

集美大學

數據結構課程設計

學 院 計算機工程學院 班 級 計算 2114

姓 名 庄佳強 學 號 202121331104

成 績 指導老師 鄭翠玲、楊艷華、張佳、王智謹

評語：

2022 年 6 月 20 日

目录

1.引言	2
《家谱管理系统》	2
2.功能和原始数据	2
2.1 主要功能.....	2
2.2 原始数据.....	3
3. 程序总体设计	3
3.1 数据结构设计.....	3
3.2 程序总体框架.....	4
3.3 函数原型清单.....	4
3.4 程序组织.....	6
4.功能算法设计和调试	6
4.1 主要功能算法设计.....	6
4.2 调试.....	8
《停车场管理系统》	18
2. 功能和原始数据	18
2.1 主要功能.....	18
2.2 原始数据.....	19
3. 程序总体设计	19
3.1 数据结构设计.....	19
3.2 程序总体框架.....	20
3.3 函数原型清单.....	20
3.4 程序组织.....	21
4. 功能算法设计和调试	21
4.1 主要功能算法设计.....	21
4.2 调试.....	23
6.课程设计总结	29
7.参考资料	29

1.引言

数据结构课程设计不仅可以加深对课程内容的理解，并且可以通过实践进一步掌握程序设计的技能与方法，学会数据的组织方法和现实世界问题在计算机内部的表示方法，并针对问题的应用背景分析，选择最佳的数据结构和算法。让学生熟练掌握 C 语言的基本知识和技能，获得基本掌握面向对象程序设计的基本思路和方法；而且能够利用所学的基本知识和技能，解决简单的程序设计问题。

《家谱管理系统》

【问题描述】

一个家谱关系由若干家谱记录构成，每个家谱记录由父亲、母亲和子女姓名构成，其中姓名是关键字。设计并实现一个简单的家谱管理系统。

2.功能和原始数据

2.1 主要功能

- 1) 从键盘输入家谱记录，建立家族关系树，并存储到外部文件 MyFamily104.txt 中。
- 2) 对于家族成员的添加、删除功能。
- 3) 查询家族成员的双亲、祖先、兄弟、孩子和后代的信息。
- 4) 按关系顺序输出家谱信息资料功能。
- 5) 通过读取外部 txt 文件，建立家族关系树；
- 6) 将添加和删除后的结果同步到外部文件。

2.2 原始数据

原参考书上无数据。

```
姓名  伴侣  性别  孩子个数
小明  小红  1  3
    小崽子  大崽子  0  1
    小比兵  0  1  0
小牛马  0  1  0
小蓝  小白  0  2
    小月  小白吃  1  1
    小行  小田  0  0
小青  小怪  0  2
    小舔  小兵  0  0
    小天  小斌  1  0
```

图 2-1MyFamily104.txt 数据图

3. 程序总体设计

3.1 数据结构设计

```
#define Maxx 30
typedef struct
{
    int num;//孩子个数
    char name[Maxx];///姓名
    int sex;//性别
    char spouse[Maxx];//配偶的姓名
}DataType;//树数据域的定义

typedef struct Tnode
{
    DataType data;
    struct Tnode* nextstr[10];//孩子结点
```

```

    struct Tnode* parent;//双亲结点
}Pedtree;//家谱树

```

3.2 程序总体框架

该程序有着查询和管理的功能，采用了模块化程序设计，其中，main 函数为放置菜单和选择子块功能及一些变量的初始化，其他封装的模块有：家谱构建模块，家谱查找模块，家谱更改模块，家谱输出模块，文件读入模块，综合成家谱管理系统。程序总体框架如下：

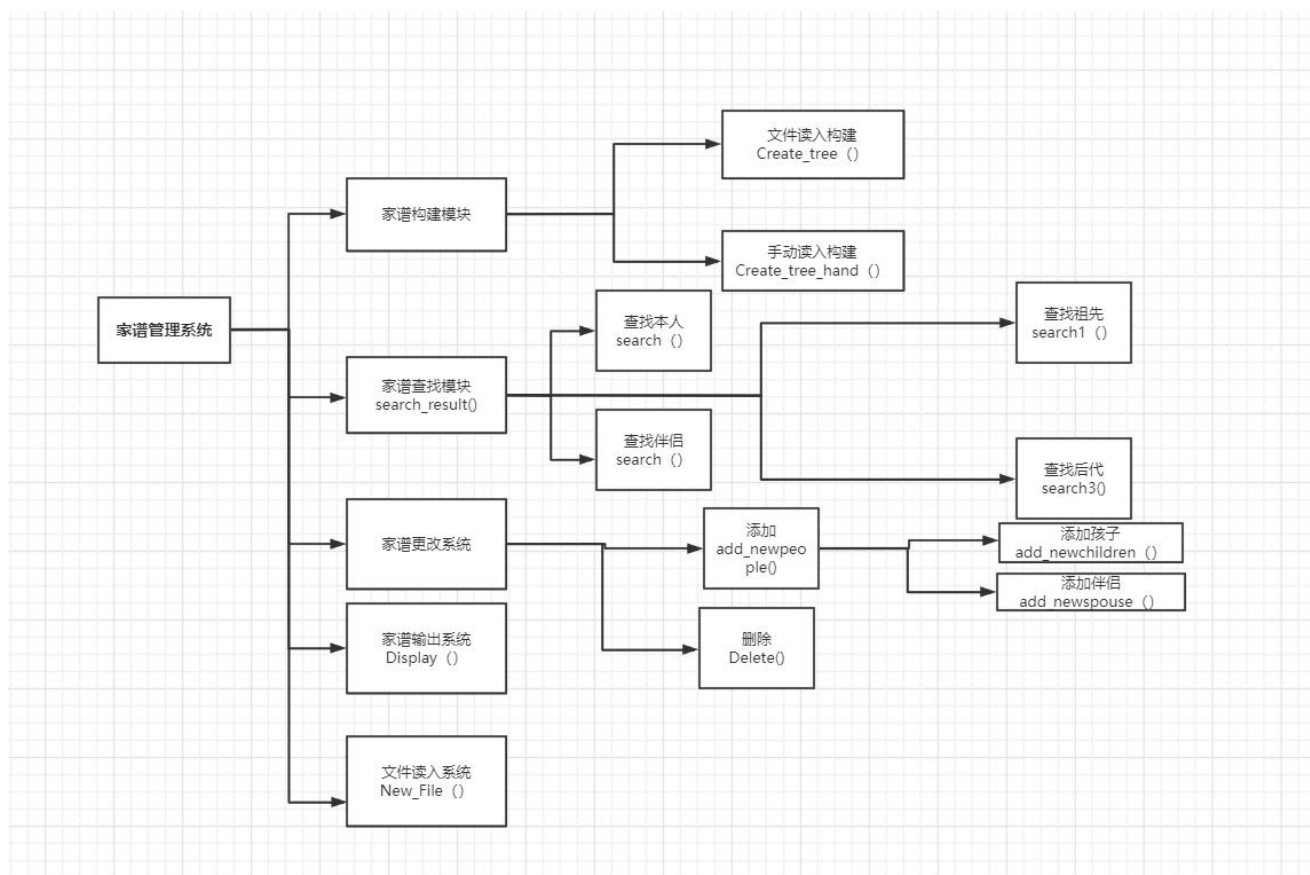


图 3-1 家谱管理系统框架图

3.3 函数原型清单

```

void welcome();//主界面
void welcome1();//在添加处的界面
void Create_tree(Pedtree*& S, int& k);//用 txt 文件中的数据构造家谱树

```

```
void Create_Newtree(Pedtree*& S, int& k); //txt 文件的第二个函数
void Create_tree_hand(Pedtree*& S, int& n); //直接输出数据构造函数
void Create_Newtree_hand(Pedtree*& ChildrenT); //直接构造的第二个函数
数
void Display(Pedtree* S); //输出家谱树
bool Search(Pedtree* S, char temp[], Pedtree *&temptree); //查找姓名
是否查找符合
void Search1(Pedtree* S); //递归查找祖先
bool Search2(Pedtree* S, char temp[], Pedtree*& temptree); //查找伴
侣的姓名是否查找符合
void Search3(Pedtree* S); //递归查找后代
void Search_result(Pedtree* S); //总的查找函数
void add_newpeople(Pedtree*& S); //添加主函数
void add_newchildren(Pedtree*& S); //添加新孩子函数
void add_newspouse(Pedtree*& S); //添加新伴侣函数
void Delete(Pedtree*& S); //删除主函数
void Deleteall(Pedtree* &S); //删除孩子部分函数
void New_File(Pedtree* S); //存储到文件的函数
```

3.4 程序组织

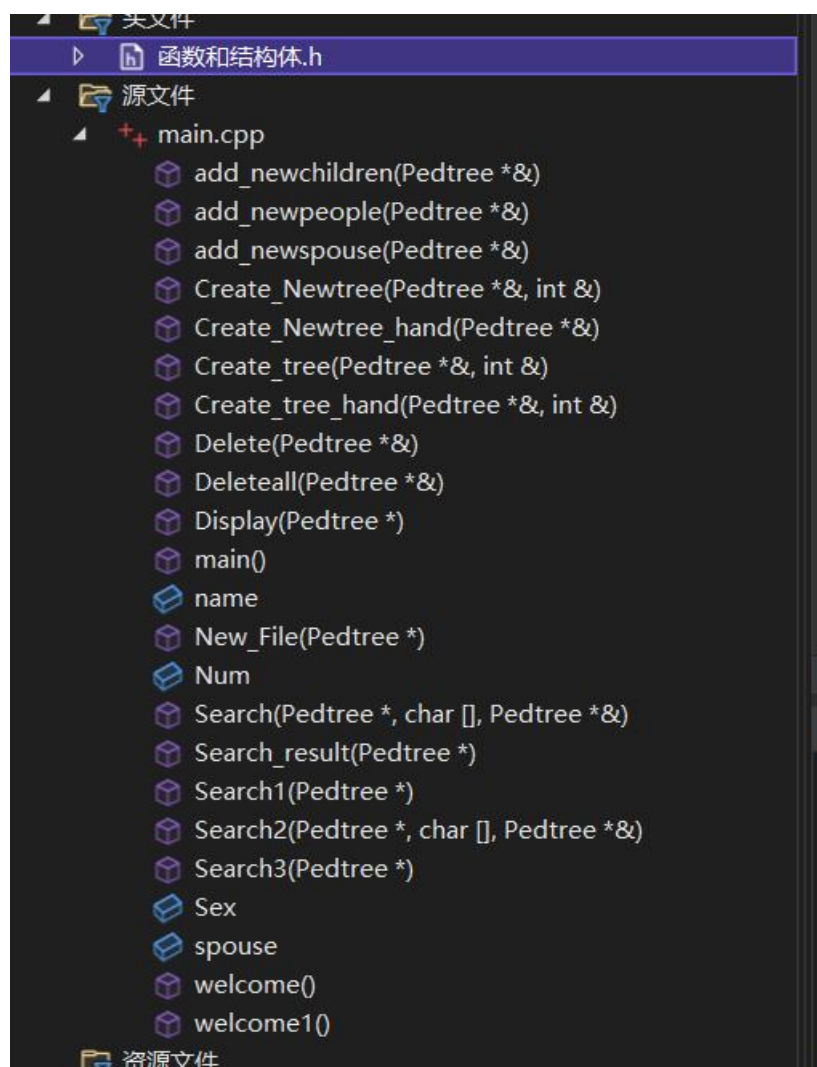


图 3-2 目录树构造图

4.功能算法设计和调试

4.1 主要功能算法设计

4.1.1 家谱查询模块

描述：先输入要查询的名字，程序会先在名字判断查找到位置，接着就会输出他的个个信息。

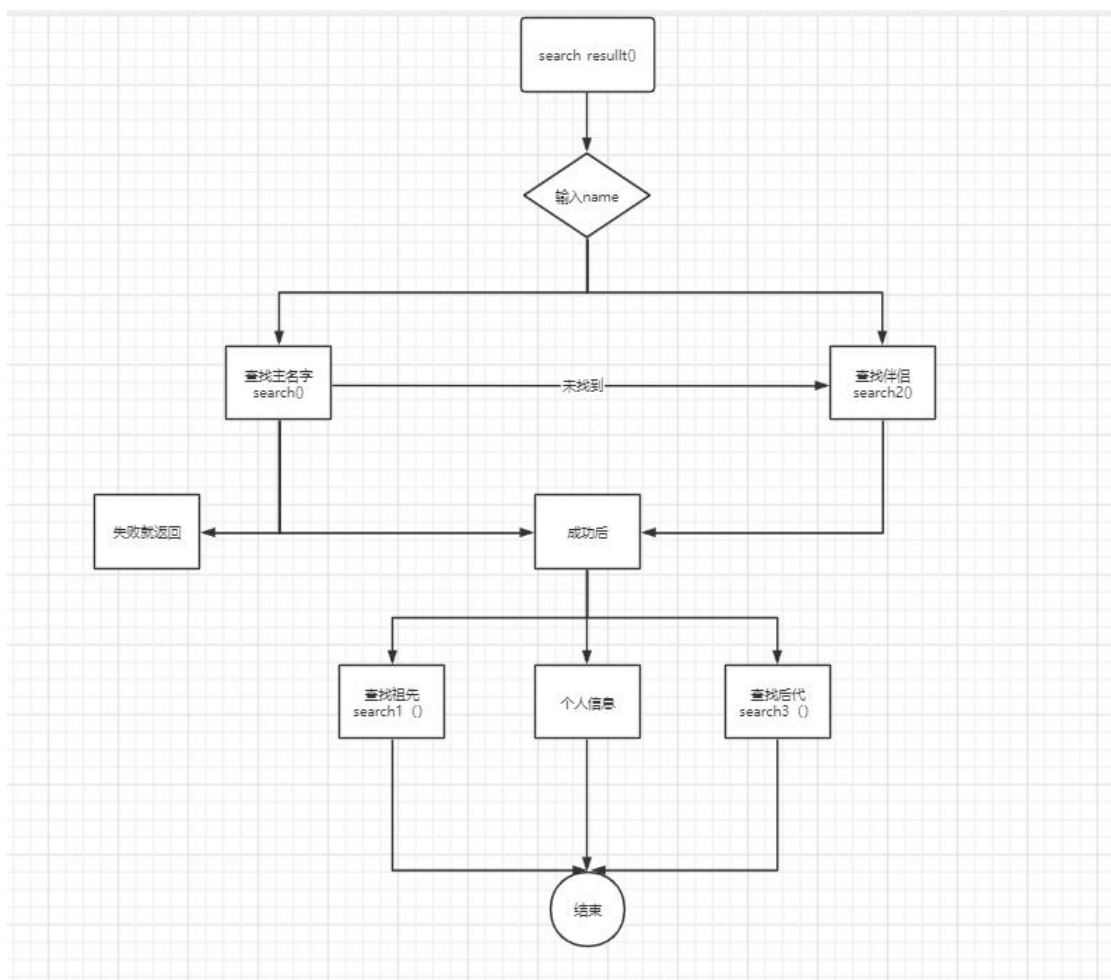
```

Void rearch_result(Pedtree *S)
{
    Pedtree *temptree;
    Cin>>name;
    If(search(S, name, temptree);
    {
        输出个个信息;

    }
    If(search2(S, name , temptree)
    {
        输出个个信息;

    }
}

```



4.1.1 查找程序流程图

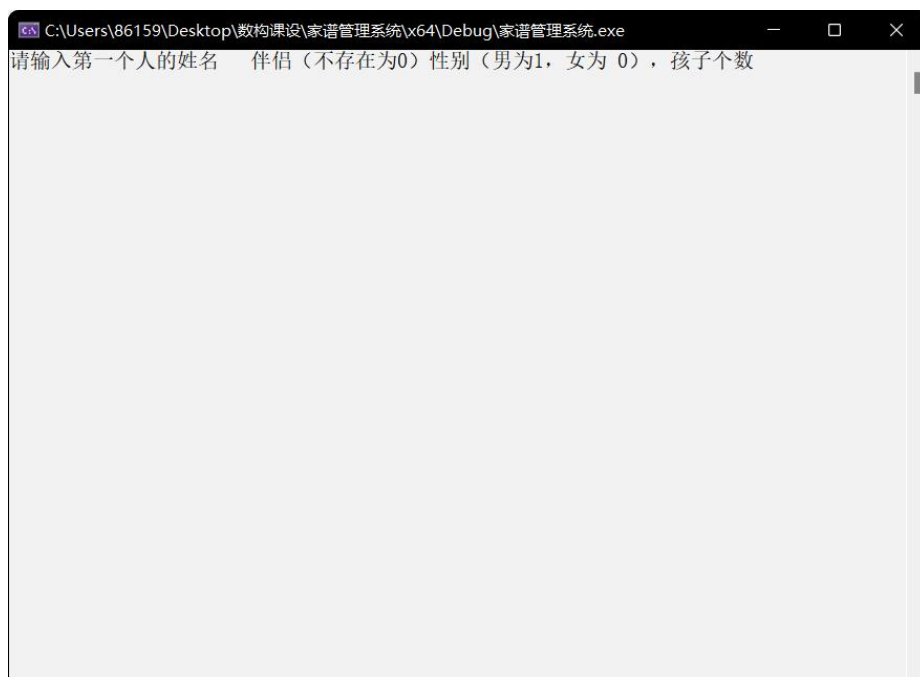
4.2 调试

先进入开始界面



4-1 开始界面图

接着选择第一个构建方法



4-2 手动模式图

输入数据构造

```
C:\Users\86159\Desktop\数构课设\家谱管理系统\64\Debug\家谱管理系统.exe
请输入第一个人的姓名  伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小明 小红 1 3

请输入小明的  第1个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小蓝 小白 0 2

请输入小蓝的  第1个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小月 小白吃 1 1

请输入小月的  第1个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小型 小田 0 0

请输入小蓝的  第2个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小青 小怪 0 2

请输入小青的  第1个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小天 小兵 0 0

请输入小青的  第2个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
啸天 小斌 1 0

请输入小明的  第2个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小崽子 大宰子 0 0

请输入小明的  第3个孩子的 姓名 伴侣（不存在为0）性别（男为1，女为 0），孩子个数
小牛妈 0 1 0

- - - - 家谱建造成功- - - - -
构建完成
输出0返回
```

4-3 手动结果图

在通过 5. 输出家谱图

```
C:\Users\86159\Desktop\数构课设\家谱管理系统\64\Debug\家谱管理系统.exe

姓名：小蓝 性别：女
老公姓名：小白
第1个子女的姓名：小月 性别：男
第2个子女的姓名：小青 性别：女

姓名：小月 性别：男
老婆姓名：小白吃
第1个子女的姓名：小型 性别：女

姓名：小型 性别：女
老公姓名：小田

姓名：小青 性别：女
老公姓名：小怪
第1个子女的姓名：小天 性别：女
第2个子女的姓名：啸天 性别：男

姓名：小天 性别：女
老公姓名：小兵

姓名：啸天 性别：男
老婆姓名：小斌

姓名：小崽子 性别：女
老公姓名：大宰子

姓名：小牛妈 性别：男
没有老婆，未婚

输出0返回
```

4-3 手动构建家谱图

测试文件读入模式，先删除



4-4 删除家谱图

这是通过文件构建出来的

```
C:\Users\86159\Desktop\数构课设\家谱管理系统\x64\Debug\家谱管理系统.exe

---***---***---***---***---***---
姓名：小明 性别：男
老婆姓名：小红
第1个子女的姓名：小逼崽子 性别：女
第2个子女的姓名：小牛马 性别：男
第3个子女的姓名：小蓝 性别：女

姓名：小逼崽子 性别：女
老公姓名：大臂崽子
第1个子女的姓名：小比兵 性别：男

姓名：小比兵 性别：男
没有老婆，未婚

姓名：小牛马 性别：男
没有老婆，未婚

姓名：小蓝 性别：女
老公姓名：小白
第1个子女的姓名：小月 性别：男
第2个子女的姓名：小青 性别：女

姓名：小月 性别：男
老婆姓名：小白
第1个子女的姓名：小行 性别：女

姓名：小行 性别：女
老公姓名：小田

姓名：小青 性别：女
老公姓名：小怪
第1个子女的姓名：小舔 性别：女
第2个子女的姓名：小天 性别：男

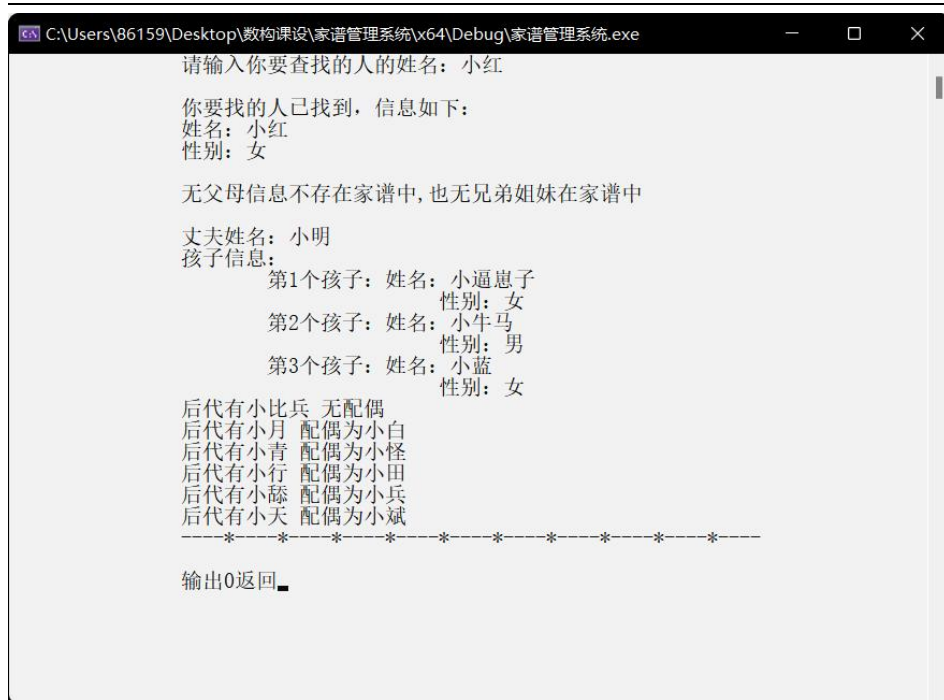
姓名：小舔 性别：女
老公姓名：小兵

姓名：小天 性别：男
老婆姓名：小斌

输出0返回
```

4-5 文件构建家谱遍历图

接着使用查找，这里举例查找“小红”



4-6 家谱查找图

再进行添加



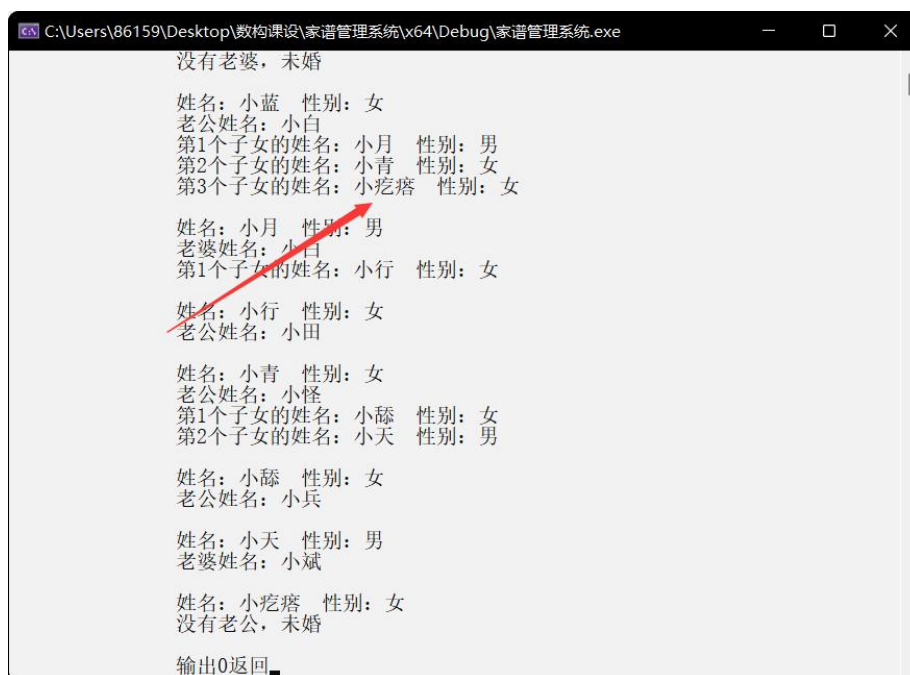
4-7 添加菜单图

先选择添加子女



4-8 子女添加图

查看结果（成功）



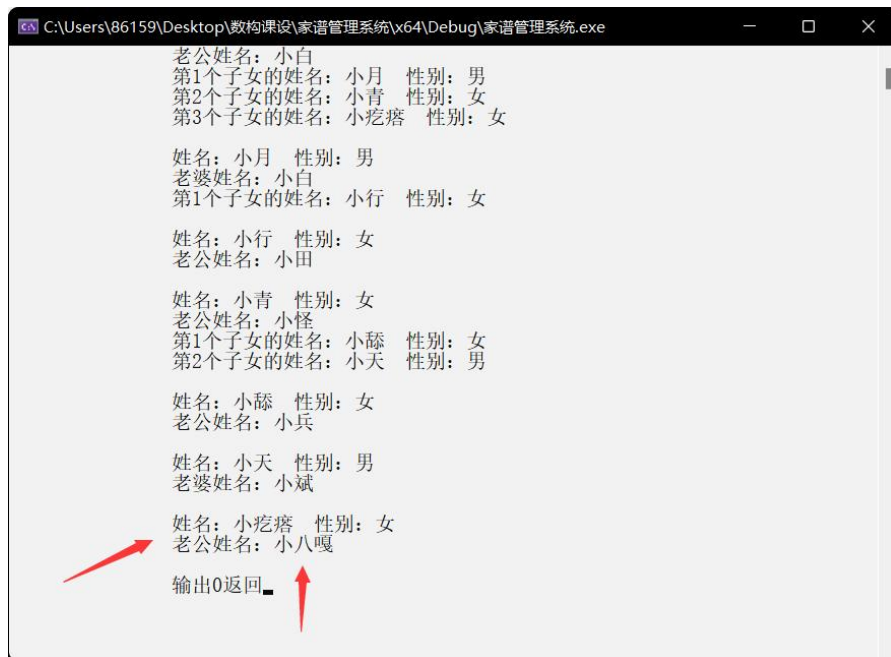
4-9 子女添加结果图

再添加伴侣



4-10 伴侣添加图

查看结果（成功）



4-11 伴侣添加结果图

再删除 例 “小牛马”



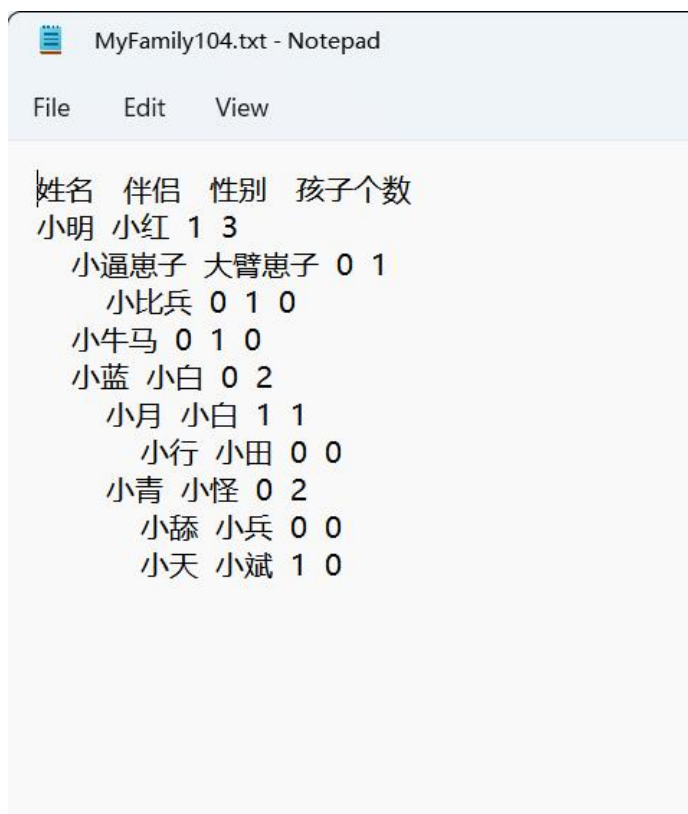
4-12 删除成员图

查看结果（成功）

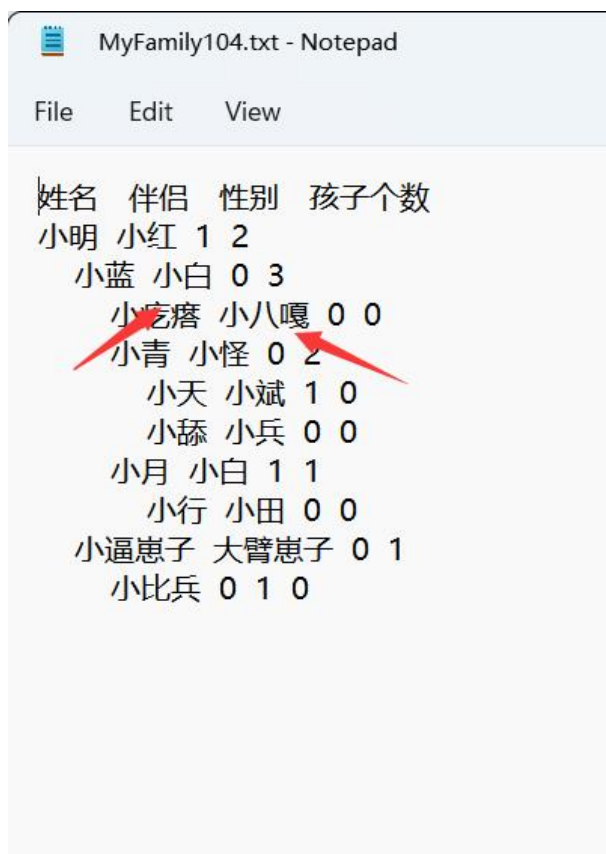


4-13 删除成员结果图

读入文件中



4-14 原文件图



4-15 更新文件图

结束



《停车场管理系统》

【问题描述】

汽车在停车场内按车辆到达时间的先后顺序依次由南向北排列(大门在最北端,最先到达的第一辆车停放在停车场的最南端),若停车场内已停满 n 辆车,则后来的汽车只能在门外的便道(即候车场上)等候,一旦有车开走,则排在便道上的第一辆车即可开入;当停车场内某辆车要离开时,在它之后进入的车辆必须先退出停车场为它让路,待该辆车开出大门外,其他车辆再按原次序进入停车场,每辆停放在停车场的车在它离开停车场时必须按停留的时间长短交纳费用。如果停留在便道上的车辆未进入停车场就要离去,允许其离去,不收停车费,并且仍然保持在便道上等待的车辆的次序。整个停车场的示意图如图 1 所示。

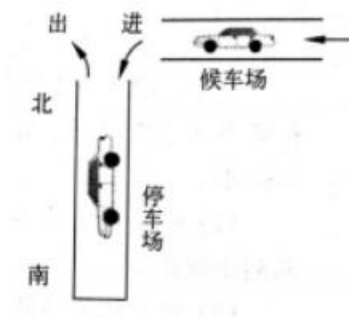


图 1 停车场示意图

2. 功能和原始数据

2.1 主要功能

1) 车辆信息的录入:每辆车“到达”或“离去”的信息、车牌号,如“闽 D 12345”车辆到达或离去的时刻(时间格式为:分:秒)。

2) 根据用户输入的车辆到达或离去的信息,系统输出:若是到达,则输出汽车在停车场内或便道上的停车位置;若是离去,则输出汽车在停车场内的停留时间和应缴纳的费用

3) 可以查看停车场内或便道上车辆相关信息:停车位、车牌号、到达时间。

4) 停车场内的车位数 n 设置 5 个,便道的车位数 4 个。

2.2 原始数据

设计书未给出数据。

3. 程序总体设计

3.1 数据结构设计

```
#define Price 0.1//每一秒的价格
#define MAX_car 5//停车场最大容量
#define MAX_wait 4//候车场最大容量
```

```
typedef struct
{
    int in_time;//进场时间
    int out_time;//出场时间
    char ct[100];//停车时间信息
    char License_plate[20];//车牌号
}Car;
```

```
typedef struct node
{
    Car stop[MAX_car];
    int top;
}car_stop;//停车栈
```

```
typedef struct //候车场
{
    Car wait[MAX_wait];
    int count;
    int front, rear;
}Waitting;//候车循环队列
```

3.2 程序总体框架

该程序有着查询和管理的功能，采用了模块化程序设计，其中，main 函数为放置菜单和选择子块功能及一些变量的初始化，其他封装的模块有：车辆查找模块，停车场信息系统，车辆出入系统。综合起来形成停车场管理系统，程序总体框架图如下：

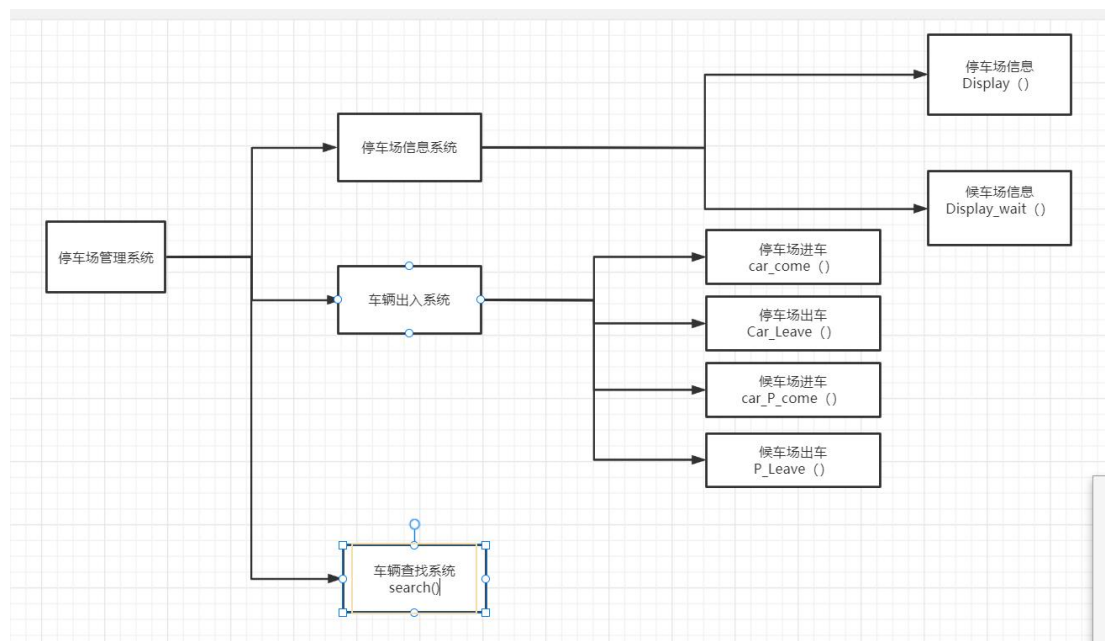
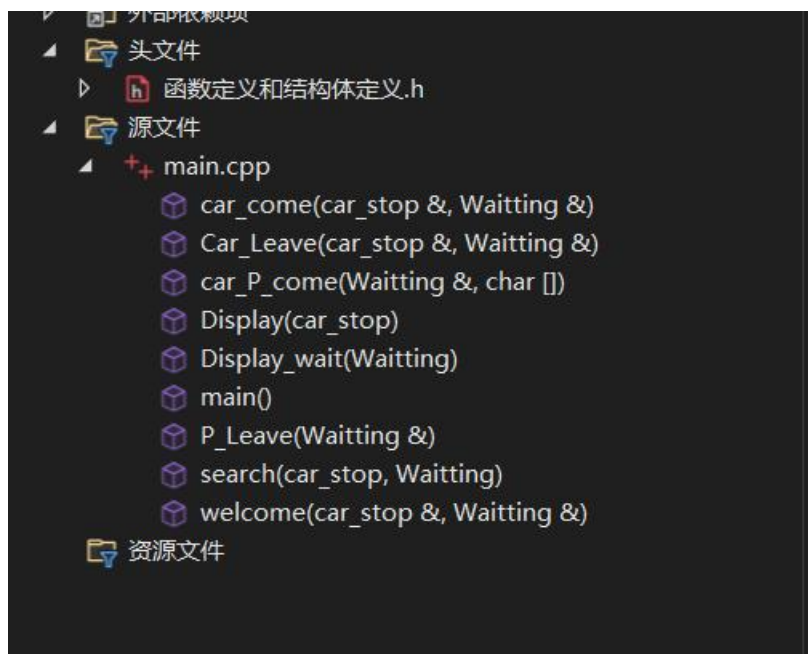


图 3-1 停车场管理系统框架图

3.3 函数原型清单

```
void Car_Leave(car_stop& G, Waitting& P); //停车场出车
void Display(car_stop G); //显示停车场车辆信息
void Display_wait(Waitting P); //显示候车场车辆信息
void P_Leave(Waitting& P); //候车场出车
void car_P_come(Waitting& P, char C[]); //候车场进车
void car_come(car_stop& G, Waitting& P); //停车场进车
void search(car_stop G, Waitting P); //查找车辆
void welcome(car_stop& G, Waitting& P); //菜单界面
```

3.4 程序组织



4. 功能算法设计和调试

4.1 主要功能算法设计

4.1.1 停车查看系统

算法描述：

先输入要查看的车辆，先在停车场中寻找，如果没有找到，就跳转到候车场寻找，用 flag 标记是否寻找到。

```
Void Search(G,P)
{
    Cin>>name;
    For 循环
        If(strcmp(name,G.stop[i].License_plate==0))
            输出相关信息;
    For 循环
        Else if(strcmp(name, P.wait[i].License_plate))
            输出相关信息;
    Else
```

没有找到;
}

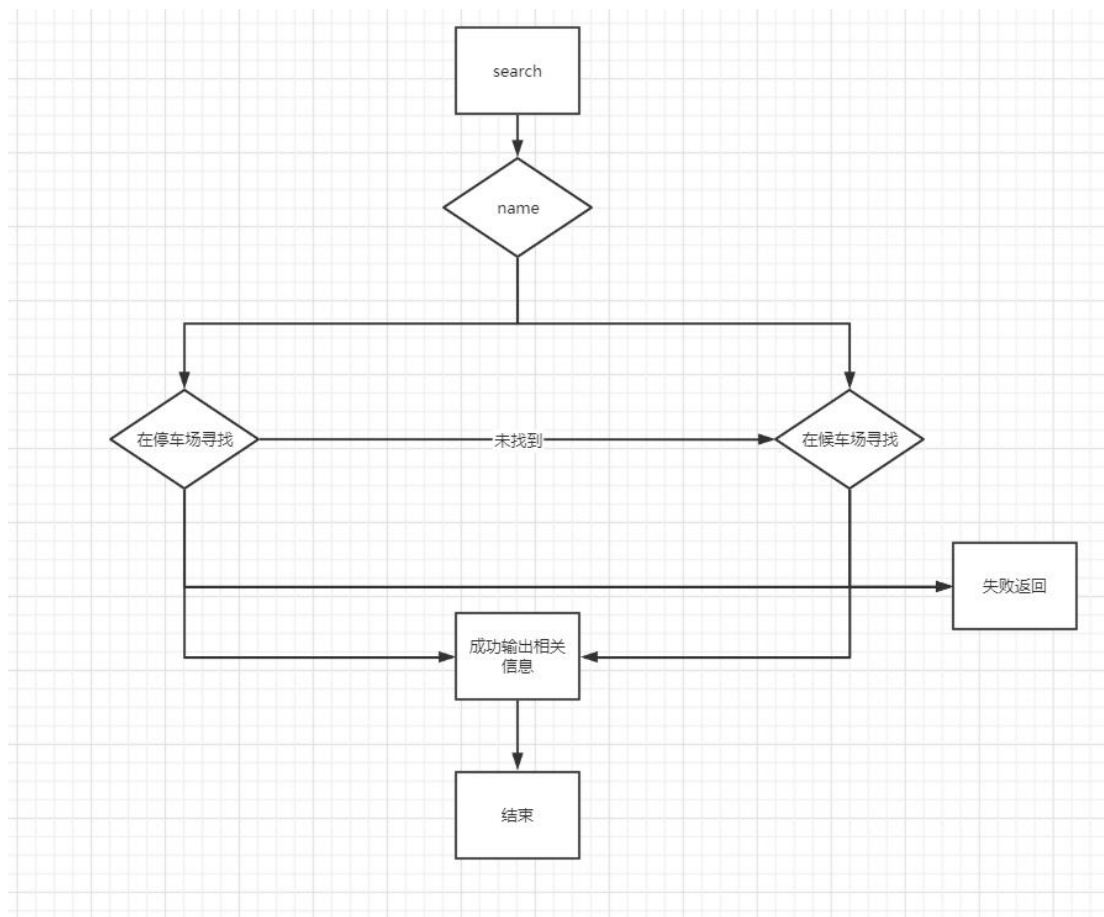


图 4-1.1 停车场查找图

4.2 调试

先打开菜单



图 4-1 菜单图

先让车辆进入停车场和候车场为操作 3

操作完后返回

接着查看结果

这是停车场的车

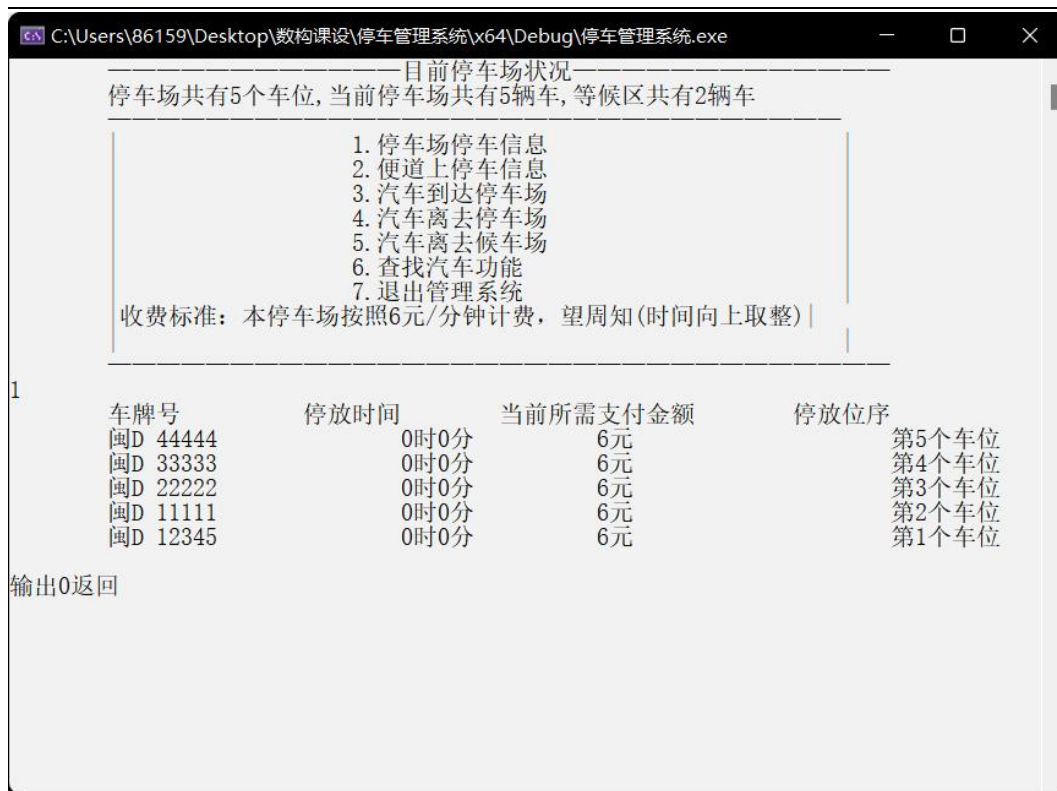


图 4-2 停车场停车图

这是候车场的车

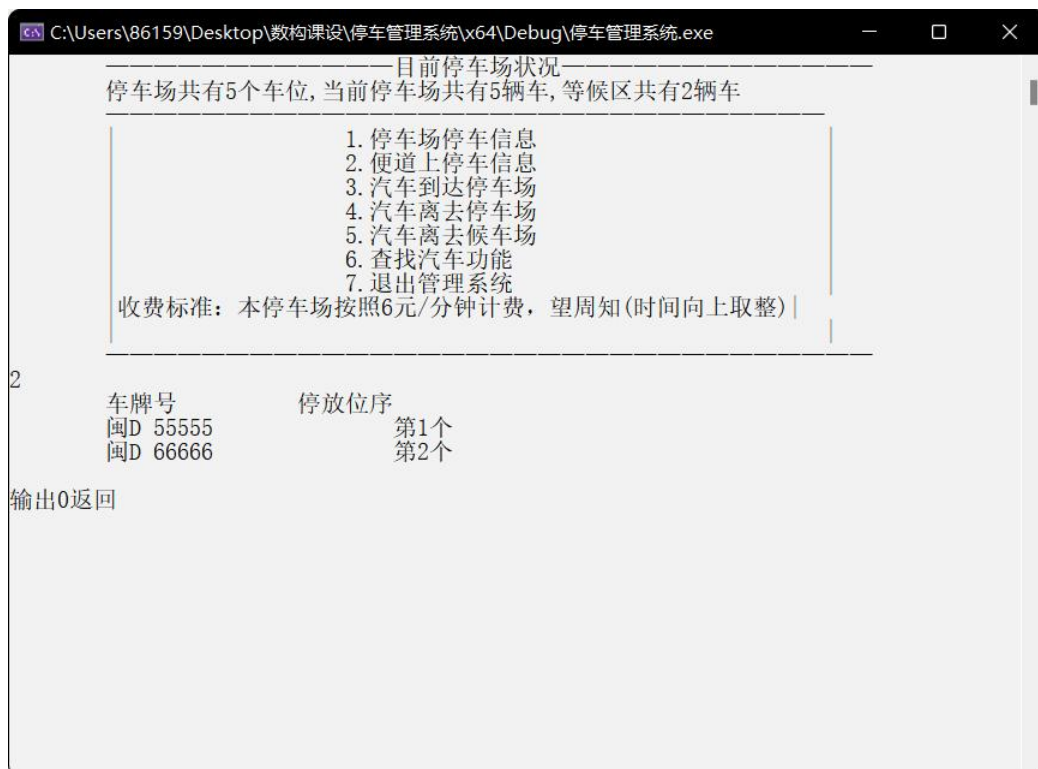


图 4-3 候车场停车图

接着使用一下查找

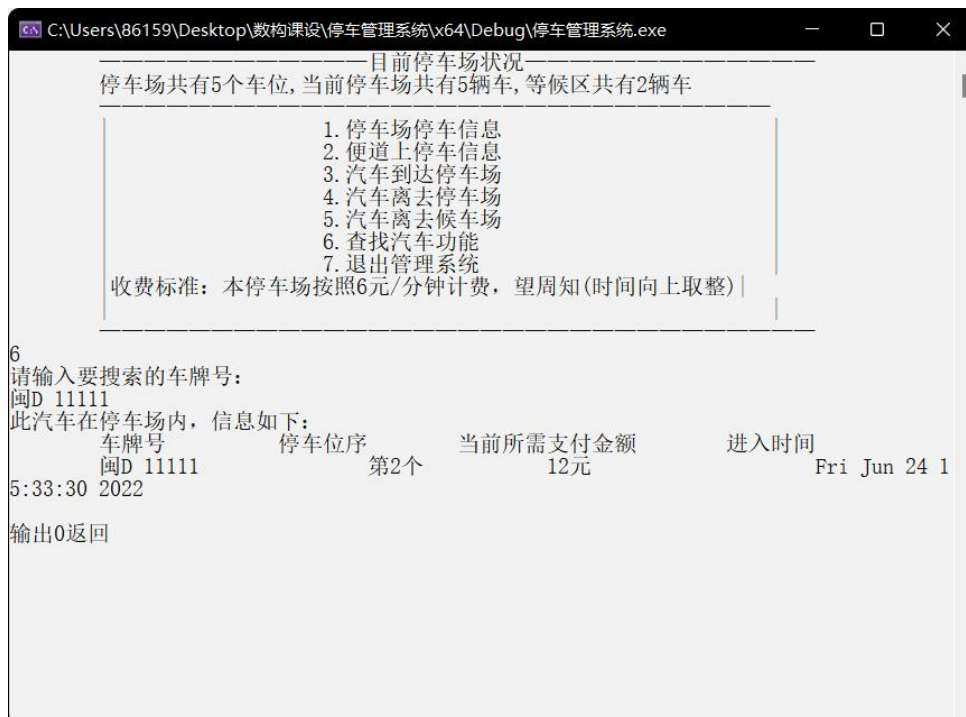


图 4-4 车辆查找图

再尝试一个

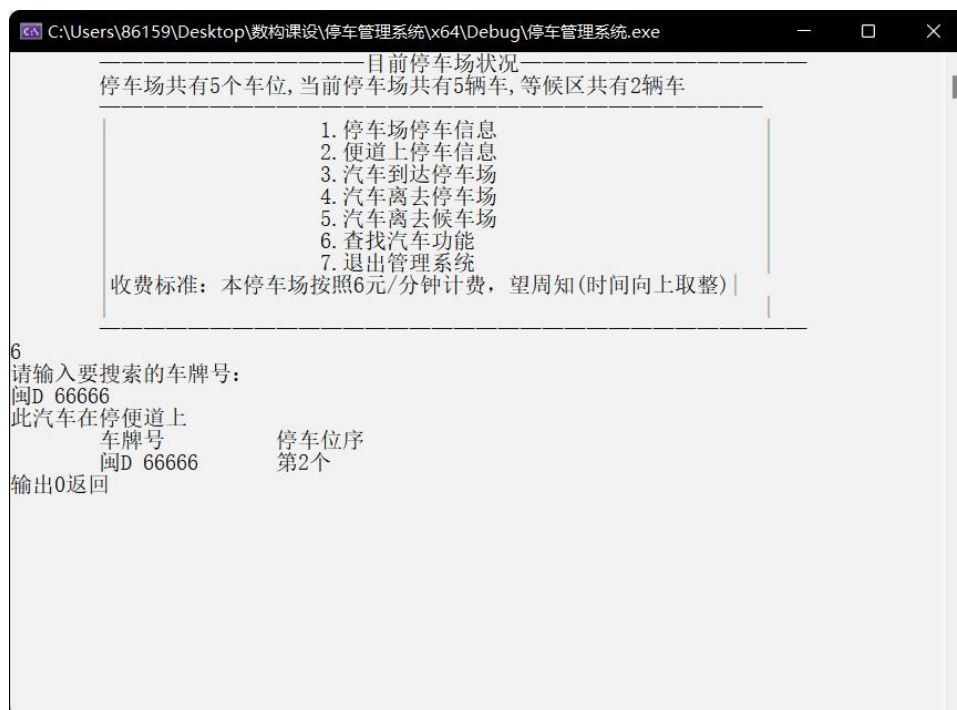


图 4-5 车辆查找图

离开停车场时，如果候车场有人会自动补入

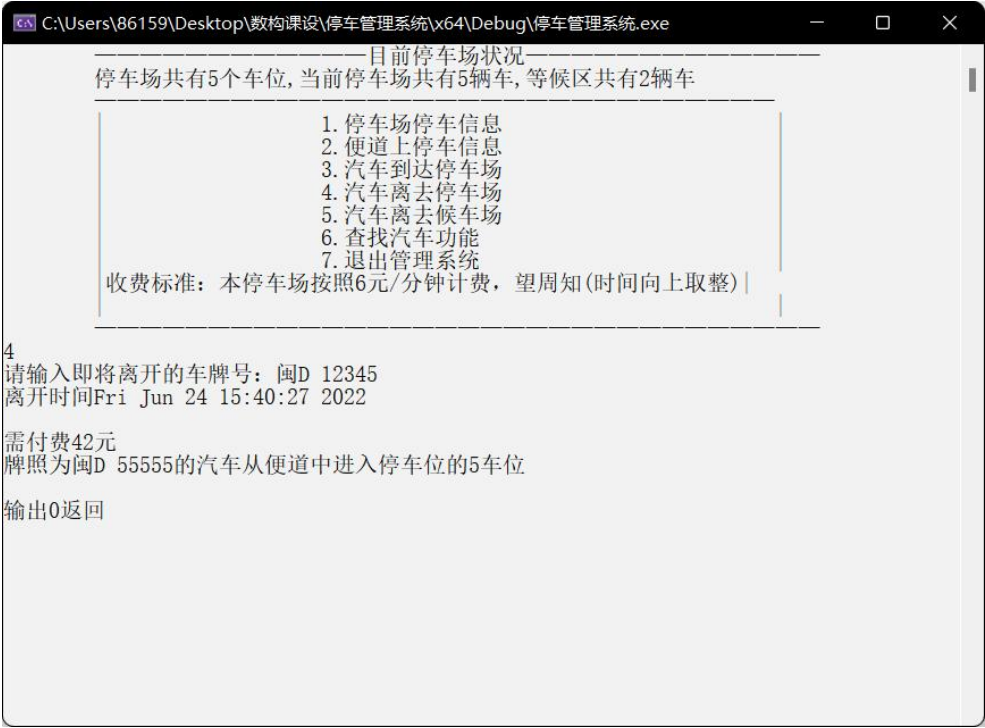


图 4-5 车辆出站图

此时通停车场和候车场变化

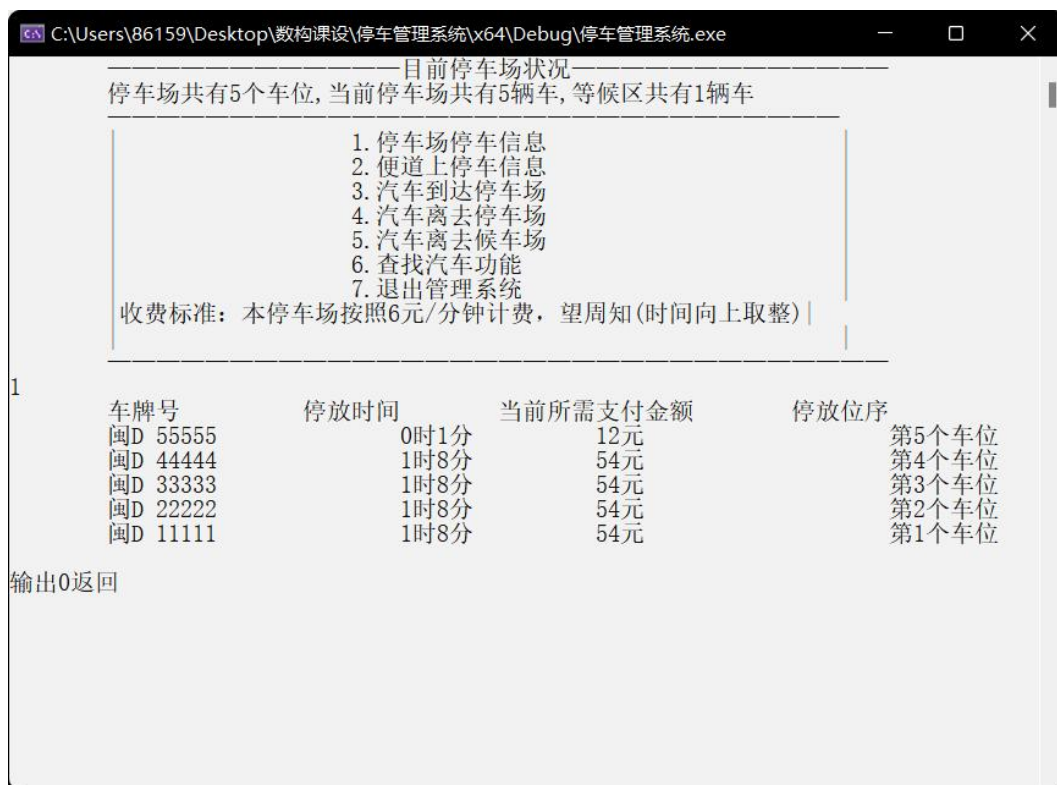


图 4-6 车辆停车场变化图

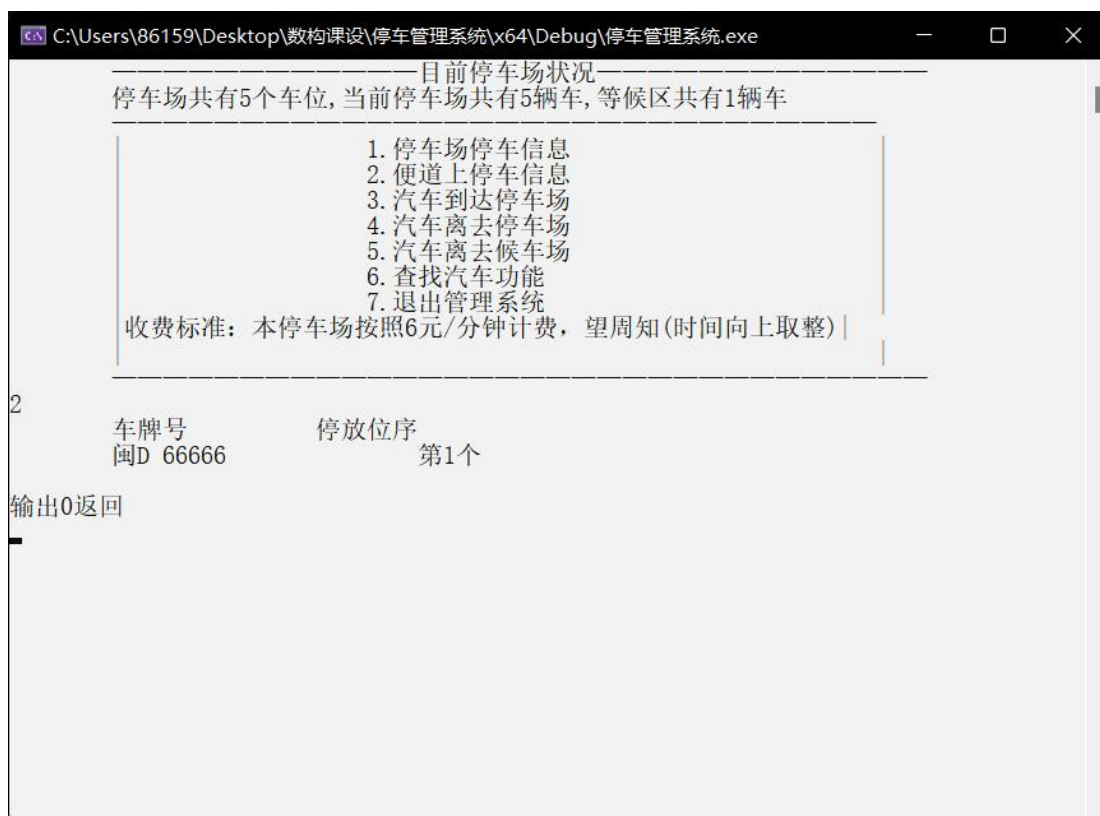


图 4-7 车辆候车场变化图

结束



图 4-8 结束图

6.课程设计总结

一周的课设即将完成，在这一周的学习当中，是我学会了不少的东西，刚开始的时候还挺兴奋的，可当做起来的时候，想象的完全不是一回事。此次的课设是由个人完成，我做的项目是家谱管理系统和停车管理系统。

在分析的过程中，把家谱管理系统和停车管理系统分了多个模块，变成模块化设计，加快了设计速度，减少了大脑负荷。

在设计的时候最头痛的莫过于文件的处理和读入，攻克了这一难关后，剩下的就较普通，没有什么难点，在不断的调试中寻找 bug, 获得 debug 的快乐。

这次的课程设计让我可以把这个学期的内容更好的复习和扩展，让我可以对数据结构更好的认识，这次课程设计很快乐！

7.参考资料

《数据结构教程》 李春葆 清华大学出版社