附件 1

杭州电子科技大学

《程序设计课程实践》

综合项目实验报告

项目名称：咖啡店订单管理系统

团队成员：**23051213** 谭舒文

23051214 张逸轩

完成时间　2024　年　6　月

订单管理系统项目实验报告

1. 团队成员组成及分工

学号	姓名	详细任务分工
23051213	谭舒文	完成订单相关函数，完成实验报告
23051214	张逸轩	创建项目，建立项目框架，完成商品相关函数

2. 开发背景

咖啡店是一个繁忙的场所，每天都会有大量的订单和交易。为了有效地管理订单和提供更好的服务，开发一个咖啡店订单管理系统是非常有必要的。通过开发咖啡店订单管理系统，咖啡店可以提高订单处理效率、提供更好的顾客体验、优化库存管理和销售策略，并且简化店内运营流程。这样，咖啡店可以更好地满足顾客需求，提高业务效益。

3. 系统功能设计

3.1 系统功能模块设计

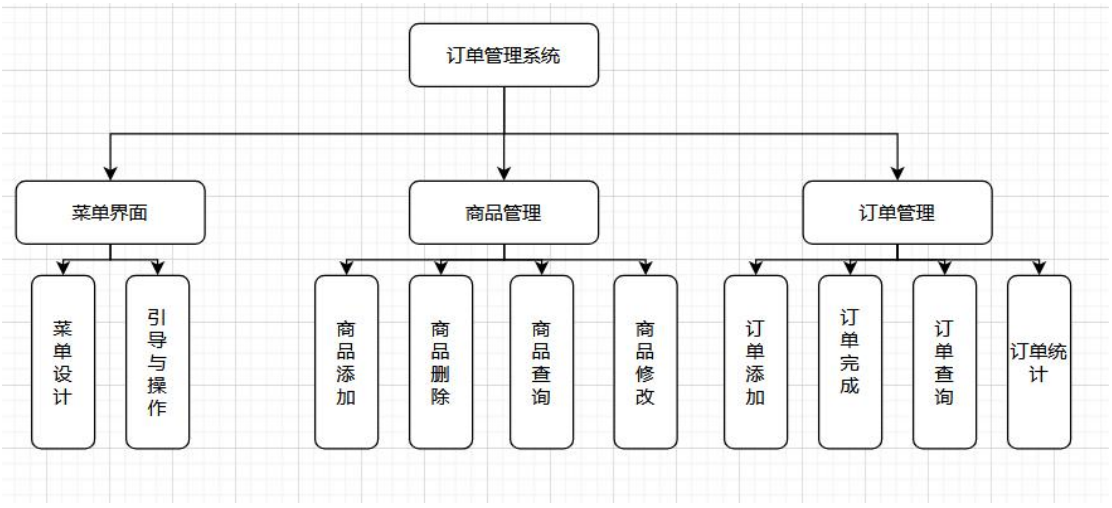

图 1 系统功能模块图

3.2 系统业务流程设计

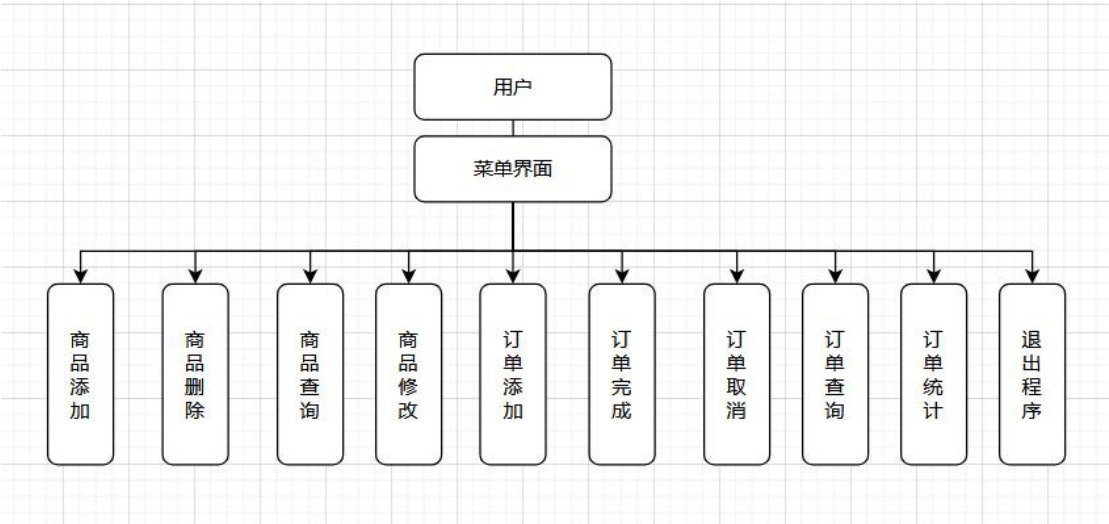

图 2 业务流程图

4. 项目创建

4.1 系统开发环境要求

本项目的开发及运行环境要求：gcc 10.3.0 make 4.4.1
操作系统：linux/windows
开发工具：VSCode
开发语言：c 语言

4.2 项目创建过程

先完成 cafe.h 的编写，确立了项目大体框架和数据结构。

```
#include <string.h>
#include <stdio.h>
#include <time.h>

const char classify[5][10] = {"coffee", "tea", "juice", "milk", "other"};
```

```
typedef struct {
    int ID;
    char name[50];
    int price;
```

```
    char description[100];
    int type;
    int quantity;
    int available;
} tGoods;
```

```
typedef struct {
    int ID;
    char customerName[50];
    time_t ordertime;
    int status;//未完成为 1，已完成为 2，已取消为 3
    int goodsNum[100]; // goodsNum[goodsID] is num
} tOrder;
```

```
void printGoods(tGoods *goods,int goodsNum);
void addGoods(tGoods *goods,int *goodsNum);
```

```
int searchGoods(tGoods *goods,int goodsNum,int printFlag);
void deleteGoods(tGoods *goods,int *goodsNum);
void modifyGoods(tGoods *goods,int goodsNum);
```

```
int searchOrder(tOrder *order,int orderNum,int printFlag);
void sortOrder(tOrder *order,int orderNum);
```

```
void printOrder(tOrder *order,int orderNum);
void addOrder(tOrder *orderm,int *orderNum);
void finishOrder(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
void cancelOrder(tOrder *order,int orderNum);
```

```
void printMenu();
```

```
void exitProgram();
```

```
int getSales(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
```

5. 预处理模块设计

5.1 文件引用

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
#include "cafe.h"  //函数声明
#include <signal.h>  //捕捉进程终止信号
```

5.2 宏定义

无

5.3 定义全局变量

```
tGoods *goods;  //指向商品结构体
tOrder *order;  //指向订单结构体
int goodsNum, orderNum;  //商品，订单总数
```

5.4 函数声明

```
void printGoods(tGoods *goods,int goodsNum);
```

输入参数：`goods, goodsNum`

输出参数：无

实现功能：打印商品

```
void addGoods(tGoods *goods,int *goodsNum);
```

输入参数：`goods, &goodsNum`

输出参数：无

实现功能：添加商品

```
int searchGoods(tGoods *goods,int goodsNum,int printFlag);
```

输入参数：`goods, goodsNum, 1`

输出参数：商品编号

实现功能：查找商品

```
void deleteGoods(tGoods *goods,int *goodsNum);
```

输入参数：`goods, &goodsNum`

输出参数：无

实现功能：删除商品

```
void modifyGoods(tGoods *goods,int goodsNum);
```

输入参数：`goods, goodsNum`

输出参数：无

实现功能：修改商品

```
int searchOrder(tOrder *order,int orderNum,int printFlag);
```

输入参数：`order, orderNum, 1`

输出参数：订单号

实现功能：查找订单

```
void sortOrder(tOrder *order,int orderNum);
```

输入参数：`order, orderNum`

输出参数：无

实现功能：订单排序

```
void printOrder(tOrder *order,int orderNum);
```

输入参数：`order, orderNum`

输出参数：无

实现功能：打印订单

```
void addOrder(tOrder *orderm,int *orderNum);
```

输入参数：`order, &orderNum`

输出参数：无

实现功能：添加订单

```
void finishOrder(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
```

输入参数：`order, orderNum, goods, goodsNum`

输出参数：无

实现功能：完成订单

```
void cancelOrder(tOrder *order,int orderNum);
```

输入参数：`order, orderNum`

输出参数：无

实现功能：取消订单

```
void printMenu();
```

输入参数：无

输出参数：无

实现功能：输出菜单

```
void exitProgram();
```

输入参数：无

输出参数：无

实现功能：退出

```
int getSales(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
```

输入参数：`order, orderNum, goods, goodsNum`

输出参数：文件是否成功打开

实现功能：统计订单并保存

```
void SIGINT_handler(int sig);
```

输入参数：`SIGINT`

输出参数：无

实现功能：安全退出

文档完成人：谭舒文

6. 项目运行效果

```
                                                    ./cafe

 1. 打印商品
 2. 添加商品
 3. 搜索商品
 4. 删除商品
 5. 修改商品
 6. 搜索订单
 7. 排序订单
 8. 打印订单
 9. 添加订单
 10. 取消订单
 11. 完成订单
 12. 获取销售记录并保存
 13. 退出
 请输入你的选择：2
 请输入商品名字：q
 请输入商品价格：12
 请输入商品描述：x
 请输入商品类型：1
 请输入商品库存：99
```

```
 1. 打印商品
 2. 添加商品
 3. 搜索商品
 4. 删除商品
 5. 修改商品
 6. 搜索订单
 7. 排序订单
 8. 打印订单
 9. 添加订单
 10. 取消订单
 11. 完成订单
 12. 获取销售记录并保存
 13. 退出
 请输入你的选择：1
 编号    名字    价格    描述    类型    库存
 0       q       12      x       tea     99
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：3
请选择搜索方式：1.编号 2.名字
1
请输入商品编号：0
编号    名字    价格    描述    类型    库存
0       q       12      x       tea     99
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：9
该订单订单号为：0
请输入顾客名字：
t
请输入购买商品种类的个数
1
请输入商品编号
0
请输入购买该商品的数量
5
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：8
订单号：0
顾客名字：t
订单状态：1
订单时间：2024-06-04 16:44
购买商品为：
商品编号：0     名字：q 数量:5
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：11
请输入订单号
0
订单完成
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：8
订单号：0
顾客名字：t
订单状态：2
订单时间：2024-06-04 16:44
购买商品为：
商品编号：0      名字：q 数量:5
```

```
1. 打印商品
2. 添加商品
3. 搜索商品
4. 删除商品
5. 修改商品
6. 搜索订单
7. 排序订单
8. 打印订单
9. 添加订单
10. 取消订单
11. 完成订单
12. 获取销售记录并保存
13. 退出
请输入你的选择：12
请输入起始时间(XXXX年X月X日X时X分):
2024
6
4
16
40
请输入结束时间(XXXX年X月X日X时X分):
2024
6
4
16
45
总销售额为：60元
销售清单如下
商品编号：0     商品名字：q     商品单价：12    销售数量：5     总销售额：60
```

7. 项目创新点

1. 使用 Signal 重定义信号处理，确保强制退出的安全性。
2. 使用 Makefile 可以确保代码在不同的操作系统和编译器上都能够顺利编译和构建,提高了代码的可移植性。

8. 收获和建议

谭舒文：

通过程序设计实践课，我学到了如何将软件需求转化为实际的程序设计，并进行模块化和结构化的设计。我学会了分析和理解客户的需求，并将其转化为清晰的功能和特性。通过使用适当的设计模式和架构原则，我能够将复杂的问题分解为更小、更易于管理的模块，从而提高代码的可读性和可维护性。我也学会了使用 UML 图和其他工具来进行系统设计和建模，以确保我能够准确地实现所需的功能。这些经验和技能对于提升我的软件开发能力和职业发展具有重要意义。

在项目中，通过学习和使用 Git，我掌握了版本控制的重要性和使用 Git 进行团队协作的技巧。我学会了如何使用 Git 来管理代码的版本和更改历史，以及如何回滚到先前的版本。我还学会了使用分支进行并行开发，并了解了如何切换、创建和合并分支。通过将本地仓库推送到远程仓库，我可以与团队成员实时同步工作，并解决可能出现的代码冲突。我还学会了如何解决不同分支之间的代码冲突，以保持代码的一致性和完整性。通过有意义的提交消息，我可以更好地回顾和理解代码更改历史。通过学习和掌握 Git，我提高了开发效率和团队合作能力，为订单管理系统项目的成功做出了贡献。

在团队项目中，我主要负责订单管理和库存管理模块的开发。我与小组成员密切合作，通过讨论，我们共同制定了项目的计划和目标，并分配了各自的任务。我们使用版本控制工具（如 Git）来管理代码，并定期进行代码审查和合并，以确保代码质量和一致性。

在团队协作过程中，我逐步完成了我的开发任务，并及时与小组成员分享进展和遇到的问题。我们通过互相交流和协作，解决了一些技术难题，并提供了优化和改进的建议。最终，我们成功地将订单管理系统开发完成。

通过参与团队项目，我不仅学到了技术方面的知识和技能，还学会了与他人合作、沟通和协调的重要性。我学会了如何有效地分工合作，互相支持和帮助，以实现共同的目标。我也意识到软件产品的质量和用户体验对于项目成功的重要性。

从团队协作的角度来看，良好的团队合作可以提高开发效率和质量，减少沟通和协调的成本，并增强团队成员的凝聚力和归属感。从项目创新的角度来看，通过

与团队成员的合作和交流，我们能够共同提出创新的想法和解决方案，为客户提供更好的产品和服务。

人工智能在订单管理系统中的应用是一个具有较好前景的软件产品设想。通过利用人工智能技术，我们可以实现自动化的订单处理和推荐系统，根据顾客的历史订单和偏好，提供个性化的推荐和优惠活动。这样可以提高用户体验，增加销售额，并为咖啡店提供更精确的数据分析和决策支持。

总之，在软件产品的策划和研发过程中，团队协作、项目创新、人工智能和产品质量都是非常重要的。通过合理的规划和组织，有效的沟通和协调，以及持续的创新和改进，我们可以开发出具有高质量和创新性的软件产品，满足客户需求，并为用户带来更好的体验和价值。

张逸轩：

首先,通过这个软件产品开发的实践,我学到了如何设计一个可扩展和可维护的系统架构。在确定整体架构时,我需要考虑模块间的耦合度、数据流向、性能需求等诸多因素。同时,为了确保代码质量,我也积极参与编码规范的制定,并严格要求自己和团队成员遵守。这对于保证软件产品的可靠性和可演化性非常重要。

在代码实现环节,我也遇到了不少挑战。比如,如何在满足功能需求的同时,优化系统性能?如何有效管理项目依赖,避免引入安全隐患?这些都需要我仔细思考并与团队成员反复讨论。通过不断学习和实践,我逐步掌握了一些有效的技巧,如采用缓存机制、依赖管理工具等,最终顺利完成了代码开发任务。

在团队协作方面,我注重与其他成员的沟通和协调。我会主动了解他们的工作进度,并及时为他们提供技术支持。同时,我也会积极听取他们的反馈意见,共同优化系统设计。这种密切的协作,不仅提高了开发效率,也增强了团队凝聚力。让我明白了团队协作的重要性。

组长：谭舒文

成员：张逸轩

附：源代码清单

cafe.h （张逸轩完成）

```
#include <string.h>
#include <stdio.h>
#include <time.h>

const char classify[5][10] = {"coffee", "tea", "juice", "milk", "other"};
```

```
typedef struct {
    int ID;
    char name[50];
    int price;
    char description[100];
    int type;
    int quantity;
    int available;
} tGoods;
```

```
typedef struct {
    int ID;
    char customerName[50];
    time_t ordertime;
    int status;//未完成为 1，已完成为 2，已取消为 3
    int goodsNum[100]; // goodsNum[goodsID] is num
} tOrder;
```

```
void printGoods(tGoods *goods,int goodsNum);
void addGoods(tGoods *goods,int *goodsNum);
```

```
int searchGoods(tGoods *goods,int goodsNum,int printFlag);
void deleteGoods(tGoods *goods,int *goodsNum);
void modifyGoods(tGoods *goods,int goodsNum);
```

```
int searchOrder(tOrder *order,int orderNum,int printFlag);
void sortOrder(tOrder *order,int orderNum);
```

```
void printOrder(tOrder *order,int orderNum);
void addOrder(tOrder *orderm,int *orderNum);
void finishOrder(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
```

```
void cancelOrder(tOrder *order,int orderNum);

void printMenu();

void exitProgram();

int getSales(tOrder *order,int orderNum,tGoods *goods,int goodsNum);
```

以下为 cafe.c

（框架及商品相关函数由张逸轩完成，订单相关函数由谭舒文完成）

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <stdlib.h>
#include "cafe.h"
#include <signal.h>

tGoods *goods;
tOrder *order;
int goodsNum, orderNum;
```

```
void init()
{
    goods = (tGoods *)malloc(sizeof(tGoods)*100);
    order = (tOrder *)malloc(sizeof(tOrder)*100);
    goodsNum = orderNum = 0;
}
```

```
void printMenu()
{
    while (1)
    {
        int choice;
        printf("\n1. 打印商品\n2. 添加商品\n3. 搜索商品\n4. 删除商品\n5. 修改商品\n6. 搜索订单\n7. 排序订单\n8. 打印订单\n9. 添加订单\n10. 取消订单\n11. 完成订单\n12. 获取销售记录并保存\n13. 退出\n");
        printf("请输入你的选择:");
        scanf("%d", &choice);
```

```
        switch (choice)
        {
            case 1:
                printGoods(goods, goodsNum);
                break;
            case 2:
                addGoods(goods, &goodsNum);
                break;
            case 3:
                searchGoods(goods, goodsNum, 1);
                break;
            case 4:
                deleteGoods(goods, &goodsNum);
                break;
            case 5:
                modifyGoods(goods, goodsNum);
                break;
            case 6:
                searchOrder(order, orderNum, 1);
                break;
            case 7:
                sortOrder(order, orderNum);
                break;
            case 8:
                printOrder(order, orderNum);
                break;
            case 9:
                addOrder(order, &orderNum);
                break;
            case 10:
                cancelOrder(order, orderNum);
                break;
            case 11:
                finishOrder(order, orderNum, goods, goodsNum);
                break;
            case 12:
                getSales(order, orderNum, goods, goodsNum);
                break;
            case 13:
                exitProgram();
                return;
            default:
                printf("无效的选择，请重新输入。\n");
```

```
        }
    }
}
```

```
void printGoods(tGoods *goods,int goodsNum)
{
    printf("编号\t 名字\t 价格\t 描述\t 类型\t 库存\n");
    for(int i=0;i<goodsNum;i++)
    {
        if(goods[i].available == 0) continue;
       printf("%d\t%s\t%d\t%s\t%s\t%d\n",goods[i].ID,goods[i].name,goods[i].price,go
ods[i].description,classify[goods[i].type],goods[i].quantity);
    }
}
```

```
void addGoods(tGoods *goods,int *goodsNum)
{
    printf("请输入商品名字:");
    scanf("%s",goods[*goodsNum].name);
    printf("请输入商品价格:");
    scanf("%d",&goods[*goodsNum].price);
    printf("请输入商品描述:");
    scanf("%s",goods[*goodsNum].description);
    printf("请输入商品类型:");
    scanf("%d",&goods[*goodsNum].type);
    printf("请输入商品库存:");
    scanf("%d",&goods[*goodsNum].quantity);
    goods[*goodsNum].available = 1;
    goods[*goodsNum].ID = *goodsNum;
    (*goodsNum)++;
}
```

```
int searchGoods(tGoods *goods, int goodsNum, int printFlag)
{
    printf("请选择搜索方式:1.编号 2.名字\n");
    int choice;
    scanf("%d", &choice);
    if(choice == 1)
    {
        printf("请输入商品编号:");
        int id;
        scanf("%d", &id);
        for(int i=0;i<goodsNum;i++)
        {
```

```
            if(goods[i].ID == id && goods[i].available == 1)
            {
                if(printFlag)
                {
                    printf("编号\t 名字\t 价格\t 描述\t 类型\t 库存\n");
                printf("%d\t%s\t%d\t%s\t%s\t%d\n",goods[i].ID,goods[i].name,goods[i].
price,goods[i].description,classify[goods[i].type],goods[i].quantity);
                }
                return i;
            }
        }
    }
    else if(choice == 2)
    {
        printf("请输入商品名字:");
        char name[50];
        scanf("%s", name);
        for(int i=0;i<goodsNum;i++)
        {
            if(strcmp(goods[i].name, name) == 0 && goods[i].available == 1)
            {
                if(printFlag)
                {
                    printf("编号\t 名字\t 价格\t 描述\t 类型\t 库存\n");
                printf("%d\t%s\t%d\t%s\t%s\t%d\n",goods[i].ID,goods[i].name,goods[i].
price,goods[i].description,classify[goods[i].type],goods[i].quantity);
                }
                return i;
            }
        }
    }
    if(printFlag)
        printf("商品不存在\n");
    return -1;
}
```

```
void deleteGoods(tGoods *goods, int *goodsNum)
{
    int id = searchGoods(goods, *goodsNum, 0);
    if(id == -1)
    {
        printf("商品不存在\n");
        return;
    }
```

```
    goods[id].available = 0;
}
```

```
void modifyGoods(tGoods *goods, int goodsNum)
{
    int id = searchGoods(goods, goodsNum, 0);
    if(id == -1)
    {
        printf("商品不存在\n");
        return;
    }
    printf("请输入商品名字:");
    scanf("%s",goods[id].name);
    printf("请输入商品价格:");
    scanf("%d",&goods[id].price);
    printf("请输入商品描述:");
    scanf("%s",goods[id].description);
    printf("请输入商品类型:");
    scanf("%d",&goods[id].type);
    printf("请输入商品库存:");
    scanf("%d",&goods[id].quantity);
}
```

```
int searchOrder(tOrder *order, int orderNum, int printFlag)
{
    sortOrder(order, orderNum);
    printf("请选择搜索方式:1.订单号 2.顾客名字\n");
    int choice;
    scanf("%d", &choice);
    if(choice == 1)
    {
        printf("请输入订单号:");
        int id;
        scanf("%d", &id);
        for(int i=0;i<orderNum;i++)
        {
            if(order[i].ID == id)
            {
                if(printFlag)
                {
                    struct tm *date = localtime(&order[i].ordertime);
                    char buffer[80];
                    strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M", date);
```

```
                    printf("订单号:%d\n 顾客名字:%s\n 订单状态:%d （未完成为 1, 已完成为 2, 已取消为 3)\n 订单时间:%s\n 购买商品为:
\n",order[i].ID,order[i].customerName,order[i].status,buffer);
                    for(int j=0;j<goodsNum;j++)
                    {
                        if(order[i].goodsNum[j] > 0)
                        {
                            printf("商品编号:%d\t 名字:%s\t 数
量:%d\n",i,goods[i].name,order[i].goodsNum[j]);
                        }
                    }
                }
                return i;
            }
        }
    }
    else if(choice == 2)
    {
        printf("请输入顾客名字:");
        char name[50];
        scanf("%s", name);
        for(int i=0;i<orderNum;i++)
        {
            if(strcmp(order[i].customerName, name) == 0 )
            {
                if(printFlag)
                {
                    struct tm *date = localtime(&order[i].ordertime);
                    char buffer[80];
                    strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M", date);
                    printf("订单号:%d\n 顾客名字:%s\n 订单状态:%d\n 订单时间:%s\n 购买商品
为:\n",order[i].ID,order[i].customerName,order[i].status,buffer);
                    for(int j=0;j<goodsNum;j++)
                    {
                        if(order[i].goodsNum[j] > 0)
                        {
                            printf("商品编号:%d\t 名字:%s\t 数
量:%d\n",i,goods[i].name,order[i].goodsNum[j]);
                        }
                    }
                }
                return i;
            }
        }
```

```
    }
    if(printFlag)
        printf("订单不存在\n");
```

```
    return -1;
}
```

```
void sortOrder(tOrder *order, int orderNum)
{
    int i, j;
    tOrder temp;
```

```
    for (i = 0; i < orderNum - 1; i++)
    {
        for (j = 0; j < orderNum - i - 1; j++)
        {
            if (order[j].ordertime > order[j + 1].ordertime)
            {
                // 交换顺序
                temp = order[j];
                order[j] = order[j + 1];
                order[j + 1] = temp;
            }
        }
    }
    return;
}
```

```
void printOrder(tOrder *order, int orderNum)
{
    sortOrder(order, orderNum);
```

```
    for(int i=0;i<orderNum;i++)
    {
        struct tm *date = localtime(&order[i].ordertime);
        char buffer[80];
        strftime(buffer, sizeof(buffer), "%Y-%m-%d %H:%M", date);
        printf("订单号:%d\n 顾客名字:%s\n 订单状态:%d\n 订单时间:%s\n 购买商品为:
\n",order[i].ID,order[i].customerName,order[i].status,buffer);
        for(int j=0;j<goodsNum;j++)
        {
            if(order[i].goodsNum[j] > 0)
            {
```

```
                printf("商品编号:%d\t 名字:%s\t 数
量:%d\n",j,goods[j].name,order[i].goodsNum[j]);
            }
        }
        printf("\n");
    }

    return;
}
```

```
void addOrder(tOrder *order, int *orderNum)
{
```

```
    printf("该订单订单号为:%d\n",*orderNum);
    printf("请输入顾客名字:\n");
    scanf("%s",order[*orderNum].customerName);
```

```
    printf("请输入购买商品种类的个数\n");
    int num=0;
    scanf("%d",&num);
```

```
    int flag=0;
    for(int i=0;i<num;i++)
    {
        int goodsID=0,goodsnum=0;
        printf("请输入商品编号\n");
        scanf("%d",&goodsID);
        printf("请输入购买该商品的数量\n");
        scanf("%d",&goodsnum);
        if(goods[goodsID].quantity<goodsnum)
        {
          printf("库存过少，请重新操作\n");
          flag=1;
          break;
        }
        order[*orderNum].goodsNum[goodsID]=goodsnum;
```

```
    }
```

```
    if(flag==0)
    {
    order[*orderNum].ordertime=time(NULL);
    order[*orderNum].status = 1;
    order[*orderNum].ID = *orderNum;
```

```
    (*orderNum)++;
    }
    return;
}
```

```
void cancelOrder(tOrder *order, int orderNum)
{
    sortOrder(order, orderNum);
```

```
    printf("请输入订单号\n");
    int num=0;
    scanf("%d",&num);
```

```
    if (order[num].status==1)
    {
        order[num].status=3;
        printf("订单成功删除\n");
    }
    else if(order[num].status==2)
    {
        printf("已完成的订单不可取消\n");
    }
    else if (order[num].status==3)
    {
        printf("订单原已删除\n");
    }
    else printf("订单不存在\n");
```

```
    return;
}
```

```
void finishOrder(tOrder *order, int orderNum, tGoods *goods, int goodsNum)
{
    sortOrder(order, orderNum);
```

```
    printf("请输入订单号\n");
    int num=0;
    scanf("%d",&num);
```

```
    if (order[num].status==1)
    {
        order[num].status=2;
```

```
        for(int j=0;j<goodsNum;j++)
```

```
        {
            if(order[num].goodsNum[j] > 0)
            {
                goods[j].quantity=goods[j].quantity-order[num].goodsNum[j];
            }
        }
        printf("订单完成\n");
    }
    else if (order[num].status==2)
    {
        printf("订单原已完成\n");
    }
    else if(order[num].status==3)
    {
        printf("已取消的订单不可完成\n");
    }
    else printf("订单不存在\n");
```

```
    return;
}
```

```
int getSales(tOrder *order, int orderNum, tGoods *goods, int goodsNum)
{
    sortOrder(order, orderNum);
```

```
    int year1=0, year2=0;
    int month1=0, month2=0;
    int day1=0, day2=0,hour1=0,hour2=0,min1=0,min2=0;
    printf("请输入起始时间(XXXX 年 X 月 X 日 X 时 X 分):\n");
    scanf("%d %d %d %d %d",&year1,&month1,&day1,&hour1,&min1);
    printf("请输入结束时间(XXXX 年 X 月 X 日 X 时 X 分):\n");
    scanf("%d %d %d %d %d",&year2,&month2,&day2,&hour2,&min2);
```

```
    struct tm date1 = {0};
    date1.tm_year = year1 - 1900; // tm_year 表示的是从 1900 年开始的年数
    date1.tm_mon = month1 - 1;    // tm_mon 表示的是 0-11 的月份
    date1.tm_mday = day1;
    date1.tm_hour = hour1;
    date1.tm_min = min1;
    time_t time1 = mktime(&date1);
    struct tm date2 = {0};
    date2.tm_year = year2 - 1900; // tm_year 表示的是从 1900 年开始的年数
    date2.tm_mon = month2 - 1;    // tm_mon 表示的是 0-11 的月份
    date2.tm_mday = day2;
```

```
    date2.tm_hour = hour2;
    date2.tm_min = min2;
    time_t time2 = mktime(&date2);
```

```
    int sales=0;
    int salesList[100];
    memset(salesList,0,sizeof(salesList));
```

```
    for(int i=0;i<orderNum;i++)
    {
        if(order[i].ordertime>time1&&order[i].ordertime<time2&&order[i].status==2)
        {
            for(int j=0;j<goodsNum;j++)
            {
                if(order[i].goodsNum[j] > 0)
                {
                    sales=sales+goods[j].price*order[i].goodsNum[j];
                    salesList[j]=salesList[j]+order[i].goodsNum[j];
                }
            }
        }
    }
```

```
    printf("总销售额为:%d 元\n 销售清单如下\n",sales);
    for(int i=0;i<goodsNum;i++)
    {
        if(salesList[i]>0)
        {
            printf("商品编号:%d\t 商品名字:%s\t 商品单价:%d\t 销售数量:%d\t 总销售
额:%d\n",goods[i].ID,goods[i].name,goods[i].price,salesList[i],goods[i].price*salesL
ist[i]);
        }
    }
```

```
    FILE *file = fopen("output.txt", "w+");
    if (file == NULL)
    {
        printf("文件打开失败，未保存\n");
        return 1;
    }
```

```
    fprintf(file, "%d 年%d 月%d 日%d 时%d 分到%d 年%d 月%d 日%d 时%d 分期间
\n",year1,month1,day1,hour1,min1,year2,month2,day2,hour2,min2);
    fprintf(file,"总销售额为:%d 元\n 销售清单如下\n",sales);
```

```
    int length = sizeof(salesList);
    for (int i = 0; i < length; i++)
    {
        if (salesList[i]>0)
        {
            fprintf(file,"商品编号:%d\t 商品名字:%s\t 商品单价:%d\t 销售数量:%d\t 总销售
额:%d\n",goods[i].ID,goods[i].name,goods[i].price,salesList[i],goods[i].price*salesL
ist[i]);
        }
    }
```

```
    fclose(file);
```

```
    printf("成功保存\n");
```

```
    return 0;
}
```

```
void exitProgram()
{
    free(goods);
    free(order);
    printf("成功退出\n");
}
```

```
void SIGINT_handler(int sig)
{
    exitProgram();
    exit(0);
}
```

```
int main()
{
```

```
    signal(SIGINT, SIGINT_handler);
    init();
    printMenu();
}
    free(goods);
    free(order);
    printf("成功退出\n");
}
```

```
void SIGINT_handler(int sig)
```

```
{
    exitProgram();
    exit(0);
}

int main()
{

    signal(SIGINT, SIGINT_handler);
    init();
    printMenu();
}
```