

河南工业大学实验报告

课程名称 数据结构 实验项目 实验一 线性结构(一)——线性表的操作
院 系 信息科学与工程学院 专业班级 计科 XXXX 班
姓 名 COS 学 号 XXXXXXXXXXXX
指导老师 XXXXXX 日 期 2020.10.12
批改日期 _____ 成 绩 _____

一 实验目的

熟练掌握线性表的顺序和链式存储结构及相关典型操作。

二 实验内容及要求

实验内容：

下列两题

1. 完成线性表链式存储结构的若干典型算法（典型算法可自选）：。
2. 有一个单链表的第一个结点指针为 head, 编写一个函数将该单链表逆置, 即最后一个结点变成第一个结点, 原来倒数第二个结点变成第二个结点。在逆置中不能建立新的单链表。

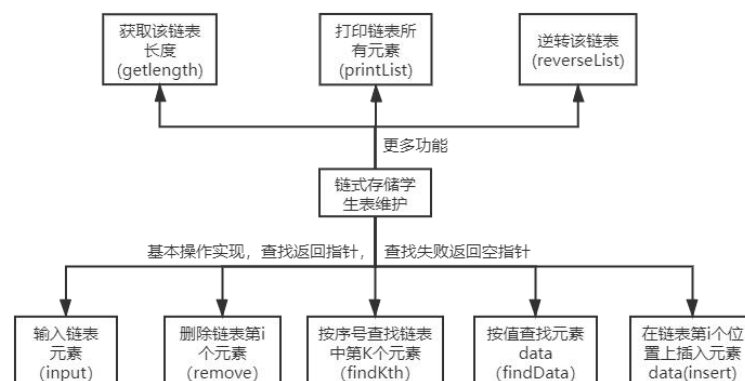
实验要求：

1. 键盘输入数据；
2. 屏幕输出运行结果。
3. 要求记录实验源代码及运行结果。
4. 运行环境：Windows 10 1903 Visual Studio 2019

三 实验过程及运行结果

1. 初步构思

封装一个链表模板类，实现基本的链表操作和逆置功能等，使用 c++ 语言，



2. 基本操作

(1) 查找第 K 个元素

findKth 函数

按序号查找：查找链表中第 K 个元素，若找到返回指向该结点的指针，找不到返回空指针
思想：若 p 后有节点且 p 未到第 K 个，将 p 指针后挪，并且位置 pos++，则结束循环时要么 p 后面无节点了（即到链表末尾了），要么是 p 到第 K 个了，所以只需判断 pos 是否等于 K
核心代码：

```
LNode<T>* p = head->next; //第一个元素
if (!p) {
    cout << "Element not found, This List is empty!" << endl;
    return nullptr;
}
int pos = 1;
while (p->next && pos < K) {
    p = p->next;
    pos++;
}
if (pos == K)
    return p; //找到第K个则返回指针
else { //否则位置不合法，返回空指针
    cout << "The location is illegal." << endl;
    return nullptr;
}
```

(2) 查找元素 data

find 函数

按值查找：查找元素 data，若找到返回指向该结点的指针，找不到返回空指针
思想：p 后有节点且当前 p 并不是要找的元素，则结束循环后 p 要么到最后一个节点都没找到，要么为已经找到的元素
核心代码：

```
LNode<T>* p = head->next;
if (!p) {
    cout << "Element not found, This List is empty!" << endl;
    return nullptr;
}
while (p->next && p->Data != data)
    p = p->next;
if (p->Data != data) { //到最后了但元素仍不是data
    cout << "Element not found!" << endl;
    return nullptr;
}
else
```

```
return p;
```

(3) 插入操作

insert 函数 默认为头插法 将 data 插入在第 pos 个元素

思想：前一个结点为 s, 要插入的结点为 p, 则先将 s 的 next 指针赋给 p 的 next 指针，然后将 s 的 next 指针指向 p

核心代码如下：

```
LNode<T>* p = new LNode<T>();
LNode<T>* p = head->next;
if (!p) {
    cout << "Deleted failed, This List is empty!" << endl;
    return;
}
if (pos == 1) {    //若要删除的是表的第一个结点
    head->next = p->next;
    if (p)
        delete p;
    return;
}
LNode<T>* s = FindKth(pos - 1);    //找到第pos-1个元素
if (!s) {
    cout << "The " << pos-1 << " node does not exist!" << endl;
    return;
}
p = s->next;
if (!p) {
    cout << "The " << pos << " node does not exist!" << endl;
    return;
} else {
    s->next = p->next;    // s指向第pos+1个结点
    delete p;            //将p从链表中删除
    cout << "Successfully deleted!" << endl;
}
```

(4) 删除操作

Delete 函数

删除链表第pos个位置上的结点, 默认为删第一个元素

思路：首先p为要删除的结点, s为p前一个结点, 删除只需先将s的next指针赋值为p的next指针，再释放p的空间，注意特判删第一个结点的情况。

核心代码如下：

```
LNode<T>* p = head->next;
if (!p) {
```

```

        cout << "Deleted failed, This List is empty!" << endl;
        return;
    }
    if (pos == 1) { //若要删除的是表的第一个结点
        head->next = p->next;
        if (p)
            delete p;
        return;
    }
    LNode<T>* s = FindKth(pos - 1); //找到第pos-1个元素
    if (!s) {
        cout << "The " << pos-1 << " node does not exist!" << endl;
        return;
    }
    p = s->next;
    if (!p) {
        cout << "The " << pos << " node does not exist!" << endl;
        return;
    } else {
        s->next = p->next; // s指向第pos+1个结点
        delete p; //将p从链表中删除
        cout << "Successfully deleted!" << endl;
    }
}

```

(5) 获取该链表长度

getLength 函数

思路：就遍历，next 指针不为空就往后走

核心代码如下：

```

LNode<T>* p = head; // p指向表的头节点
int cnt = 0;
while (p->next) {
    p = p->next;
    cnt++;
}

return cnt;

```

(6) 展示该链表

PrintList 函数

思路：遍历，不为空就往下走，为空输出链表为空咯

核心代码如下：

```

template <class T> void List<T>::PrintList() {
    LNode<T>* p = head; // p指向表的头节点

```

```

    if (!head->next)
        cout << "This List is empty!" << endl;
    int cnt = 0;
    while (p->next) {
        p = p->next;
        cnt++;
        cout << "The " << cnt << "th Data is:";
        p->showdata();
    }
    cout << "The list has " << cnt << " elements in total" << endl;
}

```

(7) 逆转链表

ReverseList 函数

删除链表第pos个位置上的结点, 默认为删第一个元素

核心代码如下:

```

LNode<T>* now = head->next;    // now指向表的第一个节点
if (!now) {
    cout << "This List is empty!";
    return;
}
LNode<T>* tmp = now->next; // 指向now之后的一个节点
while (tmp) {              //把tmp放到头结点后边
    now->next = tmp->next;
    tmp->next = head->next;
    head->next = tmp;
    tmp = now->next;
}

```

2. 几个测试数据

20191601 小于 19

20191022 小明 18

20191412 小芳 19

20191456 小田 19

20191501 小梦 17

3. 运行结果

(1) 图 3-1 链表为空时各函数操作测试正常

```
Microsoft Visual Studio 调试控制台
1 添加一名学生信息
2 删除链表第i个元素
3 按序号查找链表中第K个元素
4 按值查找元素data
5 在链表第i个位置上插入元素data
6 获取该链表长度
7 打印所有学生信息
8 逆转该链表
9 结束
菜单选择(1-6):2
请输入i(删除链表第i个元素):1
Deleted failed,This List is empty!
按回车键继续...

菜单选择(1-6):3
(按序号查找)请输入K, 查找链表中第K个元素:2
Element not found,This List is empty!
未找到!
按回车键继续...
```

```
菜单选择(1-6):4
请输入待查找学生的学号:
2312
Element not found,This List is empty!
未找到该元素!
按回车键继续...

菜单选择(1-6):6
该链表长度为:0
按回车键继续...

菜单选择(1-6):7
----- List P -----
This List is empty!
The list has 0 elements in total.
按回车键继续...

菜单选择(1-6):8
----- List P -----
This List is empty!
The list has 0 elements in total.

This List is empty!
----- After reverse, List P -----
This List is empty!
The list has 0 elements in total.
按回车键继续...

菜单选择(1-6):9
E:\Programming\数据结构\class\计科201916010728余思娴实验1\List\Debug\List.exe (进程 11192)
已退出, 代码为 0。
按任意键关闭此窗口. . .
```

(2) 测试数据全部头插法插入成功, 显示链表全部元素功能正常

```
E:\Programming\数据结构\class\计科201916010728余思娟实验1\List\Debug\List.exe
1 添加一名学生信息
2 删除链表第i个元素
3 按序号查找链表中第K个元素
4 按值查找元素data
5 在链表第i个位置上插入元素data
6 获取该链表长度
7 打印所有学生信息
8 逆转该链表
9 结束
菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191601 小于 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191022 小明 18
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191412 小芳 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191456 小田 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191501 小梦 17
Successfully Inserted!
按回车键继续...
```

```
E:\Programming\数据结构\class\计科201916010728余思婧实验1\List\Debug\List.exe
1 添加一名学生信息
2 删除链表第i个元素
3 按序号查找链表中第K个元素
4 按值查找元素data
5 在链表第i个位置上插入元素data
6 获取该链表长度
7 打印所有学生信息
8 逆转该链表
9 结束
菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191601 小于 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191022 小明 18
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191412 小芳 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191456 小田 19
Successfully Inserted!
按回车键继续...

菜单选择(1-6):1
请输入链表元素(以空格间隔,回车结束):20191501 小梦 17
Successfully Inserted!
按回车键继续...

菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191412 姓名:小芳 年龄:19
The 4th Data is:学号:20191022 姓名:小明 年龄:18
The 5th Data is:学号:20191601 姓名:小于 年龄:19
The list has 5 elements in total.
按回车键继续...
```

(3) 删除功能正常, 删除位置错误则提示

```
E:\Programming\数据结构\class\计科201916010728余思婧实验1\List\Debug\List.exe
菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191412 姓名:小芳 年龄:19
The 4th Data is:学号:20191022 姓名:小明 年龄:18
The 5th Data is:学号:20191601 姓名:小于 年龄:19
The list has 5 elements in total.
按回车键继续...
```



```

菜单选择(1-6):2
请输入i(删除链表第i个元素):4
Successfully deleted!
按回车键继续...

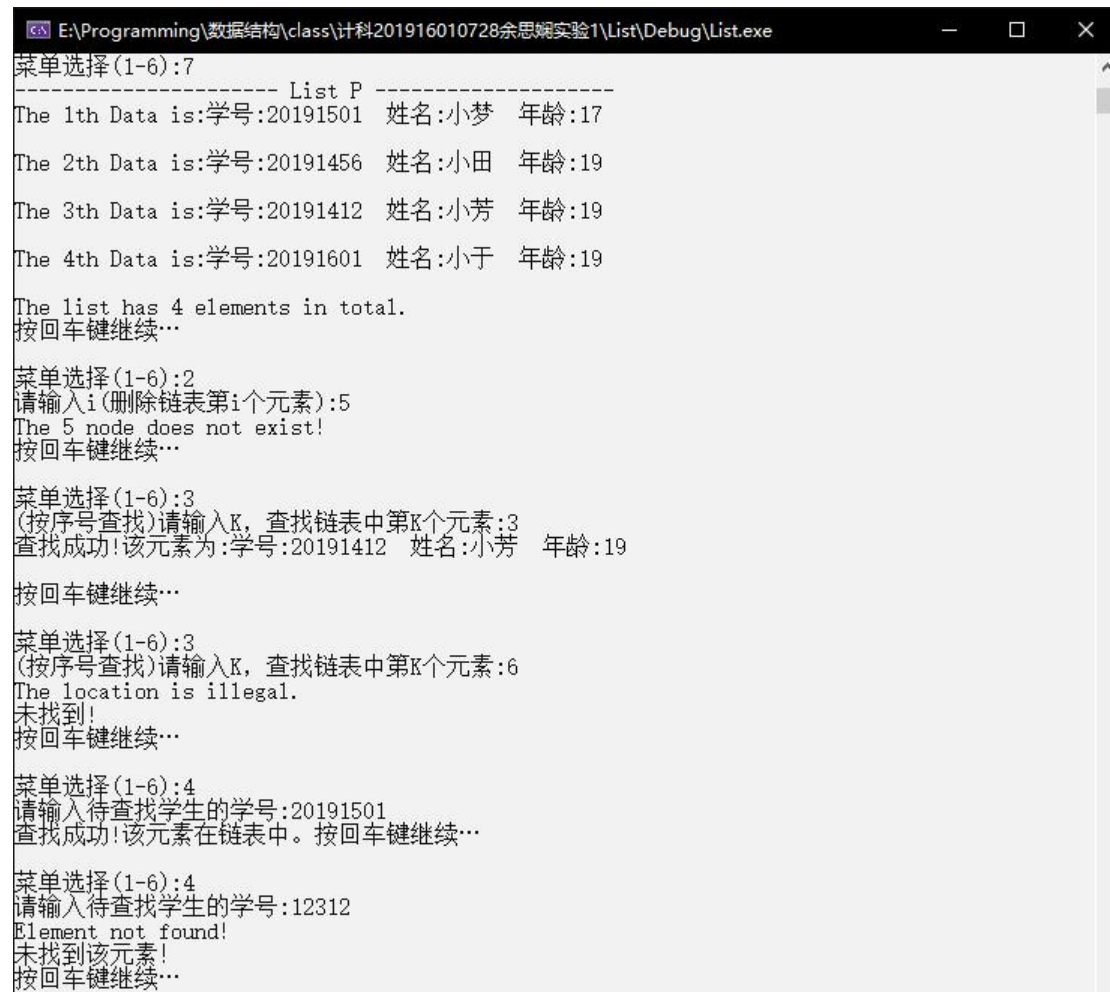
菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191412 姓名:小芳 年龄:19
The 4th Data is:学号:20191601 姓名:小于 年龄:19

The list has 4 elements in total.
按回车键继续...

菜单选择(1-6):2
请输入i(删除链表第i个元素):5
The 5 node does not exist!
按回车键继续...

```

(4) 查找功能正常，未找到元素有提示



```

E:\Programming\数据结构\class\计科201916010728余思媚实验1\List\Debug\List.exe
菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191412 姓名:小芳 年龄:19
The 4th Data is:学号:20191601 姓名:小于 年龄:19

The list has 4 elements in total.
按回车键继续...

菜单选择(1-6):2
请输入i(删除链表第i个元素):5
The 5 node does not exist!
按回车键继续...

菜单选择(1-6):3
(按序号查找)请输入K, 查找链表中第K个元素:3
查找成功!该元素为:学号:20191412 姓名:小芳 年龄:19

按回车键继续...

菜单选择(1-6):3
(按序号查找)请输入K, 查找链表中第K个元素:6
The location is illegal.
未找到!
按回车键继续...

菜单选择(1-6):4
请输入待查找学生的学号:20191501
查找成功!该元素在链表中。按回车键继续...

菜单选择(1-6):4
请输入待查找学生的学号:12312
Element not found!
未找到该元素!
按回车键继续...

```

(5) 在第 i 个位置上插入成功，未插入成功有提示

```
E:\Programming\数据结构\class\计科201916010728余思媚实验1\List\Debug\List.exe
菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191412 姓名:小芳 年龄:19
The 4th Data is:学号:20191601 姓名:小于 年龄:19
The list has 4 elements in total.
按回车键继续...

菜单选择(1-6):5
在链表第i个位置上插入一名学生信息,输入i:3
输入该学生信息,学号 姓名 年龄以空格间隔,回车结束:20191022 小明 18
Successfully Inserted!
按回车键继续...

菜单选择(1-6):7
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191022 姓名:小明 年龄:18
The 4th Data is:学号:20191412 姓名:小芳 年龄:19
The 5th Data is:学号:20191601 姓名:小于 年龄:19
The list has 5 elements in total.
按回车键继续...

菜单选择(1-6):5
在链表第i个位置上插入一名学生信息,输入i:7
输入该学生信息,学号 姓名 年龄以空格间隔,回车结束:20191022 小明 18
The location is illegal.
The location is illegal, Insertion failed!
按回车键继续...
```

(6) 获取长度、逆转链表成功

```
E:\Programming\数据结构\class\计科201916010728余思媚实验1\List\Debug\List.exe
菜单选择(1-6):6
该链表长度为:5
按回车键继续...

菜单选择(1-6):8
----- List P -----
The 1th Data is:学号:20191501 姓名:小梦 年龄:17
The 2th Data is:学号:20191456 姓名:小田 年龄:19
The 3th Data is:学号:20191022 姓名:小明 年龄:18
The 4th Data is:学号:20191412 姓名:小芳 年龄:19
The 5th Data is:学号:20191601 姓名:小于 年龄:19
The list has 5 elements in total.

----- After reverse, List P -----
The 1th Data is:学号:20191601 姓名:小于 年龄:19
The 2th Data is:学号:20191412 姓名:小芳 年龄:19
The 3th Data is:学号:20191022 姓名:小明 年龄:18
The 4th Data is:学号:20191456 姓名:小田 年龄:19
The 5th Data is:学号:20191501 姓名:小梦 年龄:17
The list has 5 elements in total.
按回车键继续...
```

四 调试情况、设计技巧

1. 调试情况

问题 1: 查找失败?

```
while (p->next && p->Data != data)
    p = p->next;
if (!p)
```

改为:

```
if (!p || p->Data != data) { //第一个就是空节点或到最后了但元素仍不是data
```

出问题的地方是没有考虑到最后了元素还不是 data 的情况, 这样不管怎么查都是存在的。

问题 2: 逆转链表没有考虑一个节点的情况

```
if (!now) {
    cout << "This List is empty!";
    return;
}
```

加个判断就好了

2. 设计技巧

这种数据结构很适合封装一个模板类, 于是我封装了一个带空头结点的链表类, 包括基本的链表操作以及逆转, 一开始有遇到很多问题, 比如模板类之间链接出错或是别的一些语法错误, 链表本身的操作不算难, 这里简述一下较难理解的东西就好。这是逆置操作的核心思想。

