

**课程设计报告**

**课程设计题目： 基于医院分诊系统的排队策略**

**学 生 姓 名： \*\*\***

**学 号： \*\*\***

**班 级： \*\*\***

**指 导 教 师： \*\*\*老师**

**2022年 月 日**

东华理工大学

课程设计评分表

学生姓名： 班级： 学号：

课程设计题目：基于医院分诊系统的排列策略

|  |  |  |  |
| --- | --- | --- | --- |
|  | 项目内容 | 满分 | 实 评 |
|
| 选  题 | 能结合所学课程知识、有一定的能力训练。符合选题要求 | 10 | 10 |
| 工作量适中，难易度合理 | 10 | 10 |
| 能  力  水  平 | 能熟练应用所学知识，有一定查阅文献及运用文献资料能力 | 10 | 9 |
| 理论依据充分，数据准确，公式推导正确 | 10 | 10 |
| 能应用计算机软件进行编程、资料搜集录入、加工、排版、制图等 | 10 | 10 |
| 能体现创造性思维，或有独特见解 | 10 | 9 |
| 成  果  质  量 | 总体设计正确、合理，各项技术指标符合要求。 | 10 | 10 |
| 说明书综述简练完整，概念清楚、立论正确、技术用语准确、结论严谨合理；分析处理科学、条理分明、语言流畅、结构严谨、版面清晰 | 10 | 10 |
| 设计说明书栏目齐全、合理，符号统一、编号齐全。　格式、绘图、表格、插图等规范准确，符合国家标准 | 10 | 10 |
| 有一定篇幅，字符数不少于5000 | 10 | 9 |
|  | 总 分 | 100 | 97 |
| 指导教师评语：  该生的课程设计报告文字表达准确，结构严谨，推理严密、逻辑强；书写格式、图表等规范准确。建议增加适当的参考文献。  指导教师签名：  2022年 月 日 | | | |

目录

[一、需求分析说明 4](#_Toc6773)

[二、总体设计 5](#_Toc17638)

[三、 详细设计 7](#_Toc32435)

[四、实现部分 17](#_Toc9051)

[五、程序运行结果 32](#_Toc1770)

[六、总结 41](#_Toc20635)

# 一、需求分析说明

1、问题的提出: 近年来，随着医疗事业的发展，医院的规模不断扩大，科室门类划分的也越来越细，随之而来的是病人排队挂号和看病变的繁琐，等待时间变长，为此，需要制定一种良好的排队策略，减少病人的等待时间，提高医院的服务质量。

2、问题的假设:

（1）医院门诊分为普通号和专家号，普通号一个科室是有数为医生，专家号一个科室仅有一个医生，并且将普通号和专家号相互独立，以方便模拟。

（2）我们将病人分为A类和B类，对于A类病人，我们仅进行一次诊断，即用户挂号后进入队列，在进行一次诊断后即完成诊治。B类病人需要进行两次诊断和一次检查，在完成一次诊断后，医生开出检查意见，病人进行检查，完成后病人需返回科室进行第二次诊断，在第二次完成后即完成诊治。

（3）虽然实际中在一定时间内病人的产生应该按照泊松分布，由于参数设定的困难，我们只给出病人最基本的信息（如姓名、病种等），同理，病人的诊治时间使确定的时间。

（4）实际中，每位医生每天接诊的人数是有一定限制的，尤其是专家门诊。但考虑病人来源的数据无论自实际数据库，还是来自根据实际情况的随机模拟数据， 各类病人的数量都对应了实际的各科室接诊病人的限制。

（5）对于由于到达时间过晚，部分病人将无法完成诊断，我们将忽略这部分病人。

3、两种策略的制定：

（1）一级分诊队列模式: 每个诊室队列只对应一个候诊区队列，我们将之记作Q，一般排队规则是：①刚完成挂号的病人，添加在Q的末尾；②按医嘱完成体检的病人， 直接插入在Q的首部。这种排队规则广泛应用于中小医院。

（2）二级分诊队列模式: 每个诊室队列对应一个候诊区队列和一个候诊厅候诊队列， 我们将前者记作Q1，后者记作Q2。一般情况下，因为候诊区空间限制，Q1的人数限制在2、3 人左右；而候诊厅空间较大，所以对Q2的人数不作限制。一般排队规则是：①刚完成挂号的病人，添加在Q2的末尾；②按医嘱完成体检的病人，直接插入在Q2的首部；③医师始终对Q1中的首位病人诊治，每位病人诊治结束时，若Q2不空，则Q2的首位病人脱离Q2，进入Q1的尾部。这种排队规则广泛应用于大中型医院。

# 二、总体设计

把程序的实现大体分为两个部分：一级分诊模式和二级分诊模式。

1、一级分诊模式：

此模式又可以分为三个子模块：

（1）挂号模式

（2）就诊模式

（3）体检模式

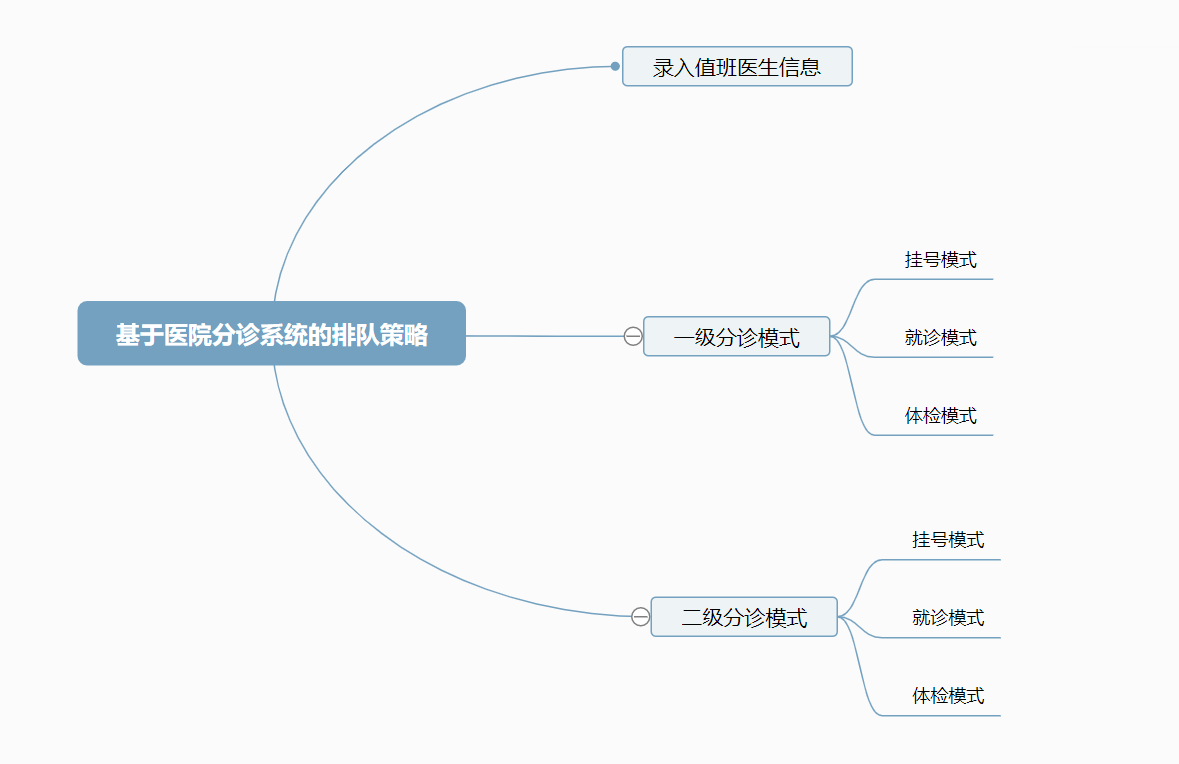
通过对题干的分析，在此模式下，患者选择挂专家/普通号，判断是A/B类病，直接进入对对应医生就诊室排队。B类病患较为特殊，拥有“插队”的能力，即在第一次就诊并完成体检后，可直接返回对应医生就诊室队列队首。

2、二级分诊模式：

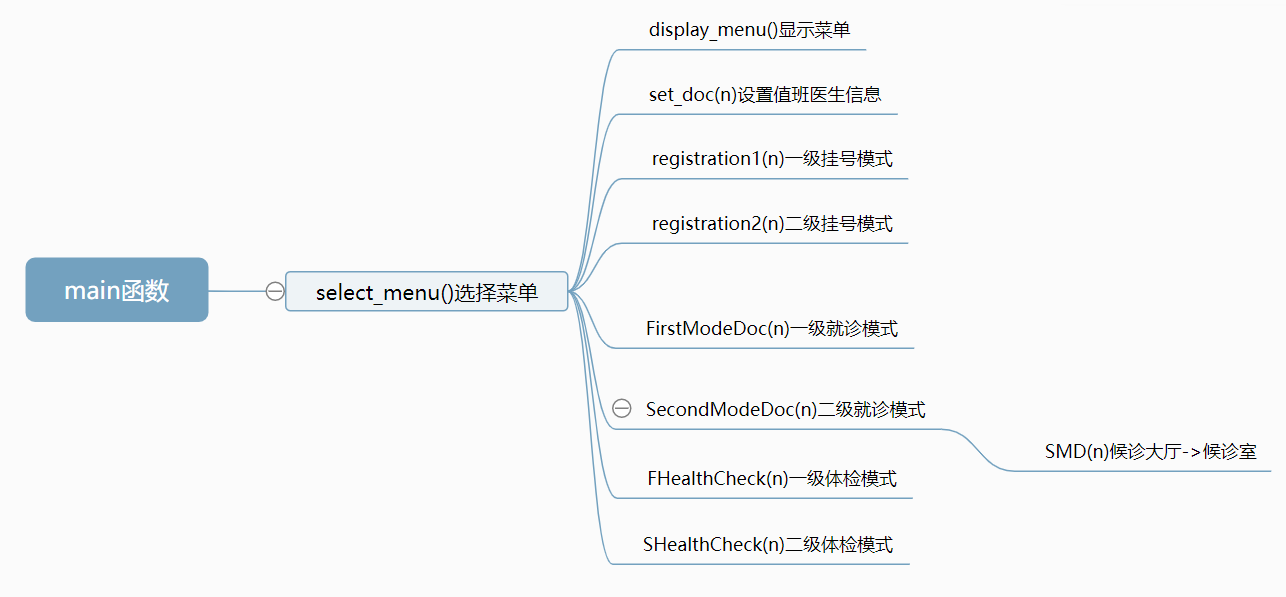
此模式相当于“一级分诊模式”（下述简称为1）的升级版，在1的基础上添加新的功能。此模式也有和1中相同的三个子模块，但每个模块有差别。在此模式中，病患并不会直接进入医生的就诊室，而是在就诊大厅（分为专家、普通候诊大厅）进行等候，等待医生叫名字进入诊室（医生诊室最多容纳3名病患）。将专家、普通大厅分开有便于管理。B类病患不直接返回就诊室，而是插在候诊大厅队首。

3、图表说明

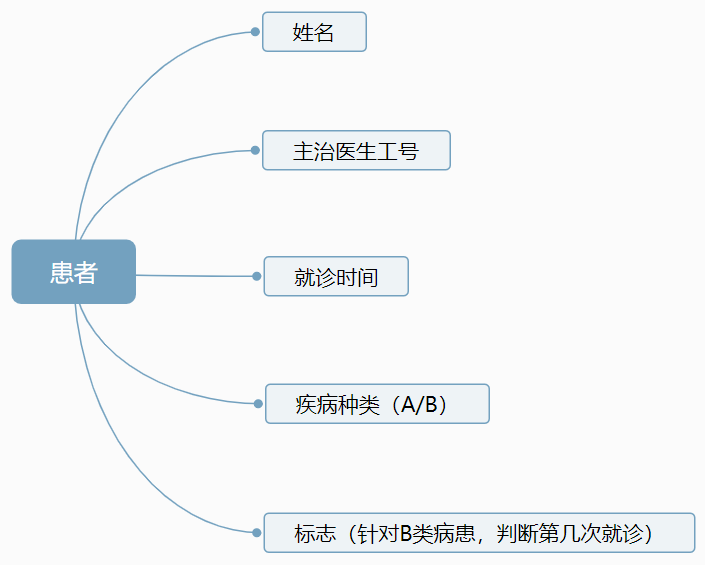
（1）大致框架

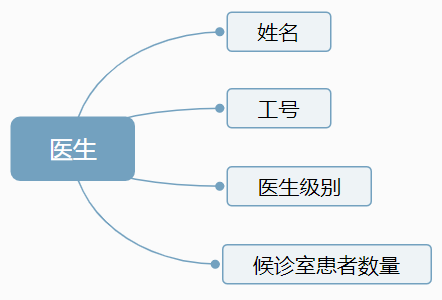


（2）程序框架



（3）医、患信息





# 详细设计

对数据结构进行详细的描述，设计好相应数据结构以及其操作功能，用文字详细描述每个功能实现的算法及思路。

1、数据结构

（1）采用三个结构体，分别是病人、医生、候诊队列，数据类型如下：

typedef struct Pat {

char name[10]; //患者姓名

int doc\_num; //医生工号

char disease; //疾病种类

int time[2]; //诊治时间

int flag; //判断第几次诊断(主要针对B类患者)

}patient;

typedef struct queue1 { //候诊队列

patient data;

struct queue1 \*next;

}\*Q1,Q1node;

typedef struct Doc {

int num; //医生工号

char name[10]; //医生姓名

int pn; //候诊室患者数量

char rank; //医生级别

Q1 link;

}doctor;

患者结构体中存有患者的各种信息，其中要注意的是time[2]和flag，他们的设置都主要针对B类患者，time[0]记录第一次就诊时长，time[1]记录第二次就诊时长，flag用于判断是第几次就诊（为1说明就诊过一次）；候诊队列结构体中有患者结构体类型的数据，用于存放患者信息，以及一个指向自己类型的指针（队列采用链式存储）。每个医生之间采用数组的方式连接（顺序存储），并且医生的结构体中除了有医生的基本信息外，设置了一个指向队列类型的指针（数据的储存采用类领接表的方式）。默认专家医生占据数组的第一个位置，普通医生的数量取决于程序使用者的设置。具体如下图所示：

专家医生 普通医生1 普通医生2

患者一

患者二

患者一

患者二

患者一

患者二

（2）设置如下全局变量：

1. 每个医生的候诊室，都有一个尾指针指向队位，此处设置一个队列类型的尾指针数组，他们的下标与医生数组的下标相对应，由此可以到达目的

doctor \*doc=NULL; //值班医生数组

Q1 \*rear=NULL; //患者尾指针数组

1. 定义体检队列及其尾指针

Q1 Q=new Q1node; //B类可插队患者队列

Q1 r=Q; //B类可插队患者队列尾指针

1. 定义普通候诊大厅队列及其尾指针

Q1 room = new Q1node; //普通候诊大厅队列

Q1 r\_room=room; //普通候诊大厅队列尾指针

1. 定义专家候诊大厅队列及其尾指针

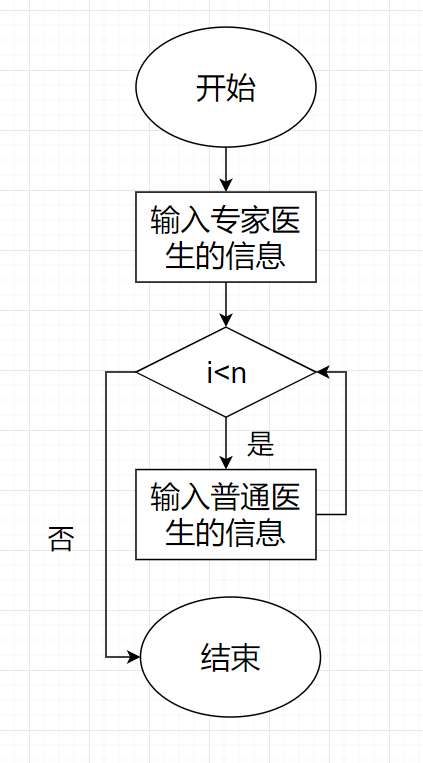
Q1 viproom=new Q1node; //专家候诊大厅队列

Q1 r\_vip=viproom; //专家候诊大厅队列尾指针

特别说明：上述所有队列都没有设置“头指针”。在医生候诊室队列中，每一名医生相当于是队伍的头，患者直接插在其后即可；其他队列均设置一个头结点，将其永远作为队首，所以也不需要设置头指针。

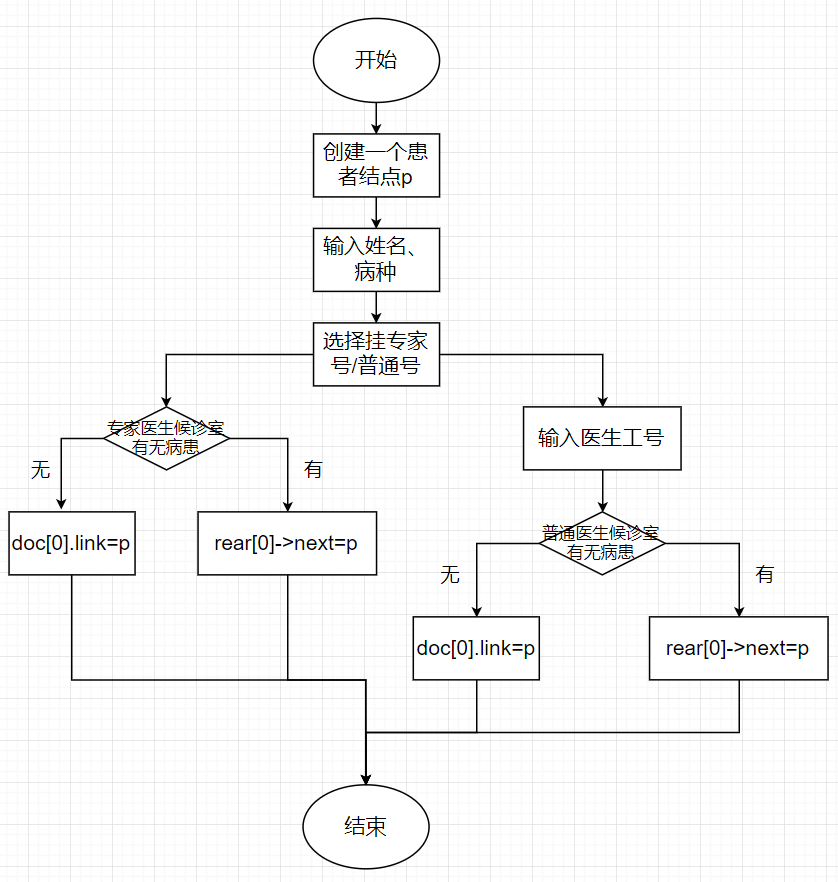
2、录入值班医生信息 void set\_doc(int n)

在select\_menu()函数中输入总共的医生数量（专家医生固定1名，普通医生n-1名）n，将n设置为静态int型数据。



在录入医生的信息过程中，部分信息自动设置：默认将doc[0]作为专家医生，将其等级设置为S，候诊室内患者数为0，next指针指向NULL；doc数组其他医生全为普通医生，将他们的等级设置为A，候诊室内患者数为0，next指针指向NULL。

3、一级分诊挂号模式 void registration1(int n)



普通号

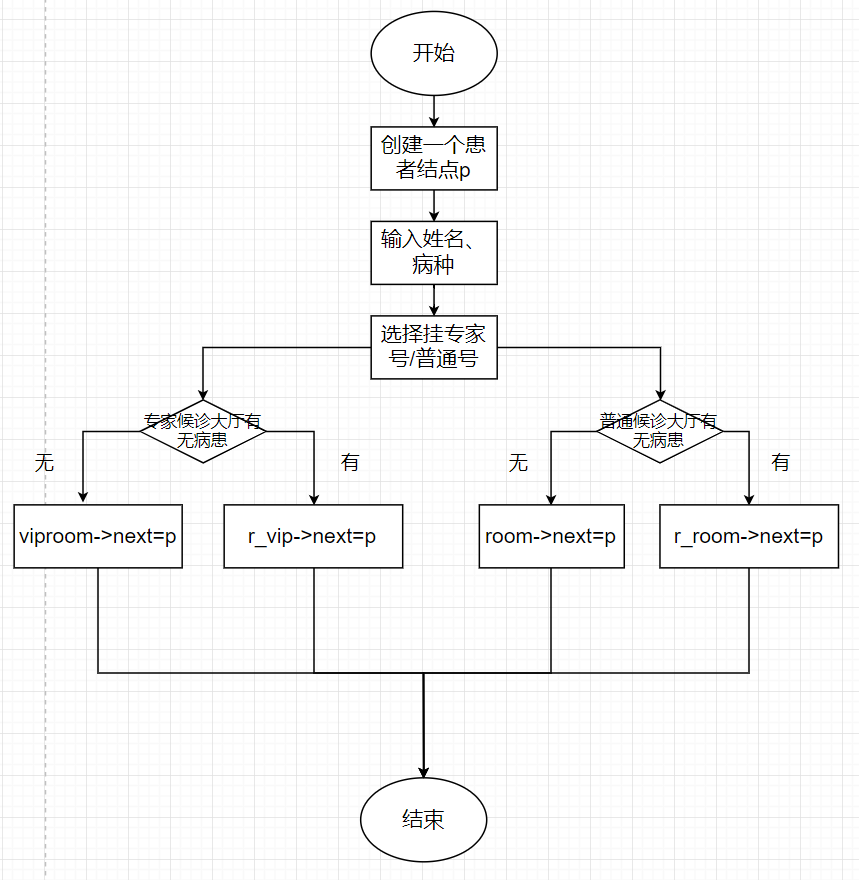
专家号

在一级挂号中，创建患者结点p，部分信息自动设置：p->next=NULL；若选择专家号，p->data.doc\_num=doc[0].num, doc[0].pn++;若选择普通号，先通过医生工号利用for循环检索出普通医生的下标i，p->data.doc\_num=doc[i].num, doc[i].pn++；将p->data.time[0]=40，若为B类病，p->data.time[1]=10（40min和10min都是预测的固定就诊时间）且p->data.flag=0。最后，给患者提示前方还有的患者数量，并让对应医生候诊室病患数+1。

对患者输入的信息，也要判断是否规范：

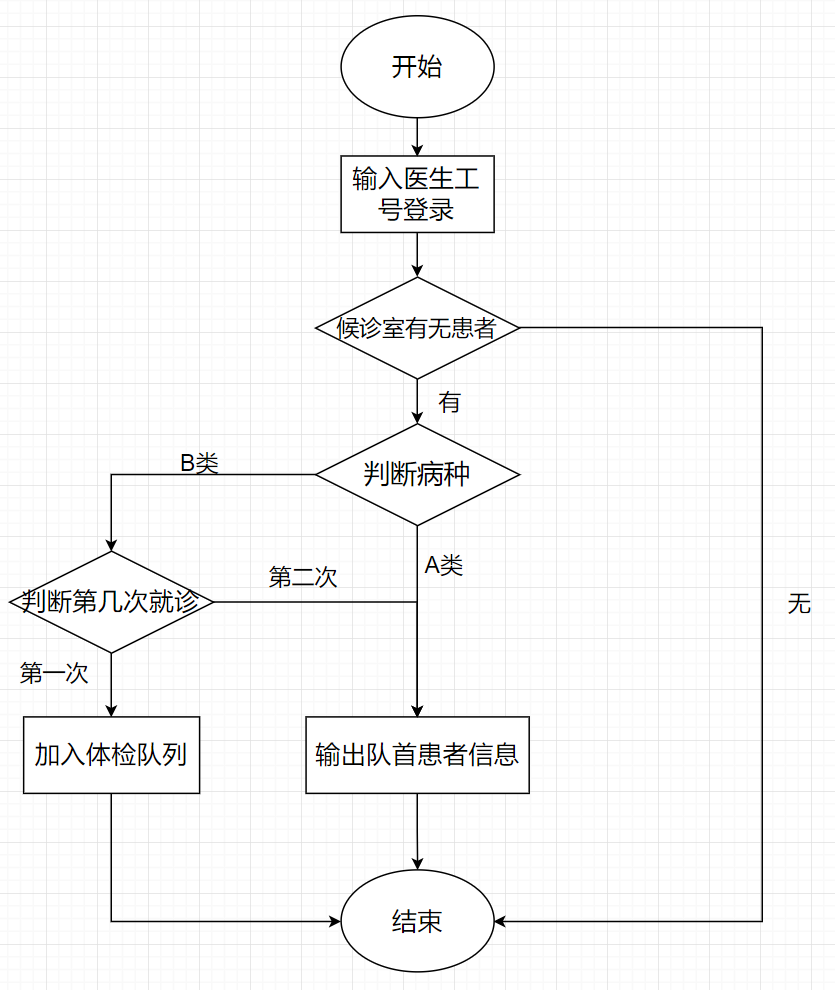
1. 病种只有A、B两种
2. 门诊类型只分专家号与普通号
3. 判断患者输入的医生工号是否存在

4、二级分诊挂号模式 void registration2(int n)



二级挂号与一级挂号基本一致。但在二级挂号下，患者不再直接进入医生的候诊室，而是在候诊大厅内进行排队，并且将专家候诊大厅和普通候诊大厅分开，有利于管理。此外，与一级挂号不同的是，选择普通号的病患不能指定医生了，而是进入患者数最少的医生的候诊室（在二级分诊医生模式中会详细介绍）。

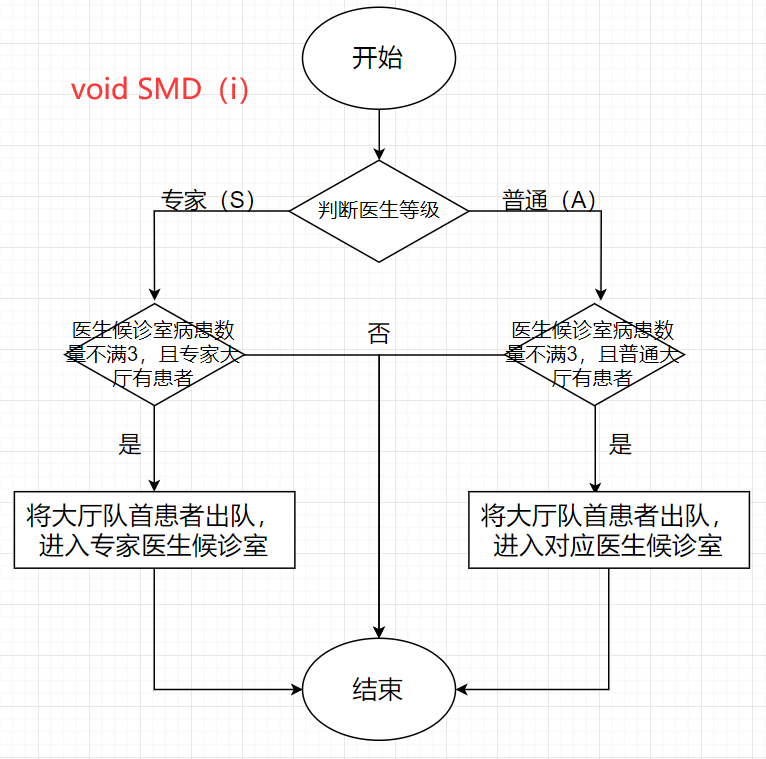
5、一级分诊医生模式 void FirstModeDoc(int n)



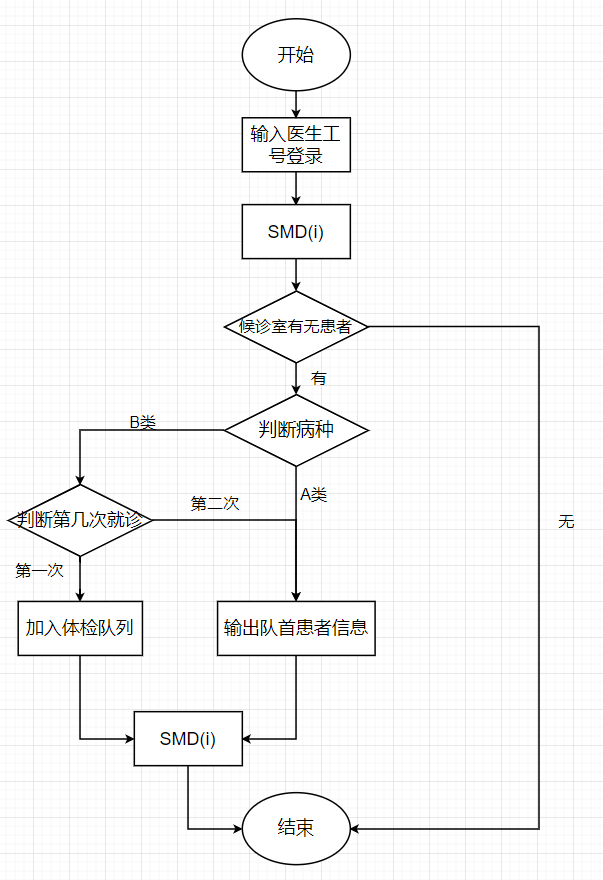
一级医生模式中，医生通过自己的工号登录系统，若查询到候诊室中无病患，则给出提示并返回主菜单。若有患者，则显示排在队首的患者信息。患者的分类通过flag进行判断：若患者的flag等于0则判断为B类患者且第一次就诊，就诊时长为time[0]，提示患者前去体检室体检，将该患者flag改为1，令他出队并加入体检队列；若患者的flag等于1则判断为B类患者且第二次就诊，就诊时长为time[1]，直接出候诊室队列；若患者的flag没有值，则判断为A类患者，在输出患者信息后，直接出候诊室队列。最后，当有患者出队时，让当前医生的候诊室患者数量-1。

6、二级分诊医生模式 void SecondModeDoc(int n)

该模式下包涵一个子函数void SMD(int i)

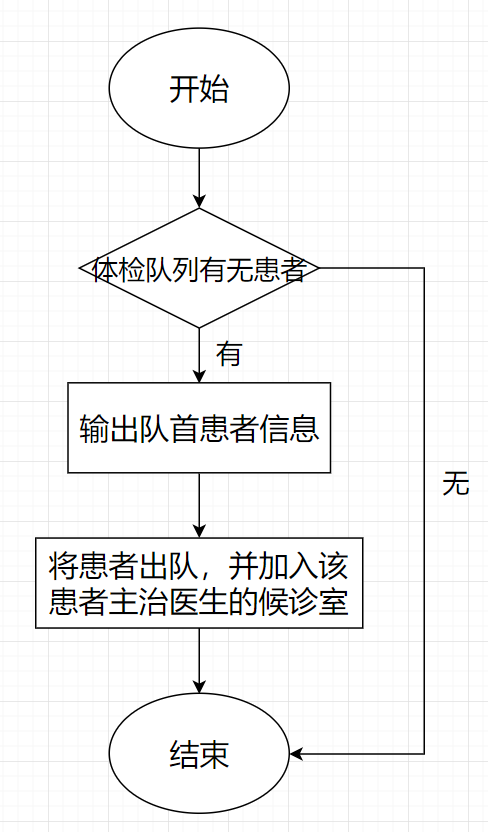


该子函数主要负责将患者从候诊大厅出队并加入对应医生的候诊室。此函数有一个形参i，是对应医生的下标。假设每名医生候诊室最多容纳三名患者，每名医生每次从候诊大厅队首叫一名患者进入候诊室，叫满为止，若候诊大厅内无患者则不叫患者。值得注意的是，若候诊大厅内患者被全部叫空，要让对应候诊大厅的尾指针重新指向队首头结点，不然下一次患者挂号时会出现错误。



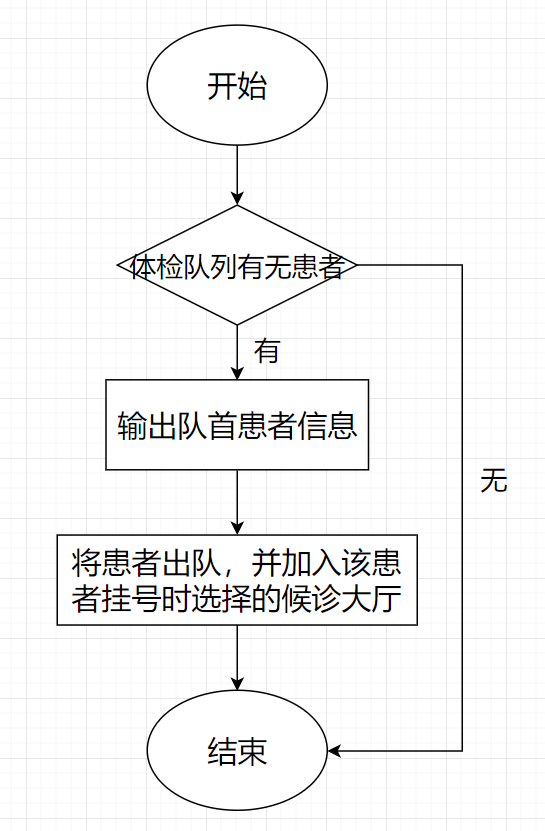
二级医生模式相当于是建立在一级的基础上，多了一步队列之间的转换。每次医生叫患者时，都能保证当前候诊室患者人数达到最大，这样相当于患者每次都会进入人数最少的医生的候诊室当中，这样可以提高医院的就诊效率。

7、1B类患者体检模式 void FHealthCheck(int n)



一级体检模式中，首先显示患者的信息，体检结束后给出提示，让患者返回他的主治医生的候诊室。患者在挂号时，在患者的信息中记录了他的主治医师的工号，通过一个for循环可以在医生数组中找到主治医师的下标，通过这个直接插在该主治医师候诊室队列的队首。值得注意的是，若患者出队后体检队列为空，要让体检队列的尾指针重新指向队首头结点，不然下一次患者加入体检队列时会出问题。

8、2B类患者体检模式 void SHealthCheck(int n)



此模式与一级体检模式大致相同。不同之处在于患者体检完之后不直接插入医生候诊室队列队首，而是插入患者挂号时选择的候诊大厅队列。可用用for循环找到患者的主治医师下标，判断该医生属于专家/普通医生。值得注意的是，若患者出队后体检队列为空，要让体检队列的尾指针重新指向队首头结点，不然下一次患者加入体检队列时会出问题。

四、实现部分

//源代码

#include <iostream>

#include <windows.h>

using namespace std;

typedef struct Pat {

char name[10]; //患者姓名

int doc\_num; //医生工号

char disease; //疾病种类

int time[2]; //诊治时间

int flag; //判断第几次诊断(主要针对B类患者)

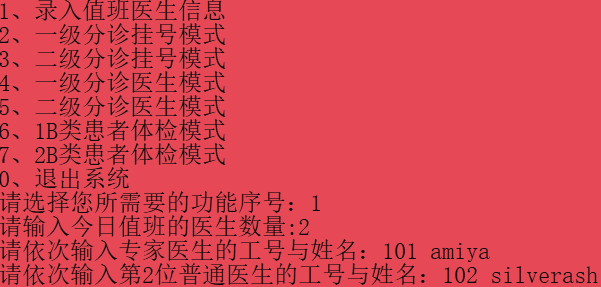
}patient;

.....

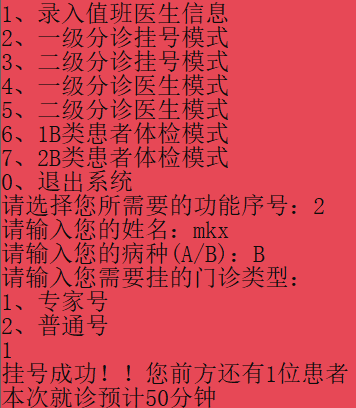
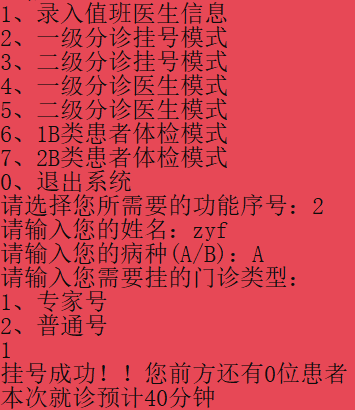
（完整代码填入）

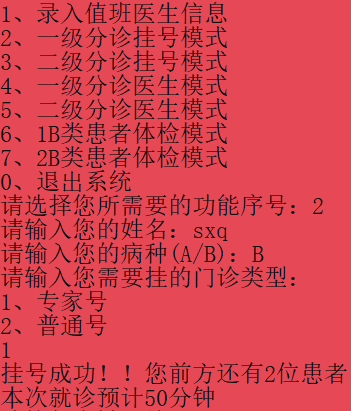
# 五、程序运行结果

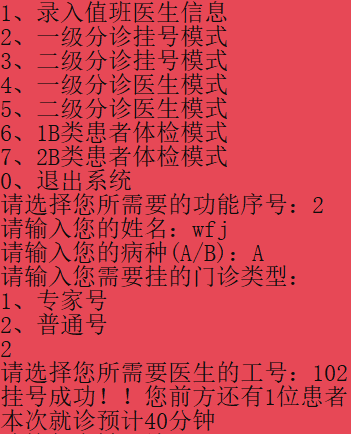
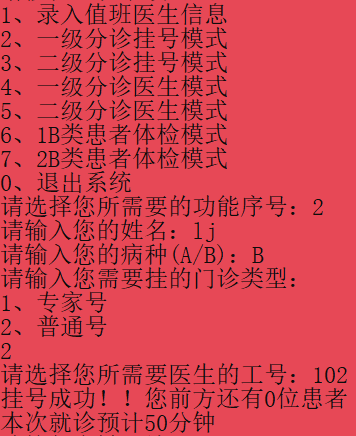
1、录入医生信息



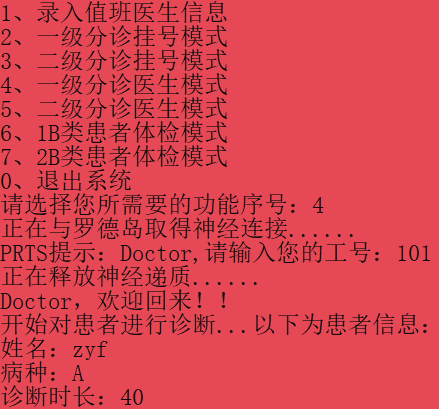
2、一级分诊模式



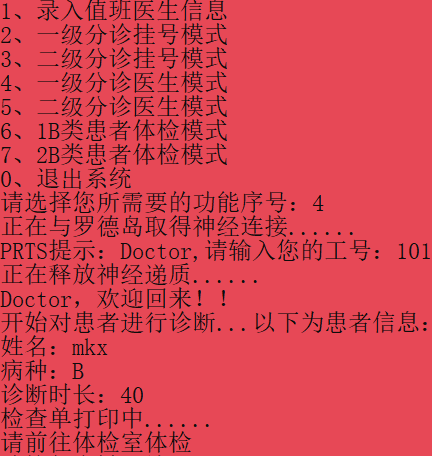
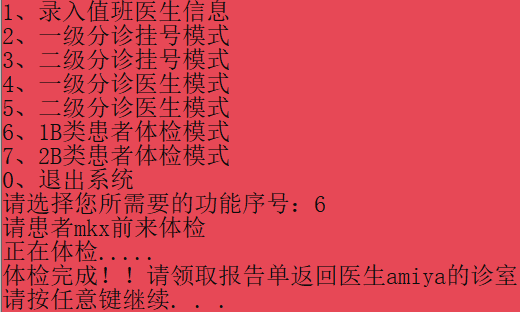


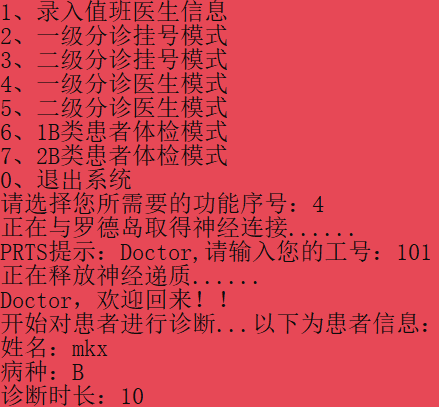


依次让多名患者挂号，3名加入专家医生候诊室，两名加入普通医生候诊室

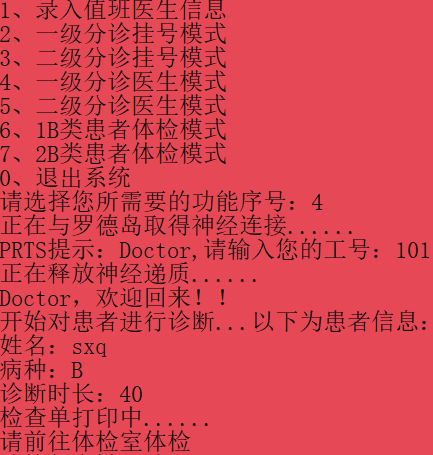
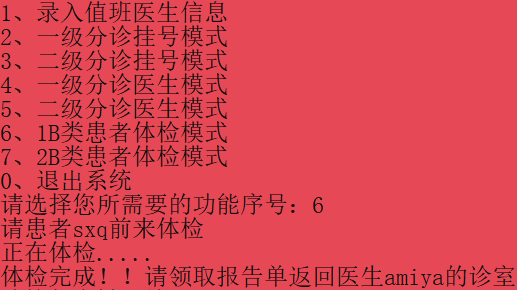


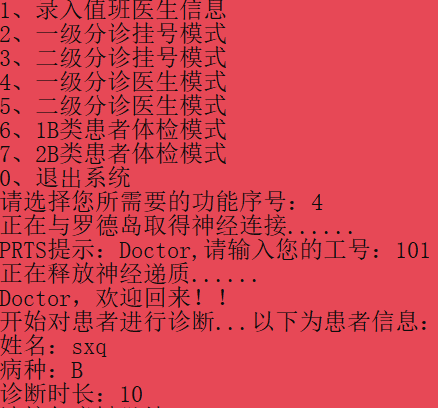
专家医生给队首患者诊断

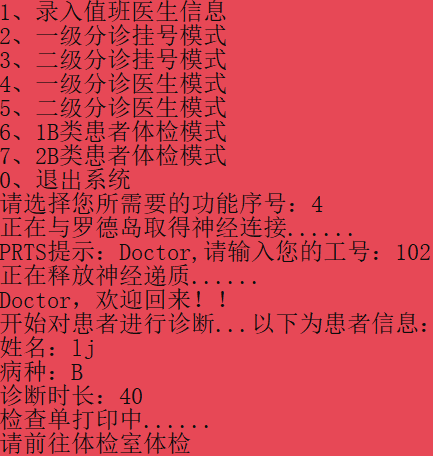
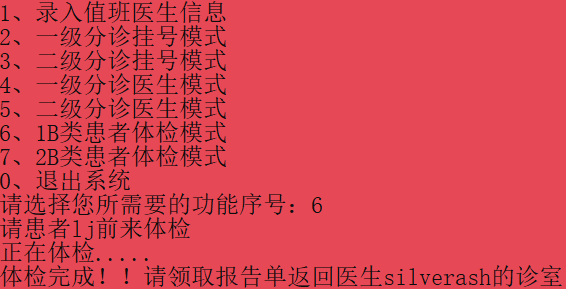


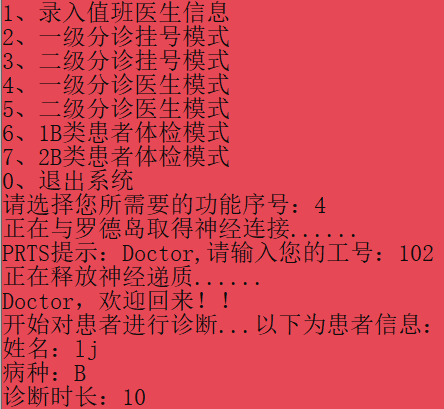


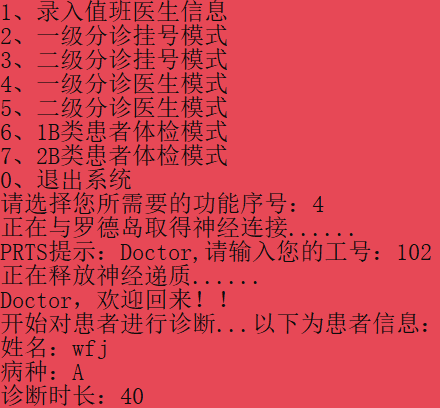
B类患者第一次诊断+体检+第二次诊断





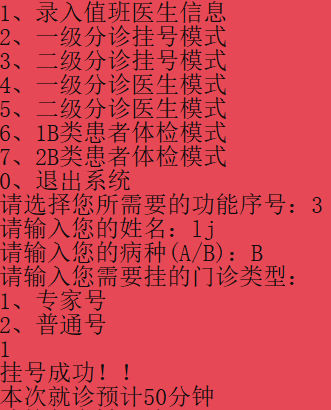
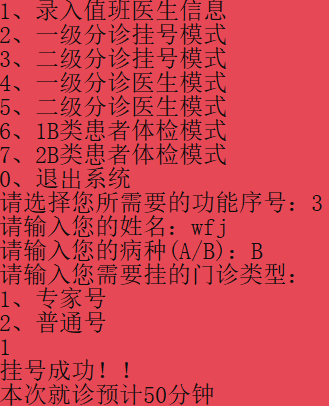


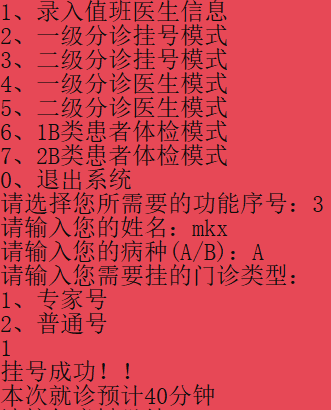
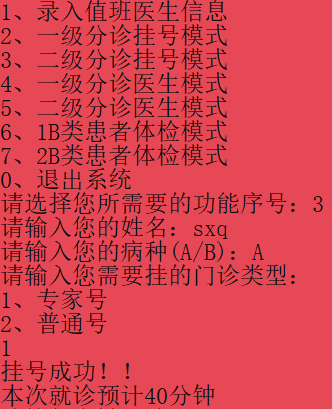




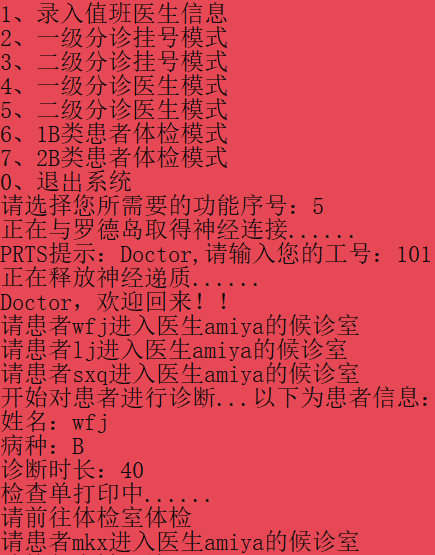
医生依次给患者就诊

3、二级分诊模式

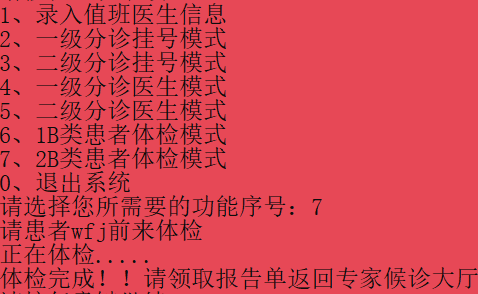




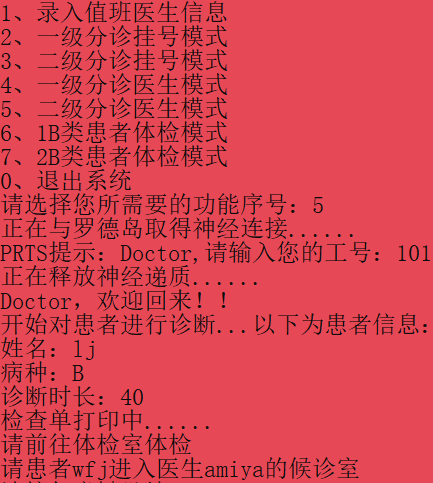
四名患者挂号，在专家候诊大厅等候



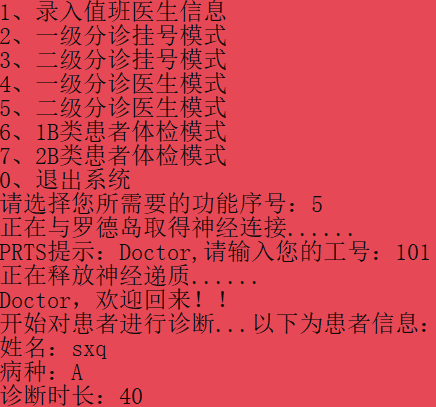
专家医生呼叫患者

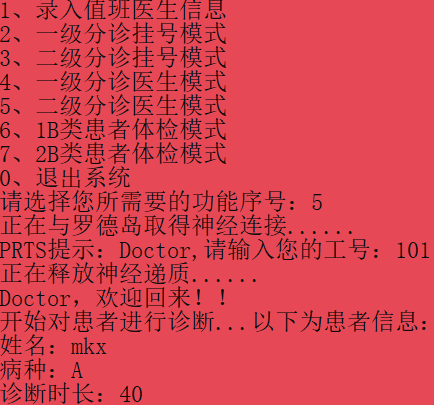


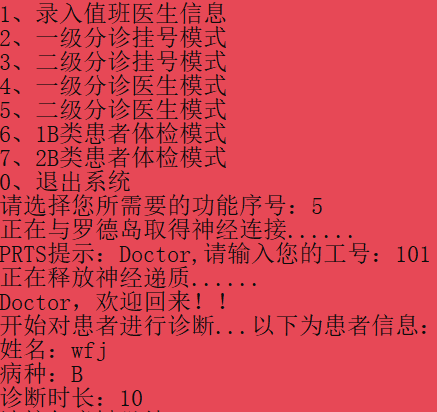
B类患者体检结束返回候诊大厅

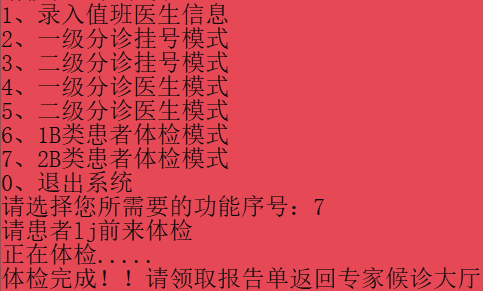


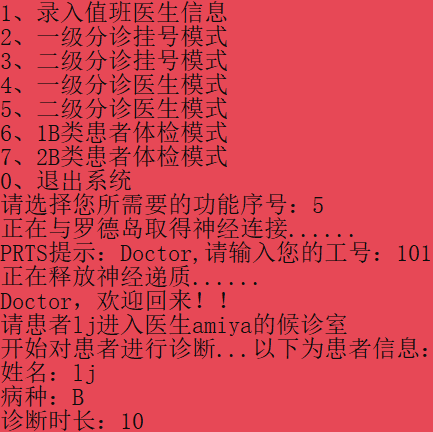
专家医生继续就诊+B类患者重新进入候诊室





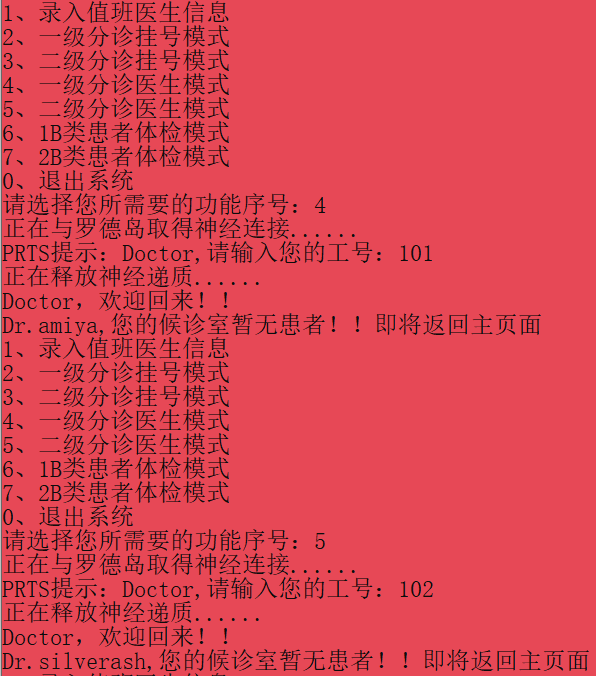


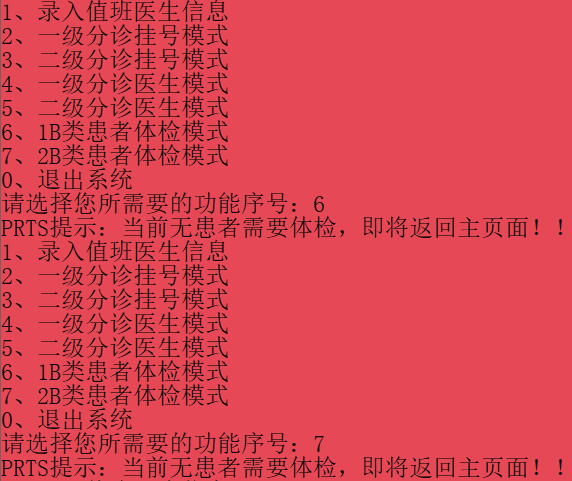




医生依次给患者就诊

4、当各个队列中没有患者时





# 六、总结

通过本次的课程设计，感受颇深。课程设计的一个重要功能，在于运用学习成果，检验学习成果。运用学习成果，把课堂上学到的系统化的理论知识，尝试性地使用于实际设计工作，并从理论的高度对设计工作的现代化提出一些有针对性的建议和设想。检验学习成果，看一看课堂学习和实际工作到底有多大距离，并通过综合分析，找出学习中存在的不足，以便为完善学习计划,改变学习内容和方法提供实践依据。对我们计算机专业的本科生来说，实际能力的培养至关重要，而这种实际能力的培养单靠课堂教学是远远不够的，必须从课堂走向实践。通过课程设计，让我们找出自身状况和实际需要的差距，并在以后的学习期间及时补充相关知识。

这次我选择的题目是基于医院分诊系统的排队策略，众所周知，医院排队在日常生活中处处可见，所以这次的课程设计很有实际价值，利用理论知识，运用到实际生活中。通过一周的努力，不仅仅让我对学过的知识加深了印象，并认识到了理论必须联系实际来解决问题，分析问题。在本次课设中，灵活运用了队列的知识，并且采用类似邻接表的形式，加强了我对链表以及队列的理解。