

**课 程 设 计**

|  |  |
| --- | --- |
| **课程设计名称：** | **数据结构课程设计** |
| **专 业 班 级 ：** | **计算机科学与技术2020级2004班** |
| **学 生 姓 名 ：** | **陈积发** |
| **学 号 ：** | **202016010425** |
| **指 导 教 师 ：** | **唐建国** |
| **课程设计时间：** | **2021.12.13-2021.12.26** |

**计算机科学与技术 专业课程设计任务书**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学生姓名** | **陈积发** | **专业班级** | | **计科2004** | **学号** | | **202016010425** |
| **题 目** | **超市货架管理系统** | | | | | | |
| **课题性质** | **工程设计** | | **课题来源** | | | **自拟** | |
| **指导教师** | **唐建国** | | **同组姓名** | | | **无** | |
| **主要内容** | 1. **基本功能:** 2. 定义商品名称、编号、品牌、单价、数量、入库时间、用户名以及用户密码等的数据类型，每件商品的相关信息采用单链表作为存储结构 3. 实现商品的录入、查询、修改、添加、删除功能。 4. 实现商品的供应与需求情况：根据商品的入库和出库功能模拟商品的进货与售货情况，查看货架上的商品库存量，提示补充库存量，知道商品的供应与需求情况。 5. **扩展功能：** 6. 定义用户类型 7. 实现用户登录系统功能 8. 实现商品的排序功能、 9. 实现将商品的所有数据保存到文件的功能 | | | | | | |
| **任务要求** | 1. 根据需求分析给出概要设计和详细设计，并编程实现相关功能模块，本系统包括以下功能模块：实现增加功能、实现删除功能、实现修改功能、实现查询功能等。 2. 结合课题利用数据结构相关知识，利用C语言实现该系统的所有上述功能，要求界面友善，程序运行正常。 3. 提交课程设计报告1份（具体写作要求参考样例），可运行的系统和源代码电子版一套。 | | | | | | |
| **参考文献** | [1] 严蔚敏.《数据结构（C语言版）》[M]. 北京：清华大学出版社, 2018.  [2] 谭浩强.《C语言程序设计》[M]. 北京：清华大学出版社, 2018  [3] 王红梅，胡明，王涛.数据结构（C++版）.北京：清华大学出版社，2011.  [4] 苏仕华.数据结构课程设计. 机械工业出版社  [5]徐慧民等.C++大学基础教程[M].北京：人民邮电出版社，2005. | | | | | | |
| **审查意见** | **指导教师签字：**  **教研室主任签字：**  **2021年12月13日** | | | | | | |

**目录**

[1 需求分析 1](#_Toc28931)

[1.1 系统概述 1](#_Toc26485)

[1.2 系统中的角色 1](#_Toc29213)

[1.3 功能性需求 1](#_Toc20121)

[2 概要设计 2](#_Toc7876)

[2.1 抽象数据类型 2](#_Toc27211)

[2.2 存储结构设计 3](#_Toc3580)

[2.3 功能结构图 3](#_Toc29641)

[2.4 系统子程序及功能设计 3](#_Toc11322)

[3 运行环境 4](#_Toc17223)

[4 开发工具和编程语言 4](#_Toc16359)

[5 详细设计 4](#_Toc9376)

[6 运行实现 13](#_Toc9600)

[7 心得体会 16](#_Toc30198)

[8 参考文献 18](#_Toc27534)

**1 需求分析**

* 1. **系统概述**

本系统是对商品进行管理，在所有的数据结构类型中，线性表是最好的选择，线性表中数据元素之间的关系是一对一的关系，即除了第一个和最后一个数据元素之外，其它数据元素都是首尾相接的，它具有集合中必存在唯一的一个“第一元素”、集合中必存在唯一的一个 “最后元素”、除最后一个元素之外，均有唯一的后继(后件)以及除第一个元素之外，均有唯一的前驱(前件)等特点。其顺序表是随机存取，插入、删除操作不方便，而链表对插入、删除很方便，不需要移动大量元素, 对于本系统，单链表是最好的选择。利用链表结构的有序表表示某商场商品的库存模型，利用链表存储商品名称、品牌、单价和数量等等，而且会以单价的升序体现链表的有序性，程序主要实现链表的初始化、创建表、插入、删除、更新数据、查询及链表数据与文件之间的转换等功能。

**1.2系统中的角色**

|  |  |  |
| --- | --- | --- |
| 角色 | 类别 | 描述 |
| 管理员 | 管理人员 | 增加、删除、修改、查询商品信息 |

表1.1 系统中的角色

**1.3功能性需求**

## 1.3.1 用户登录

管理员登陆后可以进行商品的增、删、改、查。

## 录入商品信息

录入商品的的名称、编号、数量、单价、品牌。不限制用户输入的类型。

## 1.3.4 查找商品

根据商品编号查找商品，并输出到屏幕中。

## 1.3.5 删除商品

用户输入要删除的商品编号，如在链表中找到要删除的商品记录，则将该商品对应信息从链表中删除，同时该系统中没有相匹配的商品。

## 1.3.6 修改商品

用户输入的商品编号，先在该系统中查找是否存在该商品，如果存在，则调出该编号的商品的所有信息，用户可以输入新的商品信息来替换原有的信息。

## 1.3.7 商品信息排序

根据商品的编号来进行商品的排序。根据商品的库存量进行排序。

## 1.3.8 商品入库

用户可以增加商品种类。

## 1.3.9 商品出库

模拟用户购买商品的过程。如果商品已经出售完提示进货。

**2 概要设计**

**2.1 抽象数据类型**

ADT Goods{  
数据对象：D={ |∈ElemSet,i=1,2,…,n,n≥0}  
数据关系：R={<,|,∈D,i=2,…,n}

基本操作：

Create\_List(&L)

操作结果：创建一个空的链表。

GoodsInsert(& L, Goods x)

初始条件：单链表L存在。

操作结果：在链表L中增加商品。

}ADT Goods

**2.2 存储结构设计**

**2.2.1商品信息的存储定义**

typedef struct Number //定义单链表节点类型

{

Goods data; //数据域

struct Number\* next; //指针域，指向后继结点

}Lnode, LNumber;

**2.3 功能结构图**

超市货架管理系统功能结构图如图2-1所示：

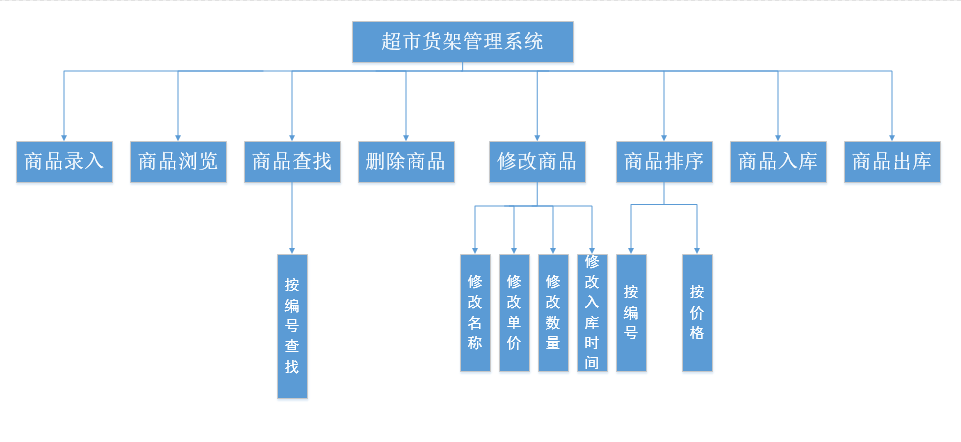
****

图2-1 系统结构图

**2.4 系统子程序及功能设计**

本系统的各个功能模块共涉及14个函数。

**2.4.1 商品相关函数**

表2-1 商品相关函数

|  |  |
| --- | --- |
| 函数名称 | 功能简述 |
| void Create\_List(LNumber\* L) | 创建链表 |
| void Browse(LNumber\* L) | 浏览商品信息 |
| void GoodsSort(LNumber\* L) | 商品信息排序 |
| void GoodsInsert(LNumber\* L, Goods x) | 插入商品 |
| void Storage(LNumber\* L, Goods x) | 商品入库 |
| void GoodsChange(LNumber\* L, Goods x) | 以商品编号进行修改 |
| int Delete(LNumber\* L, Goods x) | 删除商品 |
| void Out(LNumber\* L, Goods x) | 商品出库 |

**3 运行环境**

硬件环境： PC机内存：16G 硬盘：512G

软件环境： 操作系统：windows10

**4 开发工具和编程语言**

开发环境：Visual Studio 2019

编程语言：C语言

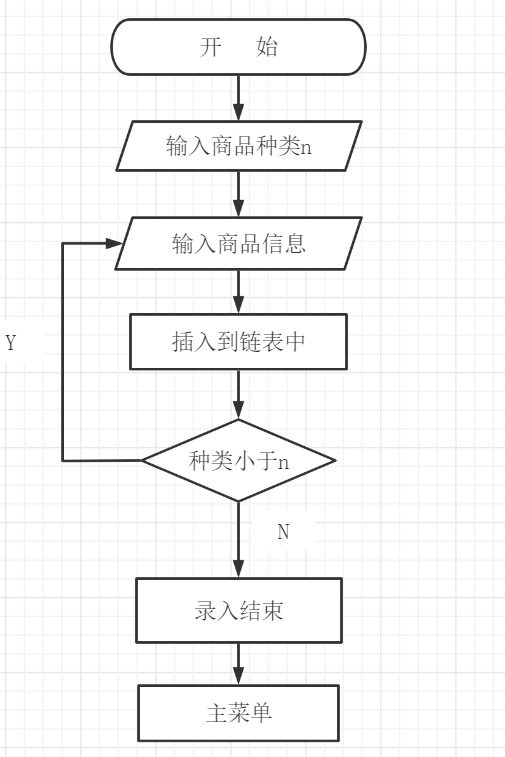
1. **详细设计**
   1. **Create\_List函数**

（1）设计思想

根据需要输入的商品种类，然后依次输入商品的相关信息，直到所有种类都输入完才返回主菜单。

（2）流程图

如图5-1所示.

 图5-1 Create\_List函数流程图

（3）关键代码

void Create\_List(LNumber\* L) //尾插法创建链表

{

int i, n;

LNumber\* r = L, \* s; //声明相关结点

Goods x;

s = (LNumber\*)malloc(sizeof(LNumber)); //开辟新结点

s->data = x; //置数据域

s->next = NULL; //后接

r->next = s; //前接

r = s; //修改尾指针的位置

}

**5.2 Browse函数**

（1）设计思路

用户选择浏览信息对应的号码，程序调用Browse函数，如果商品列表没有信息，返回主菜单，否则把商品的基本信息输出在屏幕上。

（2）流程图

如图5-2所示

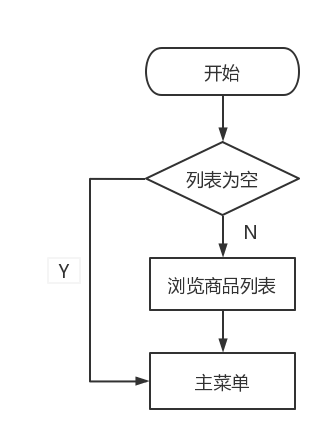


图5-2 Browse函数流程图

（3）关键代码

void Browse(LNumber\* L)

{

LNumber\* p = L->next; //确定起始位置

while (p != NULL) //输出所有商品的相关信息

{

printf("%5s%10s%12d%13s%13.2f%16s\n", p->data.name, p->data.number, p->data.num, p->data.brand, p->data.price, p->data.putdata);

p = p->next;

}

}

**5.3 Search函数**

（1）设计思想

用一个变量记录用户输入的编号，声明一个指针指向链表的第一个结点，在链表非空的情况下依次查找对应编号的商品，如果指针指向最后一个结点，说明该系统中不存在此编号的商品，否则，查找成功，把商品的信息输出到屏幕上。

（2）流程图

如图5-3所示。

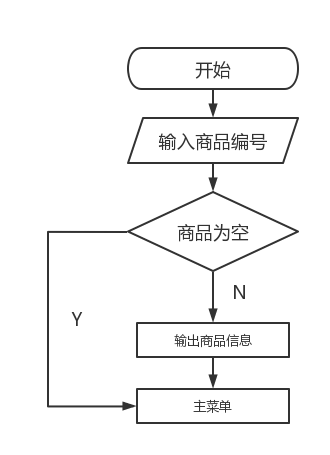


图5-3 Search函数流程图

（3）关键代码

LNumber\* Search(LNumber\* L, Goods x)

{

LNumber\* p = L->next; //查找的起始位置

while (p != NULL && (strcmp(p->data.number, x.number) != 0))

p = p->next;

if (p != NULL && (strcmp(p->data.number, x.number) == 0)) //查找到相应商品，返回指针

return p;

else

return NULL;

}

**5.4 Delete函数**

（1）设计思想

用户输入要删除的编号。找到要删除的结点p以及它的前驱结点q，然后执行q->next=p->next;e=p->data;delete p;在这个过程中声明flag，初始化为0，如果某个结点被删除，则将其的标识符置为1。

（2）流程图

如图5-4所示。

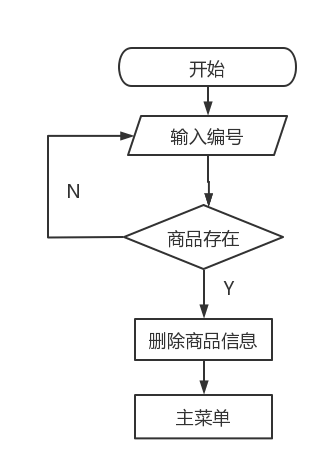


图5-4 Delete函数流程图

（3）关键代码

int Delete(LNumber\* L, Goods x) //删除商品信息

{

LNumber\* p;

int flag = 0; //flag为判断商品是否删除的标志

p = L;

p != NULL; p = p->next

p->next = p->next->next; //删除语句

flag = 1; //重置标志符

return 1;

}

**5.5 GoodsChange函数**

（1）设计思想

用户输入商品编号后，首先查找是否有此商品，若找不到，则提示用户该商品不存在。否则用修改后的信息替换原来的信息。

（2）流程图

如图5-5所示。

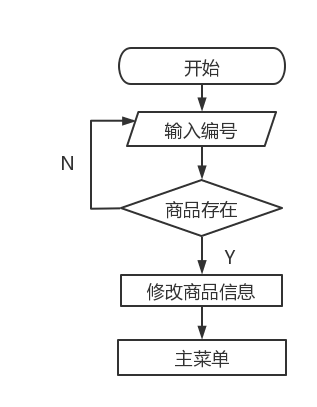


图5-6 GoodsChange函数流程图

（3）关键代码

void GoodsChange(LNumber\* L, Goods x)

{

LNumber\* p;

char NewName[30];

char NewNumber[30];

char NewBrand[30];

char NewPutData[30];

float NewPrice;

int NewNum;

p = Search(L, x); //查找是否存在该商品

printf("\n 修改商品编号： ");

scanf("%s", NewNumber);

strcpy(p->data.number, NewNumber);

scanf("%s", NewName);

strcpy(p->data.name, NewName);

p->data.num = NewNum;

strcpy(p->data.brand, NewBrand);

p->data.price = NewPrice;

strcpy(p->data.putdata, NewPutData);

}

**5.7 GoodsSort函数**

（1）设计思想

商品的排序分为两种，一种是按商品的编号排序，用了直接插入排序法，另一种是按照商品的库存量排序，运用冒泡排序法。

（2）流程图

如图5-7所示。

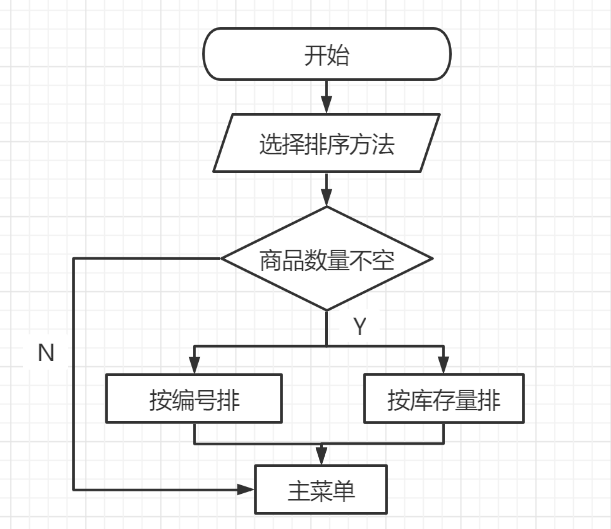


图5-7GoodsSort函数流程图

（3）关键代码

void GoodsSort(LNumber\* L)

{

LNumber\* p, \* q, \* r;

Goods x;

//按商品编号排序为直接插入排序

p = L->next->next; //p指向第一个节点的后继

L->next->next = NULL; //直接插入排序，第一个元素在有序区，从第二元素起依次插入

r = p->next; //r指针暂存p的后继

q = L;

q = q->next; //将p节点链入链表

p->next = q->next;

q->next = p;

p = r;

//按商品数量排序为冒泡排序

p = L->next;

q = p->next;

x = p->data;

p->data = q->data;

q->data = x;

q = q->next;

p = p->next;

}

**5.8 Storage函数**

（1）设计思想

用户选择入库功能，提示输入要入库商品的相关信息，同时系统调用Search(L,x)函数查找仓库中是否存有待入库商品，如果仓库中没有该商品，则调用GoodsInsert(L,r)函数进行插入到链表中；如果仓库中已经有了该商品，则执行p->data.num = p->data.num + x.num;只增加商品的数量。

（2）流程图

如图5-8所示。

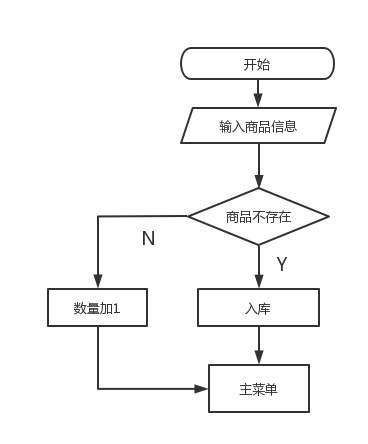


图5-8 Storage函数流程图

（3）关键代码

void Storage(LNumber\* L, Goods x)

{

LNumber\* p;

p = Search(L, x); //查找是否存在该商品

//链表中没有这个商品时可直接插入

GoodsInsert(L, x);

p->data.num = p->data.num + x.num; //链表中中有这个商品时只增加商品的数量

}

**6 运行实现**

系统的登录界面如图6-1所示。

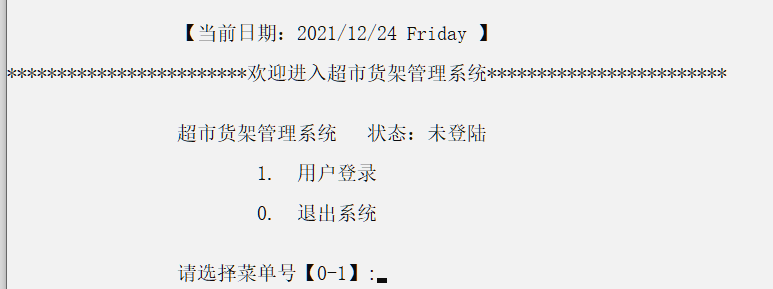


图6-1 系统登录界面

系统的商品相关操作管理页面如图6-2所示。

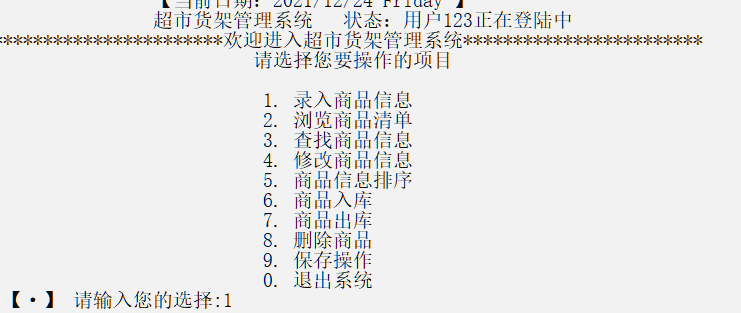


图6-2 商品管理系统主界面

录入商品信息结果如图6-3所示。

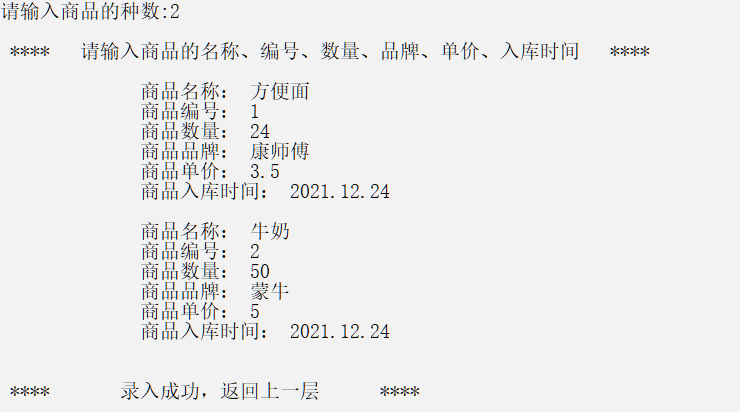


图6-3 录入商品信息

浏览商品清单结果如图6-4所示。

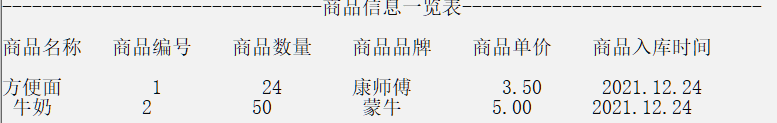


图6-4 浏览商品清单

查找商品信息结果如图6-5所示。

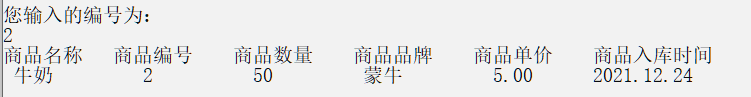


图6-5 查找商品信息

修改商品信息测试结果如图6-6所示。

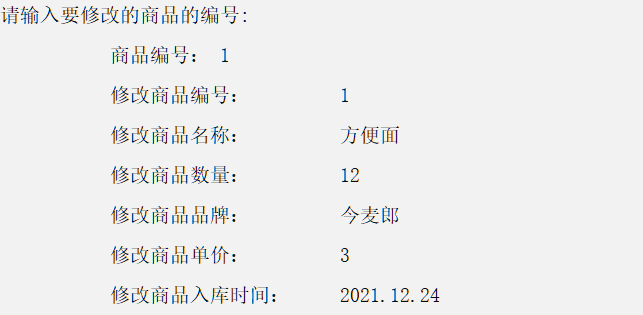


图6-6 修改商品信息

按商品编号进行排序如图6-7所示.

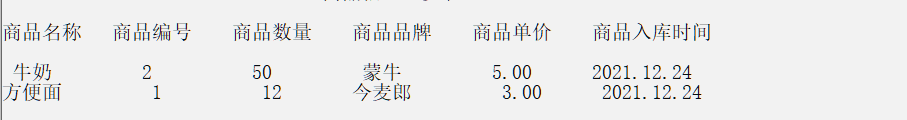


图6-7 商品排序功能之按编号排序结果

按商品库存排序如图6-8所示。

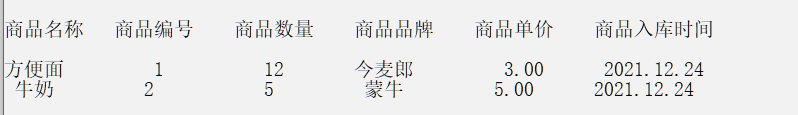


图6-8 商品排序功能之按数量排序结果

商品入库测试结果如图6-9所示。

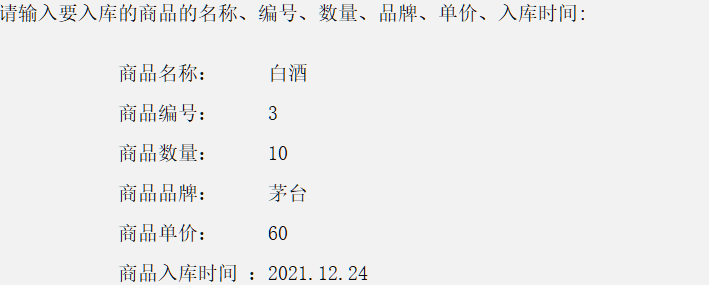


图6-9 商品入库结果

商品出库测试结果如图6-10所示。

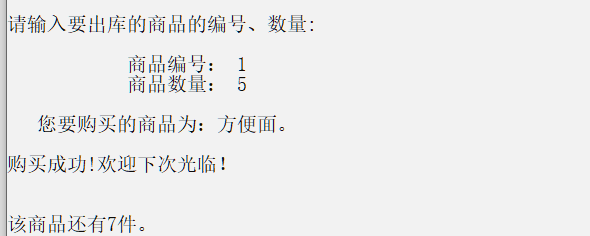


图6-10 商品出库结果

删除商品测试结果如图6-11所示。

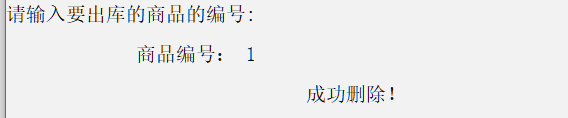


图6-11删除商品结果

数据保存测试结果如图6-12所示。

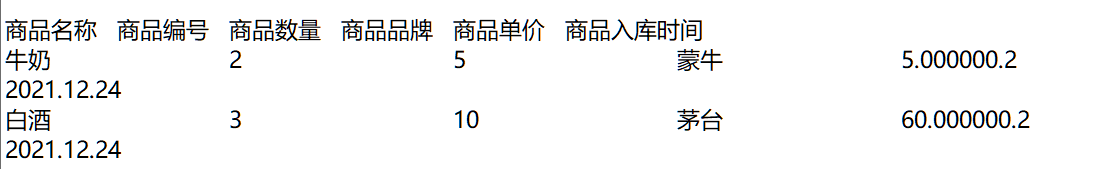


图6-12 数据保存结果

**7 心得体会**

经过两个星期的课程设计，过程曲折可谓一语难尽。整天都是对着电脑，不然就是翻阅资料。在此期间我失落过，也曾一度热情高涨。点点滴滴令我回味无长。

这次课程设计使我体会到只有做到细心耐心，恒心才能做好事情。这次的课程设计，加强了我们动手、思考和解决问题的能力。巩固和加深了对数据结构的理解，提高综合运用本课程所学知识的能力。培养了我选用参考书，查阅手册及文献资料的能力。培养独立思考，深入研究，分析问题、解决问题的能力。通过实际编译系统的分析设计、编程调试，掌握应用软件的分析方法和工程设计方法。通过课程设计，培养了我严肃认真的工作作风，逐步建立正确的生产观念、经济观念和全局观念。而且做课程设计同时也是对课本知识的巩固和加强，平时看课本时，有些问题就不是很能理解，做完课程设计，那些问题就迎刃而解了。而且还可以记住很多东西。认识来源于实践，实践是认识的动力和最终目的，实践是检验真理的唯一标准。

**10 参考文献**

[1] 严蔚敏.《数据结构（C语言版）》[M]. 北京：清华大学出版社, 2018.

[2] 谭浩强.《C语言程序设计》[M]. 北京：清华大学出版社, 2018

[3]徐慧民等.C++大学基础教程[M].北京：人民邮电出版社，2005.

[4]王红梅，胡明，王涛.数据结构（C++版）.北京：清华大学出版社，2011.

[5] 苏仕华.数据结构课程设计. 机械工业出版社