**2019级《程序设计基础课程设计》总结报告（第5组）**

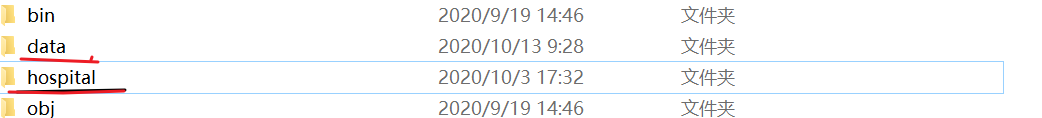
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **学号** | **姓名** | **性别** | **班级** | **具体任务分工** | **所占比例** | **成绩** |
|  |  |  |  | 结构体的设计、文件的组织、文件读写、抵抗非法输入的函数设计，菜单组织以及界面设计，加密解密模块，报告撰写 |  |  |
|  |  |  |  | 检索函数的撰写、查询打印函数的撰写、统计医生繁忙程度函数的撰写，参与代码的测试和验证 |  |  |
|  |  |  |  | 修改、删除诊疗记录功能，查询科室、医生、患者诊疗记录功能，打印科室、医生、患者诊疗记录功能，构造测试数据，参与测试和验证 |  |  |
| **填写说明：** | | | | | | |
| 1、请将首页红色部分信息填全，其中：学号为8位数字，计算机科学与技术学院通常以21开头、软件学院通常以55开头；班级为2位数字，保留首位的0；所占比例为百分数，比例值个位数为0或5，且所有人的所占比例之和为100%；人数不足的分组请保留后面的多余空行，请勿修改该表的结构。  2、请根据实际情况填写具体任务分工，主要任务包括：软件系统的总体分析与设计，具体功能的设计与实现，对应的测试与验证过程（报告正文需要列出若干组具体测试样例与对应结果），系统界面的设计与美工，以及辅助工具、视图和文件等。  3、成绩评定由指导教师填写，学生本人请勿填写和修改。 | | | | | | |

提交日期：2020年10月14日

**报告正文**

【请用小四号宋体填写，自行组织章节和段落】

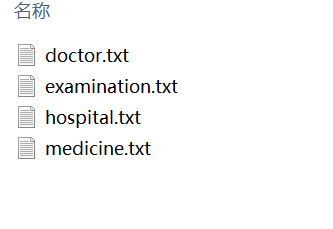
# 一、文件结构



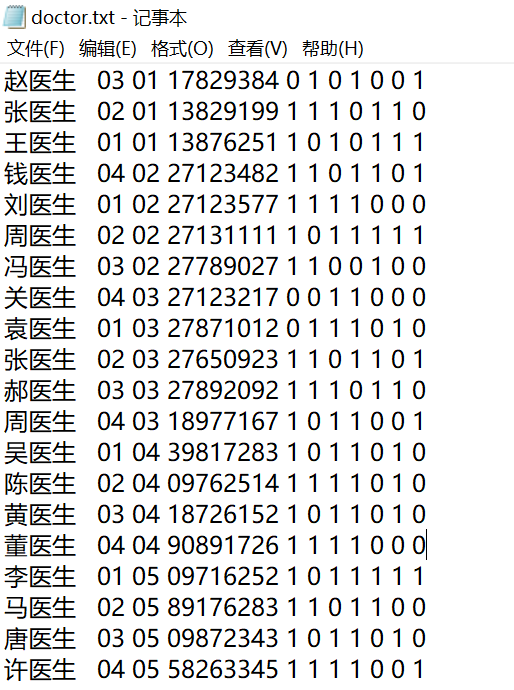
## 1、hospital文件夹

hospital文件夹存储了医院的信息。比如医院的所有检查项目、所有药品、所有医生、所有住院病人

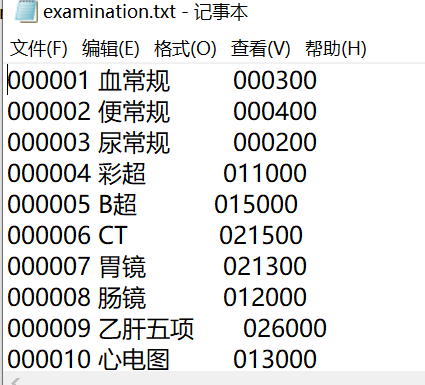
具体文件如下



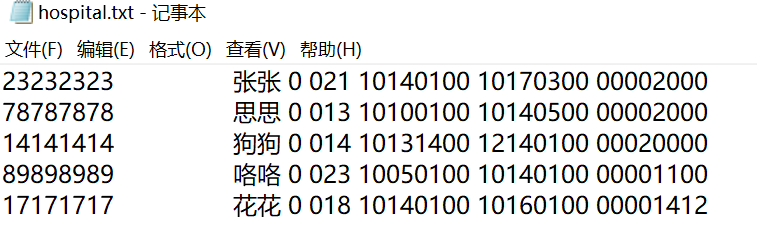
doctor.txt文件内容如下



自左至右，分别存储了所有医生的姓名、职级、科室、上班时间



自左至右，分别是序号、检查项目名、价格



这个文件夹存储了所有的住院患者信息，

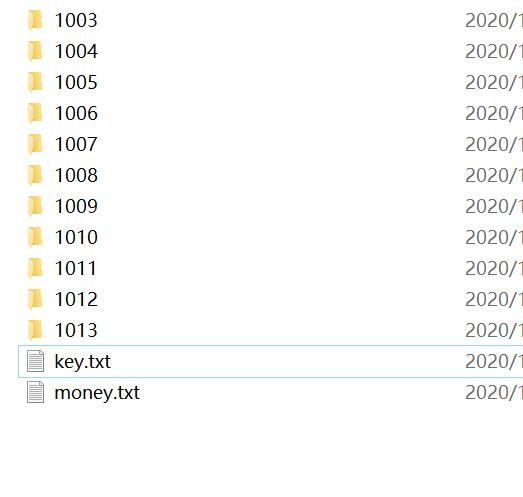
从左至右分别是 ID、姓名、性别、年龄、入院时间、出院时间、住院押金



从左至右分别是：序号、药品名、库存、价格

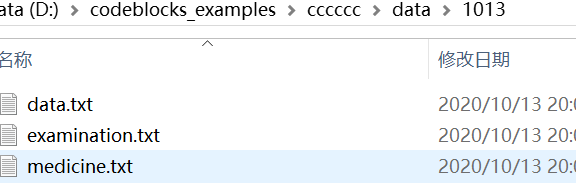
## 2、data文件夹

存储所有的诊疗记录。其中每天的数据在一个单独的文件夹中存储。



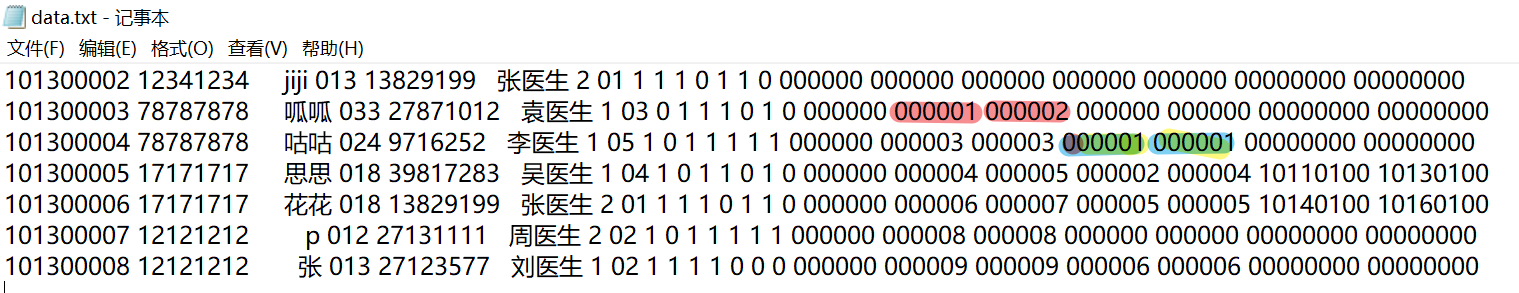
比如1003文件夹存储的是10月3日的数据

打开1013文件夹，内容是：



data.txt是诊疗记录的主体。examination.txt记录了进行的所有检查项目，medicine.txt记录了购买的所有药品。

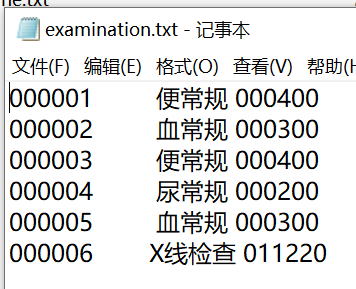
先看data.txt文件夹：



注意橙色区域：这是药品的索引值，表示中medicine.txt中第1、2条购药记录是属于患者“呱呱”的

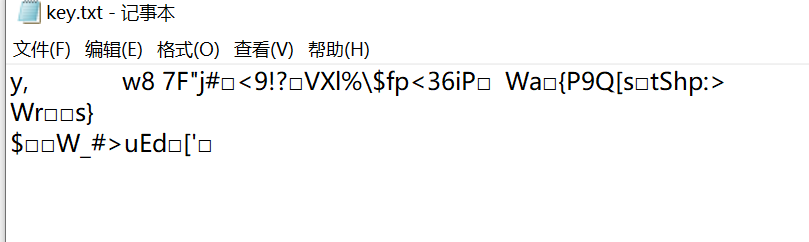
注意绿色区域：这是检查项目的索引值，表示examination.txt中第1条检查记录是属于患者“咕咕”的

而其他两个文件内容如下:

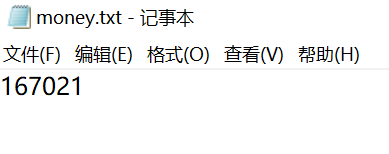




data文件夹中的其他文件：



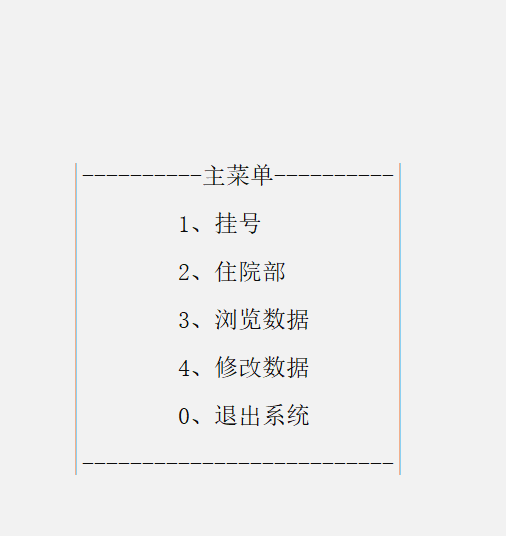
这里存储着系统密码。



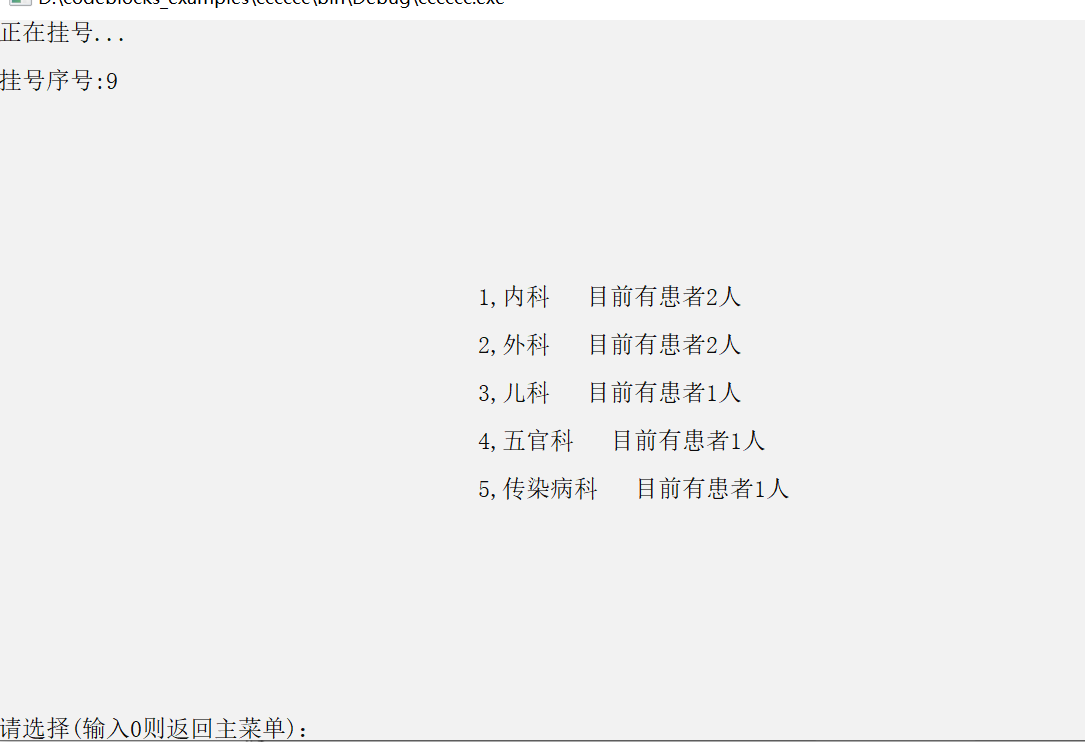
这里存储着医院的总营业额。

# 二、系统介绍：

主界面：



## 1）挂号系统：

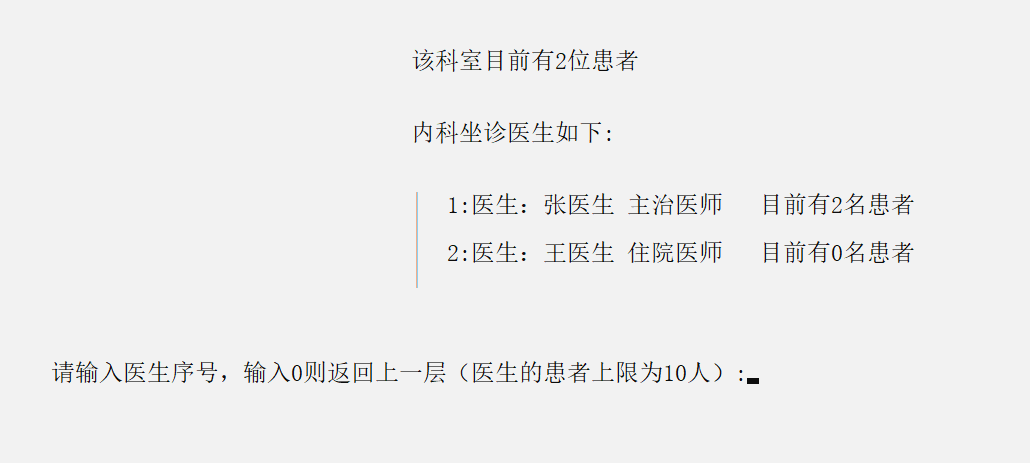


挂号系统运行前会自动加载需要的文件信息。

比如：

* 获取系统时间，并自动打开最新的文件夹，并建立链表（假如文件夹不存在就不建立链表）
* 加载所有的医生信息，并根据科室进行分类、并统计各科室已经挂了多少号

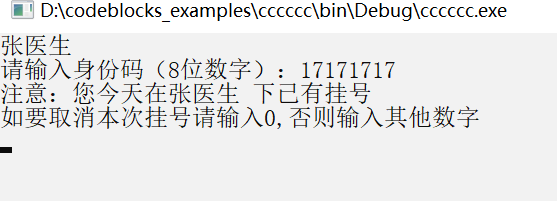
选择了“内科“，界面如下：



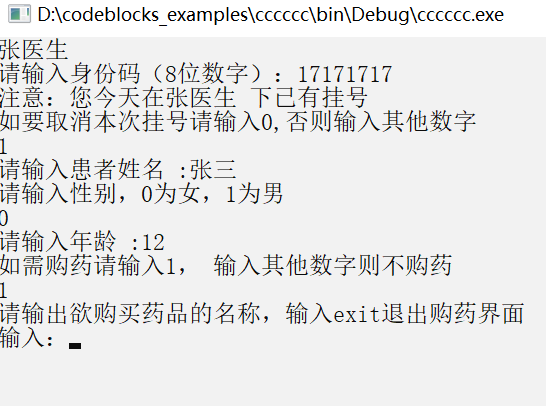
这里仅仅显示今天值班的医生，（自动获取今天时间，判断有哪些今天医生值班）

假如医生的患者上限超过10，则必须重新选择医生

选择了张医生

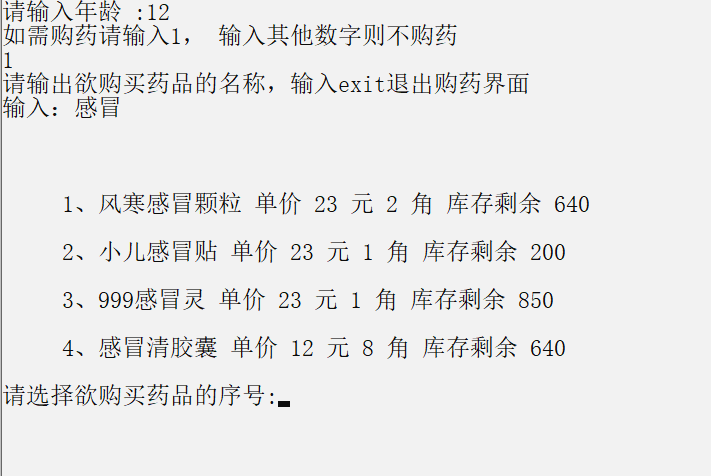


这里检测到挂号冲突，可以输入0取消挂号。但这次我们不取消

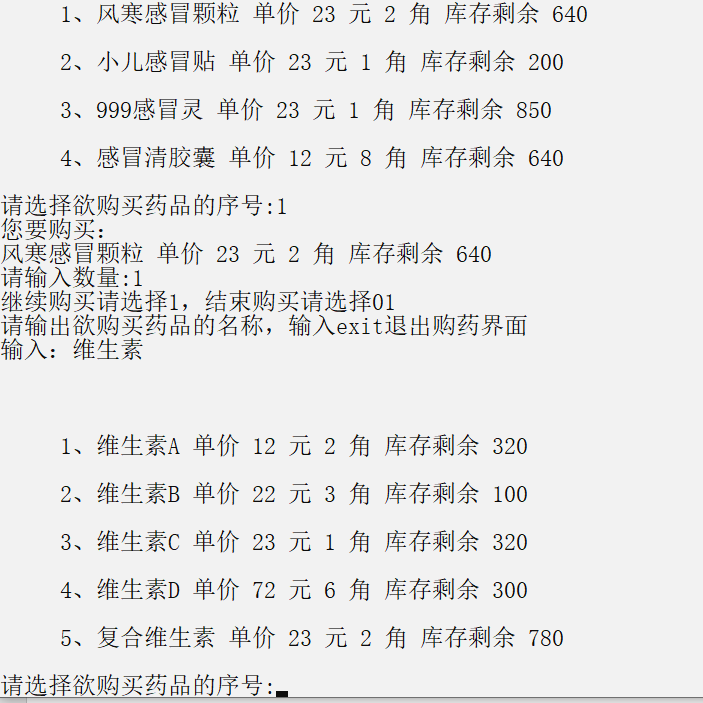


### ①开始购药：

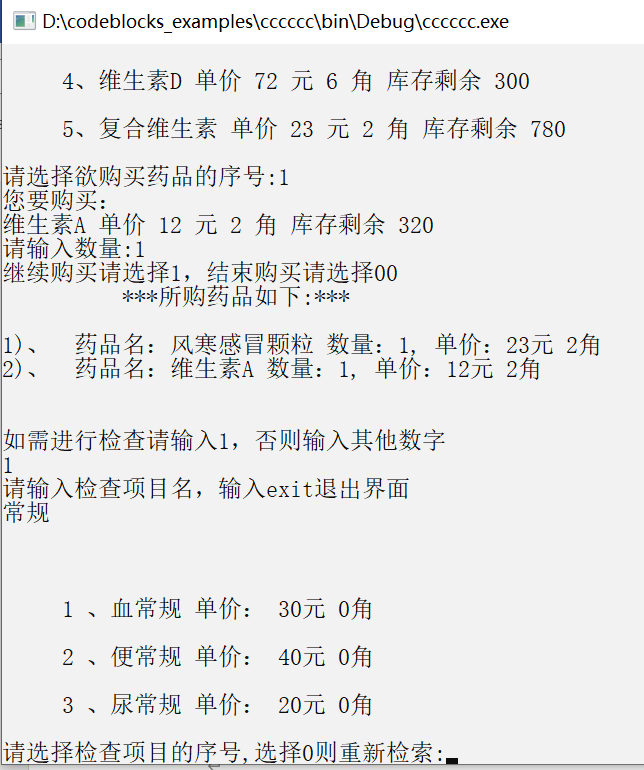
购药采用检索式：比如输入“感冒“，系统会显示所有的与之相关的药品



检索完成后，选择一个药品

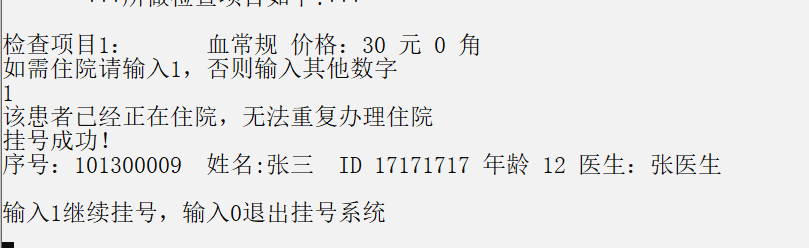


### ②选择检查项目：



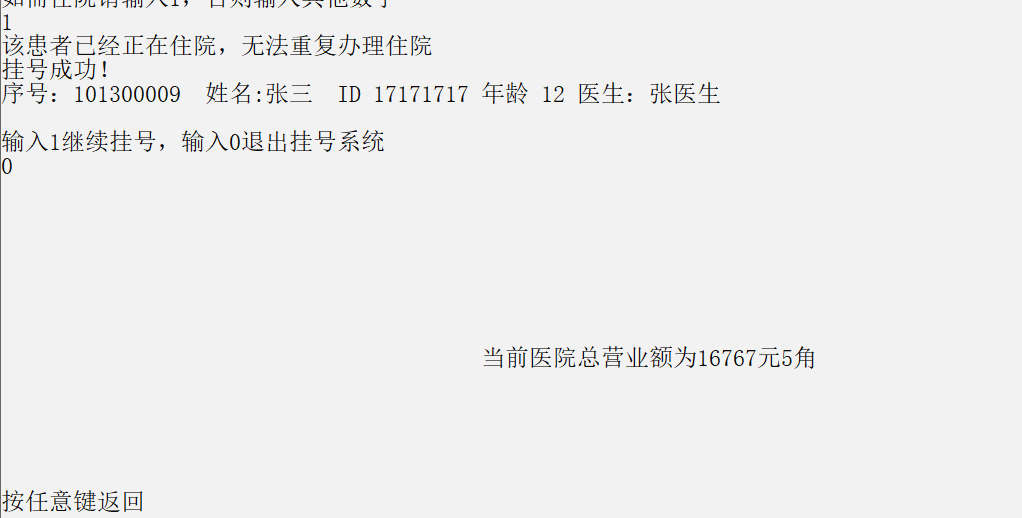
与购药模块的逻辑是相同的

住院模块：



同一个病人ID无法在住院状态下再次办理住院。

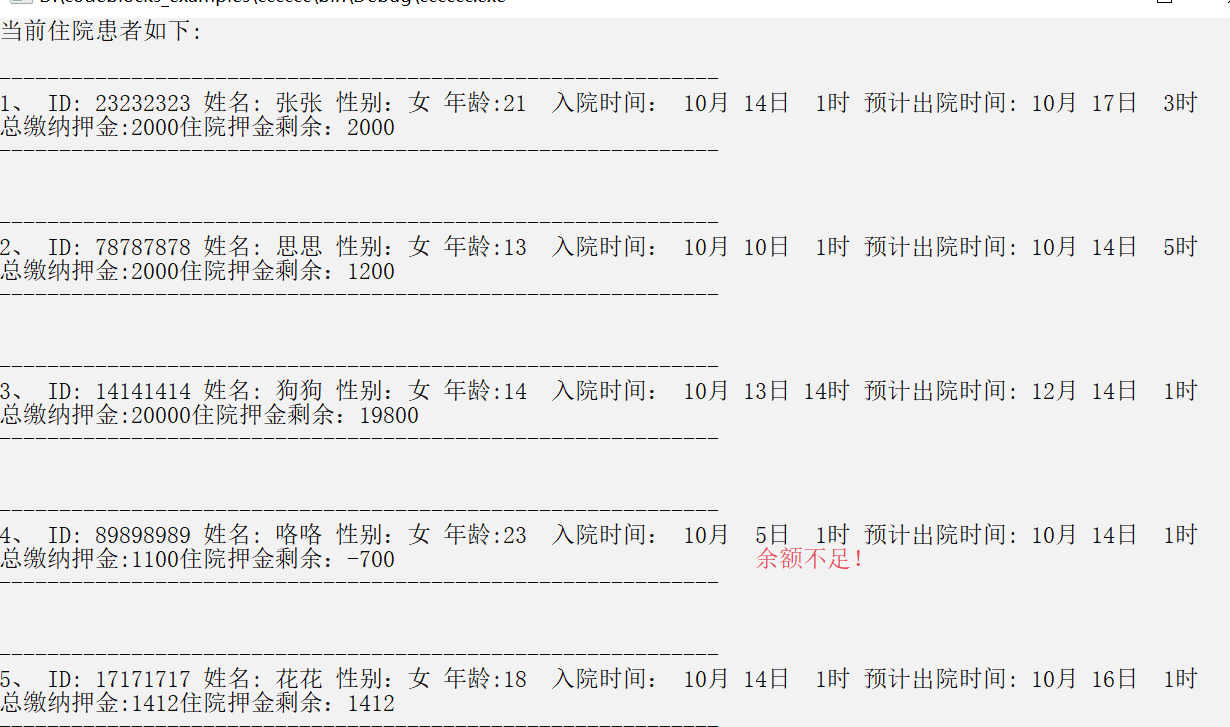
挂号完成后，系统自动计算医院的营业额



## 2、住院部



### ①病房信息：

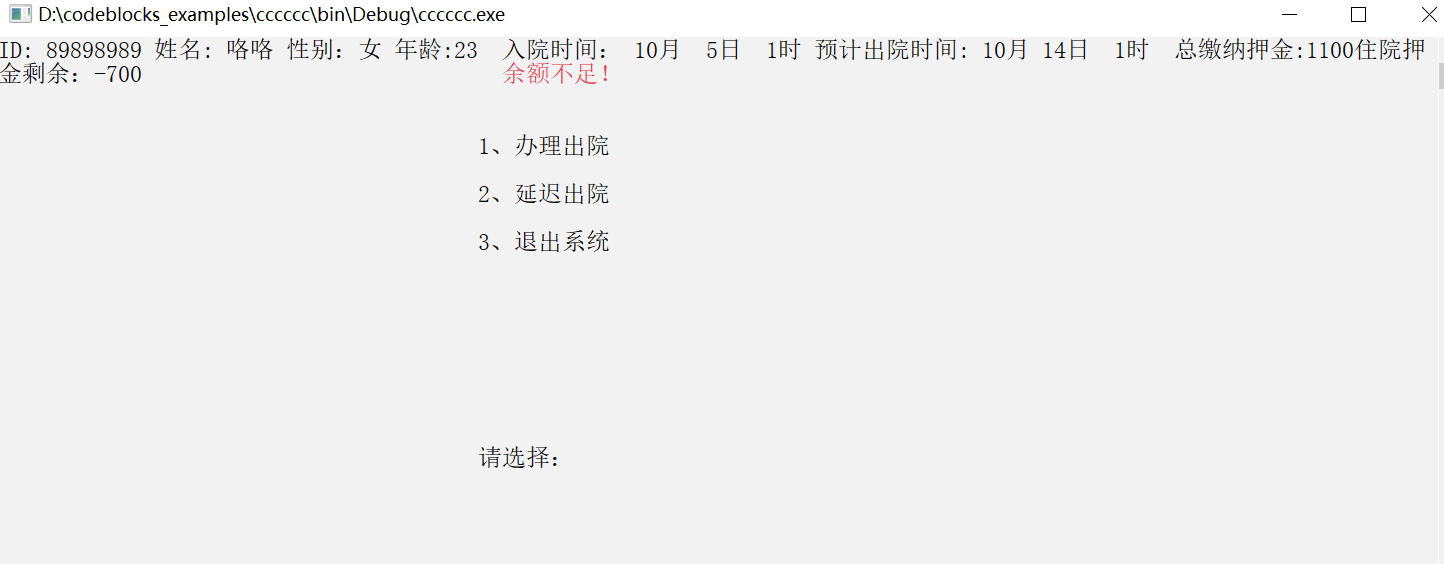


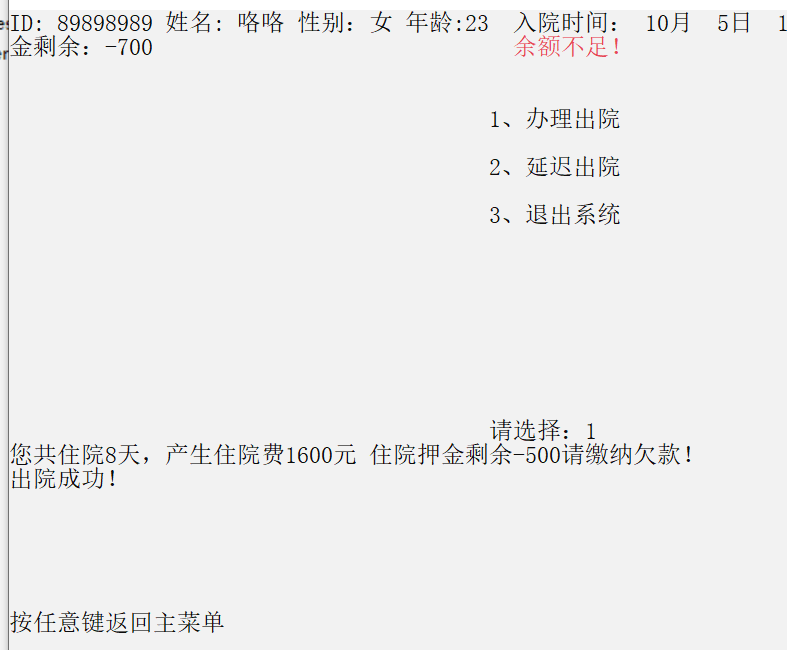
这里也会自动获取系统时间，来计算所有住院患者的押金余额。

当押金余额低于1000元后会显示”余额不足”

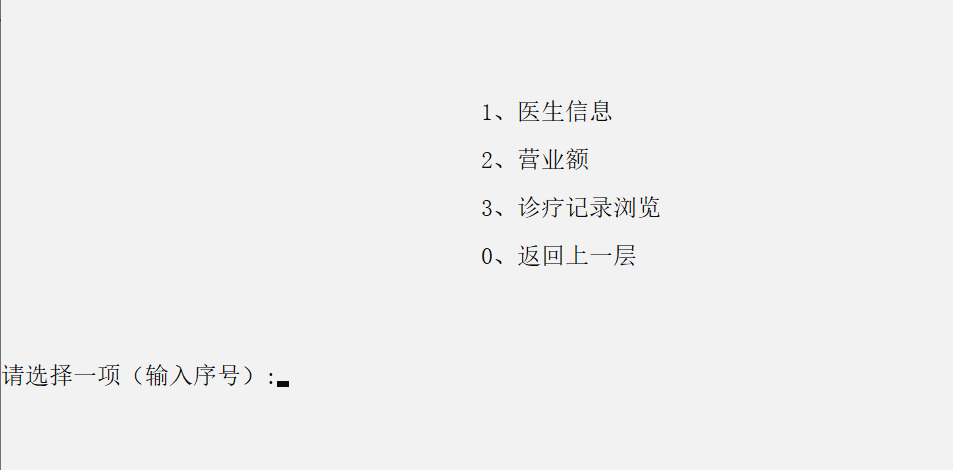
住院病人管理：

### ②办理出院：

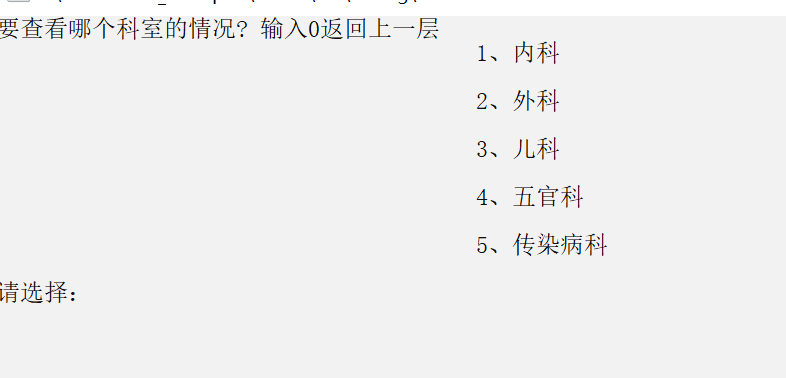


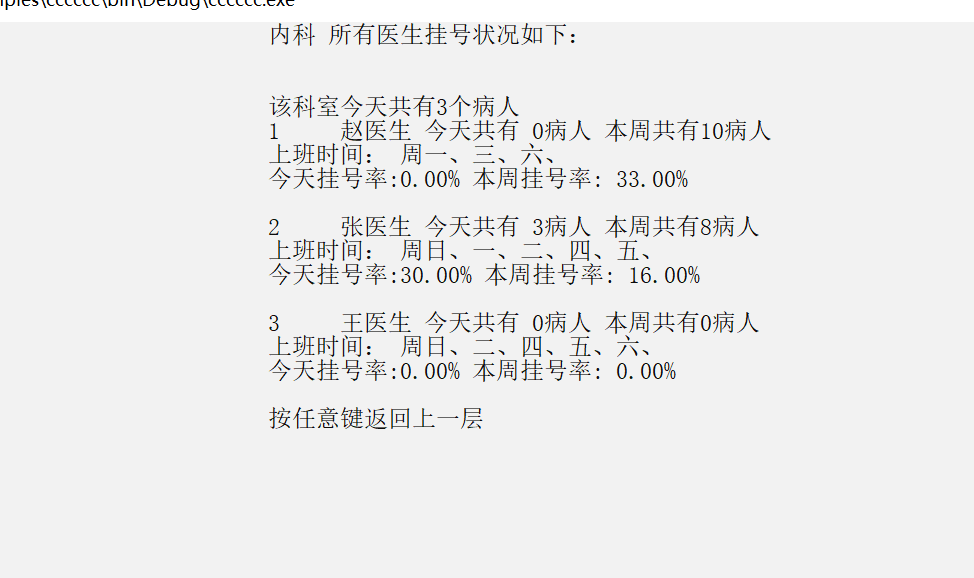


## 3、数据浏览模块：



### ①医生信息：



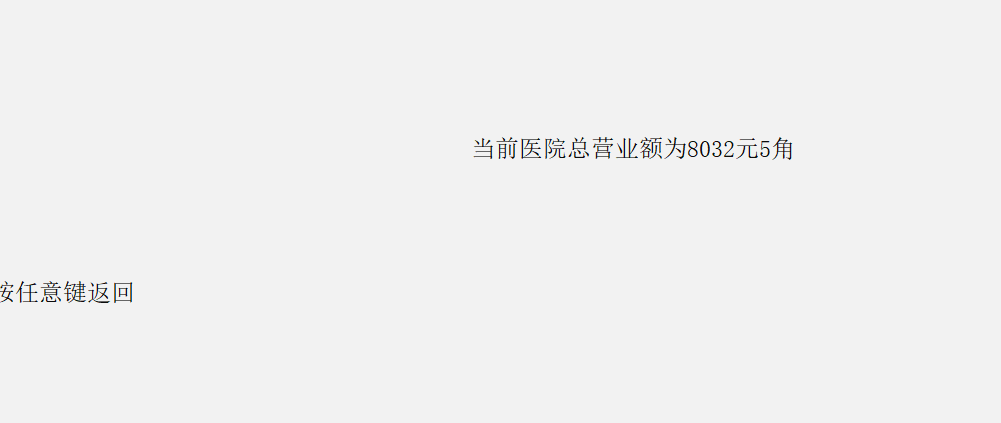


对所有医生的情况进行统计：

“今日挂号率“指的是该医生今天挂号占最高挂号数的比例，比如30%可以推算出该医生共挂了3个号

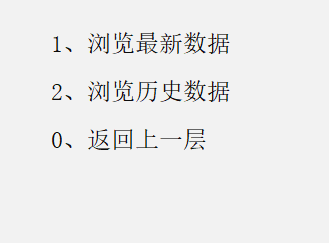
而本周挂号率是该医生7天的平均值

### ②营业额浏览：



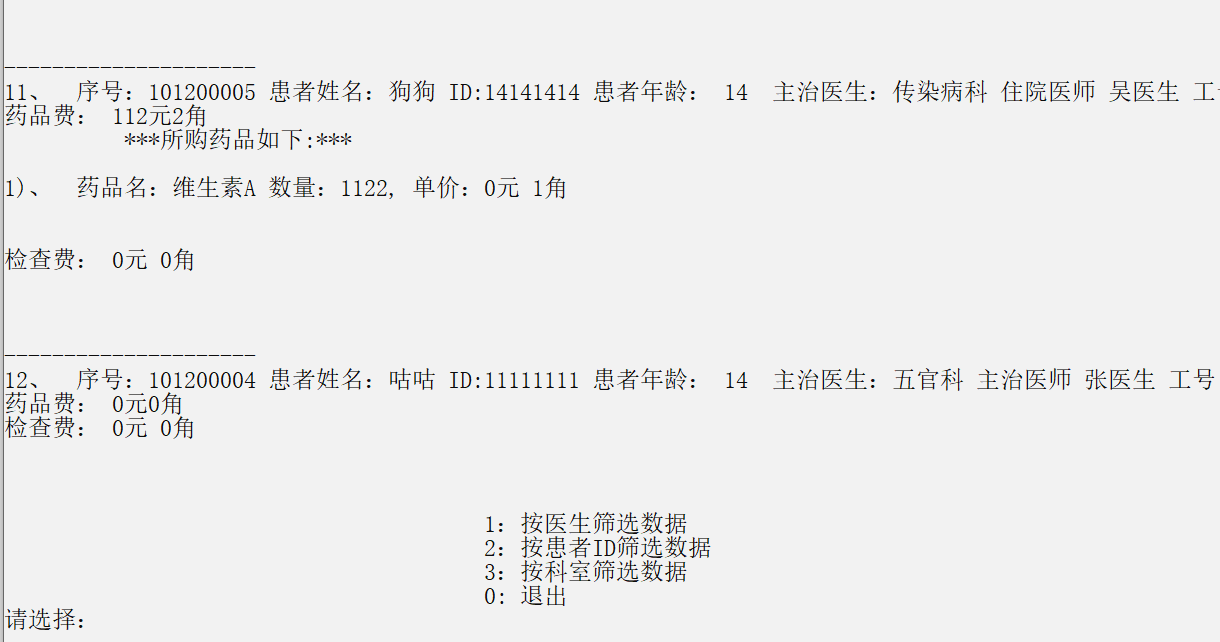
### ③诊疗记录浏览：

采用了两种浏览模式：

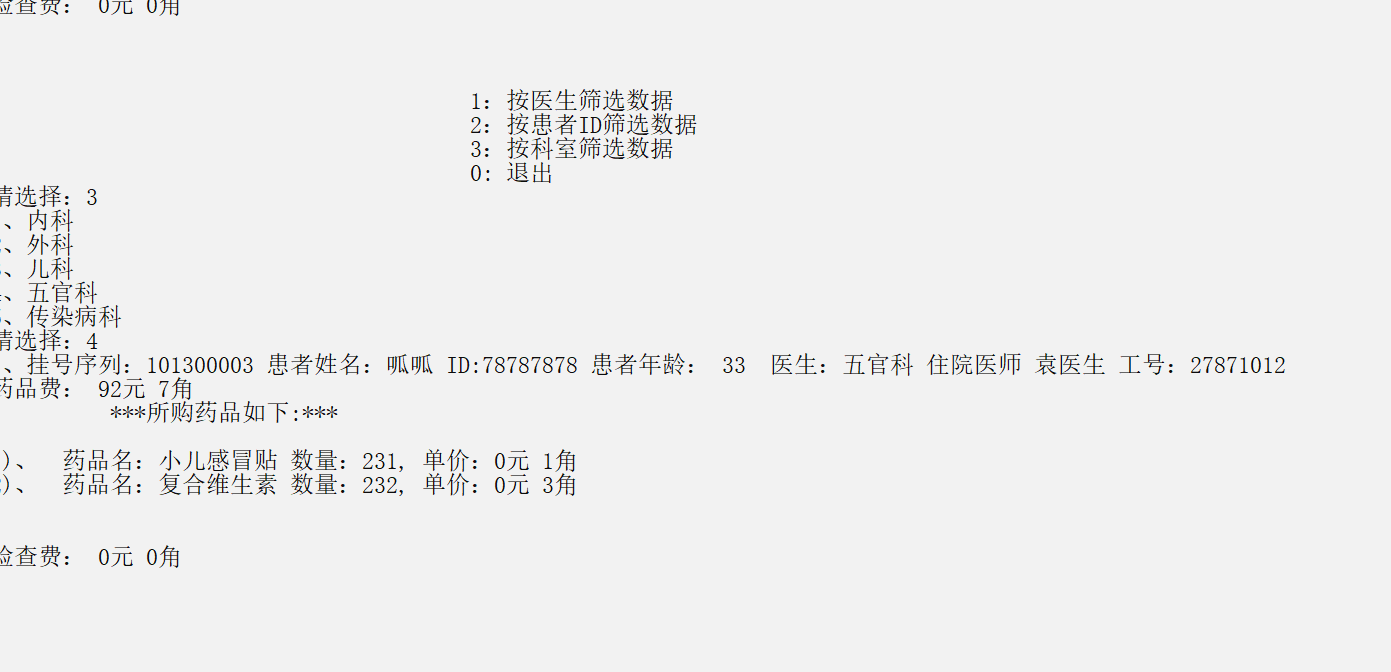


浏览最新数据：

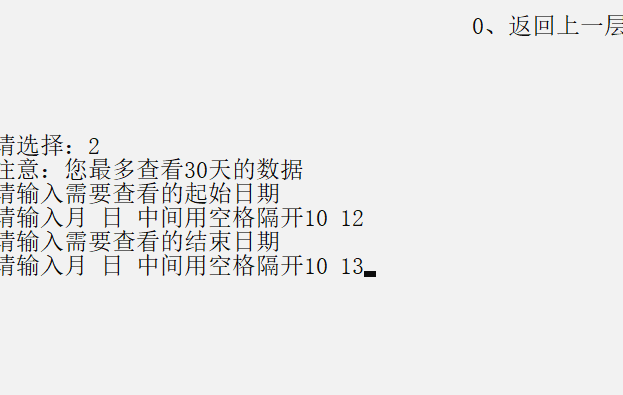
会获取系统时间、逆序遍历文件、逆序建立链表。（这样保证排在前面的数据是更新的）



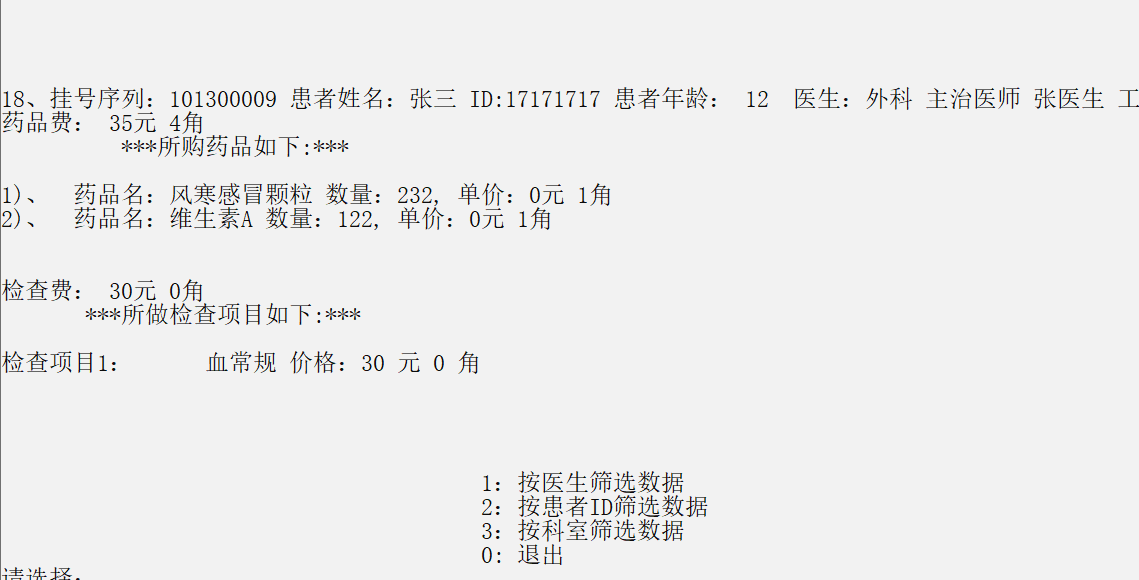
浏览的数据支持使用多种模式进行筛选。比如“五官科“：



历史数据浏览：



这里浏览的是10月12日到10月13日的数据



同样也可以进行筛选，这里不再展示

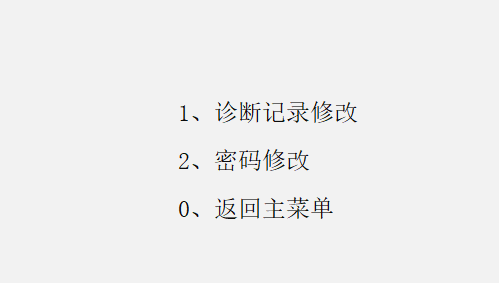
## 4、数据修改模块：

### ①加密系统

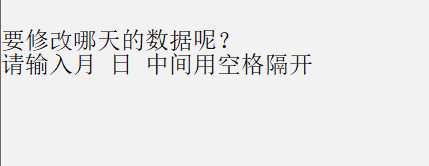


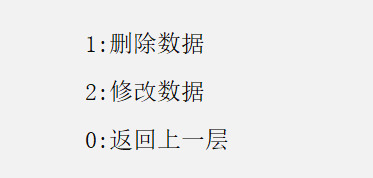
由于数据修改会破坏数据的一致性，因而需要输入密码才能提供权限

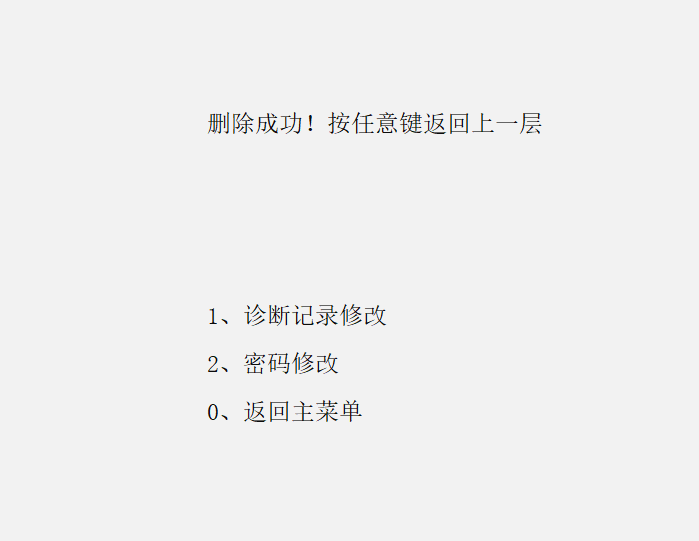
值得一提的是，这里的密码由复杂的位运算加密。实际密码是1234512345，但在文件中只有70个乱码字符



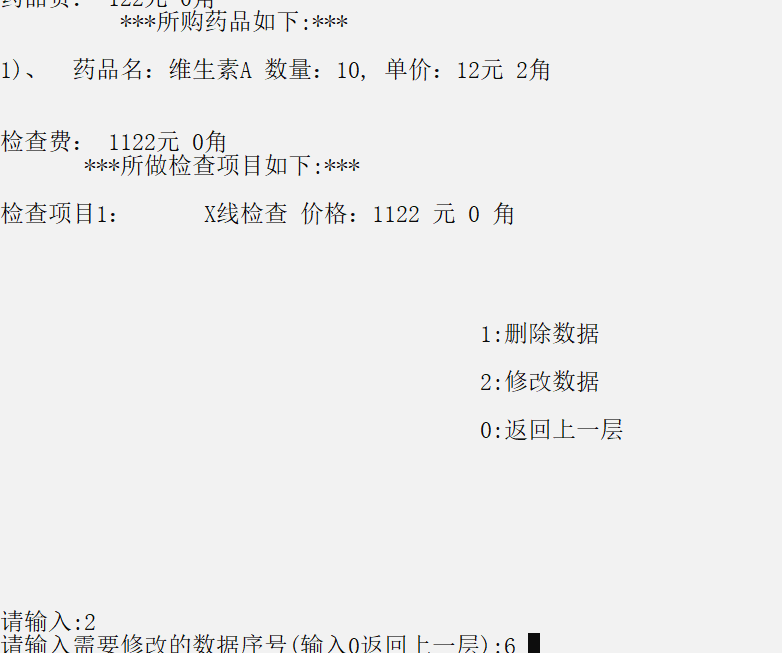
### ②修改诊疗记录：

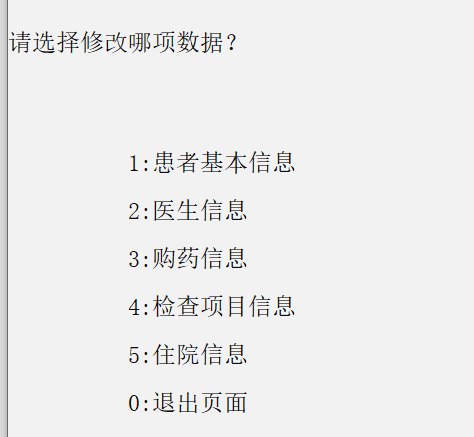






我们删除了第八条，也可以发现文件里已经没有第8条了。





都可以修改，比较简单不再赘述。

# 三、亮点

## 1）字串判断函数：用于药品和检查项目的检索

int compStr(char \*s1, int len1, char \*s2, int len2)//字符串比对

{

int count = 0;

for(int i = 0; i < len1; ++i)

{

for(int j = 0; j < len2; ++j)

{

if(s1[i] == s2[j])

{

++count;

break;

}

}

}

return count==len1;

}

## 2）加密解密函数

自创的位运算加密。

加密函数：

int setPassWord()//加密控件

{

char key[21];

int length;

for(;;)

{

printf("请输入新密码(8到16位):");

length = scanStr(key,17);

if((length > 16)||(length < 8))

{

printf("请检查输入长度\n");

continue;

}

printf("请再次输入：");

char key2[20];

scanStr(key2,18);

if(strcmp(key,key2))

{

printf("两次输入密码不一致，请检查\n");

continue;

}

break;

}

srand(time(0));

FILE \*pf = fopen("data/key.txt","wb");

for(int i = 0; i < length; ++i)

{

for(int j = 0; j < 7; ++j)

{

int num = rand()%128;

int bi = 1 << j;

if(((key[i]) & bi))

{

num |= bi;

}

else

num &= (~bi);

fwrite(&num,1,1,pf);

}

}

fclose(pf);

return 0;

}

解密函数：

char \* deCode()

{

FILE \*pf = fopen("data/key.txt","rb");

char \*password = (char \*)malloc(20);

int pos = 0;

int num;

while(!feof(pf))

{

num = 0;

for(int i = 0; i < 7; ++i)

{

int tmp;

int bi = (1 << i);

fread(&tmp,1,1,pf);

num += (tmp & bi);

}

password[pos++] = num;

}

password[pos-1] = 0;

fclose(pf);

return password;

}

3）加载动效：

void loading() {

system("color f0");

int i, j;

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

for (i = 1; i <= 10; i++)

{

printf("\t\t\t\t\t\t欢迎使用|");

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

printf("-");

printf("|'%d%%'\r", i\*10);

Sleep(50);

}

system("cls");

}