

程序设计报告

（ 2016 / 2017 学年 第 二 学期）

题 目： **店家商品信息及订单管理系统**

**专 业 数字媒体技术**

**组长 学号姓名 B16150406 吴嘉丽**

**组员 学号姓名 B16150405 杨琬婷**

**指 导 教 师 薛景**

**指 导 单 位 计算机软件教学中心**

**日 期 2016 年 6月 23 日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **成员分工** | **组长（吴嘉丽）** | **负责店家商品信息管理系统程序设计以及登陆界面程序设计，程序设计报告的整体记录和修改。** | | | | | |
| **组员（杨琬婷）** | **负责订单管理系统程序设计，程序设计报告内容的记录和修改。** | | | | | |
| **评分细则** | **评分项** | **优秀** | | **良好** | **中等** | | **差** |
| **遵守机房规章制度** |  | |  |  | |  |
| **上机时的表现** |  | |  |  | |  |
| **学习态度** |  | |  |  | |  |
| **程序准备情况** |  | |  |  | |  |
| **程序设计能力** |  | |  |  | |  |
| **团队合作精神** |  | |  |  | |  |
| **课题功能实现情况** |  | |  |  | |  |
| **算法设计合理性** |  | |  |  | |  |
| **用户界面设计** |  | |  |  | |  |
| **报告书写认真程度** |  | |  |  | |  |
| **内容详实程度** |  | |  |  | |  |
| **文字表达熟练程度** |  | |  |  | |  |
| **回答问题准确度** |  | |  |  | |  |
| **简短评语** | **教师签名：**  **年 月 日** | | | | | | |
| **评分等级** | **B16150406 吴嘉丽** | | **B16150405 杨琬婷** | | |  | |
|  | |  | | |  | |
| **备注** | **评分等级共五种：优秀、良好、中等、及格、不及格** | | | | | | |

**店家商品信息及订单管理系统**

1. **课题内容和要求**

该系统要求编写一个综合的外卖店家信息及订单管理系统，可实现店家商品信息 记录管理和·订单信息记录管理两个功能。

（1）要求能够管理若干个外卖店家的菜品信息，需要实现以下功能，实现对外卖店家 信息的管理。

系统内的所有信息必须以文件的方式存储在硬盘中，例如：

商品编号，商品名称，商品销量。格式如下：

1 芝士牛排饭团 772

2 咸蛋黄卷 309

3 粗薯条 30

……

（2）要求能够管理若干个订单信息，需要实现以下功能，实现对买家订单信息的管理。

系统内的所有信息必须以文件的方式存储在硬盘中，例如：

配送日期，配送商品，商品编号，配送地点，配送数量，交易金额格式如下：

2017.6.1 芝士牛排饭团 1 南京财经大学 1 15

2017.6.15 咸蛋黄卷 2 南京农业大学 2 28

2017.6.18 粗薯条 3 南京邮电大学 1 18

……

1. **相关理论知识**

描述该课题的相关理论知识和要点。

1. 商品信息管理
2. 在开头用typedef定义一个结构体类型的同时给出新别名，方便下面的变量声明，之后定义结构体数组方便结构数组成员的访问
3. 在菜单操作中使用选择结构if～else进行对用户输入选项进行判断。
4. 商品信息登记模块，使用for循环结构实现多个商品信息一次性登记。
5. 商品信息删除模块，使用if～else和for循环嵌套，if～else用于操作选择判断，如果选择全部删除则利用for循环将所有信息一次性删除，如果只删除一条信息使用if和for嵌套根据管理员输入关键字进行信息筛选。
6. 商品信息浏览模块，用if～else先判断系统内有无商品信息，再用for循环将所有商品信息输出。
7. 商品信息查找模块，if实现三种不同关键字查找，for对系统中信息进行筛选，最后输出。
8. 根据商品销量进行排序模块，使用了结构体指针，定义一个结构体指针后使其指向开头定义的结构题进行销量的比较，然后用选择排序法进行排序。
9. 系统信息存储，定义文件指针创建文件之后用fwrite函数指向该文件后for循环将所有数据块写入。
10. 系统信息导出，应用文件打开将存储时定义的文件指针导出文件内数据信息，if判断是否有该文件存在。
11. 最后菜单界面用选择结构switch对选项输入判断选择对应程序应用模块进行操作。

2.订单信息管理

（1）在添加信息部分，利用了条件语句if，完成了对输入月份的规范性检 查，实现将新建的订单加入订单链表，并使链表头指向新添加的订单节点。最后使用 自加运算符，实现订单数量加一。

（2）在显示信息部分，运用了循环语句for，使订单信息从订单链表的开头 显示至结束。

（3）在查询信息部分，先多次运用了条件语句if，if...else等，使得用户可以 选择“按 名称查找”或“按编号查找”。然后调用了比较函数strcmp，实现找到订单， 并输出其详细信息。

（4）在删除信息部分，利用了循环语句for来确认所待删除的信息已存储在 物流信息系统中。然后使用条件语句if...else来查找所待删除的订单信息是否为订单 列表中的第一个订单，若不是，则调用条件语句if来找到对应的订单。并在这个 条件语句中镶嵌一个条件语句if...else，实现对所选订单的删除。

（5）在存储信息部分，利用了fopen函数来实现打开存储物流信息的文件， 利用了fputc函数来实现将输入字符写入文件中，利用了fgetc函数来实现从文件中读 取一个字符，利用了fclose来实现关闭打开的文件。

（6）在main函数部分，利用了switch...case语句来实现对物流信息的操作选 择。

1. **需求分析**

提供以下几个基本功能：

（1）系统内的相关信息文件由程序设计人员预先从键盘上录入，文件中的数据记录不 得少于20条；

（2）设计并实现系统的相关界面，使用分级菜单提高人机交互性；

（3）可以添加/删除/修改管理对象的详细信息，并将改动反馈到数据文件中；

（4）统计分析店家商品信息：

⑴、xx店家商品查询：

1. 输入一个商品编号，查出此商品的相关信息并显示输出。
2. 输入一个商品名称，查出此商品的相关信息并显示输出。
3. 输入相应的商品排名，查出对应信息并显示输出。

⑵、订单状况查询：

1. 输入一个订单编号，查出此订单的相关信息并显示输出。
2. 输入一个商品名称，查出此订单的相关信息并显示输出。

（5）排序功能：能实现由用户选择按各项数据升序或降序排序对查询出的信息进行显 示。

**四、概要设计**

（1）程序的总体框图如下：

添加订单信息

修改订单信息

删除订单信息

订单信息管理

添加商品信息

修改商品信息

删除商品信息

商品基本信息管理

店家商品及订单信息管理

维护销量信息

根据销量排

名

商品销售信息统计

显示基本信息

管理员登录模块

图1店家商品信息及订单管理系统的功能模块图

2）大体数据结构：

依据给定的店家信息，设计每个店家对应的结构类型定义如下：

struct Food {

int number; //商品编号

char name[20]; //商品名称

int sale; //商品销量

}

依据给定订单信息，设计每个订单信息对应的结构类型定义如下：

struct article {

long num1; //订单号

char name[20]; //商品名称

char dest[20]; //配送地点

int num2; //配送个数

float money； //金额

struct date deliver； //配送日期

}

2 、类中主要方法的实现需要有流程图

图2-1注册登录方法流程图

进入功能选择界面

终止

否

开始

输入注册登陆用户名

是

用户名是否有效

输入注册登录密码

否

是

密码是否有效

图2-2查找商品方法流程图

显示找到商品信息

终止

否

是

关键字是否有效

输入关键字

开始

查找功能选择

按编号查找

按销量查找

按名称查找

查询选择

物品名称

物品编号

是否存在

开始

是否存在

终止

显示记录

显示记录

图2-3订单物流查找方法流程图

否

否

是

是

**五、源程序代码**

（代码需要有注释。）

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <conio.h>

#define N 1000

//订单链表

typedef struct food

{

int number; //物品编号

char name[20]; //物品名称

int sale; //销量

int sum; //总数

}FOOD;

FOOD food[N];

int shuliang=0;

//打印主菜单

void menu();

//继续操作提示

void fhzjm()

{

char biaozhi[20]; //提示响应

printf("\n");

printf("还需要操作么？如果需要操作请输入：yes，否则请输入：no\n");

scanf("%s",biaozhi); //是否需要操作

if(strcmp(biaozhi,"yes")==0) //继续操作

{

menu();

}

else if(strcmp(biaozhi,"no")==0) //退出操作

exit(0); //正常退出

else

{

printf("请输入正确的字符，谢谢！\n");

fhzjm();

}

}

//添加订单

void DengJi()

{

int rs;

int i,k=1;

system("CLS"); //清屏

printf("请输入需要输入几个商品信息：");

scanf("%d",&rs); //输入数量

for(i=shuliang;i<shuliang+rs;i++,k++)

{

printf("请输入第%d个商品的编号：",k);

scanf("%d",&food[i].number); //输入编号

printf("请输入商品的名称：");

scanf("%s",food[i].name); //输入名称

printf("请输入销量：");

scanf("%d",&food[i].sale); //输入销量

}

shuliang=shuliang+rs;

fhzjm();

}

//删除订单

void ShanChu()

{

char shanchuinfo[10];

system("CLS"); //清屏

printf("删除全部商品信息请输入\"all\"，删除指定编号的商品信息请输入\"one\"\n");

scanf("%s",shanchuinfo);

if(strcmp(shanchuinfo,"all")==0) //删除全部商品信息

{

int j;

printf("你删除的商品信息如下：\n");

printf("-----------编号-------------商品--------------销量\t\n");

for(j=0;j<shuliang;j++)

printf("----%d-------%s-------%d\t\n",food[j].number,food[j].name,food[j].sale);

shuliang=0;

printf("删除成功\n\n");

}

else if(strcmp(shanchuinfo,"one")==0) //删除指定编号的商品信息

{

struct food \*p=NULL; //指向一个空指针

int choice;

int i,j,k=0;

printf("请输入你要删除的商品的编号:");

scanf("%d",&choice);

//在系统中查找指定编号

for(i=0;i<shuliang;i++)

{

if(choice==food[i].number) //找到指定编号

{

k=1;j=i;break; //结束

}

}

if(k) //k为非零就执行

{

if(shuliang==1) //数量为1

{

p=&food[0];

free(p); //释放p

shuliang=0;

}

else //删除指定信息

{

for(i=j;i<shuliang;i++) //查找待删除信息

{

//用下一个信息覆盖待删除信息

food[i]=food[i+1];

}

shuliang=shuliang-1;

}

printf("删除成功\n\n"); //信息删除成功

}

else //否则

{

printf("输入数据错误！\n"); //信息删除失败

}

}

fhzjm();

}

//浏览订单

void LiuLan()

{

int i;

system("CLS"); //清屏

if(shuliang==0) //订单信息不在系统中

{

printf("系统里面没有任何商品的信息！\n");

}

else //订单信息存储在系统中

{

for(i=0;i<shuliang;i++)

{

printf("第%d个商品信息为:\t\n",i+1);

printf("-----------编号-------------商品--------------销量\t\n");

printf("------------%d--------------%s--------------%d\t\n",food[i].number,food[i].name,food[i].sale);

}

}

fhzjm();

}

//查找订单

void ChaZhao()

{

int xx;

char choice,yy[20]; //名称

int i,j,k=0;

system("CLS"); //清屏

if(shuliang==0) //订单信息不在系统中

{

printf("系统里面没有任何商品的信息！\n");

fhzjm();

}

printf("三种查找方式：编号，名称，销量\n");

printf("如果按编号查找请输1\n如果按名称查找请输2\n如果按销量查找请输3\n");

printf("请输入您查找的方式：");

scanf("%s",&choice);

if(choice=='1') //按编号查找

{

printf("请输入需要查找商品的编号：");

scanf("%d",&xx); //输入需要查找商品的编号

printf("您所查找的商品的信息为：\n");

printf("----编号----名称----销量----\t\n");

//查找指定编号

for(i=0;i<shuliang;i++)

{

if(xx==food[i].number)

{

j=i;k=1; //找到指定商品信息

printf("----%d-------%s-------%d----\t\n",food[j].number,food[j].name,food[j].sale);

}

}

if(k==0) //k为0

printf("输入信息有误：\n");

}

else if(choice=='2') //按名称查找

{

printf("请输入需要查找商品的名称：\n");

scanf("%s",yy); //输入需要查找商品的名称

printf("您所查找的商品的信息为：\n");

printf("----编号----名称----销量----\t\n");

for(i=0;i<shuliang;i++)

{

if(strcmp(yy,food[i].name)==0) //找到订单，输出其详细信息

{

j=i;k=1;

printf("----%d-------%s-------%d----\t\n",food[j].number,food[j].name,food[j].sale);

}

}

if(k==0)

printf("输入信息有误：\n");

}

else if(choice=='3') //按销量查找

{

printf("请输入需要查找商品的销量：\n"); //输入需要查找商品的销量

scanf("%d",&xx);

printf("您所查找的商品的信息为：\n");

printf("----编号----名称----销量----\t\n");

//查找指定商品的信息

for(i=0;i<shuliang;i++)

{

if(xx==food[i].sale)

{

j=i;k=1; //找到指定商品信息

printf("----%d-------%s-------%d----\t\n",food[j].number,food[j].name,food[j].sale);

}

}

if(k==0) //k为0

printf("输入信息有误：\n");

}

fhzjm();

}

//销量排序

void PaiXu()

{

struct food\*p1[N],\*\*p2,\*temp;

int i,j;

system("CLS"); //清屏

//对销量进行排序

p2=p1;

for( i=0;i<shuliang;i++)

{

p1[i]=food+i;

}

for( i=0;i<shuliang;i++) //第一个销量

{

for( j=i+1;j<shuliang;j++) //第二个销量

{

if((\*(p2+i))->sum<(\*(p2+j))->sum) //销量比较

{temp=\*(p2+i);\*(p2+i)=\*(p2+j);\*(p2+j)=temp;} //交换销量顺序

}

}

printf("按照销量排序之后的信息为：\n");

printf("--编号--名称--销量--\t\n");

for( i=0;i<shuliang;i++) //遍历销量排序之后的订单

{

food[i].sum=food[i].sale;

printf("--%d--%s--%d--\n",(\*(p2+i))->number,(\*(p2+i))->name,(\*(p2+i))->sum);

}

fhzjm();

}

//存储订单链表到文件

void CunChu()

{

int i;

FILE \*rs;

if((rs=fopen("food.txt","w"))==NULL) //文件打开失败

{

printf("not open");

exit(0); //正常退出

}

for(i=0;i<shuliang;i++) //遍历订单

{

fwrite(&food[i], sizeof(food[i]), 1, rs); //写入文件

}

if(ferror(rs)) //数据流错误

{

fclose(rs); //关闭打开文件

perror("写文件失败！\n");

return;

}

printf("存储文件成功！\n");

fclose(rs); //关闭打开文件

fhzjm();

}

//显示订单

void DaoChu()

{

struct food t;

int i=0;

FILE\* fp = fopen("food.txt", "r"); //以只读方式打开文件

shuliang=0;

if(NULL==fp) //文件打开失败

{

perror("读取文件打开失败！\n");

return;

}

memset(food,0x0,sizeof(food)); //将food的空间全部置0

while(1) //无线循环

{

fread(&t,sizeof(t),1,fp); //从文件中读取数据

if(ferror(fp)) //数据流错误

{

fclose(fp); //关闭打开文件

perror("读文件过程失败！\n");

return;

}

if(feof(fp)) //文件结束

{

break;

}

food[i]=t;

i++;

}

fclose(fp); //关闭打开文件

shuliang=i;

printf("导出文件成功！\n");

fhzjm();

}

//打印主菜单

void menu()

{

int n=0;

system("CLS");

printf(" 店家商品信息管理系统\n");

printf("---------MENU--------\n");

printf(" 1.登记商品信息\n");

printf(" 2.删除商品信息\n");

printf(" 3.浏览所有已经登记的商品\n");

printf(" 4.查找\n");

printf(" 5.根据销量排序\n");

printf(" 6.存储到文件\n");

printf(" 7.从文件导出\n");

printf(" 8.退出系统\n");

a: printf(" 请选择：");

scanf("%d",&n);

switch (n) //系统调用功能

{

case 1:

DengJi();break;

case 2:

ShanChu();break;

case 3:

LiuLan();break;

case 4:

ChaZhao();break;

case 5:

PaiXu();break;

case 6:

CunChu();break;

case 7:

DaoChu();break;

case 8:

exit(0);break;

default:

{

printf("请输入1-8之间的数字，谢谢！\n");

goto a;

}

}

}

//日期

struct date

{ int year;

int month;

int day;

};

//订单链表

struct article

{ char name[20]; //物品名称

int num1; //物品编号

char dest[20]; //目的地

int num2; //物品数量

float money; //配送金额

struct date deliver; //配送日期

struct article \*next; //指向上一次添加的订单

};

struct article \*head; //存储订单信息

int Recordcount; //订单数量

//添加订单

void add()

{

system("cls"); //清屏

struct article \*p=NULL; //新建一个订单节点

p=(struct article\*)malloc(sizeof(struct article)); //为新建的订单节点分配内存空间

printf("请输入配送年份:");

scanf("%d",&p->deliver.year);

printf("请输入配送月份:");

scanf("%d",&p->deliver.month);

if((p->deliver.month>0)&&(p->deliver.month<13)) //输入月份的规范性检查

{

printf("请输入配送日期:");

scanf("%d",&p->deliver.day);

printf("请输入物品名称:");

scanf("%s",p->name);

printf("请输入物品编号:");

scanf("%s",&p->num1);

printf("请输入物品配送地:");

scanf("%s",p->dest);

printf("请输入配送个数:");

scanf("%d",&p->num2);

printf("请输入配送金额:");

scanf("%f",&p->money);

p->next=head; //将新建的订单加入订单链表

head=p; //链表头指向新添加的订单节点

Recordcount++; //订单数量加一

}

system("PAUSE");

}

//显示所有订单的详细信息

void show()

{

system("cls");

//遍历订单链表

struct article \*p=NULL;

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("名称\t编号\t产地\t目的地\t模式\t数量\t金额\t日期\n");

for(p=head;p!=NULL;p=p->next)

{

printf("%s\t",p->name);

printf("%d\t",p->num1);

printf("%s\t",p->dest);

printf("%d\t",p->num2);

printf("%f\n",p->money);

printf("%d\t",p->deliver.year);

printf("%d\t",p->deliver.month);

printf("%d\t",p->deliver.day);

printf("\n");

}

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

//查找订单

void search()

{

system("cls");

char namesea[30];//名称

int type;//编号

int choice;

struct article \*p=NULL;

printf("1按姓名查找\n2按编号查找\n ");

printf("请输入你的选择：");

scanf("%d",&choice);

while(0 == scanf("%d",&choice) || (choice!=1 && choice!=2))//输入选择的规范性检查

{

while('\n' != getchar()) { }

printf("输入无效，请重新选择：");

}

if(choice==1)//按物品名称查找订单

{

printf("请输入物品名称：");

scanf("%s",namesea);

for(p=head;p!=NULL;p=p->next)

if(strcmp(p->name,namesea)==0)//找到订单，输出其详细信息

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("名称\t编号\t目的地\t数量\t金额\t日期\n");

printf("%s\t",p->name);

printf("%d\t",p->num1);

printf("%s\t",p->dest);

printf("%d\t",p->num2);

printf("%f\n",p->money);

printf("%d\t",p->deliver.year);

printf("%d\t",p->deliver.month);

printf("%d\t",p->deliver.day);

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

}

else//按物品编号查找订单

{

printf("请输入物品编号\n ");

scanf("%d",&type);

for(p=head;p!=NULL;p=p->next)

if(p->num1==type)//找到订单，输出其详细信息

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"); printf("名称\t编号\t目的地\t数量\t金额\t日期\n");

printf("%s\t",p->name);

printf("%d\t",p->num1);

printf("%s\t",p->dest);

printf("%d\t",p->num2);

printf("%f\n",p->money);

printf("%d\t",p->deliver.year);

printf("%d\t",p->deliver.month);

printf("%d\t",p->deliver.day);

printf("\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

}

system("PAUSE");

}

//根据物品名称和配送金额删除对应订单信息

void delete\_rec()

{

system("cls");

char name[20];

float money;

int choice;

struct article \*p=NULL,\*q=NULL;

p=head;

q=head;

printf("请输入要删除的物品名称：");

scanf("%s",name);

printf("请输入金额：");

scanf("%f",&money);

for(;q!=NULL;q=q->next)

{

if((head->money==money)&&(strcmp(head->name,name)==0))//要删除的订单为订单列表中的第一个订单

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"); printf("名称\t编号\t目的地\t数量\t金额\t日期\n");

printf("%d\t",q->deliver.year);

printf("%d\t",q->deliver.month);

printf("%d\t",q->deliver.day);

printf("%d\t",q->num1);

printf("%s\t",q->name);

printf("%s\t",q->dest);

printf("%d\t",q->num2);

printf("%.2f\t",q->money);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("确认删除？\n 1是\n 2否\n");

scanf("%d",&choice);

while(0 == scanf("%d",&choice) || (choice!=1 && choice!=2))//输入选择的规范性检查

{

while('\n' != getchar()) { }

printf("输入无效，请重新选择：");

}

if(choice==1)//删除

{

head=q->next;

Recordcount--;

}

else break;//取消删除

}

else//要删除的订单不是订单列表中的第一个订单

{

if((q->money==money)&&(strcmp(q->name,name)==0))//找到对应的订单进行删除

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("名称\t编号\t目的地\t数量\t金额\t日期\n");

printf("%d\t",q->deliver.year);

printf("%d\t",q->deliver.month);

printf("%d\t",q->deliver.day);

printf("%d\t",q->num1);

printf("%s\t",q->name);

printf("%s\t",q->dest);

printf("%d\t",q->num2);

printf("%.2f\t",q->money);

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("确认删除？\n 1是\n 2否\n");

scanf("%d",&choice);

while(0 == scanf("%d",&choice) || (choice!=1 && choice!=2))//输入选择的规范性检查

{

while('\n' != getchar()) { }

printf("输入无效，请重新选择：");

}

if(choice==1)//删除

{

p->next=q->next;

Recordcount--;

}

else break;//取消删除

}

}

}

}

//存储订单链表到文件

void logistic()

{

char ch;

FILE \*fp;

char filename[10];

scanf("%s",filename); //写入文件头

if((fp=fopen(filename,"w"))==NULL)

//利用文件指针打开文件来写，如果没有就创建一个，如果打开失败，就进行if里面的操作。打开后，fp指向文件第0个数据

{

printf("cannot open file\n");

exit(0);

}

while(ch=getchar()!='\n') //遍历订单链表，将各个订单信息逐次写入文件

fputc(ch,fp);

fclose(fp);

ch=fgetc(fp);

while(ch!=EOF)

{ putchar(ch);

ch=fgetc(fp);

}

fclose(fp); //关闭打开文件

}

//打印主菜单

void order()

{ int n,b=1,s;

struct article \*head=NULL;

void add();

void show();

void logistic();

void search();

void change();

void delete\_rec();

while(b==1)

{

system("cls"); //清屏

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("欢迎使用商品配送管理系统\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("1:添加记录\n");

printf("2:显示记录\n");

printf("3:查询记录\n");

printf("4:删除记录\n");

printf("5:退出\n");

printf("请输入您的选择:\n");

scanf("%d", &n);

switch(n) //系统功能调用

{case 1: add();break;

case 2: show();break;

case 3: search();break;

case 4: delete\_rec();break;

case 5: exit(0);break;

default :printf("error\n");}

printf("是否返回主菜单? 0是 1否\n");

scanf("%d",&s);b=b+s;

}

}

int main()

{

char nam[N];

char password[N]; //用于存放密码

char shop[N];

char key[N]; //用于保存确认密码

int tryAgain = 0; //重试的次数

//创建管理系统

do

{

printf("----------创建用户---------\n");

printf("请输入你的用户名：");

scanf("%s", nam);

printf("请输入你的管理密码：");

scanf("%s", password);

printf("请确认你的管理密码：");

scanf("%s", key);

}

while(strcmp(password, key)); //字符串比较函数，

printf("密码设置成功！\n您的用户名为%s\n你的密码为%s\n", nam,password);

printf("按任意键开始登录系统.....\n");

getch(); //阻塞函数，可以让程序停留在此，直到我们按下键盘上的某个按键

system("cls"); //清屏

//登录系统

printf("----------登录系统---------\n");

printf("请输入用户名：");

while(scanf("%s", shop), strcmp(nam, shop))

{

tryAgain++; //累加密码重试的次数

//密码重试的次数限制在三次以内

if(tryAgain < 3)

printf("错误%d次！请重新输入：", tryAgain);

else

break; //输入密码次数大于等于三次，退出循环

}

//输出结果

system("cls");

//两种情况 ：1输入了三次错误的密码 2输入了正确的密码

if(tryAgain >= 3)

{

printf("对不起，3次密码错误，系统退出！\n");

exit(-1); //退出程序

}

else

{

printf("请输入密码：");

while(scanf("%s", key), strcmp(password, key))

{

tryAgain++; //累加密码重试的次数

//密码重试的次数限制在三次以内

if(tryAgain < 3)

printf("错误%d次！请重新输入：", tryAgain);

else

break; //输入密码次数大于等于三次，退出循环

}

//输出结果

system("cls");

//两种情况 ：1输入了三次错误的密码 2输入了正确的密码

printf("----------输出结果---------\n");

if(tryAgain >= 3)

{

printf("对不起，3次密码错误，系统退出！\n");

exit(-1); //退出程序

}

else

{

printf("欢迎你，管理员！\n");

printf("现在进入功能选择\n"); //功能选择

lop:

printf("输入1进入商品信息管理系统\n");

printf("输入2进入订单管理系统\n");

printf("输入3退出程序\n");

int GN;

scanf("%d",&GN);

switch (GN)

{

case 1:

menu();break;

case 2:

order();break;

case 3:

exit(0);break;

default:

{

printf("请输入1-3之间的数字，谢谢！\n");

goto lop;

}

}

}

}

return 0;

}

**六、测试数据及其结果分析 预期**

**1.运行进入创建用户界面输入用户名和密码创建成功：**

**预期：输入用户名和密码创建成功后即可开始登陆。**

**结果：输入用户名和密码以及再一次密码确认后提示设置成功即可开始登录（如图3-1）**

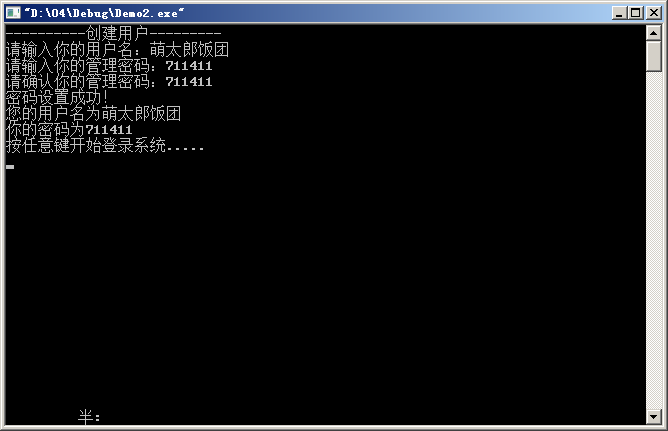


图3-1创建用户

1. 登陆成功后输出功能选择菜单可进行功能选择：

预期：可以选择进入商品信息管理系统或是进入订单管理系统。

结果：可以进行功能选择，输入1进入商品信息管理系统，输入2进入订单管理系统，

输入3直接退出程序（如图3-2）

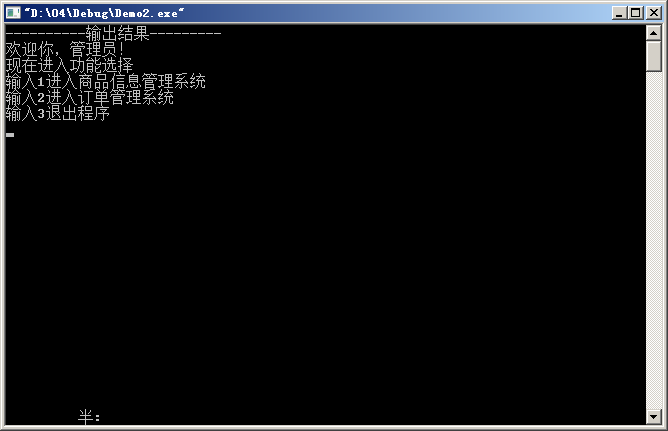


图3-2功能选择

1. 进入店家商品信息管理系统显示该系统功能菜单：

预期：菜单显示能够实现商品登记，浏览，删除，查找以及排序功能。

结果：菜单显示有8个操作功能，1登记商品信息；2删除商品信息；3浏览登记记录；

4关键字查找；5根据销量排序；6存储信息到文件；7导出文件信息；8退出系统。

（如图3-3）



图3-3商店信息管理系统

1. 商品信息登记功能：

预期：可以登记商品的编号，名称和销量信息。

结果：可以一次性登记多条商品信息，系统显示第几个商品的编号，名称和销量，依次输入信息，全部登记完成后可进行下一步操作。（如图3-4）

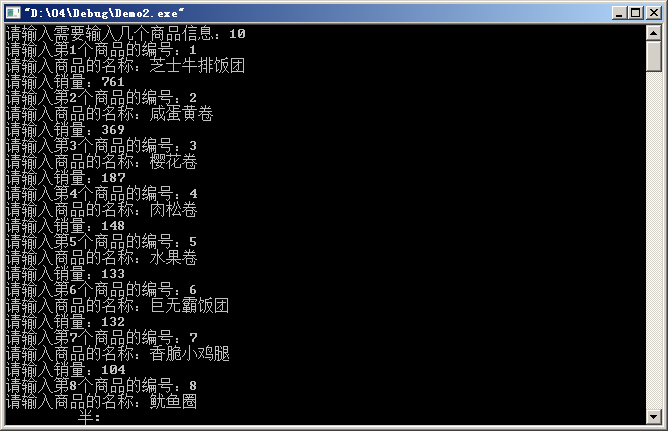


图3-4登记商品信息

1. 商品信息浏览：

预期：登记的商品信息一条条有序排列输出，清晰可观。

结果：系统输出按登记时的顺序显示第几个商品信息，然后横向依次按编号，商品，销量显示。（如图3-5）

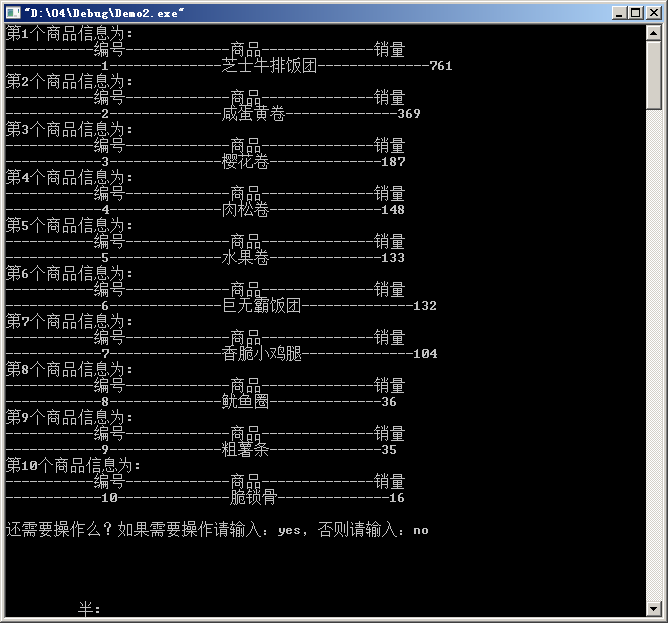


图3-5商品信息浏览

1. 按关键字查找商品信息：

预期：可以由编号，名称，销量三个关键词查找出对应信息。

结果：输入4进入查找功能，系统提示三种查找方式，按编号查找输1；按名称查找输2；按销量查找输3。输入2按名称查找，然后输入想要查找的商品名称即可显示所要查找的商品信息，显示该商品完整信息。（如图3-6）

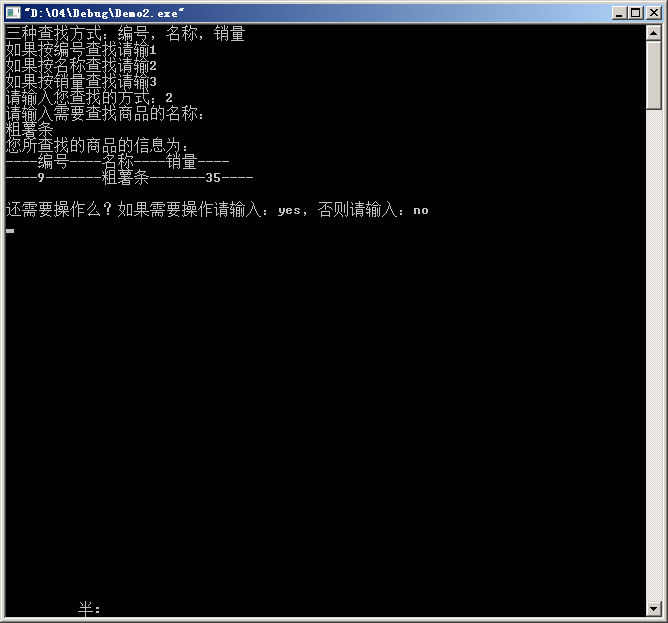


图3-6按名称查找商品信息

1. 按照商品销量进行排序：

预期：能够根据商品的销量进行排序。

结果：按商品销量从大到小降序排列输出商品信息。（如图3-7）



图3-7按销量排序

1. 删除商品信息：

预期：能够根据关键字找出要删除的信息将其删除。

结果：只能根据商品编号查找要删除的信息将其删除，且删除界面不能显示该商品信息，只能在删除后在浏览功能中查看。（如图3-8和3-9）



图3-8按编号删除商品信息



图3-9删除后的记录浏览

9．信息的存储和导出：

预期：能够将登记好的信息储存在一个文件中，下次运行时能够从文件中导出。

结果：储存成功后关闭系统，再次打开该系统进行导出，导出文件成功即可显示登记过的商品信息。（如图3-10，3-11）

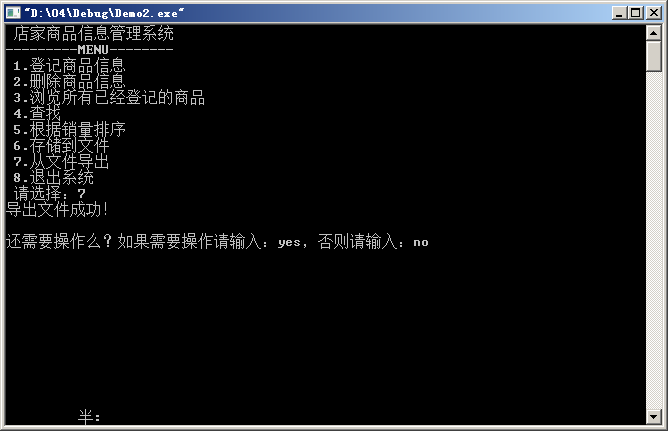


图3-10文件导出



图3-11导出的商品信息

10.物流配送管理系统主界面，用户可对物流信息进行一系列的操作：

预期：在主界面中，用户可以实现对物流信息进行管理，如添加记录、查询记录、显示记录、删除记录和退出等。

结果：如图及所示，系统除了满足实验预期中用户对物流信息的管理以外，还对主界面进行了美化，增强了用户的使用体验。（如图3-12）

****

**图3-12物流配送系统界面**

11.物流订单添加：

预期：在添加信息界面中，用户可以输入外卖物品的名称、配送地、日期，以完成对物流信息的添加。

结果：如图及所示，用户可以在物流信息系统中输入外卖物品的名称、编号、配送地、日期、数量、金额等。较实验预测相比，系统功能更加完善。（如图3-13）

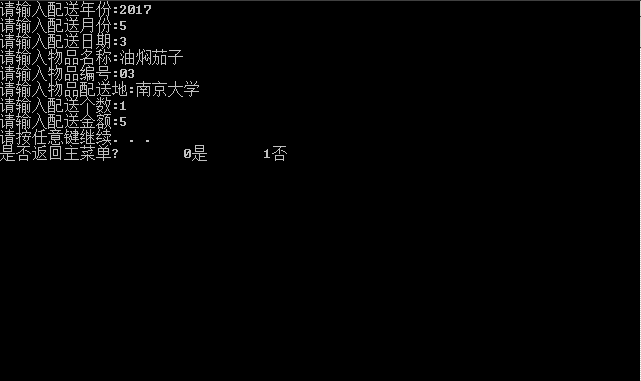


图3-13订单登记

12.查看已录入的物流订单信息：

预期：在显示信息界面中，用户可以查看自己录入的物流信息，如商品名称、商品配送地、配送日期等。

结果：如图及所示，用户可以查看已录入的商品物流信息，如商品名称、商品编号、商品配送地、配送日期、配送个数、配送金额等。（如图3-14）

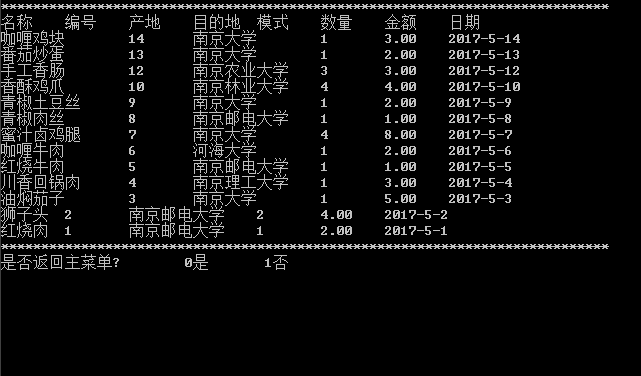


图3-14查看物流订单信息

13. 按照名称或编号查找物流订单，并查看其详细信息

预期：在查询信息界面中，用户可以查询已存储在物流信息系统中的商品信息，如商品名称、商品编号、商品配送地、配送日期、配送个数、配送金额等。

结果：如图及所示，在满足用户需求的基础上，系统提供了“按名称查找”和“按编号查找”两种查找方式，以满足用户的不同需求。（如图3-15）



图3-15关键字查找订单信息

14. 删除已经录入的物流订单信息

预期：在删除信息界面中，用户可以通过输入已存储在物流信息系统中的物品名称和金额来删除相应的物流信息。

结果：实验结果完全符合预期。（如图3-16）



图3-16删除指定订单信息

**七、调试过程中的问题 问题和解决方案**

1.问题：整个程序找不到主函数main，无法运行。

解决方案：找到程序起点给主函数main加上系统保留字int，完成完整的主函数定义。

2.问题：存储后的信息无法导出，一直找不到文件。

解决方案：查看存储文件和打开文件是否是同一个文件名，将文件名统一后即可导出储存的文件信息。

3.问题：在我第一次调试程序时，VC反馈了多条“missing storage-class or type specifiers”这个错误。

解决方案：我通过查阅上学期的C语音课本，发现是程序中的结构体出现了问题，我在变量名前遗漏了类型，当我按照定义结构变量的一般格式补全类型以后，这部分的程序就运行成功了。

4.问题：在我调试查找订单的程序时，VC反馈了一条“undeclared identifier”的错误。

解决方法：我按照VC反馈的错误行数，查找到是我错将else写为Else。这只是一个极细微的错误，但它却使整个程序运行失败。当我按照规范将else输入正确以后，程序运行成功。

5.问题：在我调试删除订单的程序时，VC反馈了“left operand must be l-value“的错误。

解决方法：我按照VC的提示查找程序错误时，却始终没有找到错因。于是我从程序开头看起，逐条分析程序的含义，最终发现因为我错将“==”写成了“=”导致了程序运行失败。究其原因，是因为我混淆了“==”与“=”的含义。当我改正程序以后，程序运行成功。

**八、课程设计总结**

我们在对店家商品信息及订单信息管理系统的编写中遇到了不少困难。在对物流配送信息系统进行分析的时候，忽略了很多功能上的问题，比如最初只是简单的设想系统包含三个功能，即物理流信息存储、物流信息显示和退出。在完成代码的过程中，逐渐丰富了系统功能，加入了查询信息和删除信息。在完成代码的过程中，发现自己已经忘记了C语言的很多知识，比如在使用结构体struct时，忘记了结构体struct的形式，后来通过复习书本内容，才顺利完成了代码。程序运行过程中，代码也出现了很多问题，其中就包括因为混淆了“=”和“==”的作用而导致程序无法运行，后来采用了分段检测代码的方式，最后将错误一一排查，完成了正确的代码。通过这些问题以及对问题的解决，我们再次感受到C语言的严谨性，即使是一点细微之处的错误都能导致整个代码的运行失败。此外还从完成代码的过程中感受到了C语言对个人思维的锻炼，这使得我在测试代码时必须保持理性和冷静，使个人得到了锻炼。

本次程序设计设计对与我们来说是一大挑战，上学期第一次接触了c语言和编程，能够进行简单的程序进行编写，实现一些简单的功能已是觉得小有建树，当真正开始接触比较正规的系统程序设计时才发现自己的知识和能力水平远远不够，编程需要强大的逻辑思维能力和丰富的相关知识掌握，以及要有极大的耐心和毅力，要在一次次运行失败中寻找错误并修改，有时甚至会被一个问题困扰许久无法解决，当发现自己能力的局限性时要善于转换思维方式或是及时请教他人，不要造成时间上的浪费，最终仍无收获。