



**计算机学院程序设计作品大赛作品**

作品题目 **校医院预约挂号管理系统**

姓 名 **DYJ**

2021年06月

目录

[作品简要介绍 1](#_Toc74999170)

[一． 总体设计 2](#_Toc74999171)

[1.1设计思想 2](#_Toc74999172)

[1.2设计原则 2](#_Toc74999173)

[1.3功能模块图 3](#_Toc74999174)

[二． 详细设计 4](#_Toc74999175)

[2.1程序流程图 4](#_Toc74999176)

[2.2模块流程图 5](#_Toc74999177)

[(1)注册模块 5](#_Toc74999178)

[(3)更改密码模块 7](#_Toc74999179)

[(4)密码寻回模块 8](#_Toc74999180)

[(5)挂号/出诊模块 9](#_Toc74999181)

[(6)查看模块 10](#_Toc74999182)

[(7)取消出诊/预约模块 11](#_Toc74999183)

[(8)注销账号模块 12](#_Toc74999184)

[(9)更改预约/出诊模块 13](#_Toc74999185)

[2.3模块细致函数介绍 14](#_Toc74999186)

[(1)文件读取函数 14](#_Toc74999187)

[(2)文件写入函数 15](#_Toc74999188)

[(3)数量读取函数 16](#_Toc74999189)

[(4)数量写入函数 17](#_Toc74999190)

[(5)链表新增函数 19](#_Toc74999191)

[(6)当下时间比较函数 20](#_Toc74999192)

[(7)时间比较函数 21](#_Toc74999193)

[2.5数据设计 22](#_Toc74999194)

[三．运行效果 23](#_Toc74999195)

[四．总结 28](#_Toc74999196)

[致谢 29](#_Toc74999197)

[参考文献： 30](#_Toc74999198)

# 

# 作品简要介绍

|  |  |
| --- | --- |
| **作品名称** | **校医院预约挂号管理系统** |
| **参赛学生** | **DYJ** |
| **作品摘要**  **（400字以内）** | **本系统采用C语言编写，分为用户端和医生端，主要实现预约和查找功能。其中数据存储采用数据流的方式存储为文本文档，读取时采用链表来方便增删查改，扩大存储范围。而在信息查找时，人性化的对表单数据按照时间进行排序，方便读取。可以较好的服务于校医院。** |
| **开发技术**  **和平台** | Dev C++ |
| **系统主要功能和特色** | **医院预约挂号系统主要包含是三个子模块：（1）医生信息（2）就诊人信息（3）预约信息单**  **一、医生信息**  **医生信息的注册，注销，查找，更新**   1. **就诊人信息**   **就诊人信息的注册，注销，查找，更新**   1. **预约单**   **预约挂号的生成、查找、删除，修改**  **医生可以管理自己的预约时间段，以及每个时间段就诊人数。**  **医生可以对就诊单查看与排序。** |

# 总体设计

## 1.1设计思想

(1)校医院管理系统是应用于多方的软件，应用方包括医生和用户。

(2)医生和用户两方的功能需要区分开来，以登录来区分。

(3)医生信息需包括科室，每一端的用户区分应以用户名为准，因为有重名

(4)登录系统应考虑到密码管理问题，包括后期的更改和找回。

(5)预约和挂号类似于注册登录的储存方式。

(6)用户端和医生端的功能的查看表单需包括查看对方数据和自我的数据。

(7)表单的查看应该是按日期排好序的。

(8)两方的主要功能是预约和出诊及查看，可再衍生出取消的功能。

(9)数据分开储存，易于管理和读取，顺序要按照链表的结构体形式，好读取。

(10)个人信息分医生和用户；预约信息也分医生和用户；存于文件。

(11)登录，预约等等功能于两方的运行思路是一致的，可利用形参来区别不同端。

## 1.2设计原则

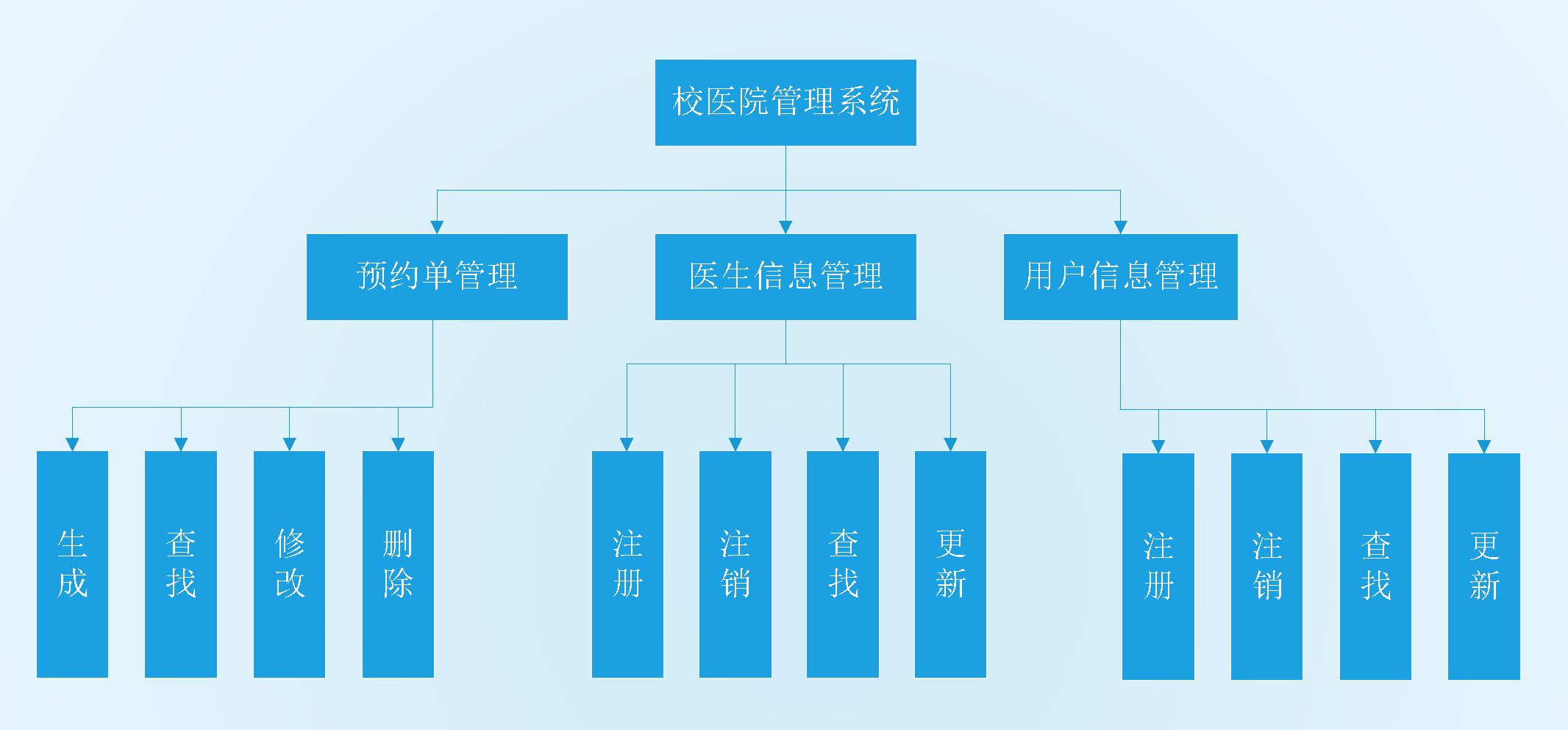
(1)分模块，分函数，分多个运行文件，利于读取和维护代码。

(2)对共性功能提取为函数，届时调用使用即可，减少代码量。

(3)代码规范，增加可读性和减少错误。

(4)运行结果美观和方便

## 1.3功能模块图

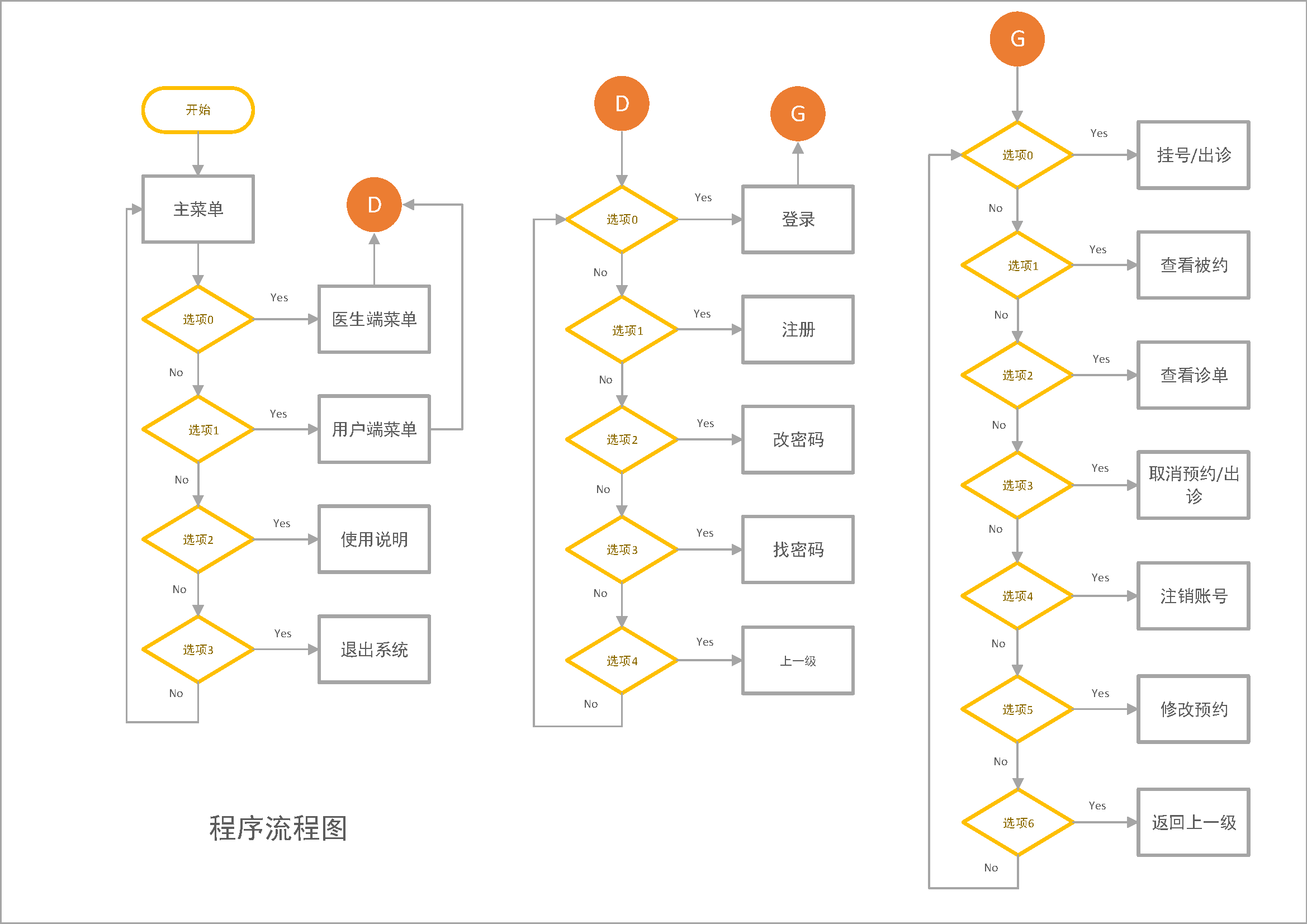


程序系统下包括三大模块，分别是医生信息、用户信息和预约单信息管理。由医生和用户端口注册和登入，然后都可以对预约单进行生成、查找、删除、修改的功能。可知，预约单亦分为两个方面，即医生的出诊单和用户的挂号单。

三个模块组成一套完整的管理系统。

# 详细设计

## 2.1程序流程图



程序共分5级菜单，采用方向键选择进入的方法，使程序易于使用。其中医生和用户共用账号的注册、登录、改密码和找密码的功能。

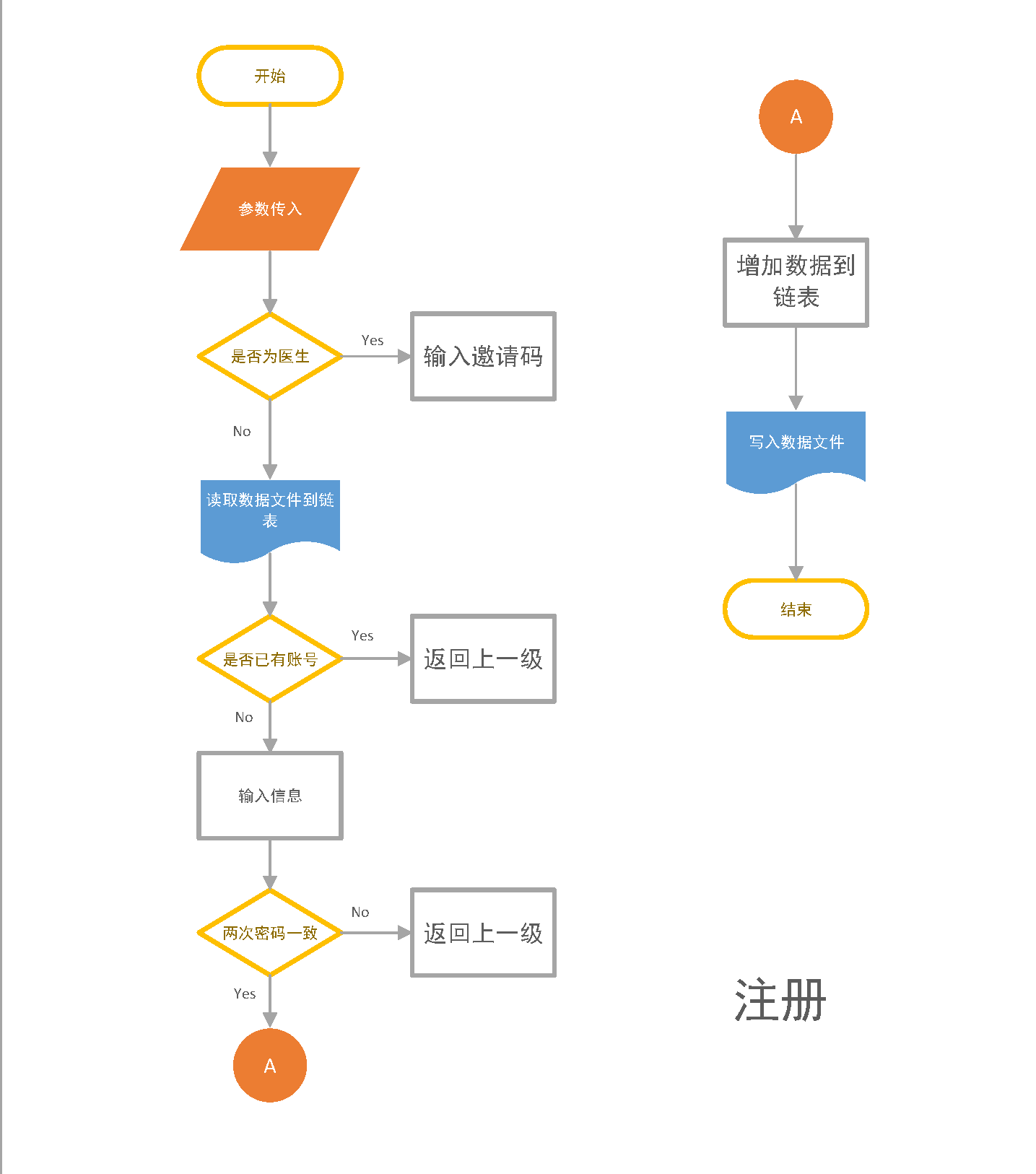
在登入进入之后便是功能界面了。其中的挂号、预约出诊、看表单以及注销自己的账号的功能会在其中。

这些功能中，有许多的功能是类似的，比如读取文件以及给链表的、输入链表保存到文件等。这类函数便可多次调用，在此便不过多赘述。

具体实现请参照上方的“程序流程图”。

## 2.2模块流程图

### (1)注册模块

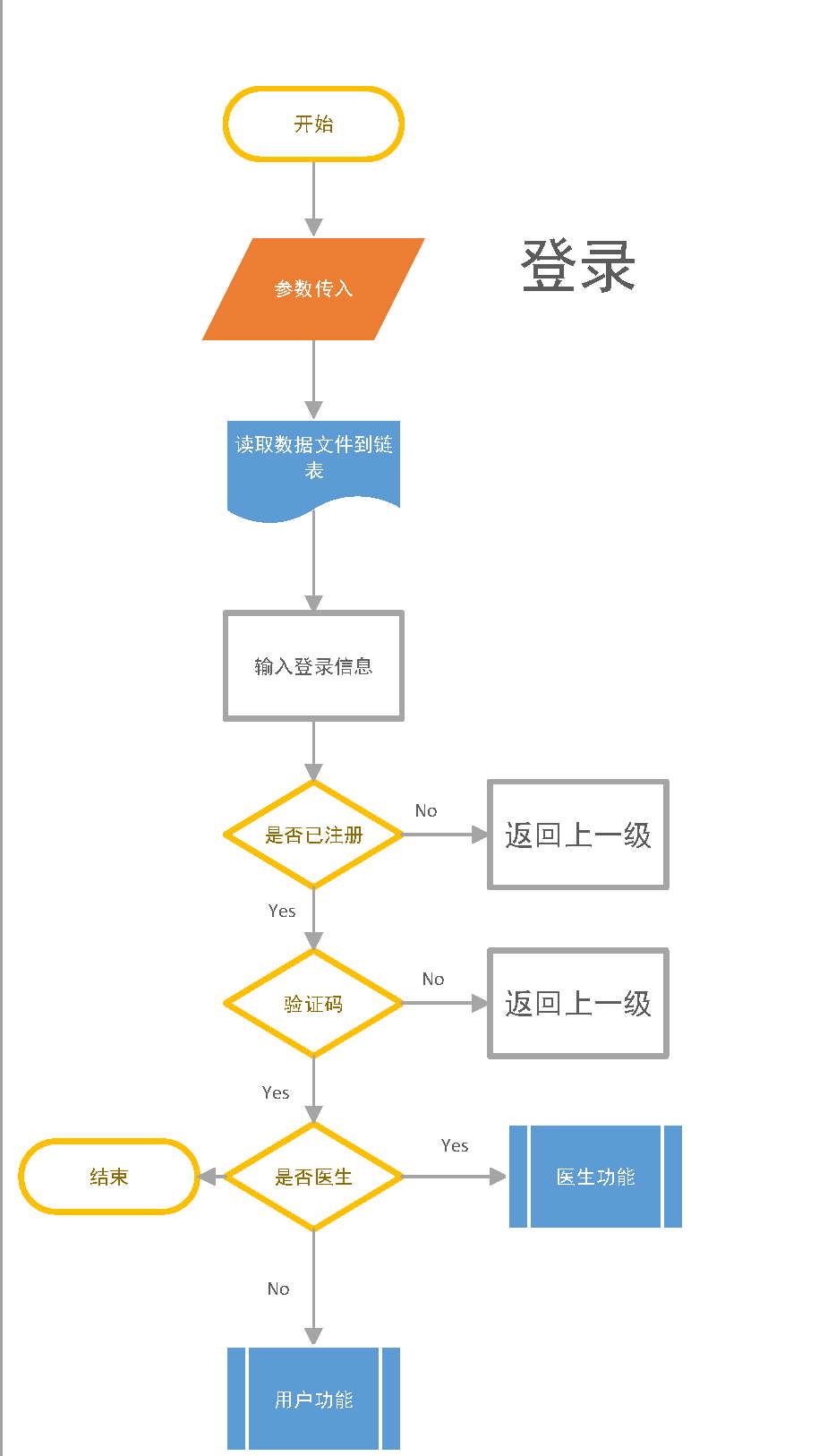


医生与用户共用的注册模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：extern void userAdd(int order)

//两端口共用的注册函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

**(2)登录模块**

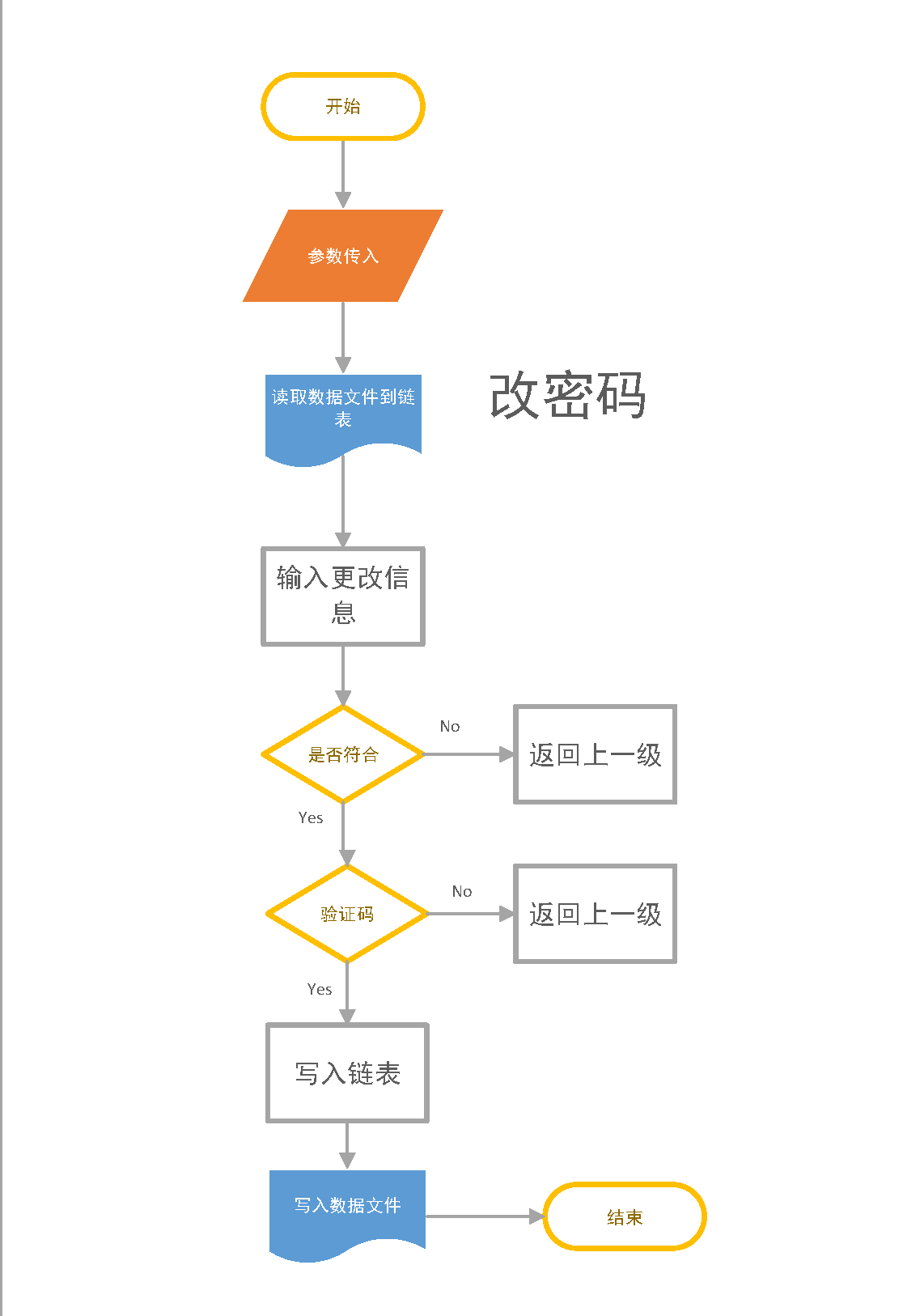


医生与用户共用的登录模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：extern void userLogin(int order)

//两端口共用的登录函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (3)更改密码模块

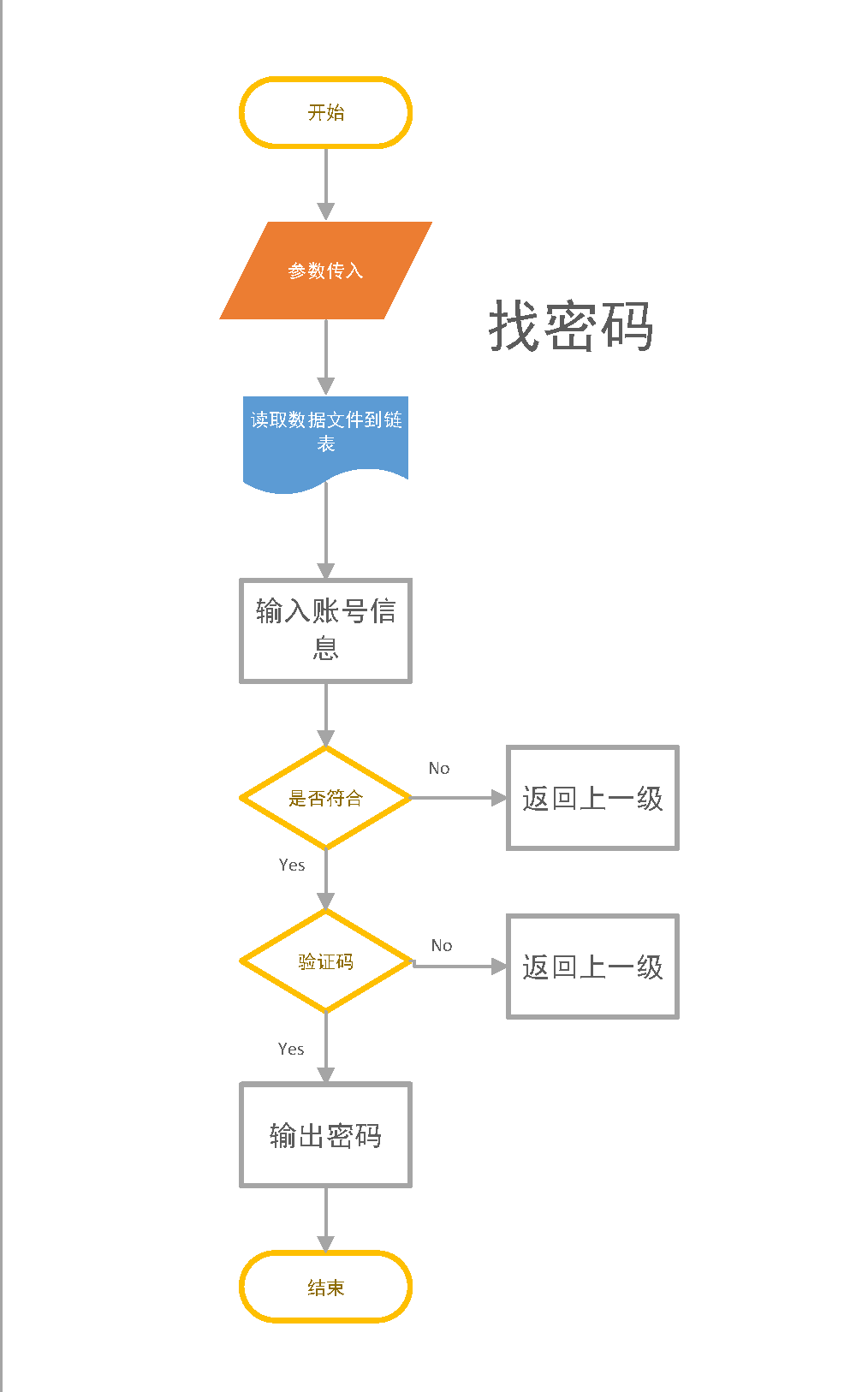


医生与用户共用的密码更改模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：extern void changPassword(int order)

//两端口共用的密码更改函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (4)密码寻回模块

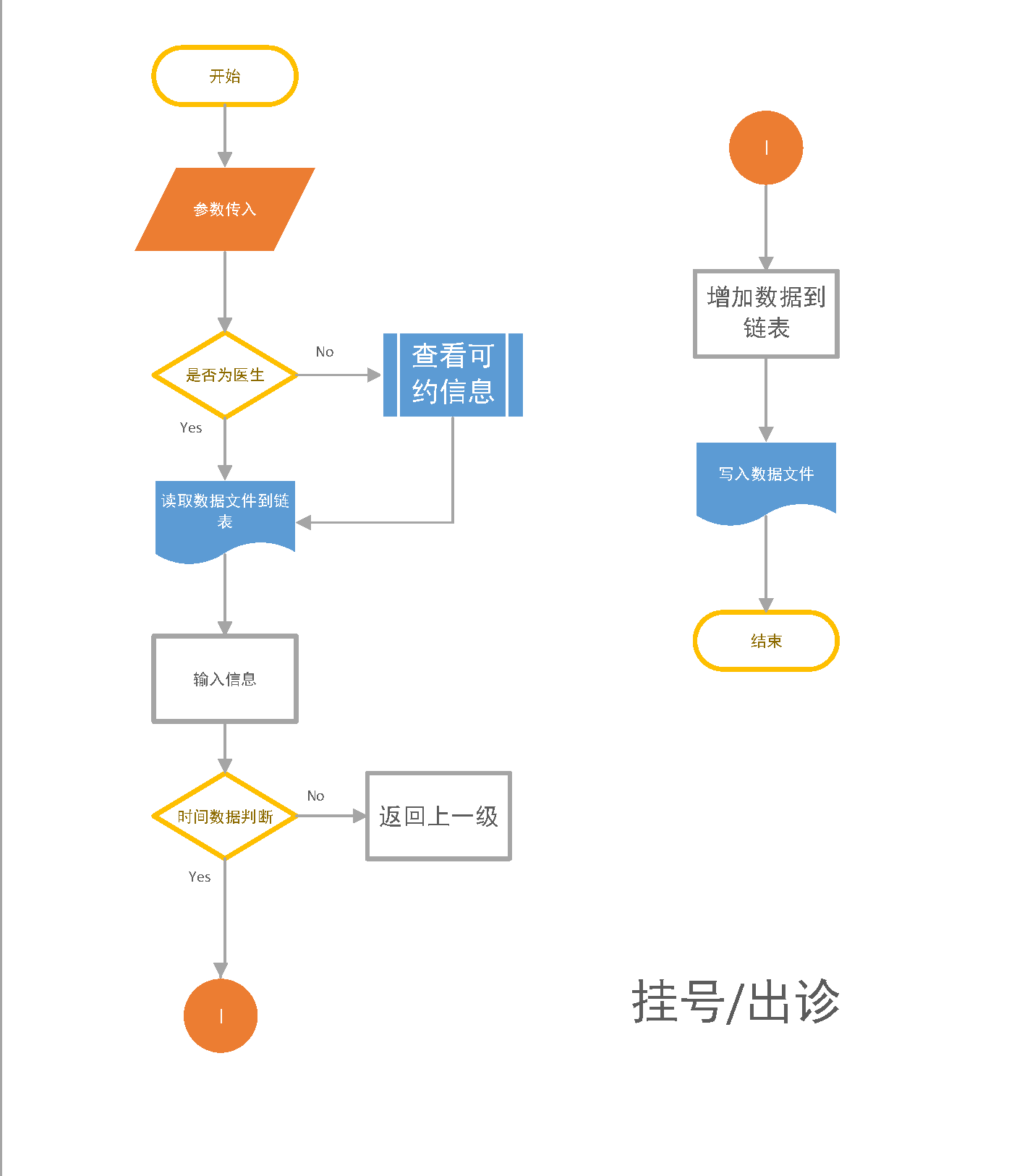


医生与用户共用的找回密码模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：extern void findPassword(int order)

//两端口共用的找回密码函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (5)挂号/出诊模块

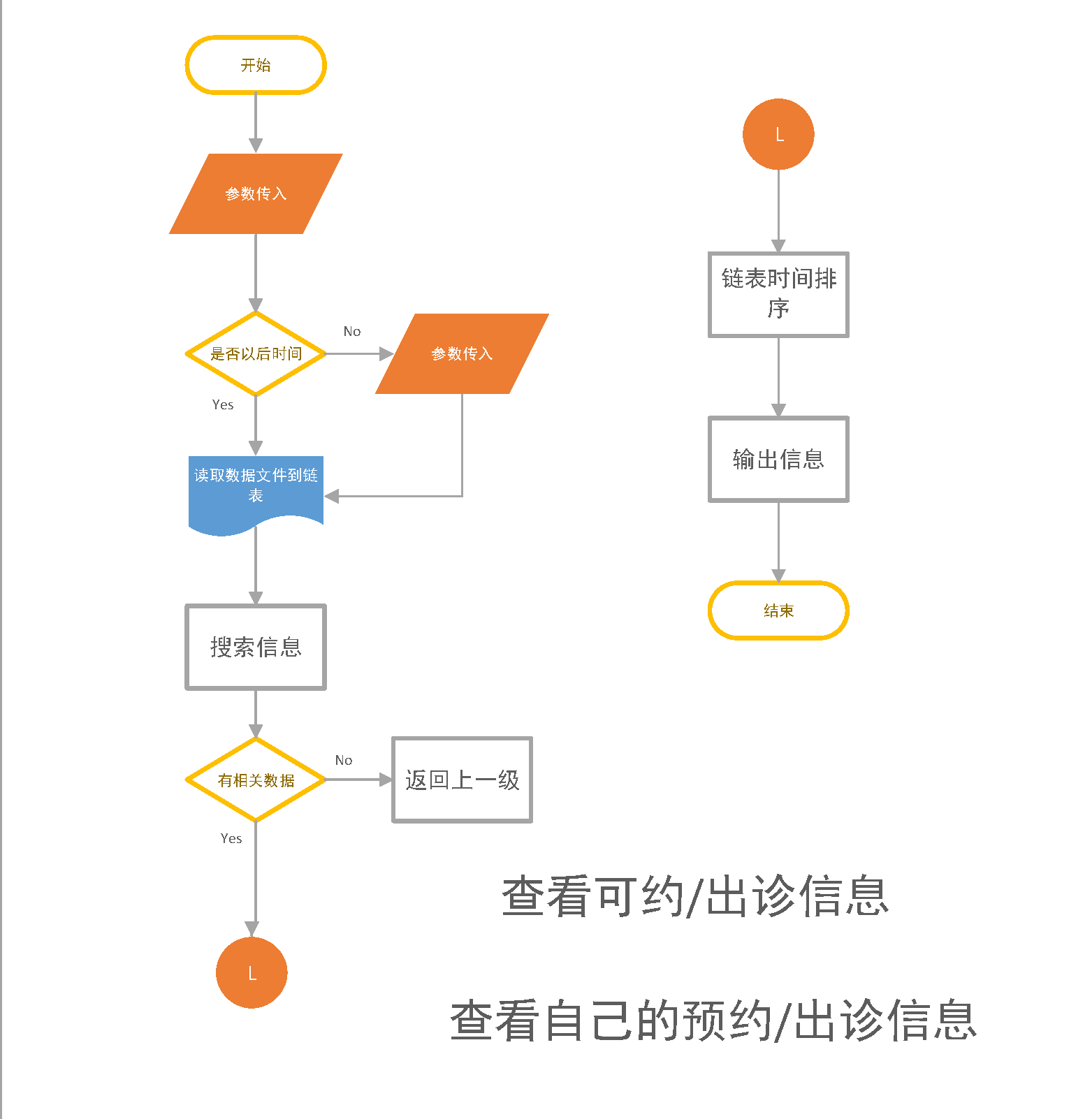


医生与用户共用的挂号/出诊模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：extern void lnput(char u[],char n[],char office[],int order)

//两端口共用的挂号/出诊模块，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；其他为传入的身份信息，包括账号、姓名、科室

### (6)查看模块



医生与用户共用的信息查看模块，通过传入的参数判断身份

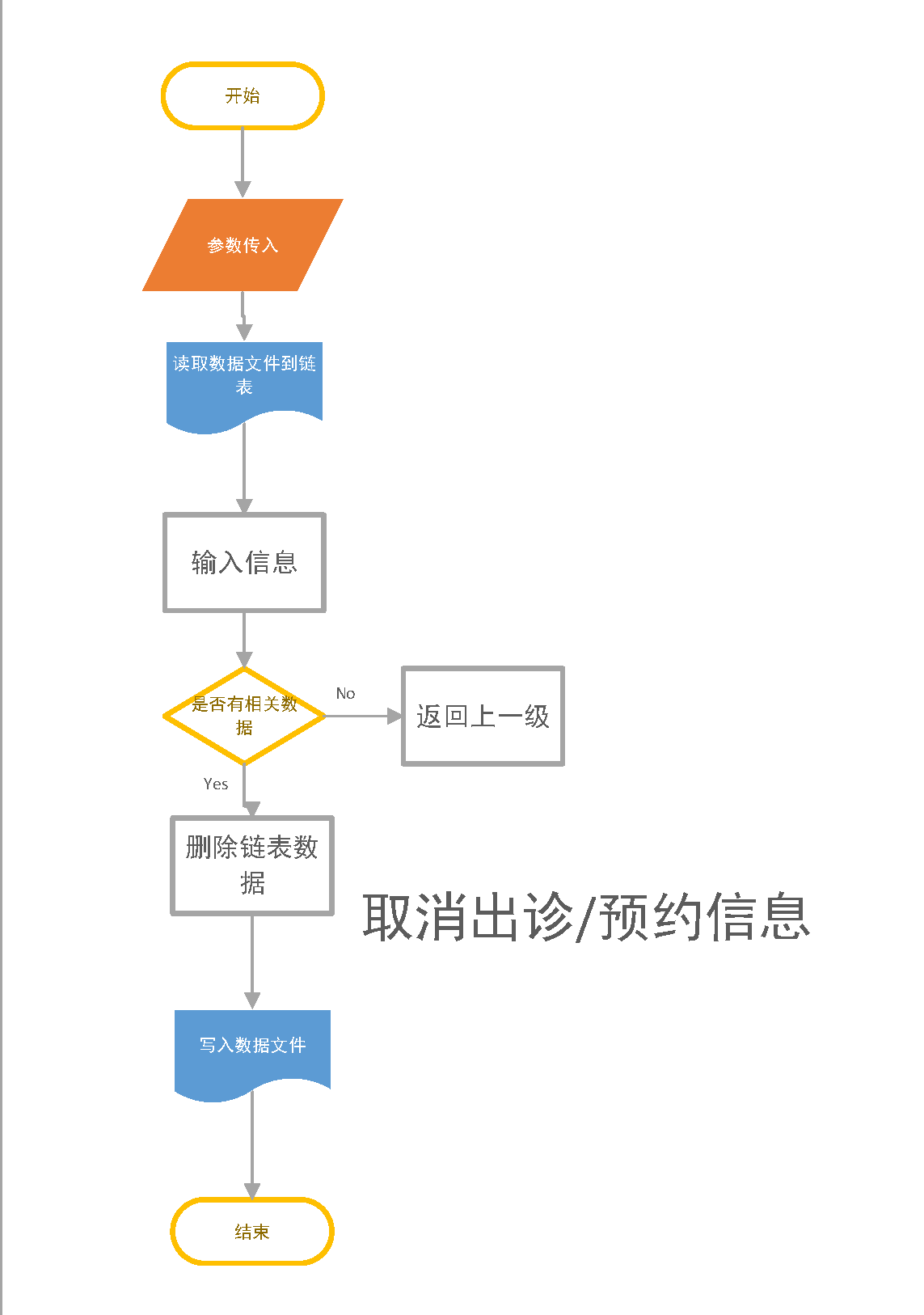
源文件中，此模块函数的声明为：

void otherLook(char u[],char n[],char doctor1[],char office[],int order,int time)

void findMe(char u[],char n[],char office[],int order,int time)

//两端口共用的查看模块，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；time为判断查看以前还是今后的参数，以前为0，以后为1；其他为传入的身份信息，包括账号、姓名、科室

### (7)取消出诊/预约模块

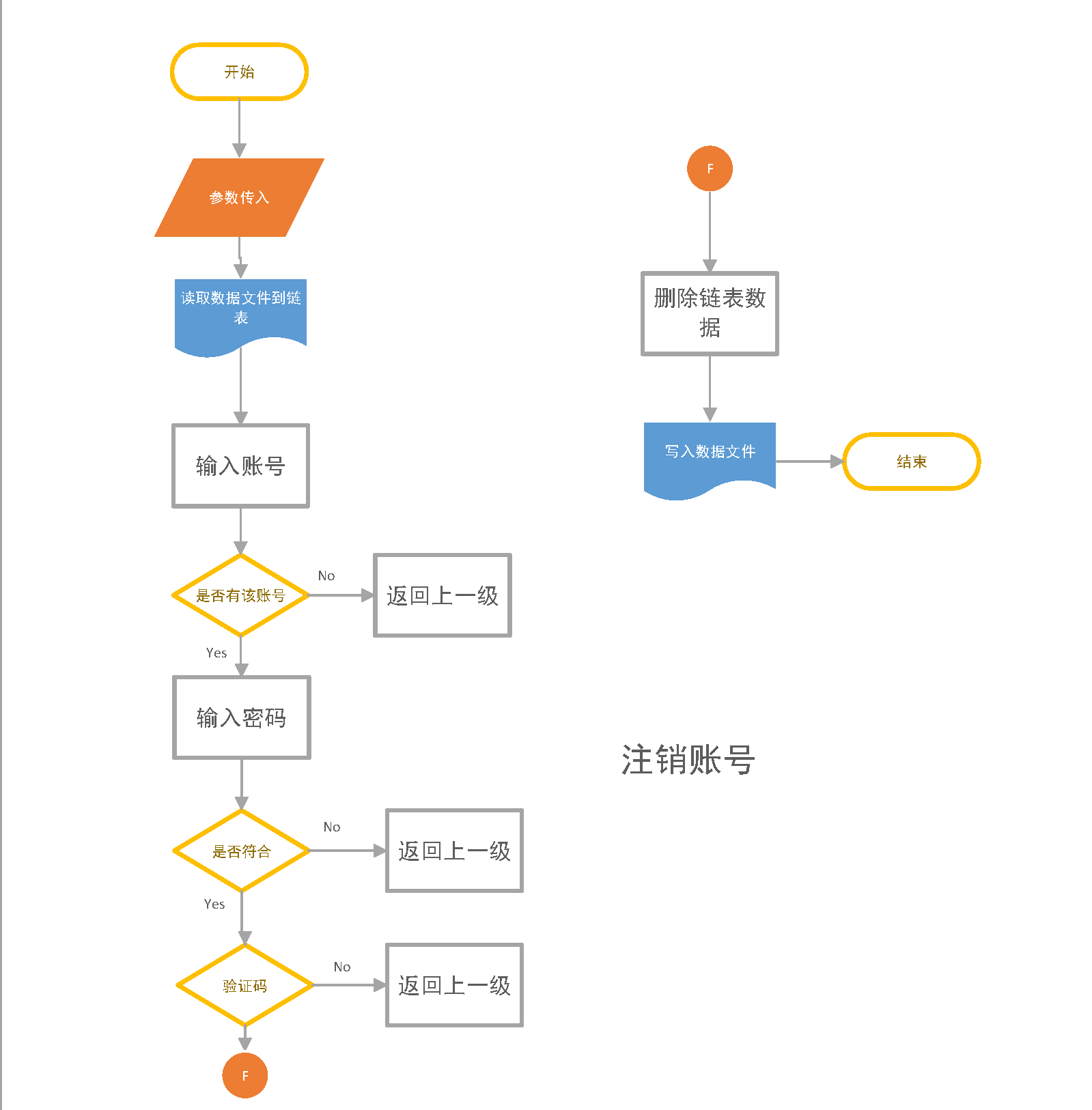


医生与用户共用的取消出诊/预约模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：void cancel(char u[],char n[],char office[],int order)

//两端口共用的取消出诊/预约模块，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；其他为传入的身份信息，包括账号、姓名、科室

### (8)注销账号模块

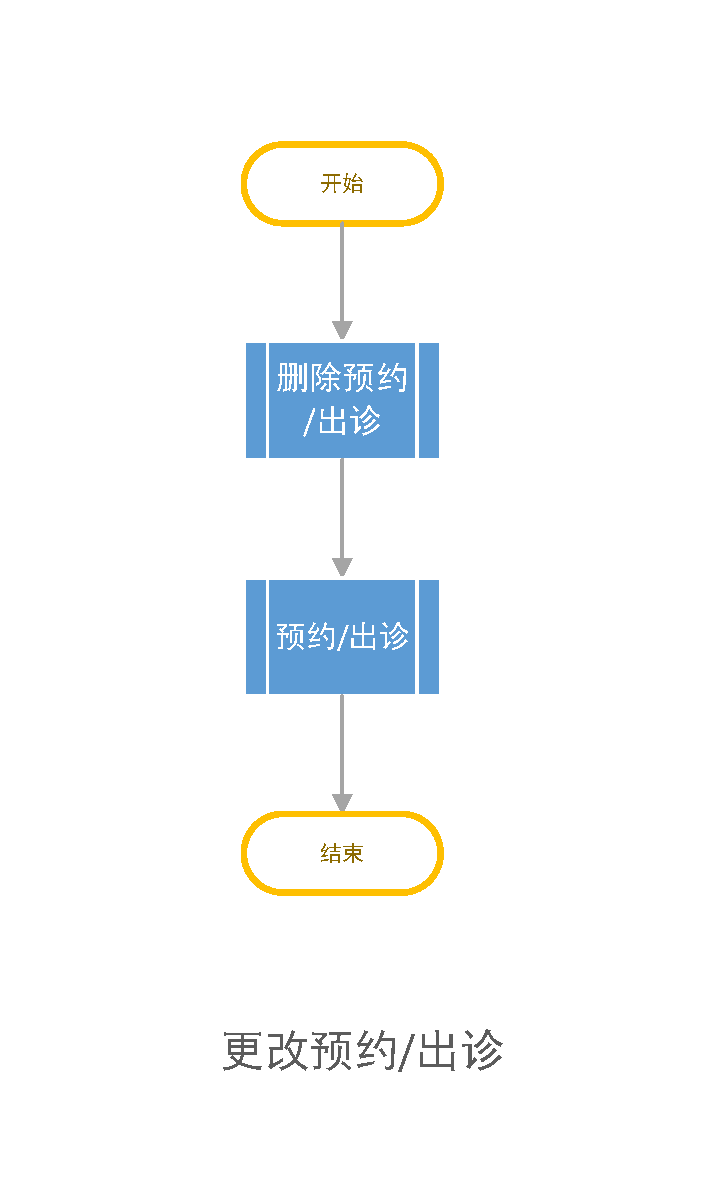


医生与用户共用的注销账号模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：void freeUser(int order)

//两端口共用的注销账号模块，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (9)更改预约/出诊模块



医生与用户共用的更改出诊/预约模块，通过传入的参数判断身份

源文件中，此模块函数的声明为：

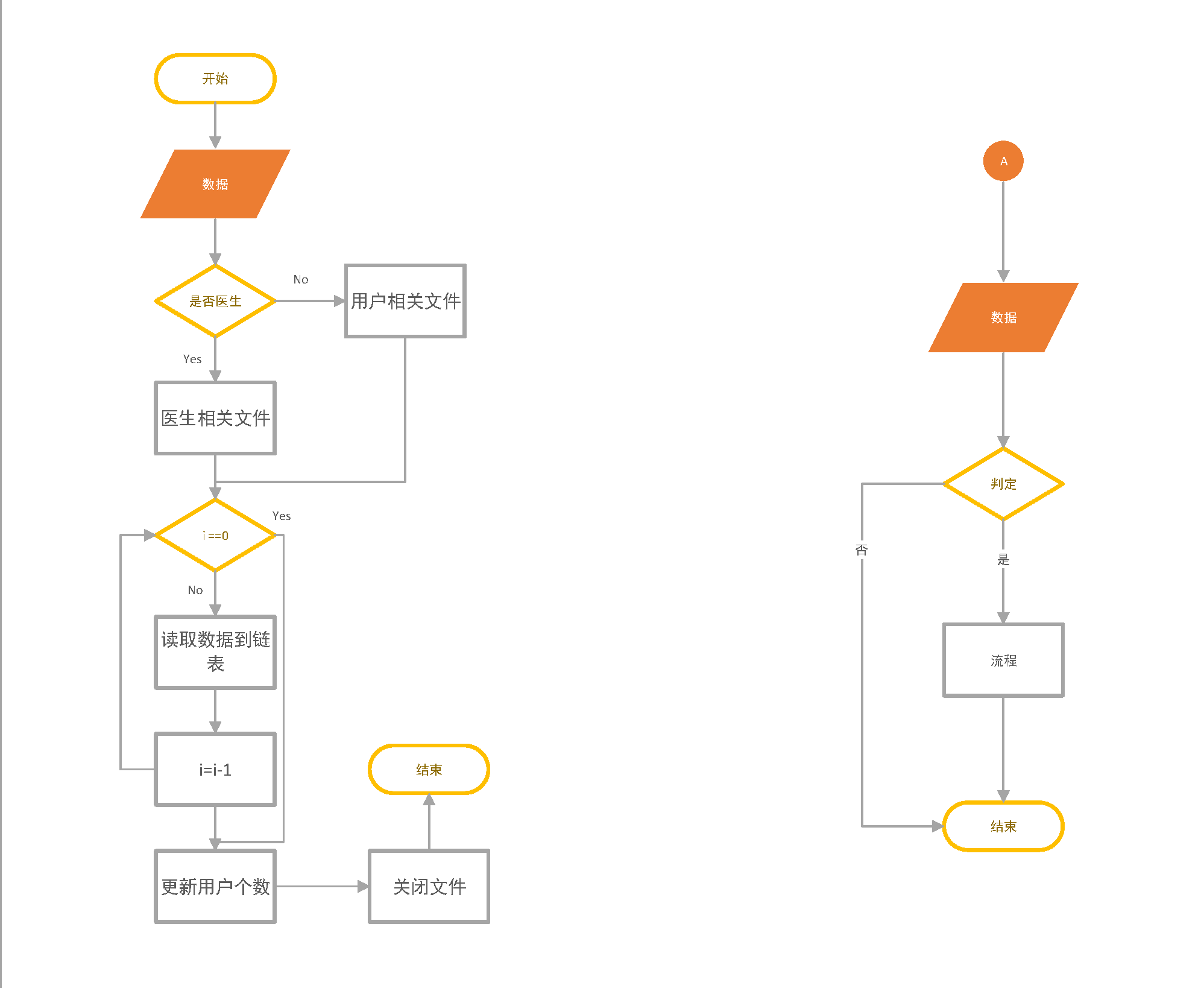
void cancel(char u[],char n[],char office[],int order)

extern void lnput(char u[],char n[],char office[],int order)

//两端口共用的更改出诊/预约模块，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；其他为传入的身份信息，包括账号、姓名、科室

## 2.3模块细致函数介绍

### (1)文件读取函数



用户和医生共用的文件读取函数，通过传入的参数判断身份

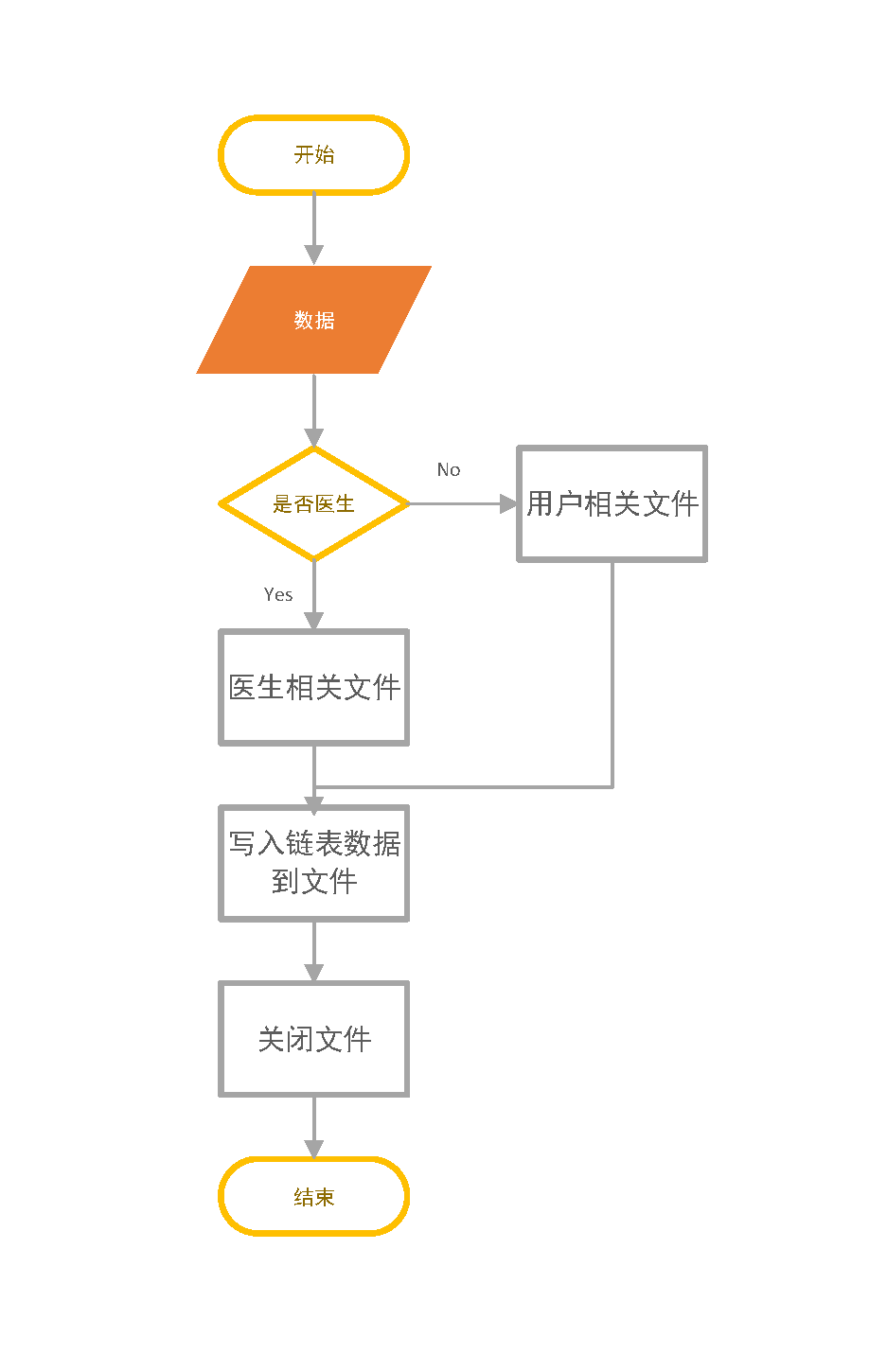
源文件中，此模块函数的声明为：

void ioRead\_input(Formlistp \*pList,int order)

void ioRead(List \*pList,int order)

//两端口共用的文件读取函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；另一个是链表指针

### (2)文件写入函数



用户和医生共用的文件写入函数，通过传入的参数判断身份

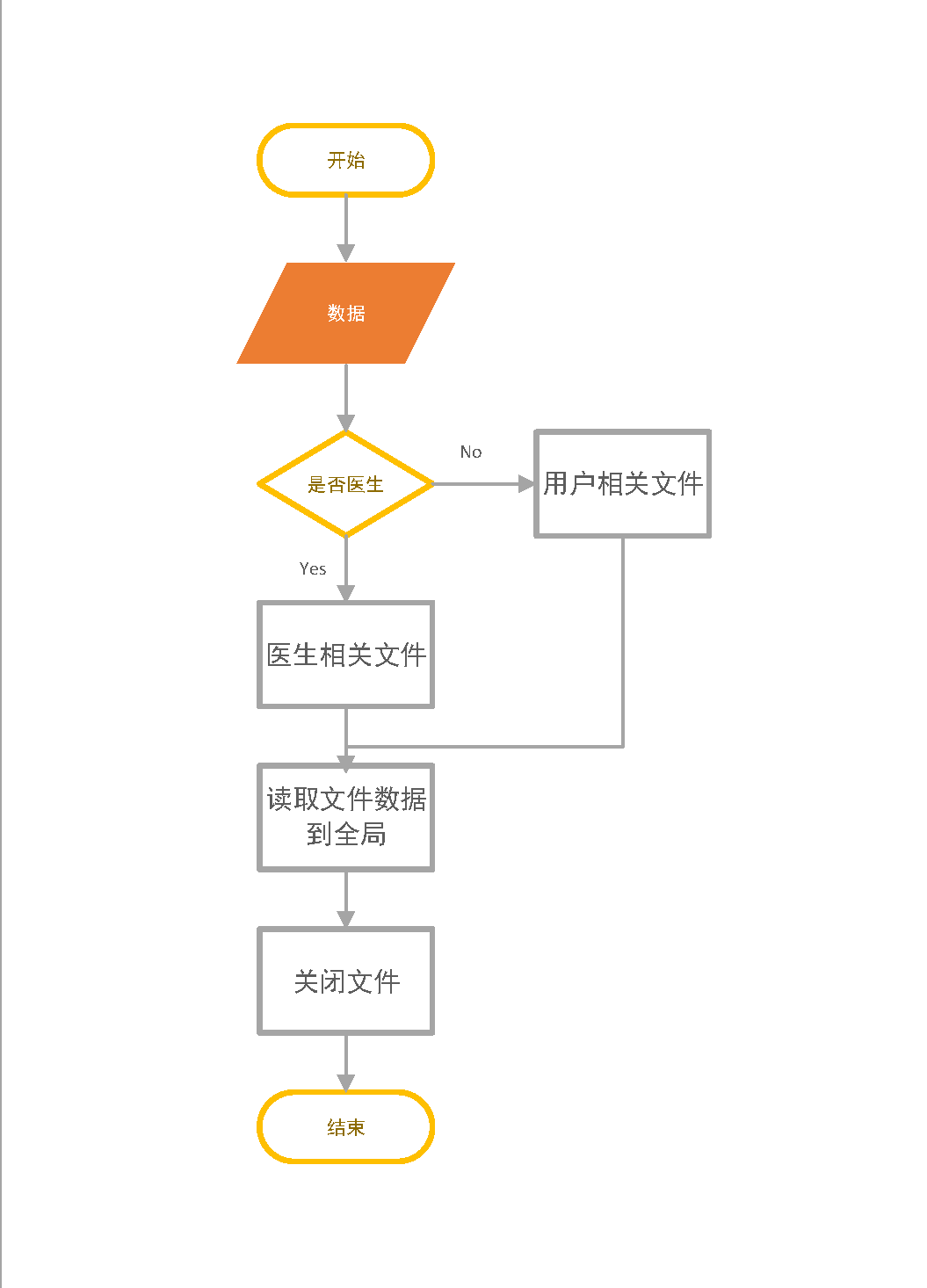
源文件中，此模块函数的声明为：

void ioWrite\_input(Formlistp \*pList,int order)

void ioWrite (List \*pList,int order)

//两端口共用的文件写入函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；另一个是链表指针

### (3)数量读取函数



用户和医生共用的数量读取函数，通过传入的参数判断身份

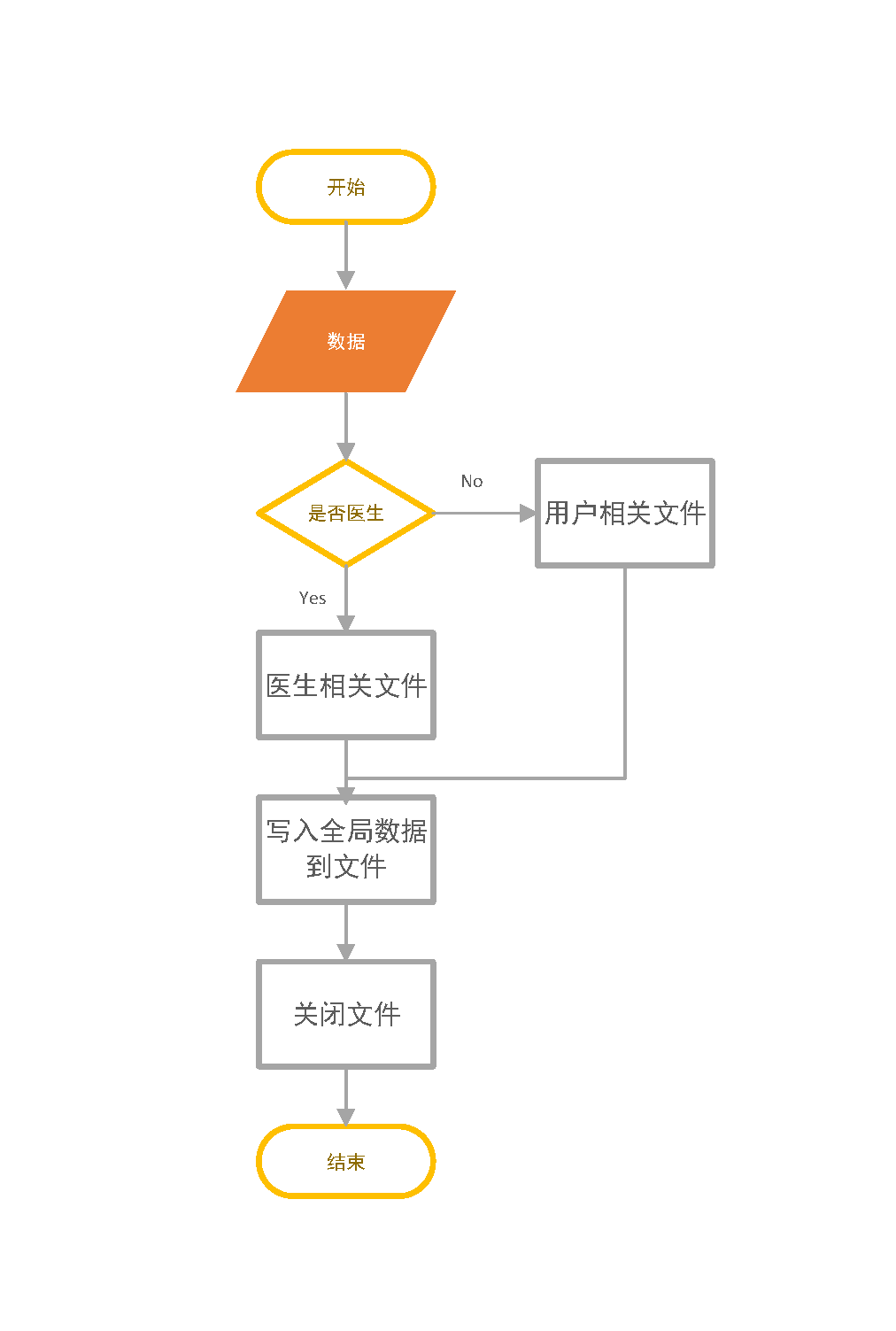
源文件中，此模块函数的声明为：

void usernumR

void inputnumR(int order)

//两端口共用的数量读取函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (4)数量写入函数



用户和医生共用的数量写入函数，通过传入的参数判断身份

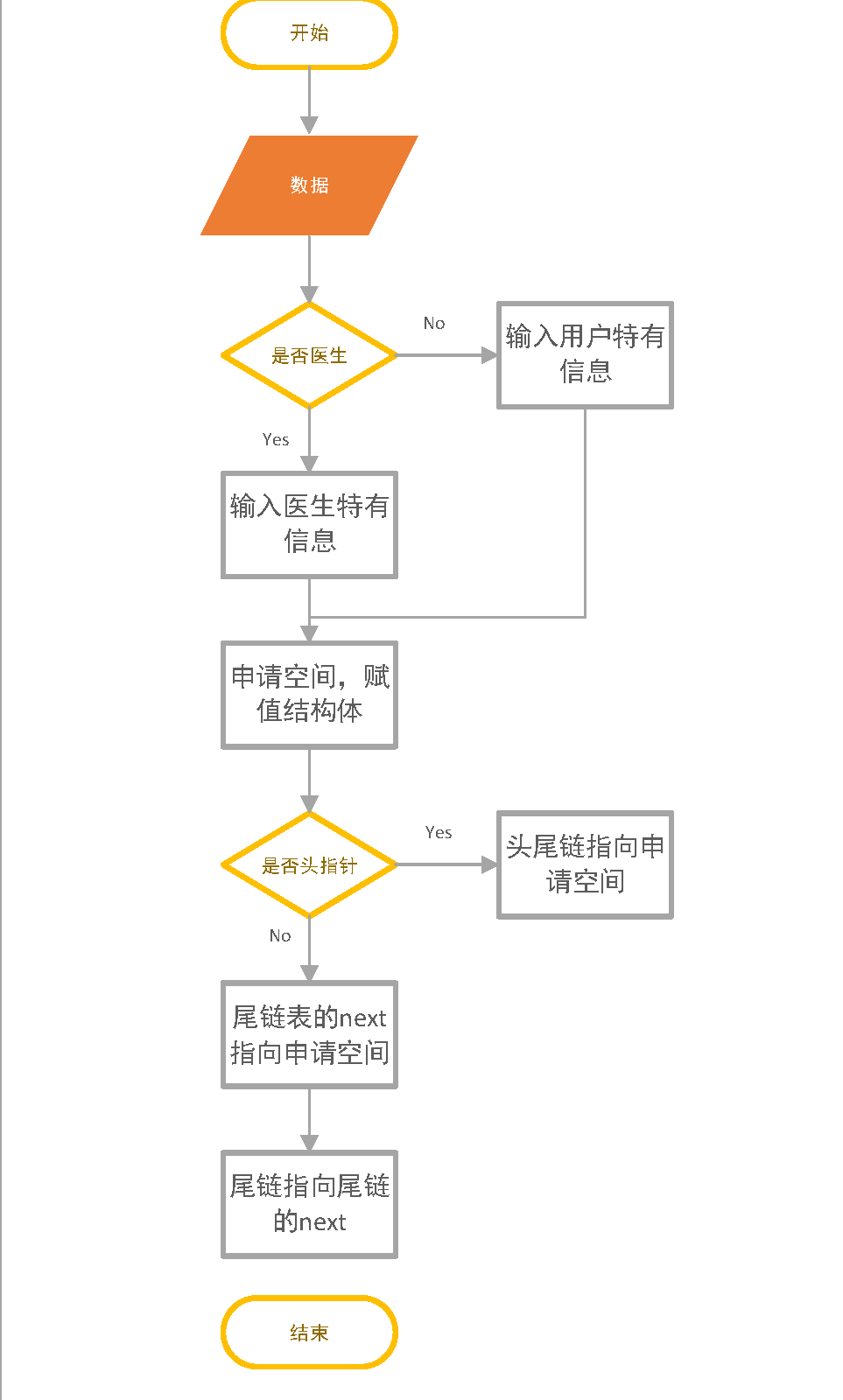
源文件中，此模块函数的声明为：

void usernumW

void inputnumW(int order)

//两端口共用的数量写入函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1

### (5)链表新增函数



医生与用户共用的链表新增函数，通过传入的参数判断身份

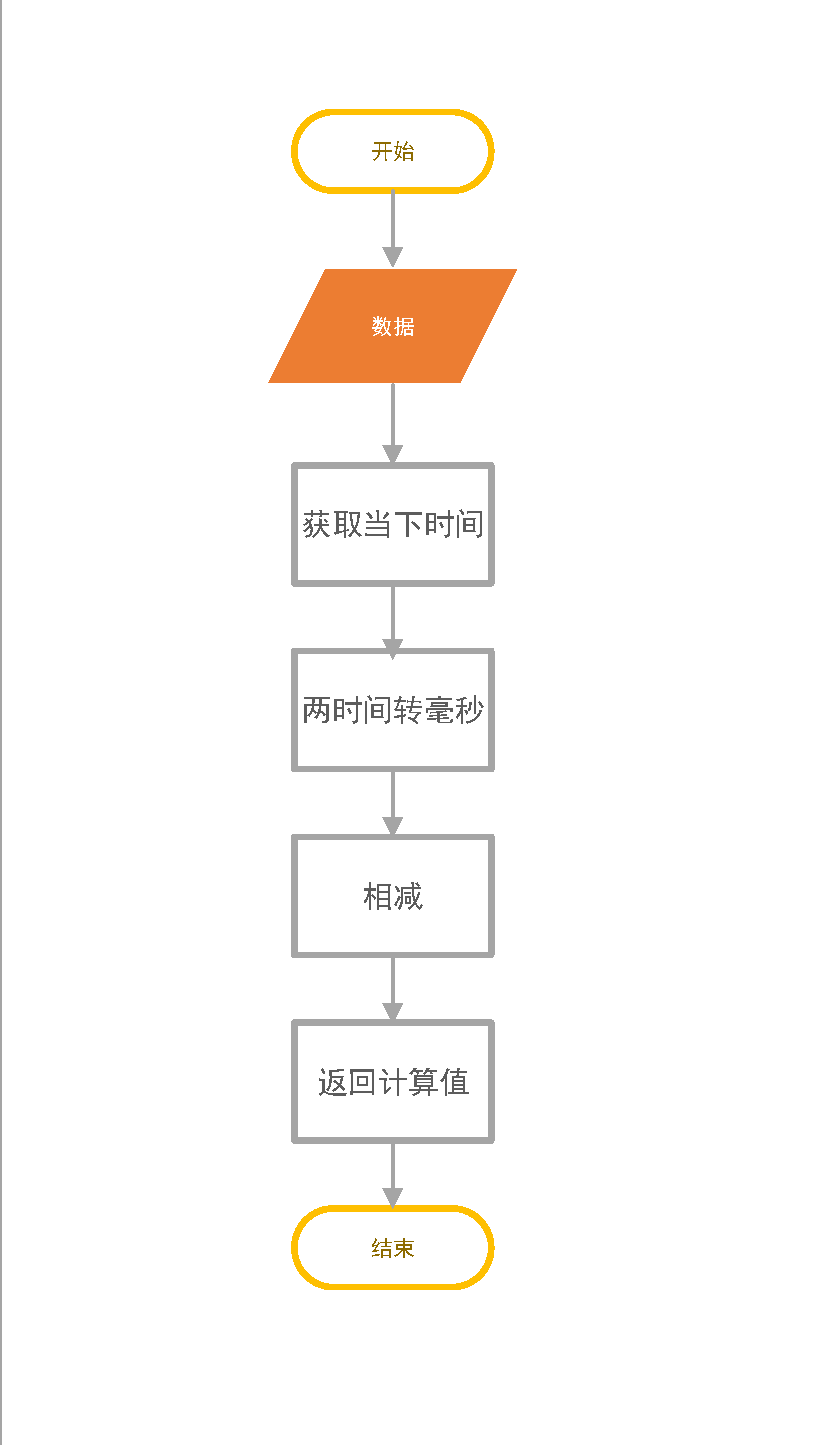
源文件中，此模块函数的声明为：

void Addinput(Formlistp \*pList,char u[],char n[],int year,int month,int day,char doctor[],char office[],int number,int order)

void Add(List \*pList,char number[],char number1[],char number2[],char number3[],int order)

//两端口共用的链表新增函数，其中形参order为判断医生与用户的依据 ；用户为0，医生为1；一开始的为传入的链表，其他为传入的身份信息，包括账号、姓名、科室等

### (6)当下时间比较函数



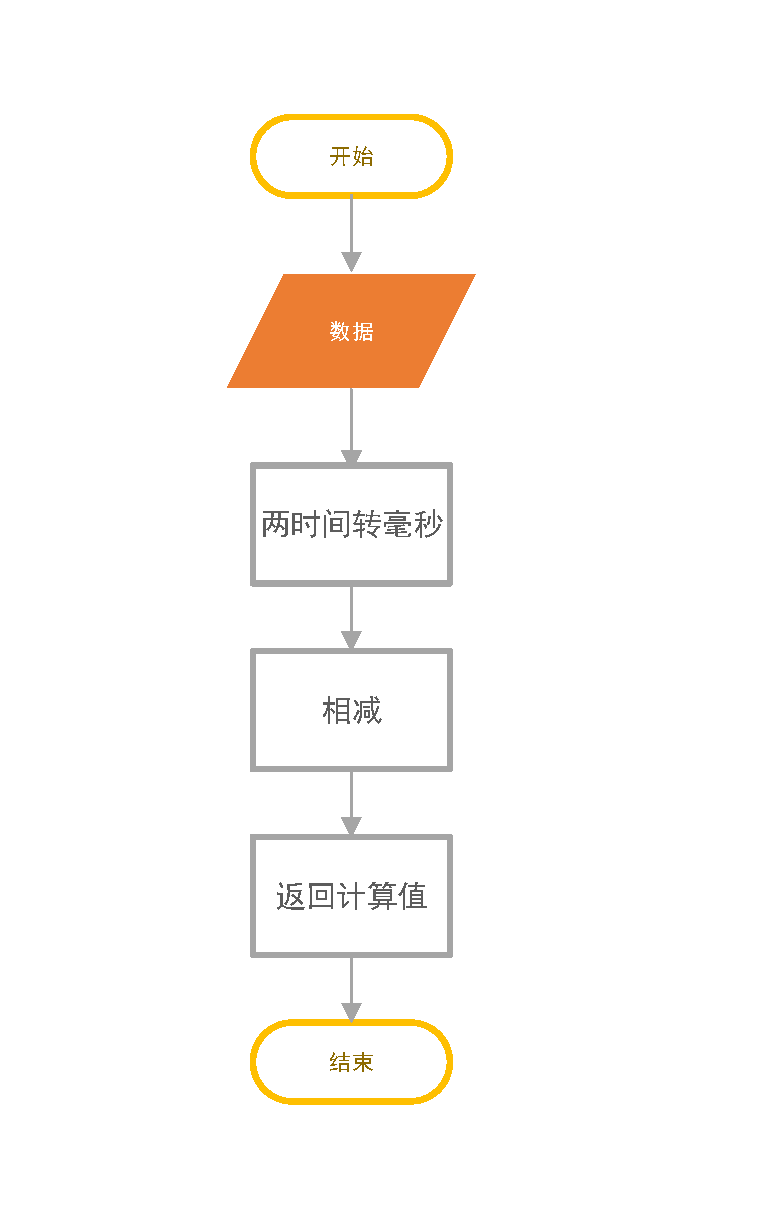
当下时间比较函数，传入的参数为要比较时间

源文件中，此模块函数的声明为：

int timeJudge(int year, int month, int day)

//当下时间比较函数，传入的信息为你需要比较的时间，包括年月日，与当下比较

### (7)时间比较函数



时间比较函数，传入的参数为要比较的两个时间

源文件中，此模块函数的声明为：

int timecpy(int year, int month, int day,int year1, int month1, int day1)

//时间比较函数，传入的信息为你需要比较的两个时间，包括年月日，相减比较

## 2.5数据设计

typedef struct patient//医生和用户登录数据共用的结构体

{

char user[30];

char name[30];

char password[30];

char office[30];

struct patient \*next;

}Patient;

typedef struct list//对结构体设置头尾指针

{

Patient \*head;

Patient \*tail;

}List;

typedef struct formp// 医生和用户的出诊及预约数据共用结构体

{

char user[30];

char name[30];

int timeYear;

int timeMon;

int timeDay;

char office[30];

char doctor[30];

int number;

struct formp \*next;

}Formp;

typedef struct formlistp//对结构体设置头尾指针

{

Formp \*head;

Formp \*tail;

}Formlistp;

struct tm //对time.h里结构体tm的再次声明，工程文件中无法自主读取

{

int tm\_sec; /\* 秒，范围从 0 到 59 \*/

int tm\_min; /\* 分，范围从 0 到 59 \*/

int tm\_hour; /\* 小时，范围从 0 到 23 \*/

int tm\_mday; /\* 一月中的第几天，范围从 1 到 31 \*/

int tm\_mon; /\* 月，范围从 0 到 11 \*/

int tm\_year; /\* 自 1900 年起的年数 \*/

int tm\_wday; /\* 一周中的第几天，范围从 0 到 6 \*/

int tm\_yday; /\* 一年中的第几天，范围从 0 到 365 \*/

int tm\_isdst; /\* 夏令时 \*/

};

int n = 0;//记录用户数量的全局变量

int m = 0;//记录医生数量的全局变量

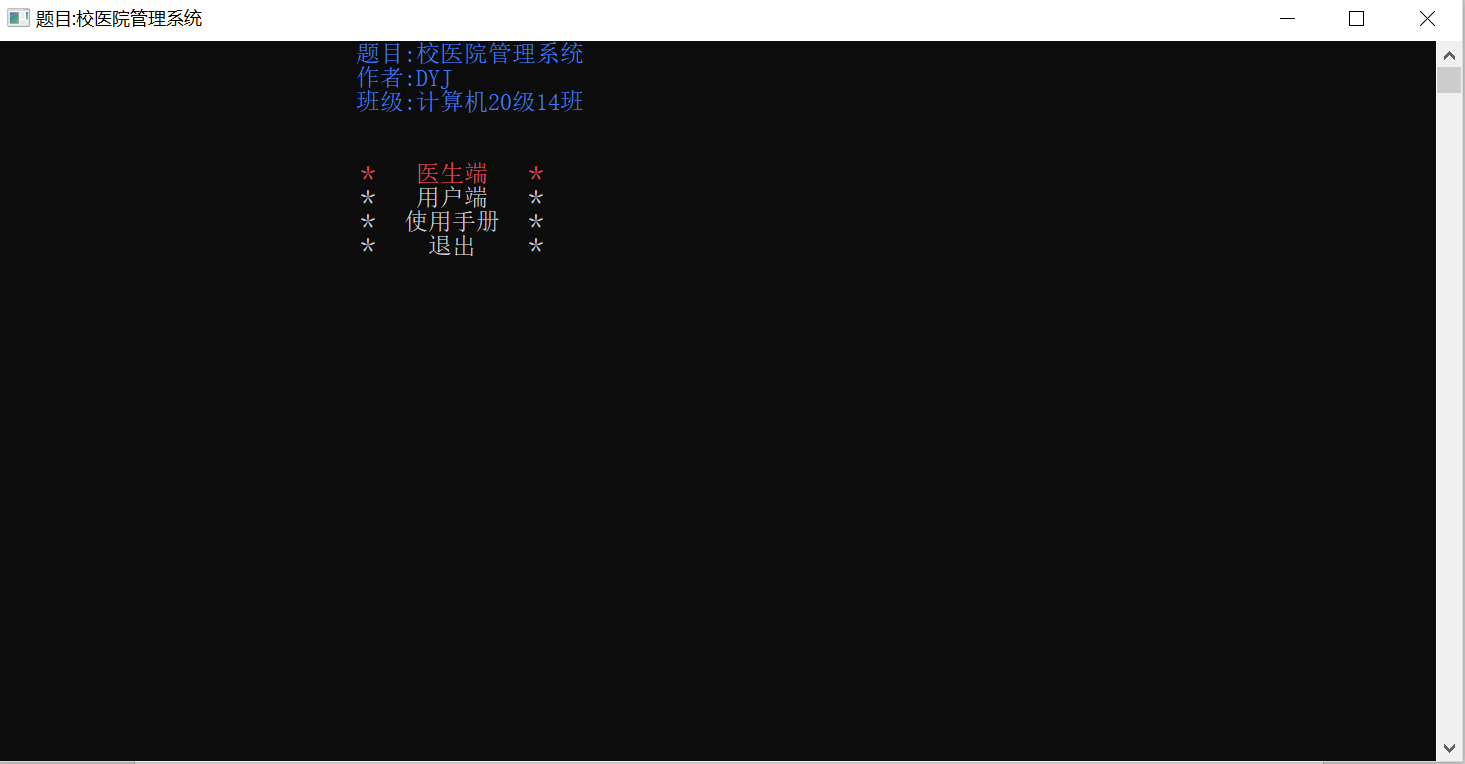
int inputp = 0;//记录预约数量的全局变量

int inputd = 0;//记录出诊数量的全局变量

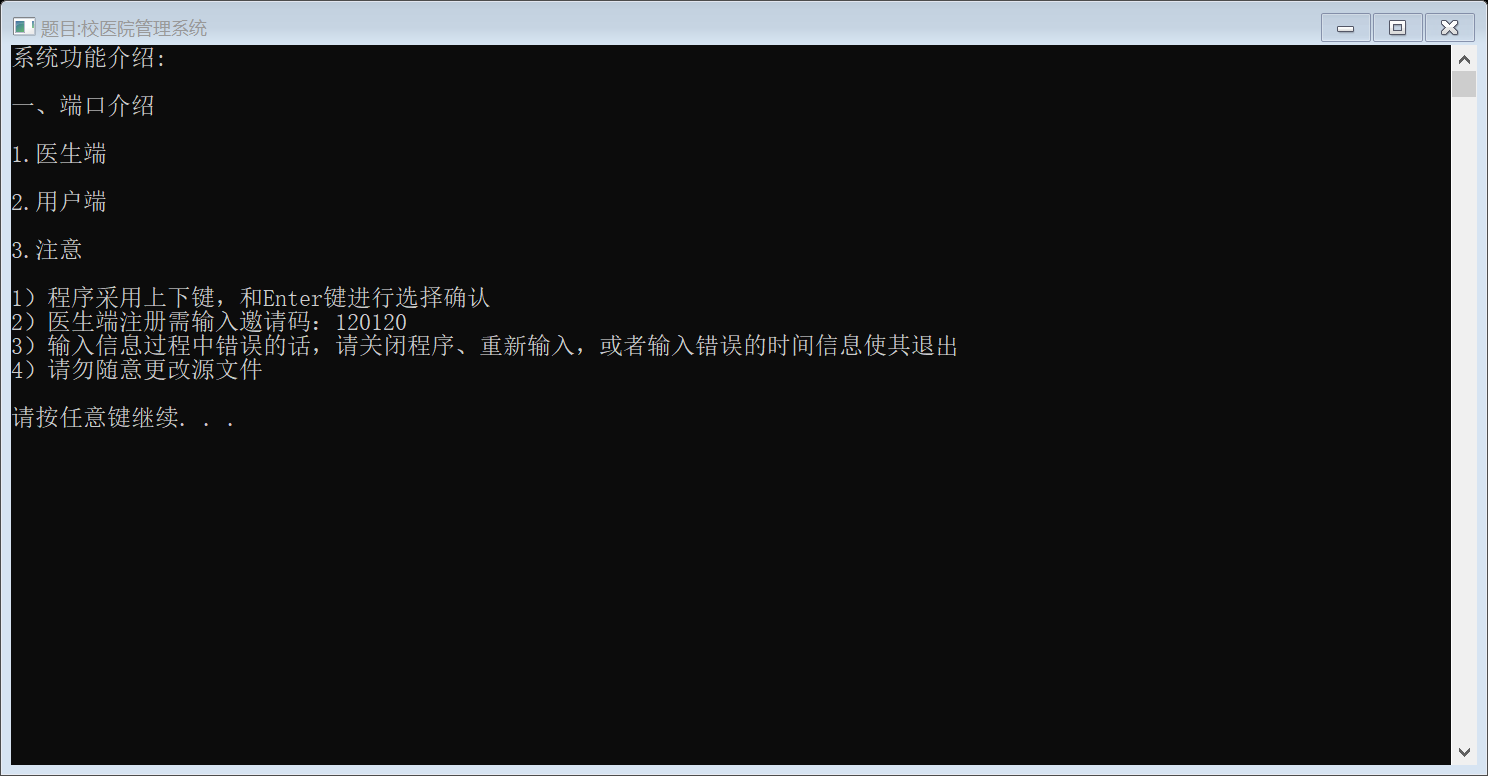
以上定义的结构体，其中结构体都是医生端和用户端共用的，分为登录信息类和预约/出诊类，另一个结构体为对time.h文件的结构声明。有意义的是对结构体的定义不是直接为指针，这样方便以后的调用，采取另一个结构为指针，其中包含头链表尾链表，里面的尾链表可以减少运算的重复。

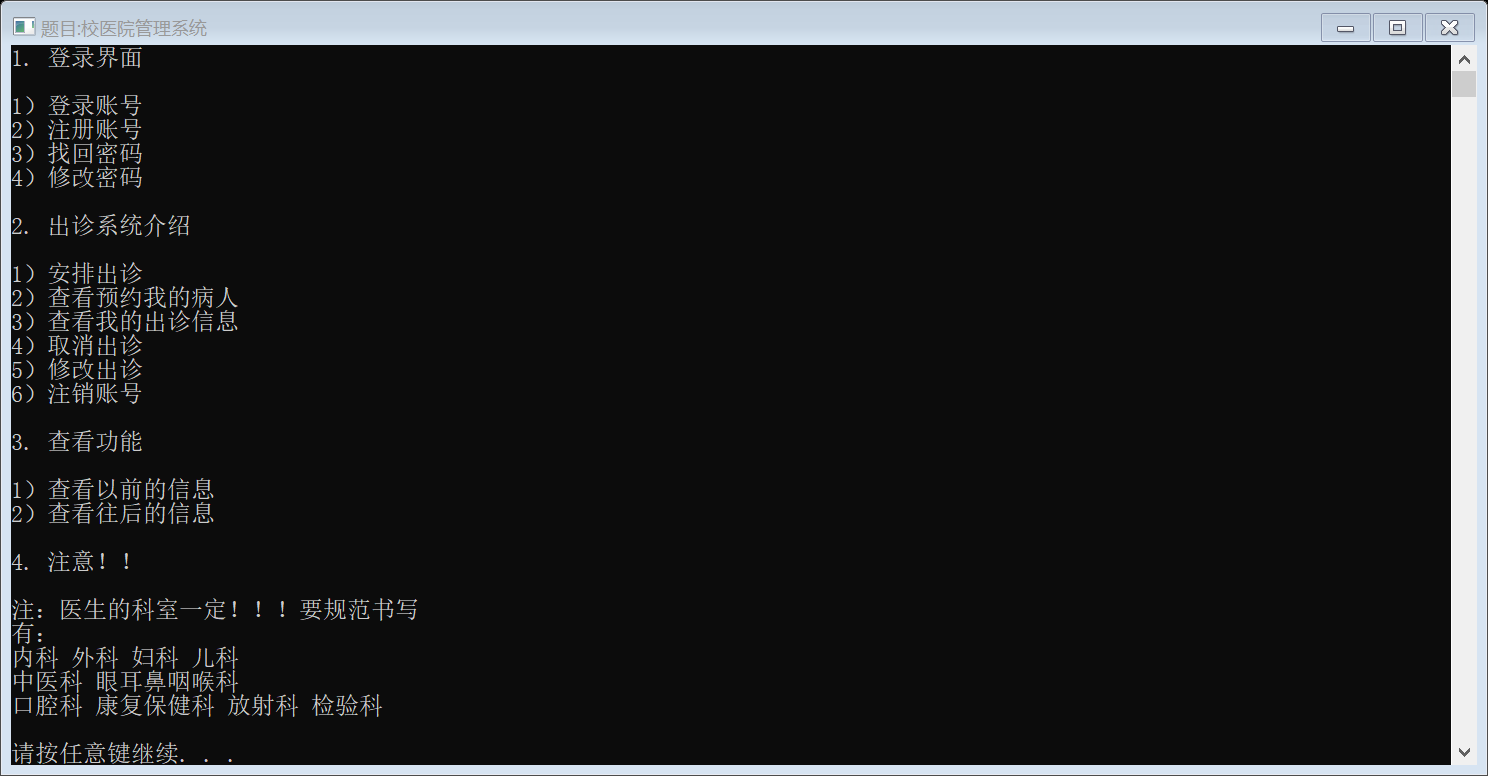
四个全局变量是分别记录四个文件里面的数据个数的，这样使读取更加容易。

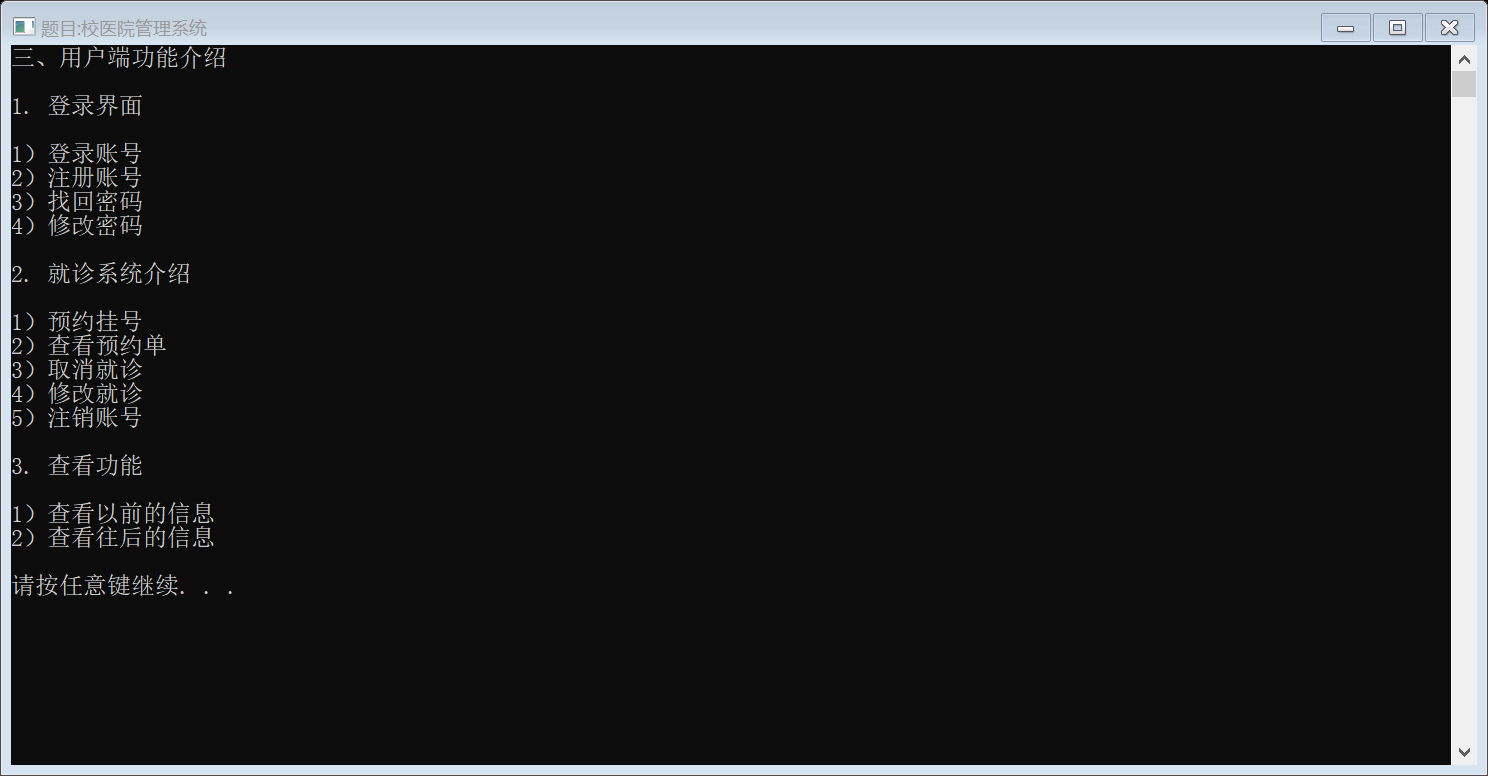
# 三．运行效果



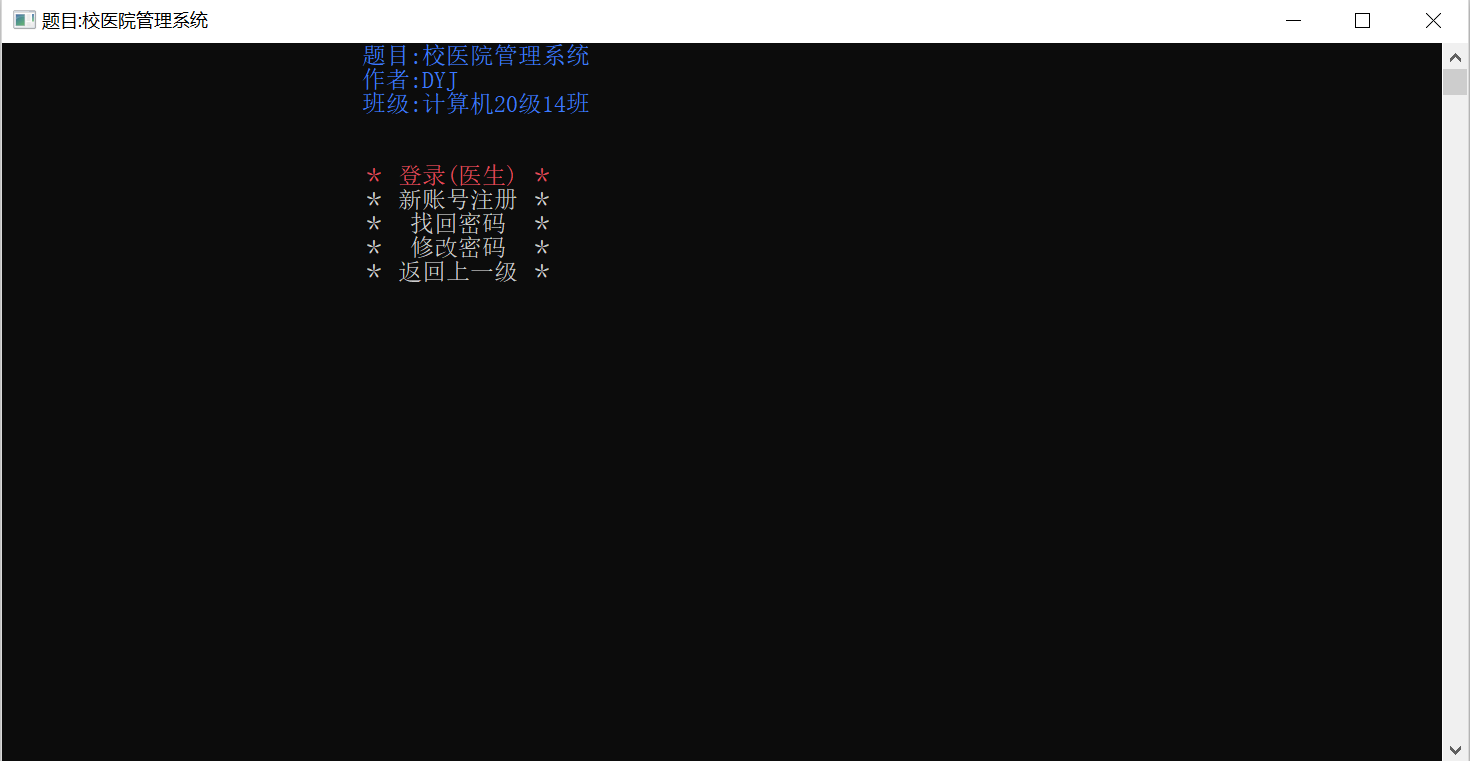
**主菜单**



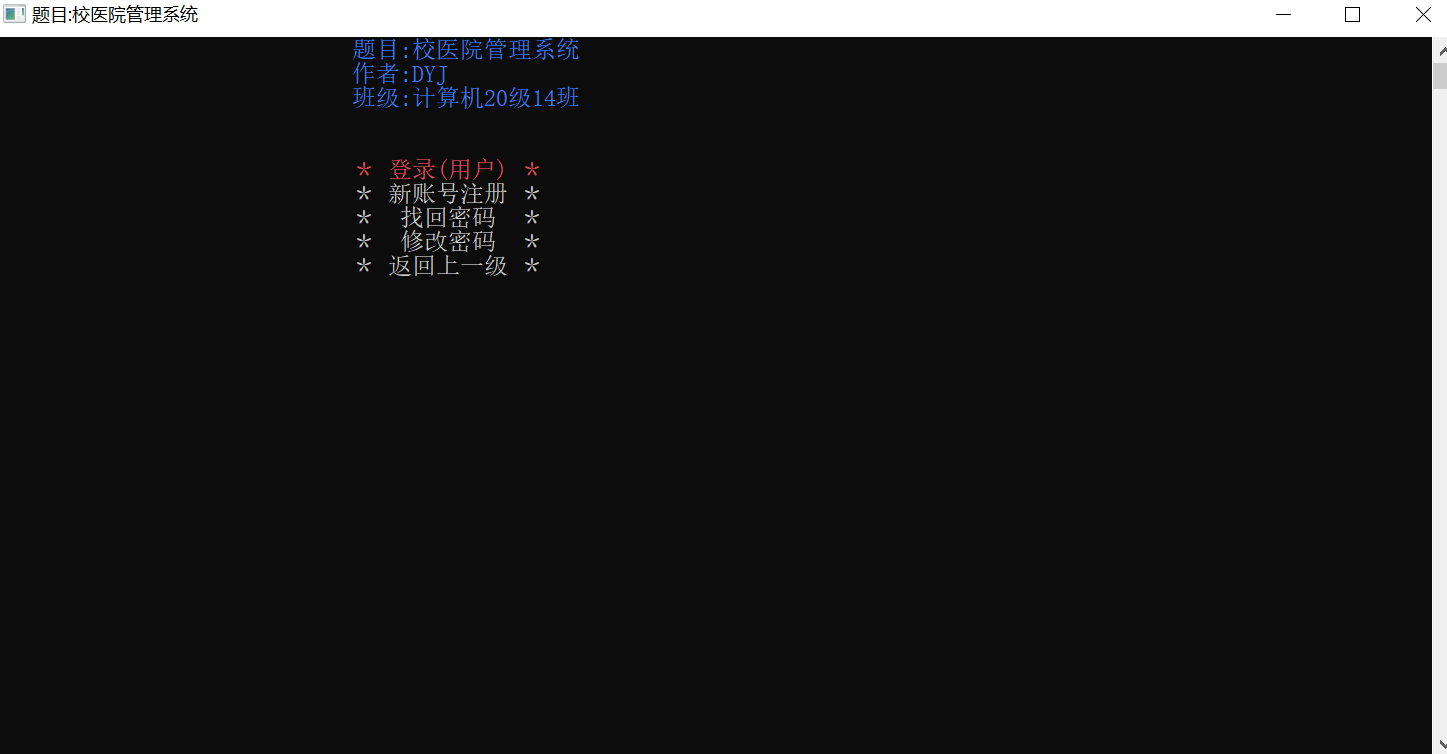




**使用手册页面**



**医生端菜单**



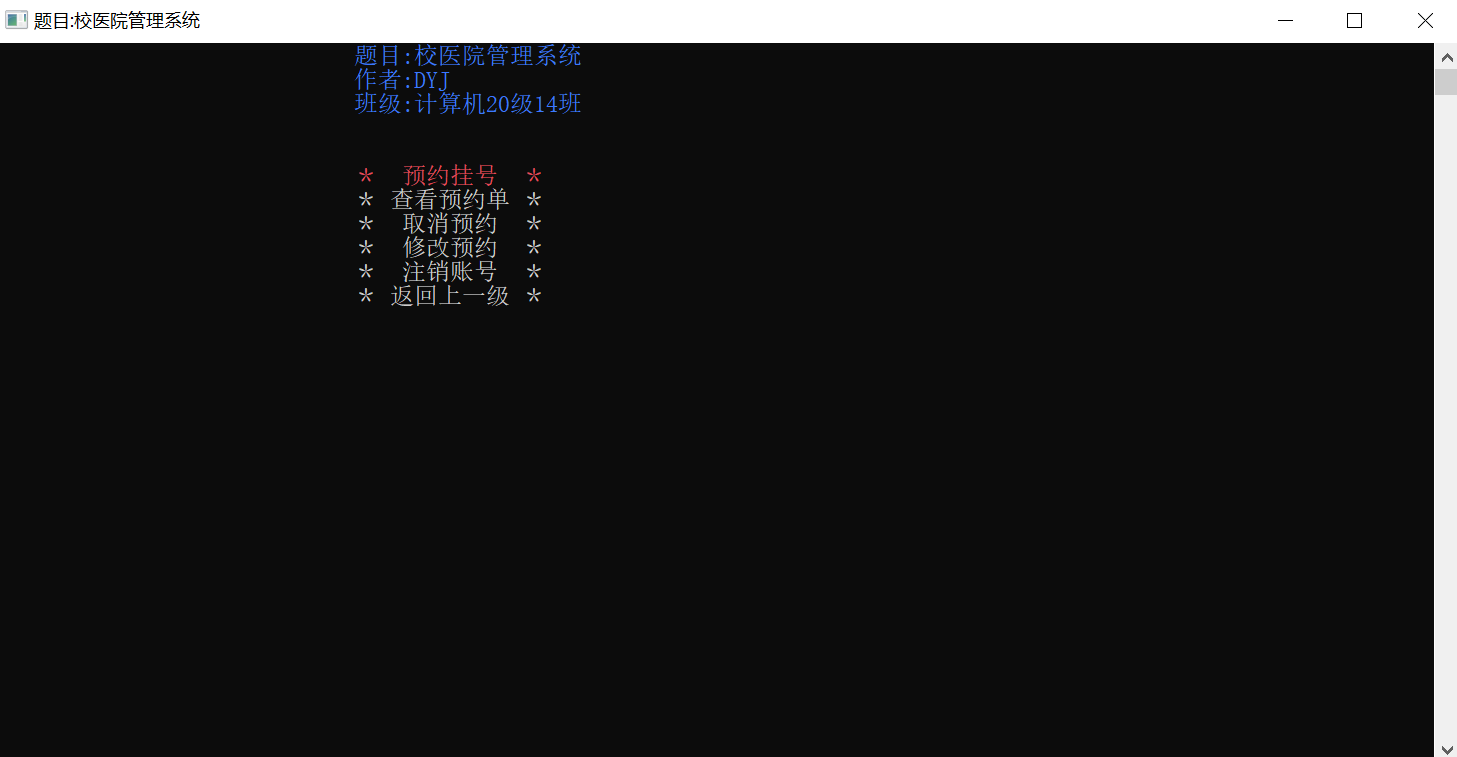
**用户端菜单**



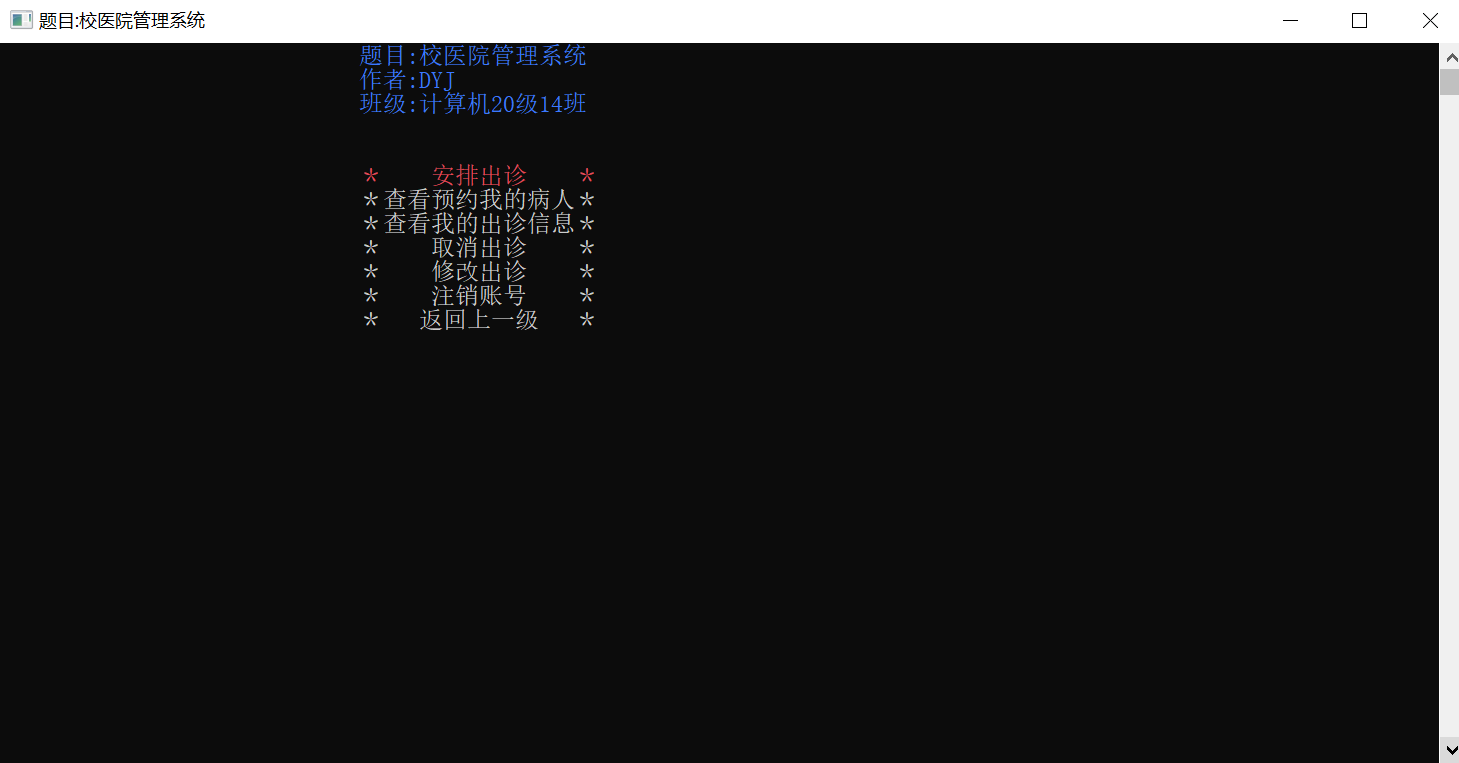
**登录页面**

****

**注册页面**

****

**用户端功能界面**

****

**医生端功能界面**

****

**数据查看**

# 四．总结

(1)分文件很重要，易于找bug

(2)多把功能设为函数，程序设计起来容易些

(3)写程序要有耐心和细心，程序崩溃了，人不能崩溃

(4)相通函数设为一个，方便改，也简单

(5)函数命名用有意义的组合单词，这样好找

# 致谢

感谢李小妹老师的悉心指导，让我有机会参加评优和把程序改的更好。也感谢坚持的自己和支持的伙伴朋友。

# 参考文献：

1.C语言图书管理系统设计报告

<https://jz-2017.blog.csdn.net/article/details/73504270>

2．用C语言计算两个日期之间的天数

<https://blog.csdn.net/CYL641/article/details/107667859>

3. C语言实现一个Window控制台带彩色

<https://yangyuanxin.blog.csdn.net/article/details/77485367>

4. C语言开发简单的学生成绩管理系统

https://summer2zz.blog.csdn.net/article/details/89320103