf("Linear List: ");
  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("\nPostion 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

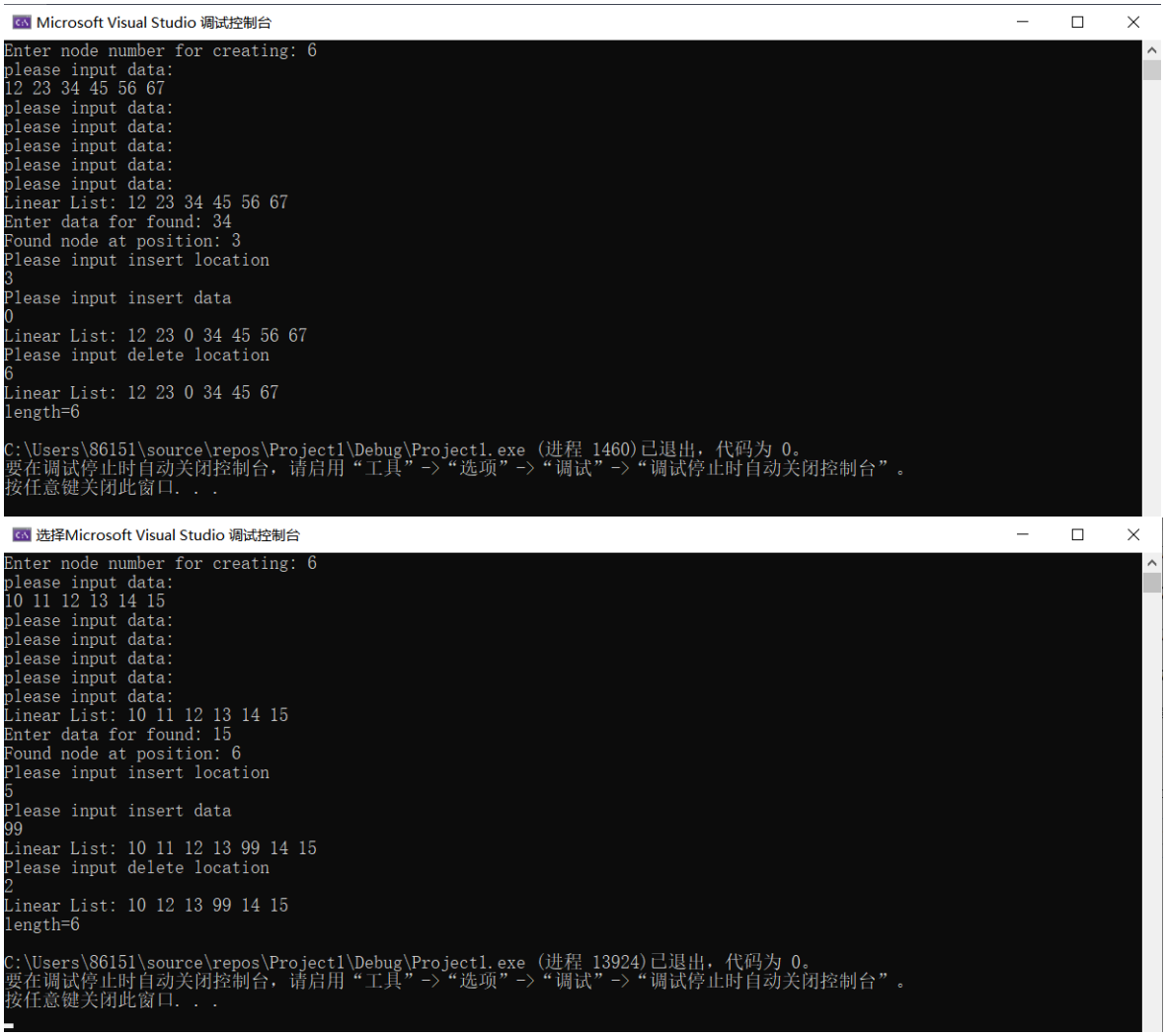

图 1: 题目 1 运行结果

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）

```
#pragma warning(disable:4996)
#include <stdio.h>
#include <stdlib.h>
struct node
{
  struct node* prev;
  char c;
  struct node* next;
};
struct node* input(struct node* top);
int main(void)
{
  struct node T, * top = &T, * bottom = &T, * p = NULL;
  T.prev = NULL;
  T.next = NULL;
  T.c = '\0';
  bottom = input(top);
  p = bottom->prev;
  while (p != NULL)
  {
    printf("%c", p->c);
    p = p->prev;
  }
  return 0;
}
//逆序//
struct node* input(struct node* top)
{
  struct node* t;
  char x;
  while ((x = getchar()) != '\n')
  {
    top->c = x;
    t = (struct node*)malloc(sizeof(struct node));
    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

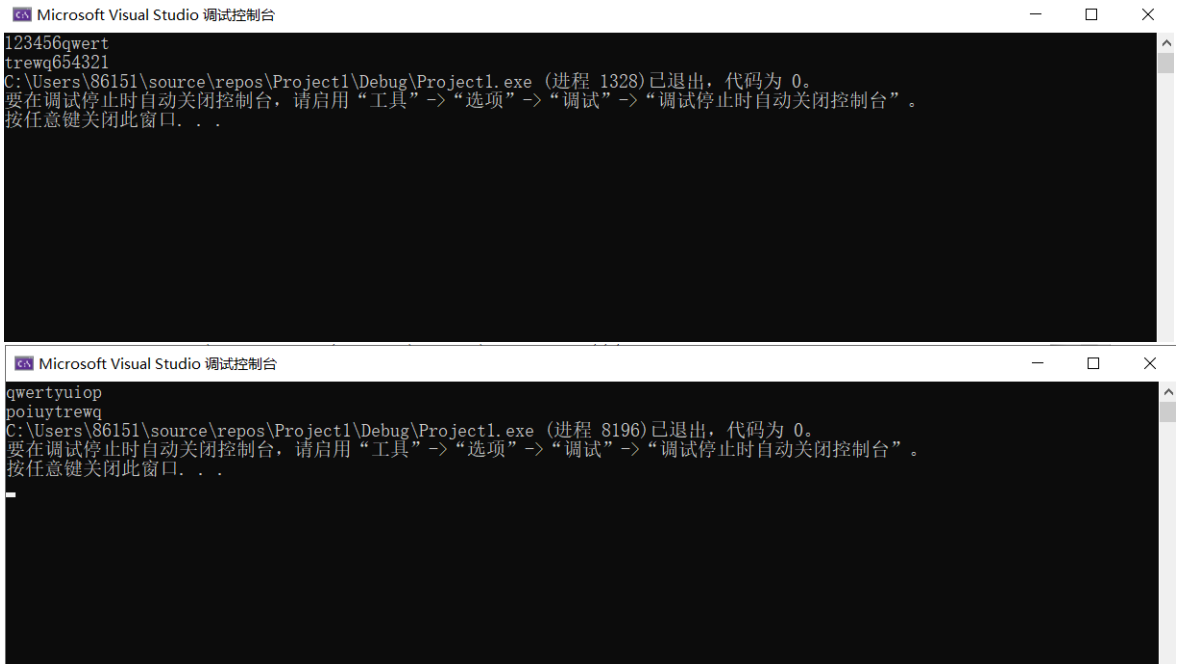

图 2: 题目 2 运行结果