

电子科技大学信息与软件工程学院

# 实 验 报 告

学 号 2018091602012

姓 名 杨贺然

(实验) 课程名称 程序设计与算法基础 I

理论教师 罗绪成

实验教师 罗绪成、朱相印

# 电子科技大学

## 实验报告

学生姓名：杨贺然      学号：2018091602012      指导教师：罗绪成

实验地点：基础实验大楼 A535      实验时间：2018.12.22

一、实验名称：超市商品管理系统链表实现

二、实验学时：4 课时

三、实验目的：

1. 掌握单链表的定义和使用方法；
2. 掌握单链表的建立方法；
3. 掌握单链表中节点的查找与删除；
4. 掌握输出单链表节点的方法；
5. 掌握链表节点排序的一种方法；
6. 掌握 C 语言创建菜单的方法；
7. 掌握结构体的定义和使用方法。

四、实验原理：

链表是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表由一系列结点（链表中每一个元素称为结点）组成，结点可以在运行时动态生成。每个结点包括两个部分：一个是存储数据元素的数据域，另一个是存储下一个结点地址的指针域。由于其可扩展性高，适用于这样需要动态添加元素的操作，因此采用链表作为数据存储方式。

对于链表节点排序，这里选择冒泡排序。冒泡排序的基本过程为：

```
for i = 1 to n - 1 do
    for j = 1 to n - i do
        if a[j] > a[j + 1]
            swap(a[j], a[j + 1])
```

分析整个算法的复杂度，对于存储链表，空间复杂度为  $O(n)$ 。对于链表的显示，节点修改，插入到中间，删除，查找，时间复杂度均为  $O(n)$ ，对于插入至链表头尾，时间复杂度为  $O(1)$ 。对于链表按价格排序，时间复杂度为  $O(n^2)$ ，可以满足商品数量较少情况下快速响应的需求。

## 五、实验内容：

用 C 语言与单链表数据结构实现一个小型的超市商品管理系统，该系统需要具备商品信息修改、商品信息删除、商品信息查找、商品信息的插入、将商品按价格排序、商品信息的查看与存盘这几个功能。具体实现步骤如下：

### 1. 软件页面的控制

实现一个数字选项式的启动界面，其中至少包含显示所有商品信息、商品信息插入、商品信息修改、商品信息删除、商品信息查找、商品价格排序、退出系统并保存 7 个选项。并且这些功能可以循环调用。

### 2. 商品信息的初始化

定义链表并初始化。实现从已有的商品信息文件中读入商品信息，并且分配内存保存至链表中。

### 3. 商品信息的增加

实现一个函数完成单个商品信息的增加，接受用户的输入的各项信息，然后保存至链表结点。同时实现可以根据用户的输入，将该结点插入到列表的头部，尾部，或者中间  $i$  号位置。

### 4. 商品信息的修改

实现一个函数完成商品信息的修改功能，实现可以根据商品的名称或者 ID 修改商品信息。其中用字符串比较的方式来查找待修改商品。

### 5. 商品信息的删除

实现一个函数，实现根据商品的名称来删除对应的商品信息的功能，商品查找通过字符串比较的方式，查找到后释放对应指针指向的内存区域，完成删除。

### 6. 商品信息的查找

实现一个函数，函数的功能是根据输入的商品名称来查找对应的商品信息，商品名称的判断用字符串比较的方式来实现，然后调用格式化显示查找到的商品信息。

## 7. 所有商品信息的显示

实现一个函数，该函数的功能是将链表中所有的商品信息以格式化的方式打印输出到屏幕上。

## 8. 对商品按价格排序

实现一个函数，可以根据链表中的商品的价格，对商品进行排序，排序算法采用冒泡排序实现，最后将排序后的链表打印至屏幕。

## 9. 退出系统，并保存

实现一个文件写入函数将所有信息的改动写入到商品信息文件，然后清理系统运行过程中已分配的内存。

# 六、实验器材（设备、元器件）：

个人电脑一台。

# 七、实验步骤：

## 1. 明确商品属性，定义商品结构与链表结构

商品信息包括商品 ID，商品名称，商品折扣，商品价格，商品数量与商品剩余量。其中用字符数组保存商品 ID，商品名称，商品价格，其余信息用整数型存储。链表结构包括商品信息与下一节点指针。

## 2. 实现主程序部分

主程序部分需要初始化商品信息，输出菜单，接收用户选择。这里用标准输入输出与用户交互。

## 3. 实现统一的字符串读入函数，商品信息读入与输出函数

对于字符串读入，需要跳过之前的空白字符，并且在输入过程中不忽略空格。对于商品信息读入，以相应方式读入相应类型数据即可。用自定义的字符串读入函数实现对于商品 ID，商品名称，商品折扣的读取，用格式化读入函数读入其他内容。用格式化输出函数进行商品信息输出。

详细实现详见附件 shopping.c 中 `void readLine(char s[],int lim), void readItem(Goods *a), void printOneInfo(Goods a)`。

## 4. 实现初始化商品信息函数

该函数需要从一个文件中读入商品信息，每次程序从文件中读取一条商品记录，并动态申请内存空间，将记录插入链表尾。一条记录中项之间用一个制表符

隔开，因此用制表符作为分隔标准。

详细实现详见附件 shopping.c 中 readAndInitialize(GoodsList \*\*L, GoodsList \*\*LL)。

## 5. 实现商品信息修改功能

该函数需要从标准输入中读入用户先要修改的商品的 ID，然后提示用户输入新的商品信息，如需要修改的商品不存在，则输出提示信息。这里只需顺序沿链表比对节点 ID 信息是否与用户的输入相同即可。若相同，则提示用户输入新的物品信息，之后存入填表退出函数，否则继续查找。若已经移动到空节点，则输出错误信息，退出函数。

详细实现详见附件 shopping.c 中 void infoChange(GoodsList \*\*L)。

## 6. 实现商品信息插入功能

该函数需要从标准输入中读入用户需要插入的商品信息，然后读入用户希望插入的位置，如果不合法，则输出提示信息。对于这项操作，首先为用户的输入申请内存，将信息初始化。插入操作只需沿链表查找需要插入位置的前一个节点，然后修改其后继为要加入的节点，要加入的节点的后继为原节点的后继即可。但是对于链表中最后一个节点需要注意特殊处理。

详细实现详见附件 shopping.c 中 void infoInsert(GoodsList \*\*L)。

## 7. 实现商品信息删除功能

该函数需要从标准输入中读入用户需要删除的商品 ID，然后在链表中删除该商品。对于这项操作，需要沿链表查找节点 ID 信息是否与用户输入完全相同，若相同，则等待用户确认后，将前一节点的后继改为这一节点的后继，然后释放该节点内存空间即可。注意第一个节点与最后一个节点需特殊处理。

详细实现详见附件 shopping.c 中 void infoDelete(GoodsList \*\*L)。

## 8. 实现商品信息查找功能

该函数需要在链表中查找与用户输入完全相同的商品名称，并将其全部信息输出，若未找到则输出其他提示信息。这里只需顺序沿链表比对节点名称信息是否与用户输入相同即可，用系统提供的函数进行比对。若该节点为空节点，则退出查找。

详细实现详见附件 shopping.c 中 void infoSearch(GoodsList \*L)。

## 9. 实现商品信息存盘功能

该函数需要将所有链表中的元素按顺序输出到文件。这里按链表顺序将每一个节点的信息输出即可，利用 3 中的函数即可很方便地输出。

详细实现详见附件 shopping.c 中 void infoFlush(GoodsList \*L)。

## 10. 实现商品按价格排序功能

该函数利用冒泡排序，对于链表中的商品按照价格从低到高排序。

详细实现详见附件 shopping.c 中 void bubbleSort(GoodsList \*\*L)。

## 11. 实现删除所有商品功能

该函数将销毁链表，并新建一个空链表。需要将原链表节点逐个回收后申请新的内存作为链表头。并且置零商品数计数器。

详细实现详见附件 shopping.c 中 void deleteAll(GoodsList \*\*L)。

## 12. 调试代码直至可以实现全部功能

# 八、实验结果与分析（含重要数据结果分析或核心代码流程分析）：

### 1. 主界面实现

当初始化文件不存在时：

```
商品信息初始化文件不存在！程序将为您新建一个。
超市商品管理系统
*****
1. 显示所有商品的信息；
2. 修改某个商品的信息；
3. 插入某个商品的信息；
4. 删除某个商品的信息；
5. 查找某个商品的信息；
6. 商品存盘并退出系统；
7. 对商品价格进行排序；
8. （慎用）删除所有内容；
其他. 不存盘并退出系统；
*****
输入您的选择：
```

当初始化文件存在时：

```
商品的链表文件已建立，有 5 个商品记录。
超市商品管理系统
*****
1. 显示所有商品的信息；
2. 修改某个商品的信息；
3. 插入某个商品的信息；
4. 删除某个商品的信息；
5. 查找某个商品的信息；
6. 商品存盘并退出系统；
7. 对商品价格进行排序；
8. （慎用）删除所有内容；
其他. 不存盘并退出系统；
*****
输入您的选择：
```

## 2. 显示所有商品的信息

```
输入您的选择：1
+++++
ID: 1000      名称: new1      价格: 90      折扣: 0.9      数量: 90      剩余: 80
+++++
ID: 1001      名称: new2      价格: 80      折扣: 0.9      数量: 80      剩余: 80
+++++
ID: 1002      名称: new3      价格: 70      折扣: 0.8      数量: 10      剩余: 10
+++++
ID: 1003      名称: new4      价格: 88      折扣: 0.8      数量: 70      剩余: 80
+++++
ID: 1004      名称: new5      价格: 100     折扣: 0.8      数量: 70      剩余: 80
+++++
```

## 3. 修改某个商品的信息

当输入退出修改指令时：

```
输入您的选择：2
输入需要修改的商品 ID (-1 退出修改)：-1
```

当存在此商品时：

```
输入您的选择：2
输入需要修改的商品 ID (-1 退出修改)：1000
输入新的商品信息：
商品 ID：1000
商品名称：MI 8 SE
商品价格：1799
商品折扣：0.00%
商品数量：100
商品剩余：1
修改商品信息成功！修改后的商品为：

+++++
ID: 1000      名称: MI 8 SE    价格: 1799    折扣: 0.00%    数量: 100    剩余: 1
+++++
```

修改后的商品列表为：

```
输入您的选择：1

+++++
ID: 1000      名称: MI 8 SE    价格: 1799    折扣: 0.00%    数量: 100    剩余: 1
+++++

+++++
ID: 1001      名称: new2      价格: 80      折扣: 0.9      数量: 80      剩余: 80
+++++

+++++
ID: 1002      名称: new3      价格: 70      折扣: 0.8      数量: 10      剩余: 10
+++++

+++++
ID: 1003      名称: new4      价格: 88      折扣: 0.8      数量: 70      剩余: 80
+++++

+++++
ID: 1004      名称: new5      价格: 100     折扣: 0.8      数量: 70      剩余: 80
+++++
```

当不存在该商品时：

```
输入您的选择：2
输入需要修改的商品 ID (-1 退出修改)：1010
您要修改的商品不存在！
```

#### 4. 删除某个商品的信息

当输入退出删除指令时：

```
输入您的选择：4
输入需要删除的商品 ID (-1 退出删除)：-1
```

当输入的商品 ID 存在，但用户取消删除时：

```
输入您的选择：4
输入需要删除的商品 ID (-1 退出删除)：1004

+++++
ID: 1004      名称: new5      价格: 100     折扣: 0.8      数量: 70      剩余: 80
+++++
是否删除该商品？(Y/N)：
n
```



当输入的商品 ID 存在，用户确定删除时：

```
输入您的选择：4
输入需要删除的商品 ID (-1 退出删除)：1002

+++++
ID: 1002      名称: new3      价格: 70      折扣: 0.8      数量: 10      剩余: 10
+++++
是否删除该商品？ (Y/N)：
Y
Tips: 删除 ID 为 1002 的商品成功！
Tips: 当前剩余商品 4 个。
```

删除后的商品列表为：

```
输入您的选择：1

+++++
ID: 1000      名称: MI 8 SE      价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++

+++++
ID: 1001      名称: new2      价格: 80      折扣: 0.9      数量: 80      剩余: 80
+++++

+++++
ID: 1003      名称: new4      价格: 88      折扣: 0.8      数量: 70      剩余: 80
+++++

+++++
ID: 1004      名称: new5      价格: 100      折扣: 0.8      数量: 70      剩余: 80
+++++
```

5. 查找某个商品信息

当用户退出查找时：

```
输入您的选择：5
输入需要查找的商品名称 (-1 退出查找)：-1
```

当查找的商品存在时：

```
输入您的选择：5
输入需要查找的商品名称 (-1 退出查找)：MI 8 SE

+++++
ID: 1000      名称: MI 8 SE      价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++
```

当查找的商品不存在时：

```
输入您的选择：5
输入需要查找的商品名称 (-1 退出查找)：UESTC
您要查找的商品不存在！
```

6. 插入某个商品

当插入至链表头时：

```
输入您的选择: 3
输入新的商品信息:
商品 ID: 1005
商品名称: iPhone XS
商品价格: 9999
商品折扣: 0.00%
商品数量: 1
商品剩余: 1
输入数字以表示您要插入的商品位置: 0. 商品列表尾部 1. 商品列表头部 i. 商品列表中间第 i 号位置: 1
```

插入后的商品列表为:

```
输入您的选择: 1

+++++
ID: 1005      名称: iPhone XS  价格: 9999      折扣: 0.00%      数量: 1  剩余: 1
+++++

+++++
ID: 1000      名称: MI 8 SE   价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++

+++++
ID: 1001      名称: new2      价格: 80        折扣: 0.9        数量: 80        剩余: 80
+++++

+++++
ID: 1003      名称: new4      价格: 88        折扣: 0.8        数量: 70        剩余: 80
+++++

+++++
ID: 1004      名称: new5      价格: 100       折扣: 0.8        数量: 70        剩余: 80
+++++
```

当插入至链表尾时:

```
输入您的选择: 3
输入新的商品信息:
商品 ID: 1002
商品名称: UESTC_SISE
商品价格: 666666666
商品折扣: 0
商品数量: 0
商品剩余: 0
输入数字以表示您要插入的商品位置: 0. 商品列表尾部 1. 商品列表头部 i. 商品列表中间第 i 号位置: 0
```

插入后的商品列表为:

```
输入您的选择: 1

+++++
ID: 1005      名称: iPhone XS  价格: 9999      折扣: 0.00%      数量: 1  剩余: 1
+++++

+++++
ID: 1000      名称: MI 8 SE   价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++

+++++
ID: 1001      名称: new2      价格: 80        折扣: 0.9        数量: 80        剩余: 80
+++++

+++++
ID: 1003      名称: new4      价格: 88        折扣: 0.8        数量: 70        剩余: 80
+++++

+++++
ID: 1004      名称: new5      价格: 100       折扣: 0.8        数量: 70        剩余: 80
+++++

+++++
ID: 1002      名称: UESTC_SISE  价格: 666666666 折扣: 0 数量: 0  剩余: 0
+++++
```

当插入到链表中间时：

```
输入您的选择：3
输入新的商品信息：
商品 ID: 1234
商品名称：HeRaNO
商品价格：0
商品折扣：0
商品数量：1
商品剩余：1
输入数字以表示您要插入的商品位置：0. 商品列表尾部 1. 商品列表头部 i. 商品列表中间第 i 号位置：3
```

插入后的商品列表为：

```
输入您的选择：1

+++++
ID: 1005      名称：iPhone XS  价格：9999      折扣：0.00%      数量：1  剩余：1
+++++

+++++
ID: 1000      名称：MI 8 SE   价格：1799      折扣：0.00%      数量：100      剩余：1
+++++

+++++
ID: 1234      名称：HeRaNO    价格：0 折扣：0 数量：1  剩余：1
+++++

+++++
ID: 1001      名称：new2       价格：80      折扣：0.9      数量：80      剩余：80
+++++

+++++
ID: 1003      名称：new4       价格：88      折扣：0.8      数量：70      剩余：80
+++++

+++++
ID: 1004      名称：new5       价格：100     折扣：0.8      数量：70      剩余：80
+++++

+++++
ID: 1002      名称：UESTC_SISE    价格：66666666 折扣：0 数量：0  剩余：0
+++++
```

当插入位置不存在时：

```
输入您的选择：3
输入新的商品信息：
商品 ID: 0000
商品名称：404 Not Found
商品价格：0
商品折扣：0
商品数量：0
商品剩余：0
输入数字以表示您要插入的商品位置：0. 商品列表尾部 1. 商品列表头部 i. 商品列表中间第 i 号位置：9
未找到该位置，请重新输入！
输入数字以表示您要插入的商品位置：0. 商品列表尾部 1. 商品列表头部 i. 商品列表中间第 i 号位置：8
```

插入后的商品列表为：

```
输入您的选择：1
+++++
ID: 1005      名称: iPhone XS  价格: 9999      折扣: 0.00%      数量: 1  剩余: 1
+++++
+++++
ID: 1000      名称: MI 8 SE   价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++
+++++
ID: 1234      名称: HeRaNO    价格: 0  折扣: 0  数量: 1  剩余: 1
+++++
+++++
ID: 1001      名称: new2      价格: 80        折扣: 0.9        数量: 80        剩余: 80
+++++
+++++
ID: 1003      名称: new4      价格: 88        折扣: 0.8        数量: 70        剩余: 80
+++++
+++++
ID: 1004      名称: new5      价格: 100       折扣: 0.8        数量: 70        剩余: 80
+++++
+++++
ID: 1002      名称: UESTC_SISE  价格: 666666666 折扣: 0  数量: 0  剩余: 0
+++++
+++++
ID: 0000      名称: 404 Not Found  价格: 0  折扣: 0  数量: 0  剩余: 0
+++++
```

## 7. 对商品列表排序

```
输入您的选择：7
```

排序后商品列表为：

```
+++++
ID: 1234      名称: HeRaNO    价格: 0  折扣: 0  数量: 1  剩余: 1
+++++
+++++
ID: 0000      名称: 404 Not Found  价格: 0  折扣: 0  数量: 0  剩余: 0
+++++
+++++
ID: 1001      名称: new2      价格: 80        折扣: 0.9        数量: 80        剩余: 80
+++++
+++++
ID: 1003      名称: new4      价格: 88        折扣: 0.8        数量: 70        剩余: 80
+++++
+++++
ID: 1004      名称: new5      价格: 100       折扣: 0.8        数量: 70        剩余: 80
+++++
+++++
ID: 1000      名称: MI 8 SE   价格: 1799      折扣: 0.00%      数量: 100      剩余: 1
+++++
+++++
ID: 1005      名称: iPhone XS  价格: 9999      折扣: 0.00%      数量: 1  剩余: 1
+++++
+++++
ID: 1002      名称: UESTC_SISE  价格: 666666666 折扣: 0  数量: 0  剩余: 0
+++++
```

## 8. 商品存盘并退出系统

输入您的选择：6

存盘后文件为：

1234	HeRaNO	0	0	1	1		
0000	404 Not Found		0	0	0	0	
1001	new2	80	0.9	80	80		
1003	new4	88	0.8	70	80		
1004	new5	100	0.8	70	80		
1000	MI 8 SE	1799	0.00%	100	1		
1005	iPhone XS		9999	0.00%	1	1	
1002	UESTC_SISE		666666666		0	0	0

## 9. 删除所有内容

输入您的选择：8

删除后，商品列表为：

输入您的选择：1

## 九、总结及心得体会：

经过本实验，对超市商品管理系统和链表有了更深刻的了解。并且破了本人曾打过最长代码的记录，锻炼了代码编写与调试能力。

## 十、对本实验过程及方法、手段的改进建议：

在插入链表结尾时，可以存下最后一个节点的指针，这样便于向后方插入节点。对于内存管理问题，还可以使用内存池管理内存，避免资源分配问题。

报告评分：

指导教师签字：