#include<stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <malloc.h>

#include <windows.h>

#include <time.h>

#define MAXSIZE 100

typedef struct //数据域定义为一个结构体

{

char name[30]; //商品名称

char number[30]; //商品编号

char brand[30]; //商品品牌

char putdata[30]; //商品入库时间

float price; //商品单价

int num; //商品数量

}Goods; //商品结构体

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; //LNumber类型的单链表

struct UserInfo //用户的相关信息的结构体

{

char UserName[30]; //用户名

char Password[30]; //用户密码

};

LNumber\* Init\_List(LNumber\* L); //初始化链表

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

void Browse(LNumber\* L); //浏览商品信息

LNumber\* Search(LNumber\* L, Goods x); //查找商品

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

void Out(LNumber\* L, Goods x); //商品出库

void Save(LNumber\* L); //保存操作

void Menu(); //功能选择窗口

void LogIn(struct UserInfo userlogin[], char tmp[]); //用户登录

void Continue(); //页面优化

//#define \_CRT\_SECURE\_NO\_WARNINGS

#include"goods.h"

/\*初始化链表\*/

LNumber\* Init\_List(LNumber\* L)

{

L->next = NULL; //初始化置为空

return L;

}

/\*录入商品信息\*/

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;

printf("\n请输入商品的种数:");

scanf("%d", &n);

printf("\n \*\*\*\* 请输入商品的名称、编号、数量、品牌、单价、入库时间 \*\*\*\*\n");

printf("\n");

for (i = 1; i <= n; i++) //循环n次进行创建单链表

{

printf(" 商品名称： ");

scanf("%s", x.name);

printf(" 商品编号： ");

scanf("%s", x.number);

printf(" 商品数量： ");

scanf("%d", &x.num);

printf(" 商品品牌： ");

scanf("%s", &x.brand);

printf(" 商品单价： ");

scanf("%f", &x.price);

printf(" 商品入库时间： ");

scanf("%s", &x.putdata);

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

printf("\n");

}

printf("\n \*\*\*\* 录入成功，返回上一层 \*\*\*\*\n");

}

/\*浏览商品信息清单（链表的遍历和输出）\*/

void Browse(LNumber\* L)

{

LNumber\* p = L->next; //确定起始位置

if (p == NULL)

{

printf("--------------该系统还未录入商品信息，请录入后在进行操作--------------\n");

Menu(); //返回主菜单

}

printf("\n--------------------------------商品信息一览表------------------------------\n");

printf("\n");

printf("商品名称 商品编号 商品数量 商品品牌 商品单价 商品入库时间\n");

printf("\n");

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;

}

printf("\n");

printf("\n");

}

/\*查找商品信息（按元素值在链表中的查询）\*/

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;

}

/\*插入新增的商品（链表的插入）\*/

void GoodsInsert(LNumber\* L, Goods x)

{

LNumber\* p = L, \* s;

while (p->next != NULL && (strcmp(p->next->data.number, x.number) != 0)) //商品不存在时可插入到已有链表上

p = p->next;

s = (LNumber\*)malloc(sizeof(LNumber));

s->data = x;

s->next = p->next;

p->next = s;

}

/\*商品信息的排序\*/

void GoodsSort(LNumber\* L)

{

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

int choose;

Goods x;

printf("请选择排序方式：");

printf("1.按商品编号排序 2.按库存数量排序 \n");

scanf("%d", &choose);

switch (choose)

{

case 1: //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

while (p != NULL)

{

r = p->next; //r指针暂存p的后继

q = L;

while (q->next != NULL && (strcmp(q->next->data.number, p->data.number) > 0)) //查找插入位置

q = q->next; //将p节点链入链表

p->next = q->next;

q->next = p;

p = r;

}

break;

case 2: //按商品数量排序为冒泡排序

p = L->next;

while (p)

{

q = p->next;

while (q)

{

if (p->data.num <= q->data.num)

{

x = p->data;

p->data = q->data;

q->data = x;

}

q = q->next;

}

p = p->next;

}

break;

}

}

/\*商品入库\*/

void Storage(LNumber\* L, Goods x)

{

LNumber\* p;

p = Search(L, x); //查找是否存在该商品

if (p == NULL) //链表中没有这个商品时可直接插入

GoodsInsert(L, x);

else

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

}

/\*商品信息的修改\*/

void GoodsChange(LNumber\* L, Goods x) //以商品编号进行修改

{

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

if (p == NULL)

printf("\n仓库中无此商品！\n");

else

{

printf("\n 修改商品编号： ");

scanf("%s", NewNumber);

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

printf("\n 修改商品名称： ");

scanf("%s", NewName);

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

printf("\n 修改商品数量： ");

scanf("%d", &NewNum);

p->data.num = NewNum;

printf("\n 修改商品品牌： ");

scanf("%s", NewBrand);

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

printf("\n 修改商品单价： ");

scanf("%f", &NewPrice);

p->data.price = NewPrice;

printf("\n 修改商品入库时间： ");

scanf("%s", NewPutData);

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

printf("\n 成功修改！\n");

printf("\n");

}

}

/\*删除商品（删除链表元素）\*/

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;

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

{

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

{

p->next = p->next->next; //删除语句

flag = 1; //重置标志符

break;

}

}

if (p == NULL && flag == 0)

{

printf("\n 不存在此商品！");

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

Menu();

}

return 1;

}

/\*商品出库（模拟购买操作）\*/

void Out(LNumber\* L, Goods x)

{

int j;

LNumber\* p;

p = Search(L, x); //查找该商品

if (p == NULL)

printf("\n仓库中无此商品！\n");

else

{

printf("\n 您要购买的商品为：%s。\n", p->data.name);

if (p->data.num > x.num) //购买的商品数量少于库存量时

{

printf("\n购买成功!欢迎下次光临！\n");

p->data.num = p->data.num - x.num; //库存数=原库存数-购买数

printf("\n");

printf("\n该商品还有%d件。\n", p->data.num); //显示现在的库存

}

else if (p->data.num == x.num) //购买的商品数量等于库存量时

{

printf("\n购买成功!欢迎下次光临！\n");

Delete(L, x); //调用删除函数，系统显示已售空

printf("\n");

printf("\n该商品已售空，请及时补充库存。\n");

}

else if (p->data.num < x.num) //购买的商品数量多于库存量时

{

printf("\n该商品数量不足，仅剩%d。是否购买\n", p->data.num);

printf("(购买请按1，不购买请按0。)\n");

printf("请输入您的的选择：");

scanf("%d", &j);

if (j == 1)

{

Delete(L, x); //若选择购买，则把剩余的该商品全部买走

printf("\n购买成功!欢迎下次光临！\n");

printf("\n");

printf("\n该商品已售空，请及时补充库存。\n");

}

else

printf("\n欢迎下次光临！\n");

}

}

}

/\*保存操作数据\*/

void Save(LNumber\* L) //保存操作结果

{

LNumber\* p = L->next, \* q = L->next;

FILE\* fp; //文件指针

if (p == NULL)

{

printf("现在没有商品信息，请先输入信息再进行操作\n");

return;

}

fp = fopen("H:\\goods.txt", "w"); //在当前目录下打开Goods.txt文件并设置只允许“写”操作

if (!fp)

{

printf("文件不存在\n");

return;

}

/\*将商品信息写入文件\*/

fprintf(fp, "===========================商品信息一览表===========================\n");

fprintf(fp, "\n");

fprintf(fp, "商品名称\t商品编号\t商品数量\t商品品牌\t商品单价\t商品入库时间\n");

while (p != NULL)

{

fprintf(fp, "%s\t\t%s\t\t%d\t\t%s\t\t%f.2\t\t%s\t\t", p->data.name, p->data.number, p->data.num, p->data.brand, p->data.price, p->data.putdata);

fprintf(fp, "\n");

p = p->next;

}

fclose(fp); //把缓冲区内最后剩余的数据输出到内核缓冲区,并释放文件指针

printf("数据已成功导入文件goods.txt！\n");

}

/\*显示系统功能界面\*/

void Menu()

{

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎进入超市货架管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" 请选择您要操作的项目 \n");

printf(" \n");

printf(" 1. 录入商品信息 \n");

printf(" 2. 浏览商品清单 \n");

printf(" 3. 查找商品信息 \n");

printf(" 4. 修改商品信息 \n");

printf(" 5. 商品信息排序 \n");

printf(" 6. 商品入库 \n");

printf(" 7. 商品出库 \n");

printf(" 8. 删除商品 \n");

printf(" 9. 保存操作 \n");

printf(" 0. 退出系统 \n");

}

void LogIn(struct UserInfo userlogin[], char tmp[])

{

FILE\* fp;

int number, checkdenglu = 1;

struct UserInfo user;

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

printf("\t\t 【当前日期：");

printf("%s】", tmp);

printf("\n\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*欢迎进入超市货架管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n");

printf("\t\t 超市货架管理系统 状态：未登陆");

printf("\n\n\t\t\t 1. 用户登录");

printf("\n\n\t\t\t 0. 退出系统\n");

printf("\n\n\t\t 请选择菜单号【0-1】:");

scanf("%d", &number);

switch (number)

{

case 1:

printf("\n\t\t 用户名:");

scanf("%s", &userlogin[0].UserName);

printf("\n\t\t 密 码:");

scanf("%s", &userlogin[0].Password);

if (!(fp = fopen("H:\\UserInfo.txt", "r")))

{

printf("文件打开失败或不存在！");

Continue();

LogIn(userlogin, tmp);

}

else

{

while (!feof(fp))

{

fread(&user, sizeof(struct UserInfo), 1, fp);

if (strcmp(userlogin[0].UserName, user.UserName) == 0 && strcmp(userlogin[0].Password, user.Password) == 0)

{

checkdenglu = 1;

}

}

}

fclose(fp);

if (checkdenglu == 1)

{

printf("登录成功！\n");

Continue();

}

else

{

printf("登录失败，请重新登录！\n");

Continue();

LogIn(userlogin, tmp);

}

break;

case 0:

printf("\n\t\t 回车键退出...");

getchar();

getchar();

printf("感谢您的使用，再见！\n");

exit(0);

default:

printf("输入有误，请重新输入:");

Continue();

LogIn(userlogin, tmp);

}

}

void Continue()

{

printf("\n\t\t 回车键继续...");

getchar();

getchar();

system("cls");

}

//#define \_CRT\_SECURE\_NO\_WARNINGS

#include"goods.h"

/\*程序入口\*/

int main()

{

struct UserInfo userlogin[10];

time\_t t = time(0);

char tmp[64];

strftime(tmp, sizeof(tmp), "%Y/%m/%d %A ", localtime(&t));

Goods x;

int chioce;

LNumber\* L, \* p;

L = (LNumber\*)malloc(sizeof(LNumber));

Init\_List(L); //初始化

LogIn(userlogin, tmp); //登录函数调用

system("cls");

do

{

printf("\t\t 【当前日期：");

printf("%s】", tmp);

printf("\n");

printf("\t\t 超市货架管理系统 状态：用户%s正在登陆中\n", userlogin[0].UserName);

Menu(); //调取菜单函数

printf(" 【·】 请输入您的选择:");

scanf("%d", &chioce);

switch (chioce)

{

case 1:

Create\_List(L); //商品录入

Continue();

break;

case 2:

printf("\n");

Browse(L); //浏览商品清单

Continue();

break;

case 3:

printf("\n");

printf("\n请输入要查找的商品的编号：\n");

printf("\n");

printf("您输入的编号为：");

printf("\n");

scanf("%s", x.number);

p = Search(L, x); //查询商品

if (p == NULL)

printf("\n该商品不存在。\n");

else

{

printf("商品名称 商品编号 商品数量 商品品牌 商品单价 商品入库时间\n");

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);

}

Continue();

break;

case 4:

printf("\n");

printf("\n请输入要修改的商品的编号:\n");

printf("\n");

printf(" 商品编号： ");

scanf("%s", x.number);

GoodsChange(L, x); //修改商品信息

Continue();

break;

case 5:

GoodsSort(L); //商品排序

printf("\n已完成排序，若要查看排序后的结果，请回到初始界面。 \n");

Continue();

break;

case 6:

printf("\n");

printf("\n请输入要入库的商品的名称、编号、数量、品牌、单价、入库时间:\n");

printf("\n");

printf("\n 商品名称： ");

scanf("%s", x.name);

printf("\n 商品编号： ");

scanf("%s", x.number);

printf("\n 商品数量： ");

scanf("%d", &x.num);

printf("\n 商品品牌： ");

scanf("%s", &x.brand);

printf("\n 商品单价： ");

scanf("%f", &x.price);

printf("\n 商品入库时间 ：");

scanf("%s", &x.putdata);

Storage(L, x); //商品入库

printf("\n 入库成功！ \n");

Continue();

break;

case 7:

printf("\n");

printf("\n请输入要出库的商品的编号、数量:\n");

printf("\n");

printf(" 商品编号： ");

scanf("%s", x.number);

printf(" 商品数量： ");

scanf("%d", &x.num);

Out(L, x); //商品出库

Continue();

break;

case 8:

printf("\n");

printf("\n请输入要出库的商品的编号:\n");

printf("\n");

printf(" 商品编号： ");

scanf("%s", x.number);

Delete(L, x); //删除某一商品

printf("\n 成功删除！");

printf("\n");

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

Continue();

break;

case 9:

printf("\n");

Save(L); //保存所有操作

printf("\n 保存成功！");

Continue();

break;

case 0:

printf("感谢您的使用，再见！\n");

exit(0);

default:

printf("输入指令有误，请重新登录进行操作:");

Continue();

main();

}

} while (1);

return 0;

}