**东北大学计算机科学与工程学院**

**数据结构课程设计报告**

题目 汽车牌照查询系统

课题组长 窦洹彰

课题组成员 蒋宇欣，郑涵中

专业名称 计算机科学与技术

班级 计1609

指导教师 张晓红

2017 年 12月

**课程设计任务书**

|  |
| --- |
| **题目：**  汽车牌照查询系统 |
| **问题描述：**  在汽车数据的信息模型中，汽车牌照是具有结构特点的一类关键字。汽车牌照是字母和数字混编的，例如01B7238。利用查找和排序算法，实现辽宁省内汽车牌照的快速查找。 |
| **设计要求：**  设计汽车牌照的快速查询程序。  （1）采用顺序表、静态链表等数据结构。  （2）可以随机、文件及人工输入数据。  （3）利用静态链表对汽车牌照进行链式基数排序。  （4）采用折半查找汽车牌照。  （5）可以按城市进行分块索引查找。  （6）其它完善性或扩展性功能。 |
| **指导教师签字：**  **年　　月　　日** |

**目录**

**1 课题概述** 5

**1.1 课题任务** 5

**1.2 课题原理** 5

**1.3 相关知识** 5

**2 需求分析** 5

**2.1 课题调研** 5

**2.2 用户需求分析** 6

**3 方案设计** 6

**3.1 总体功能设计** 6

**3.2 数据结构设计** 7

**3.3 函数原型设计** 8

**3.4 主算法设计** 9

**3.5 用户界面设计** 12

**3.6 输入输出设计** 13

**4 方案实现** 15

**4.1 开发环境与工具** 15

**4.2 程序设计关键技术** 15

**4.3 个人设计实现** 15

**4.3.1 窦洹彰设计实现** 15

**4.3.2 蒋宇欣设计实现** 17

**4.3.3 郑涵中设计实现** 17

**5 测试与调试** 18

**5.1 个人测试** 18

**5.1.1 窦洹彰测试** 18

**5.1.2 蒋宇欣测试** 18

**5.1.3 郑涵中测试** 19

**5.2 组装与系统测试** 19

**5.3 系统运行** 19

**6 课题总结** 20

**6.1 课题评价** 20

**6.2 团队协作** 20

**6.3 个人设计小结** 20

**6.3.1窦洹彰设计小结** 20

**6.3.2蒋宇欣设计小结** 21

**6.3.3郑涵中设计小结** 21

**7 附录：** 22

**A 课题程序设计分工** 22

**A-1 课题程序设计分工** 22

**A-2 课题报告分工** 23

**B 课题设计文档（光盘）** .24

**B-1课程设计报告（电子版）** .

**B-2源程序代码（\*.H，\*.CPP）** .24

**B-3工程与可执行文件** .

**B-4屏幕演示录像文件** .

**C 使用手册** 84

**C-1 运行环境说明** 84

**C-2 操作说明** 84

**1 课题概述**

* 1. **课题任务**

**汽车牌照的快速查询**

【问题描述】

在汽车数据的信息模型中，汽车牌照是具有结构特点的一类关键字。汽车牌照是字母和数字混编的，例如01B7238。利用查找和排序算法，实现辽宁省内汽车牌照的快速查找。

【设计要求】

设计汽车牌照的快速查询程序。

（1）采用顺序表、静态链表等数据结构。

（2）可以随机、文件及人工输入数据。

（3）利用静态链表对汽车牌照进行链式基数排序。

（4）采用折半查找汽车牌照。

（5）可以按城市进行分块索引查找。

（6）其它完善性或扩展性功能。

* 1. **课题原理**

结合C语言、C++、数据结构相关知识，通过输入汽车牌照信息，保存在一个文件中。程序从文件中读入信息，把车牌号码转换成相应的关键字并把车牌信息和关键字保存在链表中。再采用静态链表数据结构储存汽车牌照信息，利用了链式基数排序，采用折半查找法和分块索引查找，以及其他文件输入输出等完善功能。

**1.3相关知识**

数据结构：静态链表、链式基数排序、折半查找法、分块索引查找。

C语言、C++：文件的输入输出等。

其他知识：一个汽车的拍照相当于一辆汽车的身份证，因此通过查询汽车牌照便可知道这辆汽车的主人姓名、车辆型号、主人身份证号等信息。我们的我们的程序是有关辽宁省的汽车牌照查询系统，所以汉字都是“辽”开头只需考虑对字母和数字的排序即可。我们把0-9-A-Z作为一排列方法。

1. **需求分析**
   1. **课题调研**

汽车牌照是车辆的“身份证”，一车一牌。汽车牌照的管理是车辆管理重要的一环，而且汽车数量庞大，其牌照数量同样庞大。所以人工管理方式不仅工作效率低，错误多，劳动强度大，而且需要重复工作以检验正确增加准确性。所以需要汽车牌照查询这样的高效、底错的管理系统。

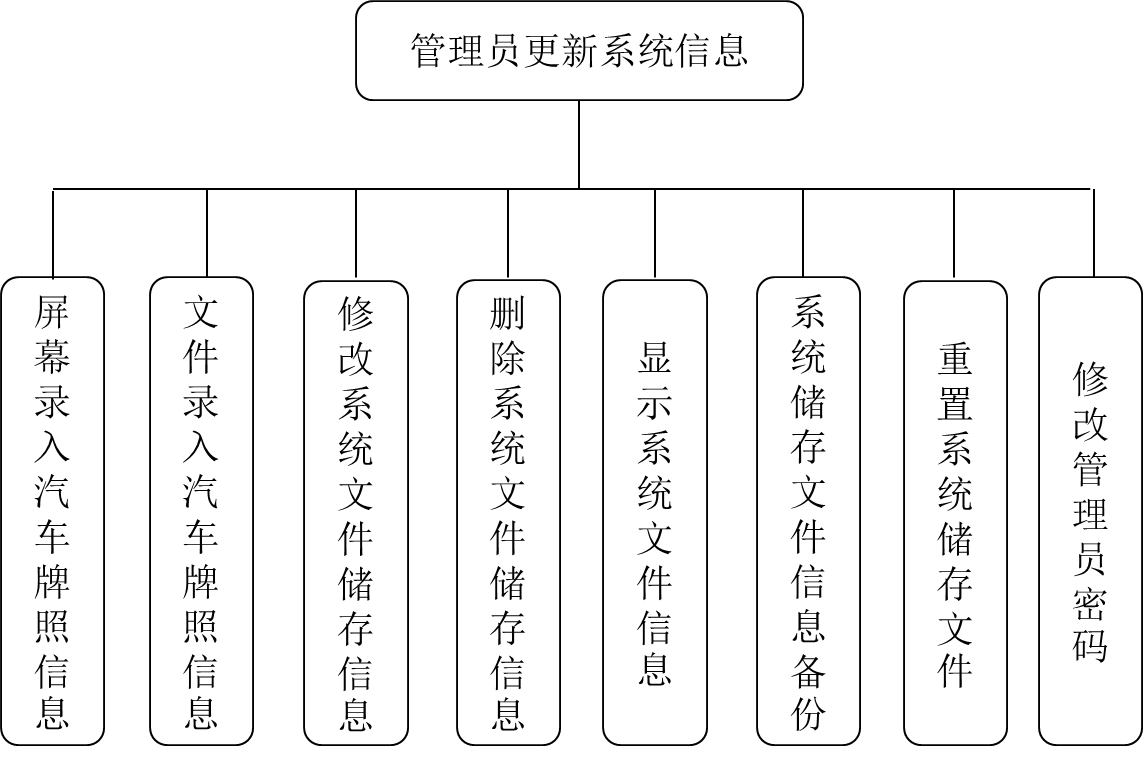
* 1. **用户需求分析**

汽车牌照查询系统不仅仅实现了汽车车牌号码的快速查询，方便了对大量汽车牌照的信息查找，而且还实现了汽车号码的选择。如果系统中没有用户输入的汽车车牌号码，就会显示可以使用，否则就不能使用。除此之外，为了更新系统数据，要拥有信息的录入功能（从文件读入和从屏幕输入）、信息的修改、删除功能、信息备份、以及系统重置等。

1. **方案设计**

**3.1 总体功能设计**

1. 从文件读入数据保存进链表
2. 对车牌进行基数排序
3. 用折半查找来查询汽车牌照
4. 用分块索引查找来查询不同城市中的汽车牌照
5. 屏幕显示输出车牌信息
6. 管理员更新系统
7. 存储文件信息的备份

****

**3.2 数据结构设计**

程序要求实现对汽车牌照的排序与查找，而如果仅仅进行牌照的排序与查找，则显得程序没有太大实用性，所以考虑在程序中假如例如车主姓名、身份证号以及车辆型号等内容来增加程序的实用性。为了能够更好的完成这些功能，在这里用链表来存储所有车辆的信息，用链表中的单个结点来存储一辆车的信息，而对应的各个结点的域中则存储器对应的车辆信息，车主的姓名及车辆型号则使用一维数组来存储。在基数排序时，每趟的数据由两个链队列来完成，链队列的个数为基数的个数，两个链队列的队列指针分别指向每组链队列的队头和队尾。

#define MAXSIZE 1000

//数据域

typedef struct ElemType

{

char plate[7];

char name[20];

char ID[19];

char car[100];

}ElemType;

//静态链表

typedef struct Node

{

ElemType data;

int cur;

}StaticLinkList[MAXSIZE];

**3.3 函数原型设计**

表 1.1函数原型设计

|  |  |
| --- | --- |
| 函数名称 | 函数功能 |
| 界面  输出主界面  void Main\_interfasce()  输出管理员更新系统信息界面  void Administrator\_interfasce()  输出表头  输出汽车牌照信息查询表头  void Header\_query()  输出汽车车牌号码选择表头  void Header\_choose()  输出管理员更新系统信息表头  void Header\_administrator()  输出空格  void Space(int n)  输出分界线  void Line() | 向屏幕输出界面 |
| 汽车牌照信息查询  void Search(StaticLinkList S,int c) | 折半查找法查询汽车牌照信息 |
| 汽车车牌号码选择  void Choose(StaticLinkList S,int c) | 分块索引查找法反映号码选择信息 |
| 屏幕录入车辆牌照信息  录入汽车车辆号码  ElemType Input\_plate(ElemType elem,int cls)  录入车主姓名  ElemType Input\_name(ElemType elem)  录入车主身份证号  ElemType Input\_ID(ElemType elem)  录入车辆型号  ElemType Input\_car(ElemType elem) | 录入车辆牌照信息 |
| 读取系统文件数据  int Read(StaticLinkList S) | 系统文件读取 |
| 储存信息到系统文件  void Store(StaticLinkList S,int s) | 系统文件存储 |
| 链式基数排序  void Sort(StaticLinkList S,int c) | 链式基数排序 |
| 屏幕录入汽车牌照信息  void Enter\_screen()  文件录入汽车牌照信息  void Enter\_file() | 录入车辆牌照信息，然后进行链式基数排序，并储存信息到系统文件 |
| 折半查找法  ElemType Search\_Binary(StaticLinkList S,  int c,ElemType elem)  int Search\_Binary\_Admin(StaticLinkList S,int c,ElemType elem)  分块索引查找  bool Search\_Block(StaticLinkList S,int c,ElemType elem,char ch) | 信息查找 |
| 管理员更新系统信息  void Administrator() | 管理员更新系统信息选择界面实现 |
| 修改系统文件储存信息  void Amend() | 修改系统文件信息 |
| 删除系统文件储存信息  void Delete() | 删除系统文件信息 |
| 显示系统文件信息  void Show() | 可以选择全部显示和逐个单独显示 |
| 系统储存文件信息备份  void Backups() | 将系统文件备份到指定文件 |
| 重置系统储存文件  void Clearfile() | 将系统文件置空 |
| 密码验证  bool PassLink()  修改管理员密码  void Password\_Change() | 密码验证及修改 |

**3.4 主算法设计**

**建立链表，添加汽车牌照信息：**

开始

申请一结点p并未其分配存储空间

head=p

head=NULL

**Y N**

输入汽车的相应信息，经过相应处理后存入结点p相应的域

将该结点按尾插法插入到链表的相应位置

返回该链表的头指针

结束

**按车牌号码查询：**

开始

输入需要查找的牌照

将待查找的牌照号处理后，存于一新开辟的存储空间中

调用折半查找函数返回i

c=0

**Y N**

没有查找成功

查找成功并输出该车的信息

结束

**车牌号码选择:**

开始

选择所在城市

**N**

选择无误

**Y**

输入想要的车牌号码

**N**

输入无误

**Y**

将输入好的牌照号处理后，存于一新开辟的存储空间中

调用分块索引函数返回k

k=1

**N Y**

没有选择成功

选择成功

结束

**各函数间的关系：**

**菜单界面，选择功能：**

开始

输入i

i=0或1或2或9

**N**

**Y**

i=1

**Y N**

调用Search()

i=2

**Y N**

调用Choose()

i=9

**Y N**

调用Administrstor()

i=0

结束

**管理员界面，选择功能：**

开始

输入i

i>=0且i<9

**N**

**Y**

i=1

**Y N**

调用Enter\_screen()

i=2

**Y N**

调用Enter\_file()

i=3

**Y N**

调用Amend()

i=4

**Y N**

调用Delete()

i=5

**Y N**

调用Show()

i=6

**Y N**

调用Backups()

i=7

**Y N**

调用Clearfile()

i=8

**Y N**

调用Password\_Change()

i=0

结束

主界面选择：1、汽车牌照信息查询2、汽车车牌号码选择0、退出9、管理员更新系统信息；选择9后，先进行密码验证，如果密码正确将进入管理员更新系统信息界面，否则3次密码错误将返回主界面。管理员更新系统信息界面选择：1、屏幕录入汽车牌照信息2、文件录入汽车牌照信息3、修改系统文件储存信息4、删除系统文件储存信息5、显示系统文件信息6系统储存文件信息备份7、重置系统储存文件8、修改管理员密码0、返回主界面

int main()

{

StaticLinkList SLL;

int s=0;

int i=100;

while(i!=0)

{

system("cls");

Header\_main();//主界面表头

Main\_interfasce();//主界面内容

s=Read(SLL);//读取系统数据

cout<<"请选择：";

cin>>i;

while(i!=0&&i!=1&&i!=2&&i!=9)

{

cout<<endl<<"输入的选择不正确！"<<endl;

cout<<"请选择1、2或者0"<<endl;

cout<<endl<<"请选择：";

cin>>i;

}

switch(i)

{

case 1:Search(SLL,s);break;// 汽车牌照信息查询

case 2:Choose(SLL,s);break;// 汽车车牌号码选择

case 9:// 管理员更新系统信息

if(PassLink())//密码验证

Administrator();// 管理员更新系统信息

break;

case 0://退出

system("cls");

Header\_main();//主界面表头

cout<<endl;

Space(10);

cout<<"退出系统！"<<endl;

cout<<endl<<endl;

Line();

break;

}

}

return 0;

}

void Administrator()

{

int i;

while(i!=0)

{

system("cls");

Header\_administrator();// 管理员更新系统信息表头

Administrator\_interfasce();//管理员更新系统信息界面

cout<<"请选择：";

cin>>i;

while(i<0||i>8)

{

cout<<endl<<"输入的选择不正确！"<<endl;

cout<<"请选择1、2、3、4、5、6、7、8或者0"<<endl;

cout<<endl<<"请选择：";

cin>>i;

}

switch(i)

{

case 1:Enter\_screen();break;// 屏幕录入汽车牌照信息

case 2:Enter\_file();break;// 文件录入汽车牌照信息

case 3:Amend();break;// 修改系统文件储存信息

case 4:Delete();break;// 删除系统文件储存信息

case 5:Show();break;// 显示系统文件信息

case 6:Backups();break;// 系统储存文件信息备份

case 7:Clearfile();break;// 重置系统储存文件

case 8:Password\_Change();break;// 修改管理员密码

case 0:// 返回主界面

system("cls");

Header\_administrator();// 管理员更新系统信息表头

cout<<endl<<endl;

Space(20);

cout<<"即将返回主菜单！"<<endl;

cout<<endl<<endl<<endl;

Line();

system("pause");

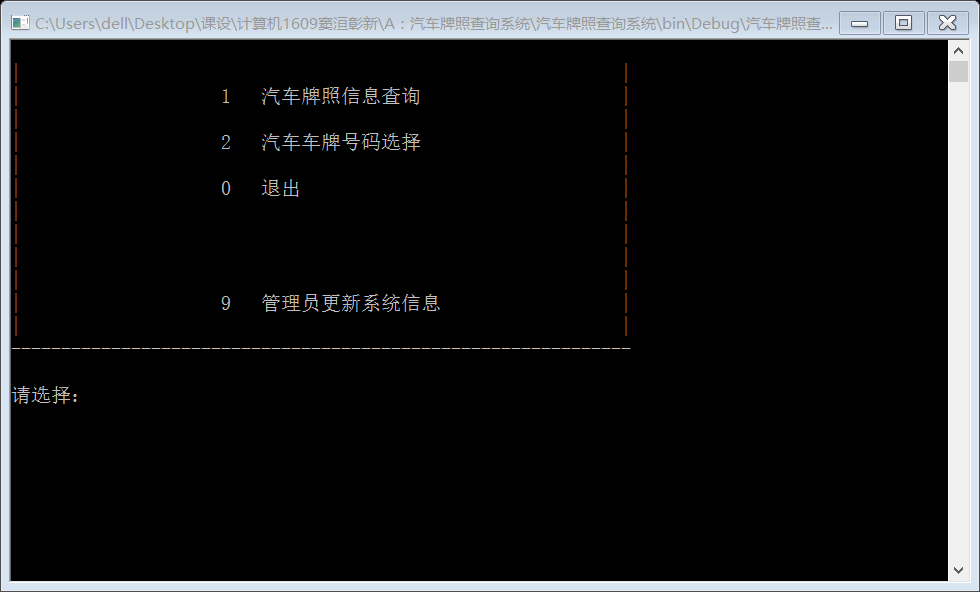
break;

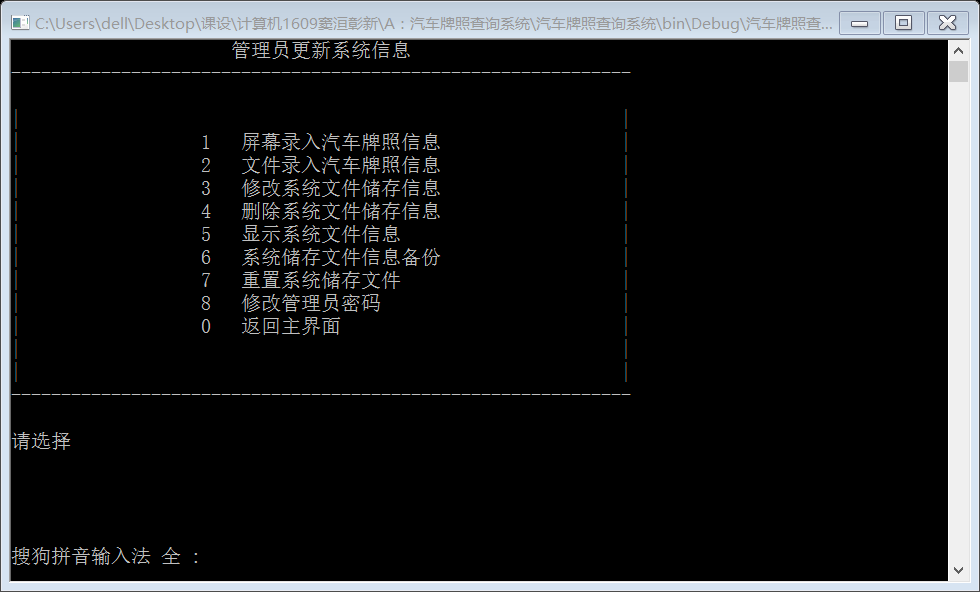
}

}

}

**3.5 用户界面设计**



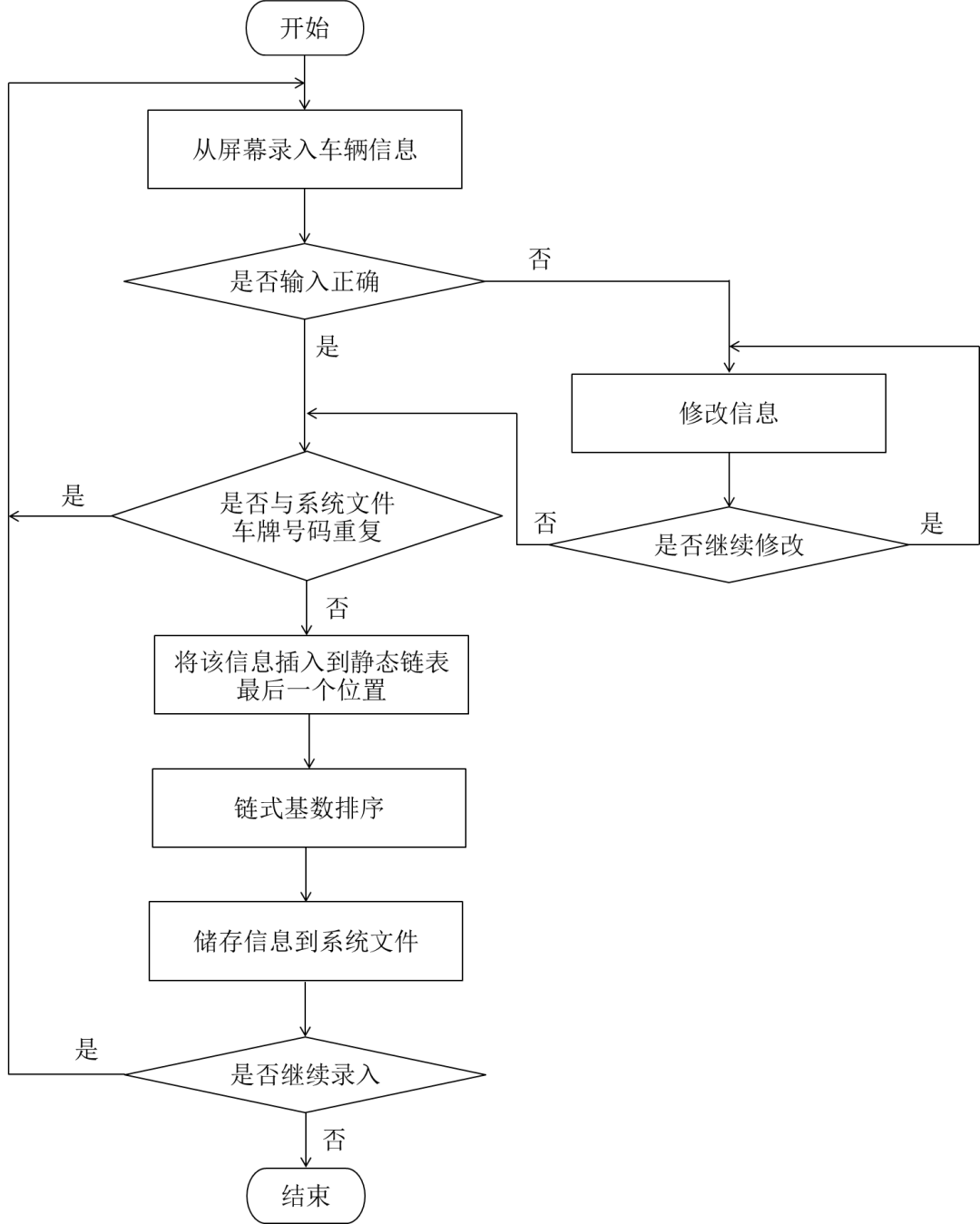
****

**3.6 输入输出设计**

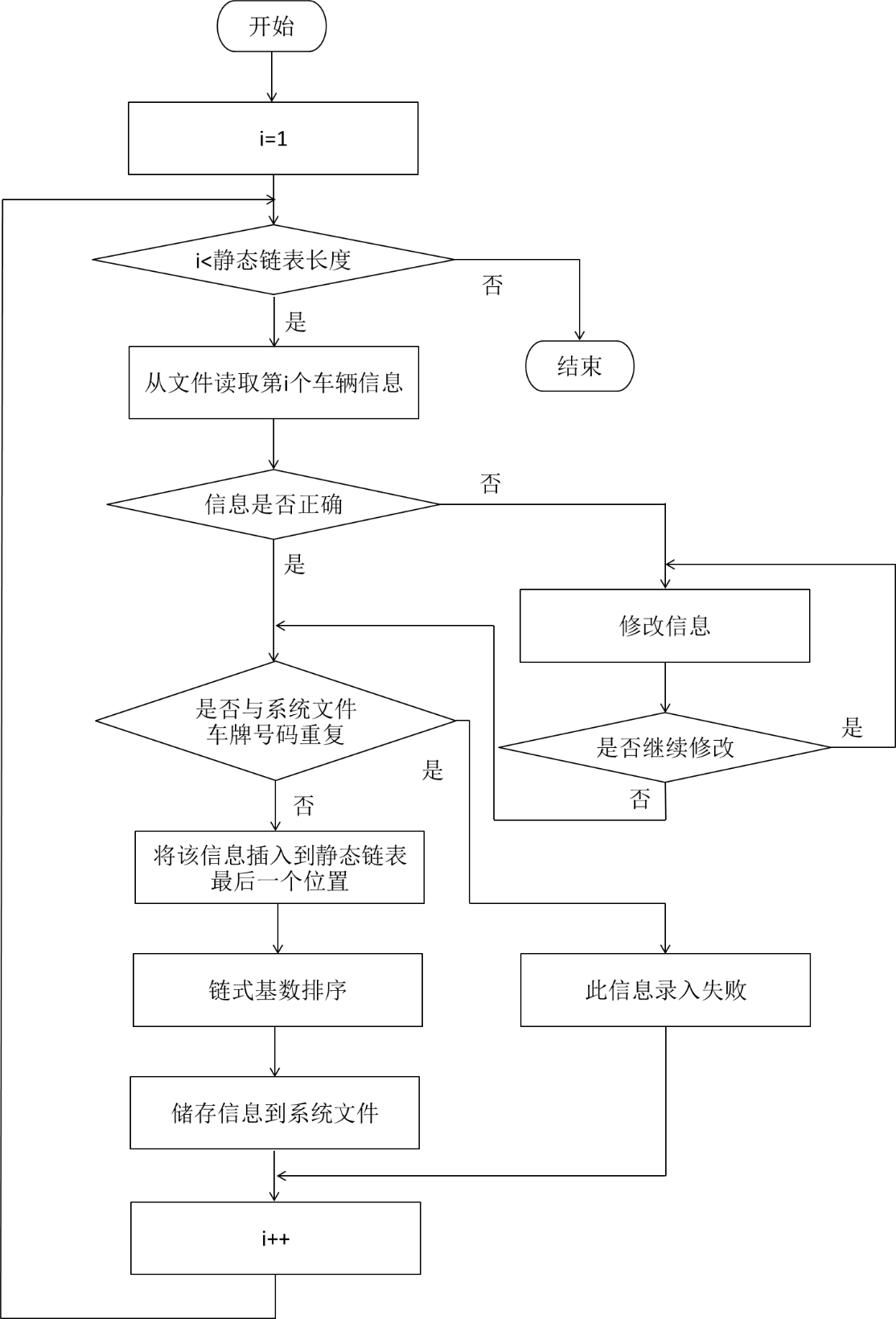
输入

输出

屏幕输入



文件输入



1. **方案实现**
   1. **开发环境与工具**

开发环境：Windows 7

工具：Code Blocks

* 1. **程序设计关键技术**

静态链表

链式基数排序

折半查找法

分块索引查找

**4.3 个人设计实现**

**4.3.1窦洹彰设计实现**

数据结构的实现：

静态链表

**折半查找法：**

运用数据结构折半查找法的相关知识，查找成功则返回查找到的汽车牌照信息，否则返回空信息。

**删除系统文件储存信息：**

主要是先调用录入汽车车辆牌照函数，再调用折半查找法（对折半查找法修改后编写一个新的函数，此函数查找成功返回的是该信息在静态链表中的位置，否则返回0）查找成功则输出该信息到屏幕，选择是否删除该信息，然后修改静态链表将删除后的结点前移，否则显示未找到该信息。同时实现循环，可以连续删除。

**链式基数排序：**

运用数据结构链式基数排序的相关知识

**分块索引查找**：

运用数据结构分块索引查找的相关知识

先按关键字提取信息，然后顺序查找信息，查找成功则返回True，否则返回False。

**主函数的实现**

**系统储存文件信息备份：**将系统文件储存到管理员指定文件

**重置系统储存文件**

**汽车牌照信息查询函数：。**

主要是调用录入汽车车辆牌照函数，再调用折半查找法，查找成功则输出该信息到屏幕，否则显示未找到该信息。同时实现循环，可以连续查找。

**汽车车牌号码选择：**

先选择14个车辆的所在城市：1、沈阳市2、大连市3、 鞍山市4、

抚顺市5、本溪市6、丹东市7、锦州市8、营口市9、阜新市10、 辽阳市11、盘锦市12、铁岭市13、朝阳市14、葫芦岛市，然后接收要查询的车牌号码，调用分块索引查找函数，查找成功则显示不可以使用，否则显示可以使用。同时实现循环，可以连续选择号码。

**修改系统文件储存信息：**

先调用车牌号码录入函数，再调用折半查找法（返回位置），查找成功则输出该信息到屏幕，然后选择：1、车主姓名2、车主身份证号3、车辆品牌及型号。同时实现循环，可以连续修改

**4.3.2蒋宇欣设计实现**

**屏幕录入车辆牌照信息函数：**

依次调用录入汽车车辆号码函数、录入车主姓名函数、录入车主身份证号函数、录入车辆型号函数，将录入结果输出到屏幕中，选择录入信息是否正确，正确则继续，否则调用数据修改函数，之后对输入好的信息进行检验是否和系统文件中的车牌号码有重复，重复则重新录入，否则储存该信息到静态链表中，然后调用链式基数排序，再调用储存文件到系统文件函数。

**文件录入汽车牌照信息**

逐个从文件中读取车辆牌照信息，将读取结果输出到屏幕中，选择录入信息是否正确，正确则继续，否则调用数据修改函数，之后对输入好的信息进行检验是否和系统文件中的车牌号码有重复，重复则录入失败，继续下一个，直至文件全部录完，最后显示录入结果，进行链式基数排序，然后储存到系统文件。

录入汽车车辆号码函数：

实现只能再屏幕显示录入大写英文字母和数字，并对输入后的车牌号码进行检验，如果错误则提示重新输入。

录入车主姓名函数

录入车主身份证号函数

实现只能再屏幕显示录入数字（最后一位可以为X）

录入车辆型号函数

**4.3.3郑涵中设计实现**

**管理员更新系统信息**

**密码验证**：

读取储存在系统中的密码，屏幕读取管理员输入的密码，屏幕上只会显示\*，将两个字符串比较，相同则密码验证成功，否则失败；三次失败返回主界面。

**修改管理员密码**

连续输入两次新的密码，两个密码一样则修改成功，否则继续重新修改。

**数据修改函数：**

选择修改：1、车牌号码2、车主姓名3、车主身份证号4、车辆品牌及型号，返回修改后的信息

**显示系统文件信息**：全部显示和单个逐一显示

**储存到系统文件函数**

**读取系统文件函数**

**界面的实现：**

主界面、管理员更新系统信息界面；输出表头：汽车牌照信息查询表头、汽车车牌号码选择表头、管理员更新系统信息表头；输出空格；输出分界线。

1. **测试与调试**
   1. **个人测试** 
      1. **窦洹彰测试**

测试并修改了折半查找法、读取系统文件函数、数据修改函数、链式基数排序、系统储存文件信息备份、重置系统储存文件、汽车车牌号码选择。其中，

折半查找法：

循环折半时出现死循环，不断测试更改循环结束条件，不断测试是否能准确查找到指定信息。又在编写删除系统文件储存信息时发现调用返回信息的函数不能满足需要，又对折半查找法进行修改，编写新的函数，其返回值为在静态链表中的位置。

汽车车牌号码选择：

借鉴录入车牌号码函数，调试本部分的车牌号码是否准确录入，车牌号码是否符合实际，调用分块索引查找是否能正确反映相应信息。

链式基数排序：

因为有6位关键字，需要排序6次，而且关键字从A-Z（除I、O），0-9共34个关键字，关键字队列数量庞大，要采用循环结构，期初出现大量的错误，如：对循环条件的约束，队列连接的错误，需要不断调试修改。

* + 1. **蒋宇欣测试**

测试并修改了录入汽车车辆号码函数、录入车主姓名函数、录入车主身份证号函数、录入车辆型号函数。

其中：录入车辆型号函数：

进行了反复的修改，逐步实现了对车牌号码的实际贴近，即辽宁省车牌样式为\"辽XXXXXX\"，其中第一位为A,B,C,D,E,F,G,H,J,K,L,M,N,P,代表14个地区，第二位到第五为除I和O外的24个大写英文字母以及0~9数字组成，但英文字母不能超过2个，最后一位为只能是0~9的数字。

* + 1. **郑涵中测试**

测试并修改了储存到系统文件函数、分块索引查找、修改管理员密码、密码验证、显示系统文件信息。

其中分块索引查找：

把辽宁省14座城市的车牌首字母作为关键字，不断调试修改按关键字提取信息是否正确、是否有漏掉，查找信息是否能正确查找。

* 1. **组装与系统测试**

窦洹彰在汽车牌照信息查询函数中数据修改函数、链式基数排序组合起来测试与修改，不断调试发现是否有信息录入后未排序或未储存到系统文件中等。在文件录入汽车牌照信息中将储存到系统文件函数、链式基数排序、折半查找法组合起来测试与修改，不断调试发现是否有信息录入后未排序或未储存到系统文件中等。

蒋宇欣在屏幕录入和文件录入汽车牌照信息中录入汽车车辆号码函数、录入车主姓名函数、录入车主身份证号函数、录入车辆型号函数、将录入汽车车辆号码函数、储存到系统文件函数等组合起来测试与修改，测试是否发现信息修改错误。

郑涵中在修改系统文件储存信息中将录入汽车车辆号码函数、将储存到系统文件函数、数据修改函数等等组合起来测试与修改，反复测试是否有信息查找的错误。在删除系统文件储存信息中，将录入汽车车辆号码函数、储存到系统文件函数等结合起来，测试并修改删除后数据前移的错误。

最后窦洹彰将三人的程序组合成一个工程文件，加入了界面等函数，三人一起测试并修改。

**5.3 系统运行**

三人不断运行该系统，一项一项的检测是否在运行时发生错误或不足，发现不足或错误就细化到个人，是哪个人的任务就哪个人负责完善修改程序，最后直到每人都测试多遍发现无误。汽车牌照查询系统最终完成。

1. **课题总结** 
   1. **课题评价**

汽车牌照查询系统是一个内容比较多的课题。它要求我们要熟练掌握和运用C语言、C++、数据结构等相关知识。而且该课题包含的函数较多，要经常使用函数调用，所以要求组内人员要配合默契，制定格式化的函数命名，便于调用。该课题很大程度上提高了我们对C语言、C++运用的熟练度，提高了我们之间的合作能力，并对实际工作中大型软件结构化、模块化的设计有了一定的了解。总的来说，这次程序实践课题对我们学好编程语言有很大的帮助。

* 1. **团队协作**

这次的程序设计实践是以小组合作的形式共同完成一个课题，这就需要我们组员间有良好的配合。而且我们选中的汽车牌照查询系统课题，内容要求多，函数多，函数间互相调用频繁，这就进一步要求我们要有非常好的团队协作能力。可能开始的时候，我们效率不高，分工不是很明确，但随着我们之间互相磨合，不断了解，配合越来越默契，最后很好的完成了程序实践设计的任务。

* 1. **个人设计小结**

**6.3.1窦洹彰设计小结**

通过这次程序实验设计，在大一学年的C语言、C++的基础上，有了进一步熟练的运用。尤其在学习数据结构之后，通过将数据结构的算法实际应用，发现许多本以为很简单的算法，在实际编程运行时却有许多的错。所以通过这出课程设计，是我更加意识到，实践是检验真理的唯一标准。计算机的许多算法与理论，更是要通过实践来加深理解。只有不断的编程、修改错误，才能真正对一些算法和理论融会贯通。今后我会尽可能多的通过实际操作来深刻理解。

本次课程设计我认识到了自己平时的不足，对课本知识掌握不够扎实，不能灵活变通，写程序过程中考虑不够细致，严密。因此在今后的学习中，我将认真的学习课本知识，掌握课本中的知识点，并在此基础上学会灵活应用。认真上好专业实验课，多在实践中锻炼自己， 并且在写程序的过程中尽可能的要考虑周到，严密。

**6.3.2蒋宇欣设计小结**

通过本次数据结构实验课程设计，使我对数据结构和C++有了更深的认识，我负责编写的分块索引部分，使我对分块索引的了解更加透彻虽然在实验中发生了一些错误，面对了一些困难，但凭借自己的努力和同伴们的帮助，成功的解决了这些问题，更使我认识到团队合作的重要相互的帮助与鼓励是完成这次课题的最大动力，遇到问题时的讨论，使对问题的理解更加深刻。这次试验使我获益颇丰，希望结果令人满意。

**6.3.3郑涵中设计小结**

这次课程设计使我对C语言、C++以及数据结构有了更深刻的了解，增强了程序的编写能力，巩固了专业知识，对程序的模块化观念也又模糊逐渐变的清晰了。在程序的运行与调试过程中出现了很多错误，通过反复地复习课本上的相关知识，不停地修改与调试，最终战胜了“bug”。在调试过程中，我认识到了数据结构的灵活性与严谨性，同一个功能可以由不同的语句来实现，但编写程序时要特别注意细节方面的问题，因为一个小小的疏忽就能导致整个程序不能运行。

同时，这次课程设计使我认识到团队合作的重要性，我们三个讨论商量，一块学习，共同解决了好多问题，体会到共同努力，完成目标的喜悦。

**附录A-1：**

**课题程序设计分工**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 程序设计函数原型、类 | 功能说明 |
| 20162514 | 窦洹彰  （60%） | 静态链表  链式基数排序  系统文件  折半查找法  分块索引查找  汽车车牌号码选择  删除系统文件储存信息  系统存储文件备份  重置系统存储文件  修改系统文件储存信息  汽车牌照信息查询 | 数据结构  数据排序  系统文件读取  查找信息  查找信息  选择号码是否可用  删除系统文件信息  对车牌信息进行备份  清空文件中所有车牌信息  修改系统文件信息  查找汽车牌照信息 |
| 20162386 | 蒋宇欣  （30%） | 屏幕录入车辆信息函数  文件录入车辆信息函数  车辆牌照信息输入函数  显示系统文件信息 | 录入车辆信息  输入车辆牌照信息  单个或全部输出车牌信息 |
| 20164497 | 郑涵中  （10%） | 管理员更新系统信息  界面  系统文件  密码 | 对系统文件进行操作  界面显示  查找汽车牌照信息  密码验证及修改 |

**附录A-2：**

**课题报告分工**

窦洹彰（45%）蒋宇欣（35%）郑涵中（20%）

|  |  |  |
| --- | --- | --- |
| 章节 | 内容 | 完成人 |
| 1 课题概述 | 1.1 课题任务  1.2 课题原理  1.3 相关知识 | 窦洹彰  蒋宇欣  蒋宇欣 |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 | 郑涵中  郑涵中 |
| 3 方案设计 | 3.1 总体功能设计  3.2 数据结构设计  3.3 函数原型设计  3.4 主算法设计  3.5 用户界面设计  3.6 输入输出设计 | 窦洹彰 蒋宇欣 郑涵中  窦洹彰 蒋宇欣 郑涵中  窦洹彰 蒋宇欣 郑涵中  窦洹彰 蒋宇欣 郑涵中  窦洹彰 蒋宇欣 郑涵中  窦洹彰 蒋宇欣 郑涵中 |
| 4 方案实现 | 4.1 开发环境与工具  4.2 程序设计关键技术  4.3 个人设计实现  4.3.1窦洹彰设计实现  4.3.2蒋宇欣设计实现  4.3.3郑涵中设计实现 | 窦洹彰  窦洹彰  窦洹彰  蒋宇欣  郑涵中 |
| 5 测试与调试 | 5.1 个人测试  5.1.1窦洹彰测试  5.1.2蒋宇欣测试  5.1.3郑涵中测试  5.2 组装与系统测试  5.3 系统运行 | 窦洹彰  蒋宇欣  郑涵中  窦洹彰  窦洹彰 |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 个人设计心得  6.3.1窦洹彰设计心得  6.3.2蒋宇欣设计心得  6.3.3郑涵中设计心得 | 窦洹彰  窦洹彰  窦洹彰  蒋宇欣  郑涵中 |

**附录B-2：程序源代码**

#ifndef ADMIN\_H\_INCLUDED

#define ADMIN\_H\_INCLUDED

void Header\_administrator();

void Administrator\_interfasce();

void Space(int n);

void Line();

void Enter\_screen();

void Enter\_file();

void Amend();

void Delete();

void Show();

void Backups();

void Clearfile();

void Password\_Change();

#endif // ADMIN\_H\_INCLUDED

#ifndef CHOICE\_H\_INCLUDED

#define CHOICE\_H\_INCLUDED

void Header\_choose();

void Line();

void Space(int n);

bool Search\_Block(StaticLinkList S,int c,ElemType elem,char ch);

#endif // CHOICE\_H\_INCLUDED

#ifndef INPUT\_H\_INCLUDED

#define INPUT\_H\_INCLUDED

ElemType Input\_plate(ElemType elem,int cls);

ElemType Input\_name(ElemType elem);

ElemType Input\_ID(ElemType elem);

ElemType Input\_car(ElemType elem);

ElemType Revise(ElemType elem);

#endif // INPUT\_H\_INCLUDED

#ifndef MAIN\_H\_INCLUDED

#define MAIN\_H\_INCLUDED

void Header\_main();

void Main\_interfasce();

void Space(int n);

void Line();

int Read(StaticLinkList S);

void Search(StaticLinkList S,int c);

void Choose(StaticLinkList S,int c);

void Administrator();

bool PassLink();

#endif // MAIN\_H\_INCLUDED

#ifndef OTHER\_H\_INCLUDED

#define OTHER\_H\_INCLUDED

#include<iostream>

using namespace std;

void Header\_query();

void Store(StaticLinkList S,int s);

void Header\_administrator();

void Header\_choose();

void Space(int n);

void Line();

void Sort(StaticLinkList S,int c);

ElemType Revise(ElemType elem);

ElemType ScreenInput(ElemType elem);

ElemType Revise(ElemType elem);

ElemType Search\_Binary(StaticLinkList S,int c,ElemType elem);

ElemType Input\_plate(ElemType elem,int cls);

int Search\_Binary\_Admin(StaticLinkList S,int c,ElemType elem);

int Read(StaticLinkList S);

void mohusearch(string a)

{

int i,k;

string s;

ofstream fout;

freopen("licence\_plate.txt","r",stdin);

while (cin>>s)

{

cin>>s;

k=1;

for (i=0;i<a.size();i++)

{

if (s[i]!=a[i]) k=0;

}

if (k=1)

{

cout<<s<<endl;

}

cin>>s;

cin>>s;

cin>>s;

}

fclose("licence\_plate.txt");

}

#endif // OTHER\_H\_INCLUDED

#ifndef QUERY\_H\_INCLUDED

#define QUERY\_H\_INCLUDED

void Header\_query();

void Space(int n);

void Line();

ElemType Input\_plate(ElemType elem,int cls);

ElemType Search\_Binary(StaticLinkList S,int c,ElemType elem);

#endif // QUERY\_H\_INCLUDED

#ifndef STATICLINKLIST\_H\_INCLUDED

#define STATICLINKLIST\_H\_INCLUDED

#define MAXSIZE 1000

class ElemType

{

public:

char plate[7];

char name[20];

char ID[19];

char car[100];

};

typedef struct Node

{

ElemType data;

int cur;

}StaticLinkList[MAXSIZE];

#endif // STATICLINKLIST\_H\_INCLUDED

#include<stdio.h>

#include<iostream>

#include<stdlib.h>

#include"StaticLinkList.h"

#include"Main.h"

using namespace std;

int main()

{

StaticLinkList SLL;

int s=0;

int i=100;

while(i!=0)

{

system("cls");

Header\_main();//主界面表头

Main\_interfasce();//主界面内容

s=Read(SLL);//读取系统数据

printf("请选择：\n");

scanf("%d,",&i);

while(i!=0&&i!=1&&i!=2&&i!=9)

{

printf("输入的好像不正确哟！\n");

printf("请输入选择 1、2、0；\n");

printf("请选择：\n");

fflush(stdin);

scanf("%d",&i);

}

switch(i)

{

case 1:Search(SLL,s);break;

case 2:Choose(SLL,s);break;

case 9:

if(PassLink())

Administrator();

break;

case 0:

system("cls");

Header\_main();

cout<<endl;

Space(10);

cout<<"退出系统！"<<endl;

cout<<endl<<endl;

Line();

break;

}

}

return 0;

}

#include<iostream>

#include<cstdio>

#include<conio.h>

#include<cstring>

#include<cstdlib>

#include<fstream>

#include"StaticLinkList.h"

#include"Other.h"

#define M 15

using namespace std;

void Password\_Change()

{

char ch1,ch2,password[M],password1[M],password2[M];

int i=0,j=0,x;

ifstream fin;

fin.open("password.txt",ios::in);

fin>>password;

fin.close();

do{

i=0,j=0;

system("cls");

Header\_administrator();

printf("\n\n");

Space(20);

printf("请输入新密码");

while((ch1=getch())!='\r' && i<=M)

{

if(ch1=='\b')

{

if(i>0)

{

i--;

printf("\b \b");// 密码支持退格的实现

}

else

putchar(7);

}

else

{

password1[i++]=ch1;

printf("\*");

}

}

password1[i]='\0';

printf("\n");

Space(20);

printf("请确认新密码");

while((ch2=getch())!='\r' && j<=M)

{

if(ch2=='\b')

{

if(j>0)

{

j--;

printf("\b \b");// 密码支持退格的实现

}

else

putchar(7);

}

else

{

password2[j++]=ch2;

printf("\*");

}

}

password2[j]='\0';

if(strcmp(password1,password2)==0)

{

x=0;

printf("\n");

Space(20);

printf("密码修改成功！\n");

system("pause");

}

else

{

x=1;

printf("\n");

Space(15);

printf("两次输入密码不同，请再设置一遍\n");

system("pause");

}

}while(x);

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

{

password[i]=password1[i];

}

ofstream fout;

fout.open("password.txt",ios::out);

fout<<password;

fout.close();

}

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include"StaticLinkList.h"

#include"Other.h"

#include"Input.h"

using namespace std;

void Amend()

{

StaticLinkList S;

ElemType amend;

int i=1;

int c=0;

int j,k,t,test;

while(i==1)

{

i=0;

test=0;

t=0;

c=Read(S);//读取系统文件数据

system("cls");

Header\_administrator();

printf("修改汽车牌照的信息\n");

amend=Input\_plate(amend,9);//接收要修改信息的车牌号码

system("cls");

Header\_administrator();

printf("\n");

Space(10);

cout<<"车牌辽"<<amend.plate<<"信息查找中..."<<endl;

printf("\n");

Line();

test=Search\_Binary\_Admin(S,c,amend);//通过车牌号码查找信息

system("cls");

Header\_administrator();

printf("查找结果...：\n");

if(test!=0)

{

k=1;

while(k==1)

{

system("cls");

Header\_administrator();

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",S[test].data.plate[j]);

printf("\n车主姓名：\t\t");

printf("%s",S[test].data.name);

printf("\n车主身份证号：\t\t");

printf("%s",S[test].data.ID);

printf("\n车辆品牌及型号：\t");

printf("%s",S[test].data.car);

printf("\n");

Line();

printf("选择：1.\t\t修改该信息 0.\t\t返回\n");

printf("请选择：\n");

scanf("%d",&k);

while(k!=0&&k!=1)

{

printf("请走心地输入一遍数据！\n");

printf("选择：1.\t\t修改该信息 0.\t\t返回\n");

printf("请选择：\n");

scanf("%d",&k);

}

if(k==1)

{

printf("\n请选择要修改的信息\n");

printf("1.车主姓名\n");

printf("2.身份证号\n");

printf("3.车的品牌和型号\n");

printf("0.返回\n");

printf("请选择：\n");

scanf("%d",&j);

while(j!=0&&j!=1&&j!=2&&j!=3)

{

printf("输入的选择不正确啊\n\n请选择1、2、3、4、0\n请选择：\n");

fflush(stdin);

scanf("%d",&j);

}

if(j!=0)

{

printf("请输入新的信息哟\n");

t++;

}

switch(j)

{

case 1:S[test].data=Input\_name(S[test].data);break;

case 2:S[test].data=Input\_ID(S[test].data);break;

case 3:S[test].data=Input\_car(S[test].data);break;

case 0:

if(t==0)

{

i=1;

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("！！！您没有修改信息呀！\n");

printf("\n");

Line();

system("pause");

}

else

{

printf("\n");

Line();

}

break;

}

if(i!=1)

{

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("恭喜！信息修改成功！\n");

printf("\n");

Line();

i=0;

system("pause");

}

}

else

{

if(t==0)

{

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("一点也没有修改信息的痕迹！\n");

printf("\n\n");

Line();

system("pause");

}

else

{

printf("\n");

Line();

}

}

}

}

else

{

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("未找到该车牌的信息哟~\n");

printf("\n");

Line();

}

printf("\n");

printf("请选择~：1.\t继续修改\t0.\t返回上一界面\n");

printf("请选择：\n");

scanf("%d",&i);

while(i!=0&&i!=1)

{

printf("输入的好像不正确哟，请认真一点输入：\n");

printf("请选择~：1.\t继续修改\t0.\t返回上一界面\n");

printf("请选择：\n");

fflush(stdin);

scanf("%d",&i);

}

}

Store(S,c);//储存修改后的信息

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员的系统界面了哟~\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<cstring>

#include"StaticLinkList.h"

using namespace std;

bool Search\_Block(StaticLinkList S,int c,ElemType elem,char ch)

{

StaticLinkList SBlock;

int i,j,k;

int block=0;

char str[10]={"0"};

for(i=0;i<MAXSIZE;i++)//初始化分块查找链表

{

for(j=0;j<6;j++)

SBlock[i].data.plate[j]='0';

SBlock[i].data.plate[6]='\0';

strcpy(SBlock[i].data.name,str);

for(j=0;j<18;j++)

SBlock[i].data.ID[j]='0';

SBlock[i].data.ID[18]='\0';

strcpy(SBlock[i].data.car,str);

SBlock[i].cur=i+1;

}

k=1;

for(i=0;i<=c;i++)//提取索引信息

{

if(S[i].data.plate[0]==ch)

{

block++;

for(j=0;j<6;j++)

SBlock[k].data.plate[j]=S[i].data.plate[j];

SBlock[k-1].cur=k;

k++;

}

}

k=0;

i=1;

while(k==0&&i<=block) //顺序查找

{

k=1;

for(j=0;j<6;j++)

{

if(elem.plate[j]!=SBlock[i].data.plate[j])

k=0;

}

i++;

}

if(k==1) return true;

else return false;

}

#include<iostream>

#include<cstdlib>

#include<cstdio>

#include<cstring>

#include"StaticLinkList.h"

#include"Other.h"

using namespace std;

void Delete()

{

StaticLinkList S,Snew;

ElemType del,Test;

char str[10]={"0"};

int c=0;

int i=1;

int j,k,test;

while(i==1)

{

k=0;

c=Read(S);

system("cls");

Header\_administrator();

printf("这里是要删除车牌的信息\n");

del=Input\_plate(del,9);//接收要删除信息的车牌号码

system("cls");

Header\_administrator();

printf("\n");

Space(10);

cout<<"车牌辽"<<del.plate<<"信息查找中..."<<endl;

printf("\n");

Line();

Test=del;

test=Search\_Binary\_Admin(S,c,Test);

strcpy(Test.plate,S[test].data.plate);

strcpy(Test.name,S[test].data.name);

strcpy(Test.ID,S[test].data.ID);

strcpy(Test.car,S[test].data.car);

system("cls");

Header\_administrator();

printf("\n查找结果是\n");

if(test!=0)

{

system("cls");

Header\_administrator();

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",Test.plate[j]);

printf("\n车主姓名：\t\t");

printf("%s",Test.name);

printf("\n车主身份证号：\t\t");

printf("%s",Test.ID);

printf("\n车辆品牌及型号：\t");

printf("%s",Test.car);

printf("\n");

Line();

printf("\n请选择：1.\t\t删除该信息 0.\t\t返回\n");

printf("\n请选择\n");

scanf("%d",&k);

while(k!=0&&k!=1)

{

printf("\n输入的不正确哟~\n");

printf("\n请选择：1.\t\t删除该信息 0.\t\t返回\n");

printf("请选择\n");

fflush(stdin);

scanf("%d",&k);

}

if(k==1)

{

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("信息删除中\n");

printf("\n");

Line();

if(test==c)

S[test-1].cur=0;

else

S[test-1].cur=test+1;

c--;

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("\n系统文件储存中\n");

printf("\n");

Line();

for(i=0;i<MAXSIZE;i++)//初始化链表

{

for(j=0;j<6;j++)

Snew[i].data.plate[j]='0';

Snew[i].data.plate[6]='\0';

strcpy(Snew[i].data.name,str);

for(j=0;j<18;j++)

Snew[i].data.ID[j]='0';

Snew[i].data.ID[18]='\0';

strcpy(Snew[i].data.car,str);

Snew[i].cur=i+1;

}

k=S[0].cur;

i=1;

Snew[0].cur=S[0].cur;

while(k!=0)//将删除后的链表复制

{

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

Snew[i].data.plate[j]=S[k].data.plate[j];

strcpy(Snew[i].data.name,S[k].data.name);

for(j=0;j<19;j++)

Snew[i].data.ID[j]=S[k].data.ID[j];

strcpy(Snew[i].data.car,S[k].data.car);

Snew[i].cur=i+1;

k=S[k].cur;

i++;

}

Snew[c].cur=0;

Store(Snew,c);//储存删除后新的链表

system("cls");

Header\_administrator();

printf("\n");

Space(5);

printf("\n信息删除成功\n");

}

else

{

printf("\n");

Space(5);

printf("\n未删除该信息\n");

}

}

else

{

Space(5);

printf("未找到这个车牌的信息\n");

}

printf("\n");

Line();

printf("\n");

system("pause");

printf("选择:1.\t\t继续删除\t 0.\t\t返回上一界面\n");

printf("请选择\n");

scanf("%d",&i);

while(i!=0&&i!=1)

{

printf("输入的数据不正确啊~！\n");

printf("选择:1.\t\t继续删除\t 0.\t\t返回上一界面\n");

printf("请选择\n");

fflush(stdin);

scanf("%d",&i);

}

}

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员管理界面\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<fstream>

#include"StaticLinkList.h"

using namespace std;

void Store(StaticLinkList S,int s)

{

int i;

ofstream fout;

fout.open("license\_plate.txt",ios::out);

fout<<s<<" ";

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

{

fout<<S[i].data.plate<<" "<<S[i].data.name<<" "<<S[i].data.ID<<" "<<S[i].data.car<<" "<<S[i].cur<<" ";

}

fout.close();

}

#include<iostream>

#include<cstdio>

#include<cstring>

#include<cstdlib>

#include<conio.h>

#include<fstream>

#include"StaticLinkList.h"

#include"Other.h"

#define M 15 //设置密码位数为(M-1)

using namespace std;

bool PassLink()

{

char password[M];

int n=0;

int x=0;

ifstream fin;

fin.open("password.txt",ios::in);

fin>>password;

fin.close();

while(n<=3&&x==0)

{

n++;

if(n<=3)

{

char ch,password0[M];

int i=0;

system("cls");

Header\_administrator();

printf("\n\n");

Space(20);

printf("请输入密码:");

while((ch=getch())!='\r' && i<=M)

{

if(ch=='\b')

{

if(i>0)

{

i--;

printf("\b \b");// 密码支持退格的实现

}

else

putchar(7);

}

else

{

password0[i++]=ch;

printf("\*");

}

}

password0[i]='\0';

if(!strcmp(password0,password))

{

system("cls");

Header\_administrator();

printf("\n\n");

Space(5);

printf("密码正确 请继续操作\n");

system("pause");

x=1;

}

else

{

printf("\n");

Space(15);

printf("密码错误 您还有%d次机会\n",3-n);

system("pause");

}

}

else

{

system("cls");

Header\_administrator();

printf("\n\n");

Space(5);

printf("密码均错误 无法进入管理员系统!\n");

system("pause");

x=0;

}

}

return x;

}

#include<iostream>

#include<cstdlib>

#include<fstream>

#include<cstring>

#include<cstdio>

#include"StaticLinkList.h"

#include"Other.h"

#include"input.h"

using namespace std;

void Enter\_screen()

{

StaticLinkList S;

ElemType elem,check;

StaticLinkList Extra;

char file[100];

int e=0;

int c=0;

int i,j,k;

char str[10]={"0"};

for(i=0;i<MAXSIZE;i++)//初始化链表

{

for(j=0;j<6;j++)

Extra[i].data.plate[j]='0';

Extra[i].data.plate[6]='\0';

strcpy(Extra[i].data.name,str);

for(j=0;j<18;j++)

Extra[i].data.ID[j]='0';

Extra[i].data.ID[18]='\0';

strcpy(Extra[i].data.car,str);

Extra[i].cur=i+1;

}

i=1;

while(i==1)

{

c=Read(S);

system("cls");

Header\_administrator();

printf("开始录入车牌的信息\n");

k=2;

elem=Input\_plate(elem,9);//录入车牌号码

elem=Input\_name(elem);//录入车主姓名

elem=Input\_ID(elem);//录入车主身份证号

elem=Input\_car(elem);//录入车辆型号

while(k==2)

{

system("cls");

Header\_administrator();

j=0;

printf("车牌号：\t\n");

for(j=0;j<6;j++)

printf("%c",elem.plate[j]);

printf("\n车主姓名：\t\t");

cout<<elem.name<<endl;

printf("车主身份证号：\t\t");

printf("%s",elem.ID);

printf("\n车辆品牌及型号：\t");

cout<<elem.car<<endl;

printf("\n请仔细核对上面的信息是否有误哟~\n1.准确无误\t2.有错误\n");

printf("请选择：\n");

scanf("%d",&k);

while(k!=1&&k!=2)

{

printf("输入的不正确哟~\n");

printf("\n请仔细核对上面的信息是否有误哟~\n1.准确无误\t2.有错误\n");

fflush(stdin);

scanf("%d",&k);

}

if(k==2)

{

elem=Revise(elem);//修改已经输入的信息

}

else

{

check=elem;

check=Search\_Binary(S,c,check);//检查是否与系统文件里的信息重复

if(check.plate[0]!=0)

{

system("cls");

Header\_administrator();

printf("对不起,此车牌号码与之前系统文件里的车牌号码重复！\n");

cout<<"输入的车牌号码为辽"<<elem.plate<<endl;

printf("请仔细核对号码是否有误！\n");

printf("请输入新的号码\n");

elem=Input\_plate(elem,9);

k=2;

}

else

{

S[c].cur=c+1;

c++;

strcpy(S[c].data.plate,elem.plate);

strcpy(S[c].data.name,elem.name);

strcpy(S[c].data.ID,elem.ID);

strcpy(S[c].data.car,elem.car);

S[c].cur=0;

Extra[e].cur=e+1;

e++;

strcpy(Extra[e].data.plate,elem.plate);

strcpy(Extra[e].data.name,elem.name);

strcpy(Extra[e].data.ID,elem.ID);

strcpy(Extra[e].data.car,elem.car);

Extra[e].cur=0;

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("新汽车牌照信息录入到系统文件中...");

printf("\n");

Line();

Sort(S,c);//链式基数排序

Store(S,c);//存储信息到系统文件

system("cls");

Header\_administrator();

printf("\n");

Space(20);

printf("信息录入成功\n");

printf("\n");

Line();

}

}

}

printf("是否继续录入车辆牌照信息\n");

printf("1.继续录入\t0.退出\n");

printf("请选择：\n");

scanf("%d",&i);

while(i!=0&&i!=1)

{

printf("输入的不正确啊\n");

printf("是否继续录入车辆牌照信息\n");

printf("1.继续录入\t0.退出\n");

printf("请选择\n");

fflush(stdin);

scanf("%d",&i);

}

}

system("cls");

Header\_administrator();

printf("\n");

printf("对新录入的汽车牌照信息是否备份？\n");

printf("1.备份\t0.不备份\n");

scanf("%d",&i);

while(i!=1&&i!=0)

{

printf("输入的有点问题啊\n");

printf("1.备份\t0.不备份\n");

printf("请选择\n");

fflush(stdin);

scanf("%d",&i);;

}

if(i==1)//将新输入的信息输入到其他文件中

{

system("cls");

Header\_administrator();

printf("\n");

ofstream fout;

printf("对输入的文件进行备份操作\n");

printf("请输入文件路径\n");

cin>>file;

fout.open(file,ios::out);

while(fout.fail())

{

printf("小哥哥输入的地址有误哟~\n");

printf("请输入正确的地址啦~\n");

cin>>file;

fout.open(file,ios::out);

}

fout<<e<<" ";

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

{

fout<<Extra[i].data.plate<<" "<<Extra[i].data.name<<" "<<Extra[i].data.ID<<" "<<Extra[i].data.car<<" "<<Extra[i].cur<<" ";

}

fout.close();

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("新牌照成功录入了哟~\n");

printf("\n");

Line();

system("pause");

}

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员操作系统\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<cstring>

#include"StaticLinkList.h"

ElemType Search\_Binary(StaticLinkList S,int c,ElemType elem)

{

int low,mid,high;

int i=0;

int j;

low=1;

high=c;

mid=(low+high)/2;

while((i==0)&&(low<=high))

{

j=0;

while((S[mid].data.plate[j]==elem.plate[j])&&(j<6))

j++;

if(j==6) i=1;

else

{

if(S[mid].data.plate[j]<elem.plate[j]) low=mid+1;

else high=mid-1;

mid=(low+high)/2;

}

}

if(i==0)

{

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

{

elem.plate[j]=0;

}

return elem;

}

else

{

strcpy(elem.plate,S[mid].data.plate);

strcpy(elem.name,S[mid].data.name);

strcpy(elem.ID,S[mid].data.ID);

strcpy(elem.car,S[mid].data.car);

return elem;

}

}

int Search\_Binary\_Admin(StaticLinkList S,int c,ElemType elem)

{

int low,mid,high;

int i=0;

int j;

low=1;

high=c;

mid=(low+high)/2;

while((i==0)&&(low<=high))

{

j=0;

while((S[mid].data.plate[j]==elem.plate[j])&&(j<6))

j++;

if(j==6) i=1;

else

{

if(S[mid].data.plate[j]<elem.plate[j]) low=mid+1;

else high=mid-1;

mid=(low+high)/2;

}

}

if(i==0)

return 0;

else

return mid;

}

#include<iostream>

#include<cstdio>

#include<conio.h>

#include<cstdlib>

#include"StaticLinkList.h"

#include"Input.h"

#include"Other.h"

using namespace std;

ElemType Revise(ElemType elem)

{

int i=0,j;

printf("\n\n请选择要修改的信息\n");

printf("1.车牌号码\n");

printf("2.车主姓名\n");

printf("3.车主身份证号\n");

printf("4.车辆品牌以及型号\n");

printf("0.完全正确无需修改\n");

printf("请选择\n");

scanf("%d",&j);

while(j<0||j>5)

{

printf("输入的选择不正确哟~\n\n请选择1、2、3、4、0输入\n请选择：\n");

scanf("%d",&j);

fflush(stdin);

}

system("cls");

Header\_administrator();

if(j!=0)

printf("请输入新的信息\n");

switch(j)

{

case 1:elem=Input\_plate(elem,9);break;

case 2:elem=Input\_name(elem);break;

case 3:elem=Input\_ID(elem);break;

case 4:elem=Input\_car(elem);break;

case 0:i=1;printf("没有修改信息\n");break;

}

if(i!=1)

{

printf("信息修改成功\n");

i=0;

}

system("pause");

return elem;

}

#include<iostream>

#include<cstdlib>

#include<fstream>

#include<cstring>

#include<cstdio>

#include"StaticLinkList.h"

#include"Other.h"

using namespace std;

void Enter\_file()

{

StaticLinkList S;

ElemType elem,check;

int n=0;

int c=0;

int m=0;

int unused,i,j,k;

char file[100];

c=Read(S);

ifstream fin;

system("cls");

Header\_administrator();

printf("请输入要读取的文件路径\n");

cin>>file;

fin.open(file,ios::in);

while(fin.fail())

{

system("cls");

Header\_administrator();

printf("文件打开失败\n");

printf("请输入正确的文件路径:\n");

cin>>file;

fin.open(file,ios::in);

}

system("cls");

Header\_administrator();

printf("文件打开成功啦~\n");

printf("读取数据信息如下所示：\n");

fin>>n;

fin>>elem.plate;

fin>>elem.name;

fin>>elem.ID;

fin>>elem.car;

fin>>unused;

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

{

fin>>elem.plate;

fin>>elem.name;

fin>>elem.ID;

fin>>elem.car;

fin>>unused;

k=2;

while(k==2)

{

system("cls");

Header\_administrator();

cout<<"正在录入文件中第"<<i<<"个信息，已成功录入"<<m<<"个信息，共"<<n<<"个信息"<<endl<<endl;

j=0;

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",elem.plate[j]);

printf("\n车主姓名：\t\t");

cout<<elem.name<<endl;

printf("车主身份证号：\t\t");

printf("%s",elem.ID);

printf("\n车辆品牌及型号：\t");

cout<<elem.car<<endl;

printf("请认真地再审查一遍上述信息是否有误哟~\n1:没有错误\t2：有错误\n请选择:");

scanf("%d",&k);

while(k!=1&&k!=2)

{

printf("输入的信息不正确哟~\n");

printf("请认真地再审查一遍上述信息是否有误哟~\n1:没有错误\t2：有错误\n请选择:");

fflush(stdin);

scanf("%d",&k);

}

if(k==2)

{

elem=Revise(elem);

}

else

{

check=elem;

check=Search\_Binary(S,c,check);

if(check.plate[0]!=0)

{

system("cls");

Header\_administrator();

printf("\n对不起,此车牌号码与之前系统文件里的车牌号码重复！\n");

printf("此汽车牌照信息无法录入到系统文件！\n");

printf("\n");

Line();

system("pause");

}

else

{

S[c].cur=c+1;

c++;

strcpy(S[c].data.plate,elem.plate);

strcpy(S[c].data.name,elem.name);

strcpy(S[c].data.ID,elem.ID);

strcpy(S[c].data.car,elem.car);

S[c].cur=0;

m++;

system("cls");

Header\_administrator();

printf("\n");

Space(20);

printf("信息录入成功\n");

printf("\n");

Line();

system("pause");

}

}

}

}

fin.close();

system("cls");

Header\_administrator();

printf("\n");

Space(5);

cout<<"此文件共"<<n<<"个信息,成功录入"<<m<<"个信息"<<endl;

printf("\n");

Line();

if(m!=0)

{

printf("即将更新系统存储文件\n");

system("pause");

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("更新系统存储文件中...");

printf("\n");

Line();

Sort(S,c);

Store(S,c);

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("系统存储文件更新成功！\n");

printf("\n");

Line();

printf("即将返回管理员更新系统信息界面\n");

system("pause");

}

else

{

printf("未录入新的汽车牌照信息\n");

printf("即将返回管理员更新系统信息界面\n");

system("pause");

}

}

#include<iostream>

#include<cstdlib>

#include<cstdio>

#include"StaticLinkList.h"

#include"Other.h"

using namespace std;

void Show()

{

StaticLinkList S;

string s1;

int i,j,k;

int c=0;

c=Read(S);

system("cls");

Header\_administrator();//表头和界面

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"1 单个显示汽车牌照信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"2 全部显示汽车牌照信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"0 返回管理员更新系统信息界面";

Space(12);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

Line();

cout<<endl;

cout<<"请选择：";

cin>>s1;

i=3;

if (s1.size()==1)

{

if (s1[0]=='0') i=0;

if (s1[0]=='1') i=1;

if (s1[0]=='2') i=2;

}

while(i!=0&&i!=1&&i!=2)

{

printf("输入的选择并不正确\n");

printf("请选择1、2或者0\n");

cout<<endl<<"请选择：";

cin>>i;

}

switch(i)

{

case 1:

if(c==0)

{

printf("没有信息\n");

}

system("pause");

for(k=1;k<=c;k++)

{

system("cls");

Header\_administrator();

cout<<endl<<"当前显示第"<<k<<"个信息，共"<<c<<"个信息."<<endl<<endl;

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",S[k].data.plate[j]);

printf("\n车主姓名：\t\t");

cout<<S[k].data.name<<endl;

printf("车主身份证号：\t\t");

printf("%s",S[k].data.ID);

printf("\n车辆品牌及型号：\t");

cout<<S[k].data.car<<endl;

printf("\n");

Line();

cout<<"1、继续显示下一辆车牌信息"<<endl<<"0、返回管理员更新系统信息界面"<<endl;

int h;

h=2;

cin>>s1;

if (s1.size()==1)

{

if (s1[0]=='0') h=0;

if (s1[0]=='1') h=1;

}

while(!(h==0||h==1))

{

system("cls");

Header\_administrator();

cout<<endl<<"当前显示第"<<k<<"个信息，共"<<c<<"个信息."<<endl<<endl;

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",S[k].data.plate[j]);

printf("\n车主姓名：\t\t");

cout<<S[k].data.name<<endl;

printf("车主身份证号：\t\t");

printf("%s",S[k].data.ID);

printf("\n车辆品牌及型号：\t");

cout<<S[k].data.car<<endl;

printf("\n");

Line();

cout<<"1、继续显示下一辆车牌信息"<<endl<<"0、返回管理员更新系统信息界面"<<endl;

cout<<"输入的不对请重新输入："<<endl;

cin>>h;

}

if(h==0) break;

}

break;

case 2:

system("cls");

Header\_administrator();

for(k=1;k<=c;k++)

{

printf("\n");

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",S[k].data.plate[j]);

printf("\n车主姓名：\t\t");

cout<<S[k].data.name<<endl;

printf("车主身份证号：\t\t");

printf("%s",S[k].data.ID);

printf("\n车辆品牌及型号：\t");

cout<<S[k].data.car<<endl;

}

Line();

cout<<"共"<<c<<"个信息！"<<endl;

system("pause");

break;

case 0:

break;

}

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员更新系统信息界面！\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<cstdlib>

#include<cstdio>

#include"StaticLinkList.h"

#include"Query.h"

using namespace std;

void Search(StaticLinkList S,int c)

{

ElemType Test;

int i=1;

int j;

string s1;

while(i==1)

{

system("cls");

Header\_query();

Test=Input\_plate(Test,1);//接收要查找汽车车牌号码

system("cls");

Header\_query();

printf("\n");

Space(10);

cout<<"车牌辽"<<Test.plate<<"信息查找中..."<<endl;

printf("\n");

Line();

Test=Search\_Binary(S,c,Test);

system("cls");

Header\_query();

printf("查找结果：\n");

if(Test.plate[0]!=0)

{

printf("车牌号：\t\t辽");

for(j=0;j<6;j++)

printf("%c",Test.plate[j]);

printf("\n车主姓名：\t\t");

printf("%s",Test.name);

printf("\n车主身份证号：\t\t");

printf("%s",Test.ID);

printf("\n车辆品牌及型号：\t");

printf("%s",Test.car);

}

else

{

Space(5);

printf("未找到该车牌信息！\n");

}

printf("\n");

Line();

printf("\n");

system("pause");

printf("选择：1.\t继续查询\t0.\t返回主界面\n");

printf("请选择：\n");

cin>>s1;

i=2;

if (s1[0]=='0')i=0;

if (s1[0]=='1')i=1;

if (s1.size()!=1) i=2;

fflush(stdin);

while(i!=0&&i!=1)

{

printf("输入的选择不正确！\n");

printf("选择：1.\t继续查询\t0.\t返回主界面\n");

printf("请选择：\n");

fflush(stdin);

scanf("%d",&i);

}

}

system("cls");

Header\_query();

printf("\n");

Space(20);

printf("即将返回主菜单！\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<cstdlib>

#include<cstdio>

#include<conio.h>

#include"StaticLinkList.h"

#include"Choice.h"

#include<stdio.h>

using namespace std;

void Choose(StaticLinkList S,int c)

{

ElemType Test;

int t=1;

int i,j,k,z;

char ch,th;

while(t==1)

{

system("cls");

Header\_choose();

printf("选择汽车所在城市：\n");

cout<<'\t'<<"1 沈阳市"<<'\t'<<"2 大连市"<<'\t'<<"3 鞍山市"<<endl;

cout<<'\t'<<"4 抚顺市"<<'\t'<<"5 本溪市"<<'\t'<<"6 丹东市"<<endl;

cout<<'\t'<<"7 锦州市"<<'\t'<<"8 营口市"<<'\t'<<"9 阜新市"<<endl;

cout<<'\t'<<"10 辽阳市"<<'\t'<<"11 盘锦市"<<'\t'<<"12 铁岭市"<<endl;

cout<<'\t'<<"13 朝阳市"<<'\t'<<"14 葫芦岛市"<<endl;

cout<<endl<<'\t'<<"0 返回主菜单"<<endl;

//printf("\t0 返回主菜单\n");

printf("\n");

Line();

printf("请选择：\n");

scanf("%d",&k);

while(k<0||k>14)

{

printf("输入的选择不正确！\n");

printf("请选择1~14或者0\n");

printf("请选择：\n");

fflush(stdin);

scanf("%d",&k);

}

switch(k)

{

case 1:th='A';break;

case 2:th='B';break;

case 3:th='C';break;

case 4:th='D';break;

case 5:th='E';break;

case 6:th='F';break;

case 7:th='G';break;

case 8:th='H';break;

case 9:th='J';break;

case 10:th='K';break;

case 11:th='L';break;

case 12:th='M';break;

case 13:th='N';break;

case 14:th='P';break;

case 0:t=0;break;

}

if(k!=0)

{

j=0;

while(j==0)//接收要选择的车牌号码

{

system("cls");

Header\_choose();

for(i=0;i<6;i++)

Test.plate[i]='O';

Test.plate[6]='\0';

cout<<"请输入你想选择的汽车车牌号码："<<endl;

printf("辽%c",th);

Test.plate[0]=th;

i=1;

while((ch=getch())!='\r')

{

if(ch=='\b')

{

if(i>1)

{

i--;

printf("\b \b");// 退格的实现

Test.plate[i+1]='O';

}

else

putchar(7);

}

else

{

if(i!=6)

{

if((ch>='A'&&ch<='Z')||(ch>='0'&&ch<='9'))

{

Test.plate[i++]=ch;

cout<<ch;

}

else putchar(7);

}

}

}

Test.plate[6]='\0';

j=1;

z=0;

for(i=0;i<6;i++)

{

ch=Test.plate[i];

if(i==0)

{

if(ch>='A'&&ch<='P'&&ch!='I'&&ch!='O') j=j;

else j=0;

}

if(i>0&&i<5)

{

if(ch>='A'&&ch<='Z'&&ch!='I'&&ch!='O')

{

j=j;

z++;

}

else if(ch>='0'&&ch<='9') j=j;

else j=0;

}

if(i==5)

{

if(ch>='0'&&ch<='9') j=j;

else j=0;

}

if(z>2) j=0;

}

if(j==0)

{

printf("输入的车牌号不符合规定！\n");

printf("注意：辽宁省车牌样式为\"辽XXXXXX\"\n");

printf("其中第一位为A,B,C,D,E,F,G,H,J,K,L,M,N,P,代表14个地区\n");

printf("第二位到第五为除I和O外的24个大写英文字母以及0~9数字组成，\n");

printf("但英文字母不能超过2个\n");

printf("最后一位为只能是0~9的数字\n");

printf("请重新输入车牌号!\n");

system("pause");

}

}

system("cls");

Header\_choose();

printf("\n");

Space(20);

printf("车牌号码查找中...\n");

printf("\n");

Line();

if(!Search\_Block(S,c,Test,th))

{

system("cls");

Header\_choose();

printf("\n");

Space(5);

cout<<"恭喜您！车牌号码 辽"<<Test.plate<<"可以使用！"<<endl;

printf("\n");

Line();

}

else

{

system("cls");

Header\_choose();

printf("\n");

Space(5);

cout<<"非常抱歉！车牌号码 辽"<<Test.plate<<"已被他人使用！"<<endl;

printf("\n");

Line();

}

}

if(k!=0)

{

printf("选择：1.\t继续选择\t\t0.\t返回主界面\n");

printf("请选择：\n");

scanf("%d",&t);

while(t!=0&&t!=1)

{

printf("输入的选择不正确！\n");

printf("选择：1.\t继续选择\t\t0.\t返回主界面\n");

printf("请选择：\n");

fflush(stdin);

scanf("%d",&t);

}

}

}

system("cls");

Header\_choose();

printf("\n");

Space(20);

printf("即将返回主菜单！\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

using namespace std;

void Space(int n)//输出空格

{

int i;

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

cout<<" ";

}

void Line()

{

cout<<"--------------------------------------------------------------"<<endl;

}

/\*效果图

辽宁省汽车牌照查询系统

--------------------------------------------------------------

\*/

void Header\_main()//输出主界面表头

{

Space(20);

cout<<"辽宁省汽车牌照查询系统"<<endl;

Line();

cout<<endl;

}

/\*效果图

|

| 1 汽车牌照信息查询

|

| 2 汽车车牌号码选择

|

| 0 退出

|

|

|

|

| 9 管理员更新系统信息

|

--------------------------------------------------------------

\*/

void Main\_interfasce()//输出主界面

{

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(20);

cout<<"1 汽车牌照信息查询";

Space(20);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(20);

cout<<"2 汽车车牌号码选择";

Space(20);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(20);

cout<<"0 退出 ";

Space(20);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(20);

cout<<"9 管理员更新系统信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

Line();

cout<<endl;

}

/\*效果图

汽车牌照信息查询

--------------------------------------------------------------

\*/

void Header\_query()//输出汽车牌照信息查询表头

{

Space(23);

cout<<"汽车牌照信息查询"<<endl;

Line();

cout<<endl;

}

/\*效果图

汽车车牌号码选择

--------------------------------------------------------------

\*/

void Header\_choose()//输出汽车车牌号码选择表头

{

Space(23);

cout<<"汽车车牌号码选择"<<endl;

Line();

cout<<endl;

}

/\*效果图

管理员更新系统信息

--------------------------------------------------------------

\*/

void Header\_administrator()//输出管理员更新系统信息表头

{

Space(22);

cout<<"管理员更新系统信息"<<endl;

Line();

cout<<endl;

}

/\*效果图

| |

| |

| 1 屏幕录入汽车牌照信息 |

| 2 文件录入汽车牌照信息 |

| 3 修改系统文件储存信息 |

| 4 删除系统文件储存信息 |

| 5 显示系统文件信息 |

| 6 系统储存文件信息备份 |

| 7 重置系统储存文件 |

| 8 修改管理员密码 |

| 0 返回主界面 |

| |

| |

--------------------------------------------------------------

\*/

void Administrator\_interfasce()//输出理员更新系统信息界面

{

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"1 屏幕录入汽车牌照信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"2 文件录入汽车牌照信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"3 修改系统文件储存信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"4 删除系统文件储存信息";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"5 显示系统文件信息";

Space(22);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"6 系统储存文件信息备份";

Space(18);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"7 重置系统储存文件";

Space(22);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"8 修改管理员密码";

Space(24);

cout<<"|"<<endl;

cout<<"|";

Space(18);

cout<<"0 返回主界面";

Space(28);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

cout<<"|";

Space(60);

cout<<"|"<<endl;

Line();

cout<<endl;

}

#include<iostream>

#include<cstdlib>

#include"StaticLinkList.h"

#include"Admin.h"

#include<stdio.h>

using namespace std;

void Administrator()

{

int i;

while(i!=0)

{

system("cls");

Header\_administrator();

Administrator\_interfasce();

printf("请选择\n");

scanf("%d",&i);

while(i<0||i>8)

{

printf("输入的选择不正确！\n");

printf("请选择1、2、3、4、5、6、7、8或者0\n");

printf("请选择:\n");

fflush(stdin);

scanf("%d",&i);

}

switch(i)

{

case 1:Enter\_screen();break;

case 2:Enter\_file();break;

case 3:Amend();break;

case 4:Delete();break;

case 5:Show();break;

case 6:Backups();break;

case 7:Clearfile();break;

case 8:Password\_Change();break;

case 0:

system("cls");

Header\_administrator();

printf("\n");

Space(20);

printf("即将返回主菜单！\n");

printf("\n");

Line();

system("pause");

break;

}

}

}

#include<iostream>

#include<fstream>

#include<cstdlib>

#include"StaticLinkList.h"

#include"Other.h"

#include<stdio.h>

using namespace std;

void Backups()

{

StaticLinkList S;

int c=0;

int i;

char file[100];

system("cls");

Header\_administrator();

printf("\n");

printf("对系统储存文件信息进行备份！\n");

printf("请输入文件路径：\n");

cin>>file;

ofstream fout;

fout.open(file,ios::out);

while(fout.fail())

{

printf("文件存储地址不正确！\n");

printf("请输入正确的文件路径：\n");

cin>>file;

fout.open(file,ios::out);

}

system("cls");

Header\_administrator();

printf("\n");

Space(20);

printf("系统储存文件信息备份中...\n");

printf("\n");

Line();

c=Read(S);

fout<<c<<" ";

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

{

fout<<S[i].data.plate<<" "<<S[i].data.name<<" "<<S[i].data.ID<<" "<<S[i].data.car<<" "<<S[i].cur<<" ";

}

fout.close();

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("系统储存文件信息成功备份！\n");

printf("\n");

Line();

system("pause");

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员更新系统信息界面\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<fstream>

#include"StaticLinkList.h"

using namespace std;

int Read(StaticLinkList S)

{

int i,s;

ifstream fin;

fin.open("license\_plate.txt",ios::in);

fin>>s;

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

{

fin>>S[i].data.plate;

fin>>S[i].data.name;

fin>>S[i].data.ID;

fin>>S[i].data.car;

fin>>S[i].cur;

}

fin.close();

return s;

}

#include<iostream>

#include<cstdio>

#include<conio.h>

#include<cstdlib>

#include"StaticLinkList.h"

#include"Other.h"

#include<stdio.h>

using namespace std;

ElemType Input\_plate(ElemType elem,int cls)

{

int i;

int j=0;

int z=0;

char ch;

while(j==0)

{

for(i=0;i<6;i++)

{

elem.plate[i]='O';

}

elem.plate[6]='\0';

i=0;

if(z==100)

{

system("cls");

if(cls==1)

Header\_query();

else if(cls==2)

Header\_choose();

else if(cls==9)

Header\_administrator();

}

printf("请输入车牌号：辽");

while((ch=getch())!='\r')//保证车牌号码的格式

{

if(ch=='\b')

{

if(i>0)

{

i--;

printf("\b \b");// 退格的实现

elem.plate[i+1]='O';

}

else

putchar(7);

}

else

{

if(i!=6)

{

if((ch>='A'&&ch<='Z')||(ch>='0'&&ch<='9'))

{

elem.plate[i++]=ch;

cout<<ch;

}

else putchar(7);

}

}

}

elem.plate[6]='\0';

j=1;

z=0;

/\*for(i=0;i<6;i++)//检验输入的车牌号码是否符合实际情况

{

ch=elem.plate[i];

if(i==0)

{

if(ch>='A'&&ch<='P'&&ch!='I'&&ch!='O') j=j;

else j=0;

}

if(i>0&&i<5)

{

if(ch>='A'&&ch<='Z'&&ch!='I'&&ch!='O')

{

j=j;

z++;

}

else if(ch>='0'&&ch<='9') j=j;

else j=0;

}

if(i==5)

{

if(ch>='0'&&ch<='9') j=j;

else j=0;

}

if(z>2) j=0;

}\*/

string s;

cin>>s;

if (s.size()>6) j=1;

if(j==0)

{

printf("输入的车牌号不符合规定！\n");

printf("注意：辽宁省车牌样式为\"辽XXXXXX\\n");

printf("其中第一位为A,B,C,D,E,F,G,H,J,K,L,M,N,P,代表14个地区\n");

printf("第二位到第五为除I和O外的24个大写英文字母以及0~9数字组成，\n");

printf("但英文字母不能超过2个\n");

printf("最后一位为只能是0~9的数字\n");

printf("请重新输入车牌号!\n");

system("pause");

z=100;

}

if (j==1)

{

mohusearch(s);

}

}

return elem;

}

ElemType Input\_name(ElemType elem)

{

printf("\n车主姓名:\n");

cin>>elem.name;

return elem;

}

ElemType Input\_ID(ElemType elem)

{

int i;

char ch;

for(i=0;i<18;i++)

{

elem.ID[i]=0;

}

i=0;

printf("\n车主身份证号：\n");

while((ch=getch())!='\r')//保证身份证号码是18位的数字（最后一位可以是X）

{

if(ch=='\b')

{

if(i>0)

{

i--;

printf("\b \b");// 退格的实现

elem.ID[i+1]='\0';

}

else

putchar(7);

}

else

{

if(i==17)

{

{

if((ch>='0'&&ch<='9')||ch=='X'||ch=='x')

{

if(ch=='X'||ch=='x')

{

elem.ID[i++]='X';

cout<<'X';

}

else

{

elem.ID[i++]=ch;

cout<<ch;

}

}

else putchar(7);

}

}

if(i<17)

{

if(ch>='0'&&ch<='9')

{

elem.ID[i++]=ch;

cout<<ch;

}

else putchar(7);

}

}

}

elem.ID[18]='\0';

while(i!=18)

{

cout<<"身份证位数输入不正确"<<endl;

cout<<"请重新输入身份证号："<<endl;

i=0;

while((ch=getch())!='\r')//保证身份证号码是18位的数字（最后一位可以是X）

{

if(ch=='\b')

{

if(i>0)

{

i--;

printf("\b \b");// 退格的实现

elem.ID[i+1]='\0';

}

else

putchar(7);

}

else

{

if(i==17)

{

{

if((ch>='0'&&ch<='9')||ch=='X'||ch=='x')

{

if(ch=='X'||ch=='x')

{

elem.ID[i++]='X';

cout<<'X';

}

else

{

elem.ID[i++]=ch;

cout<<ch;

}

}

else putchar(7);

}

}

if(i<17)

{

if(ch>='0'&&ch<='9')

{

elem.ID[i++]=ch;

cout<<ch;

}

else putchar(7);

}

}

}

}

return elem;

}

ElemType Input\_car(ElemType elem)

{

printf("\n车辆品牌及型号：\n");

cin>>elem.car;

return elem;

}

#include<iostream>

#include<cstring>

#include<cstdlib>

#include"StaticLinkList.h"

#include"Other.h"

#include<stdio.h>

using namespace std;

void Clearfile()

{

StaticLinkList S;

int i,j;

char q;

char str[10]={"0"};

int c=0;

system("cls");

Header\_administrator();

printf("\n");

Space(5);

printf("确定重置系统储存文件？（输入1确定，其他则退出）\n");

cin>>q;

if(q=='1')

{

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("重置系统储存文件中...\n");

printf("\n");

Line();

c=Read(S);

for(i=0;i<MAXSIZE;i++)//将链表初始化

{

for(j=0;j<6;j++)

S[i].data.plate[j]='0';

S[i].data.plate[6]='\0';

strcpy(S[i].data.name,str);

for(j=0;j<18;j++)

S[i].data.ID[j]='0';

S[i].data.ID[18]='\0';

strcpy(S[i].data.car,str);

S[i].cur=i+1;

}

c=0;

Store(S,c);//储存到系统文件

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("系统文件重置完毕\n");

printf("\n");

Line();

}

system("cls");

Header\_administrator();

printf("\n");

Space(10);

printf("即将返回管理员更新系统信息界面\n");

printf("\n");

Line();

system("pause");

}

#include<iostream>

#include<cstring>

#include"StaticLinkList.h"

using namespace std;

void Sort(StaticLinkList S,int c)

{

StaticLinkList Ssort;

int Front[34];//关键字队列头指针（位置）

int Rear[34];//关键字队列尾指针（位置）

int i,j,k;

char ch;

for(i=0;i<MAXSIZE;i++)//提取要排序的信息

{

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

Ssort[i].data.plate[j]=S[i].data.plate[j];

strcpy(Ssort[i].data.name,S[i].data.name);

for(j=0;j<19;j++)

Ssort[i].data.ID[j]=S[i].data.ID[j];

strcpy(Ssort[i].data.car,S[i].data.car);

Ssort[i].cur=S[i].cur;

}

for(i=0;i<6;i++)

{

for(k=0;k<34;k++)

{

Front[k]=0;

Rear[k]=0;

}

j=0;

k=1;

while(k!=0)

{

ch=Ssort[j].data.plate[5-i];

switch(ch)

{

case '0':

if(Front[0]==0)

Front[0]=j;

else

Ssort[Rear[0]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[0]=j;

break;

case '1':

if(Front[1]==0)

Front[1]=j;

else

Ssort[Rear[1]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[1]=j;

break;

case '2':

if(Front[2]==0)

Front[2]=j;

else

Ssort[Rear[2]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[2]=j;

break;

case '3':

if(Front[3]==0)

Front[3]=j;

else

Ssort[Rear[3]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[3]=j;

break;

case '4':

if(Front[4]==0)

Front[4]=j;

else

Ssort[Rear[4]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[4]=j;

break;

case '5':

if(Front[5]==0)

Front[5]=j;

else

Ssort[Rear[5]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[5]=j;

break;

case '6':

if(Front[6]==0)

Front[6]=j;

else

Ssort[Rear[6]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[6]=j;

break;

case '7':

if(Front[7]==0)

Front[7]=j;

else

Ssort[Rear[7]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[7]=j;

break;

case '8':

if(Front[8]==0)

Front[8]=j;

else

Ssort[Rear[8]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[8]=j;

break;

case '9':

if(Front[9]==0)

Front[9]=j;

else

Ssort[Rear[9]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[9]=j;

break;

case 'A':

if(Front[10]==0)

Front[10]=j;

else

Ssort[Rear[10]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[10]=j;

break;

case 'B':

if(Front[11]==0)

Front[11]=j;

else

Ssort[Rear[11]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[11]=j;

break;

case 'C':

if(Front[12]==0)

Front[12]=j;

else

Ssort[Rear[12]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[12]=j;

break;

case 'D':

if(Front[13]==0)

Front[13]=j;

else

Ssort[Rear[13]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[13]=j;

break;

case 'E':

if(Front[14]==0)

Front[14]=j;

else

Ssort[Rear[14]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[14]=j;

break;

case 'F':

if(Front[15]==0)

Front[15]=j;

else

Ssort[Rear[15]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[15]=j;

break;

case 'G':

if(Front[16]==0)

Front[16]=j;

else

Ssort[Rear[16]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[16]=j;

break;

case 'H':

if(Front[17]==0)

Front[17]=j;

else

Ssort[Rear[17]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[17]=j;

break;

case 'J':

if(Front[18]==0)

Front[18]=j;

else

Ssort[Rear[18]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[18]=j;

break;

case 'K':

if(Front[19]==0)

Front[19]=j;

else

Ssort[Rear[19]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[19]=j;

break;

case 'L':

if(Front[20]==0)

Front[20]=j;

else

Ssort[Rear[20]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[20]=j;

break;

case 'M':

if(Front[21]==0)

Front[21]=j;

else

Ssort[Rear[21]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[21]=j;

break;

case 'N':

if(Front[22]==0)

Front[22]=j;

else

Ssort[Rear[22]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[22]=j;

break;

case 'P':

if(Front[23]==0)

Front[23]=j;

else

Ssort[Rear[23]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[23]=j;

break;

case 'Q':

if(Front[24]==0)

Front[24]=j;

else

Ssort[Rear[24]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[24]=j;

break;

case 'R':

if(Front[25]==0)

Front[25]=j;

else

Ssort[Rear[25]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[25]=j;

break;

case 'S':

if(Front[26]==0)

Front[26]=j;

else

Ssort[Rear[26]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[26]=j;

break;

case 'T':

if(Front[27]==0)

Front[27]=j;

else

Ssort[Rear[27]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[27]=j;

break;

case 'U':

if(Front[28]==0)

Front[28]=j;

else

Ssort[Rear[28]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[28]=j;

break;

case 'V':

if(Front[29]==0)

Front[29]=j;

else

Ssort[Rear[29]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[29]=j;

break;

case 'W':

if(Front[30]==0)

Front[30]=j;

else

Ssort[Rear[30]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[30]=j;

break;

case 'X':

if(Front[31]==0)

Front[31]=j;

else

Ssort[Rear[31]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[31]=j;

break;

case 'Y':

if(Front[32]==0)

Front[32]=j;

else

Ssort[Rear[32]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[32]=j;

break;

case 'Z':

if(Front[33]==0)

Front[33]=j;

else

Ssort[Rear[33]].cur=j;

k=Ssort[j].cur;

Ssort[j].cur=0;

Rear[33]=j;

break;

}

j=k;

}

j=0;

while(Front[j]==0)

{

j++;

}

Ssort[0].cur=Front[j];

k=j;

j++;

for(;j<34;j++)

{

if(Front[j]!=0)

{

Ssort[Rear[k]].cur=Front[j];

k=j;

}

}

}

k=0;

S[k].cur=1;

k++;

i=0;

i=Ssort[i].cur;

while(i!=0)//将排序后的链表返回

{

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

S[k].data.plate[j]=Ssort[i].data.plate[j];

strcpy(S[k].data.name,Ssort[i].data.name);

for(j=0;j<19;j++)

S[k].data.ID[j]=Ssort[i].data.ID[j];

strcpy(S[k].data.car,Ssort[i].data.car);

S[k].cur=k+1;

k++;

i=Ssort[i].cur;

}

S[c].cur=0;

}

//新加的模糊查找

void carmanage::dim\_search()

{

int k;

string s;

cout<<"\t\t\t"<<"请输入车牌号";

cin>>s;

if(s.size()<=8)

{

for(k=1;k<carnumber;k++)

{

string key;

string ss;

ss=L[k].ornumber;

for(int j=0;j<8;j++)

{

key=ss.substr(j,s.size());

if(key==s)

{

for(int i=0;i<8;i++)

{

cout<<L[k].ornumber[i];

}

cout<<endl;

}

}

}

}

else

cout<<"无";

}

**附录C：**

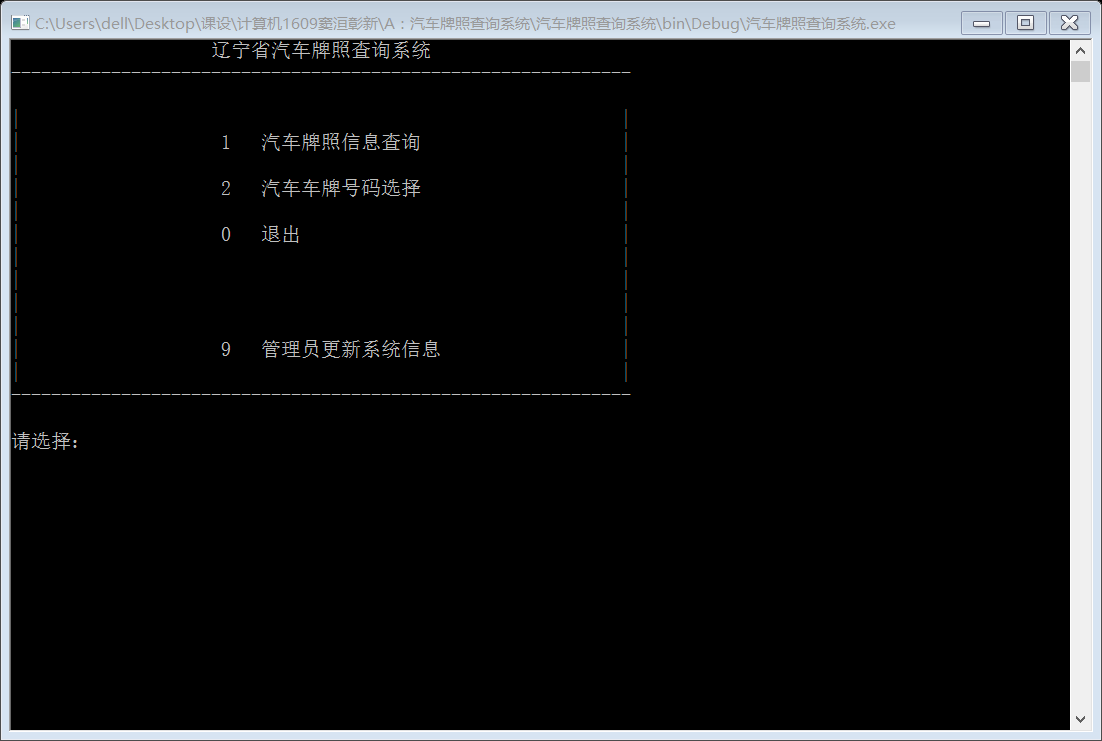
**使用手册**

**1.运行环境说明**

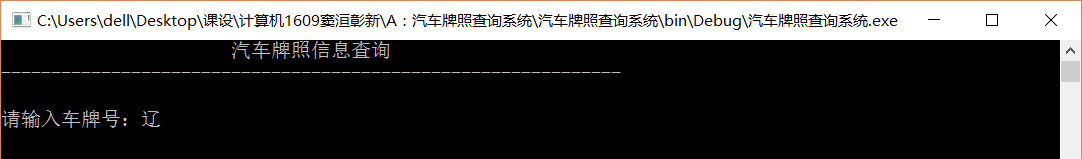
此程序可以在Windows XP、win7等操作平台用Code Blocks运行

**2.操作说明**

运行程序将看到主界面如下：



选择1：汽车牌照信息查询

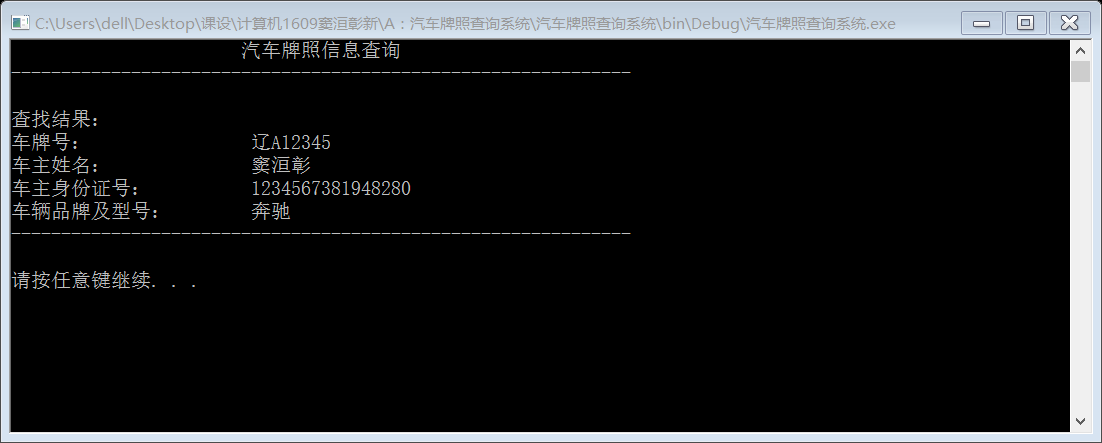


输入车牌号码即可查找

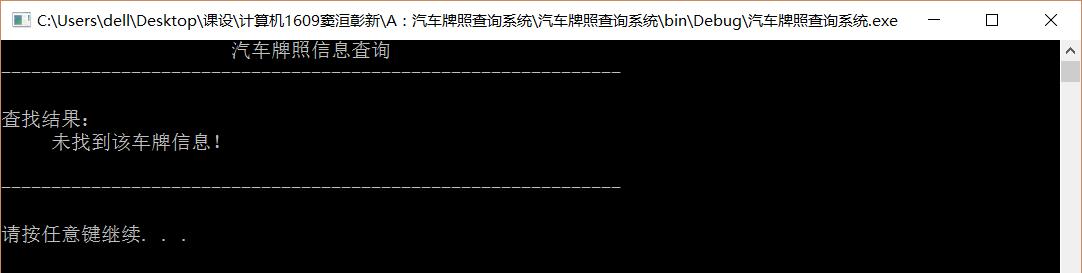
（注意：辽宁省车牌样式为\"辽XXXXXX\"，其中第一位为A,B,C,D,E,F,G,H,J,K,L,M,N,P,代表14个地区，第二位到第五为除I和O外的24个大写英文字母以及0~9数字组成，但英文字母不能超过2个，最后一位为只能是0~9的数字。

但不用担心啦，输入错误系统会提示重新输入）



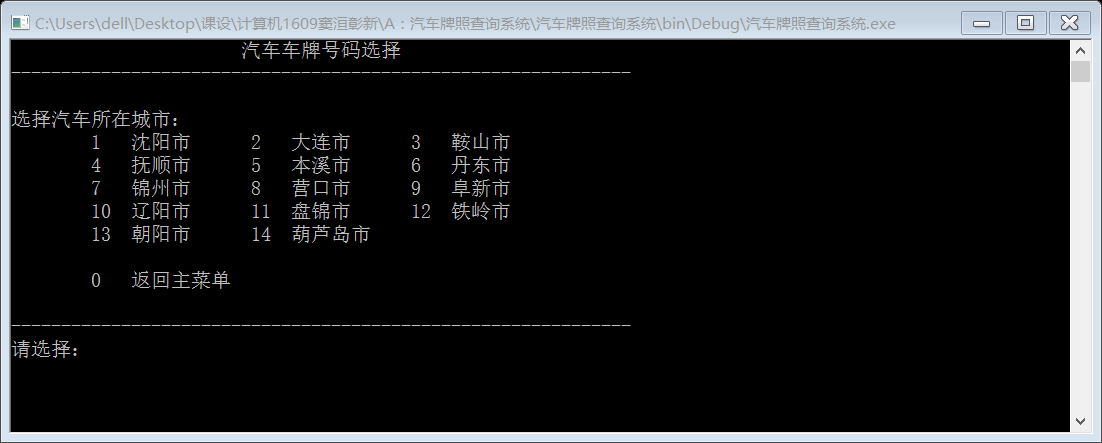
正确找到会这样：如输入A12345

没有找到：如A11111

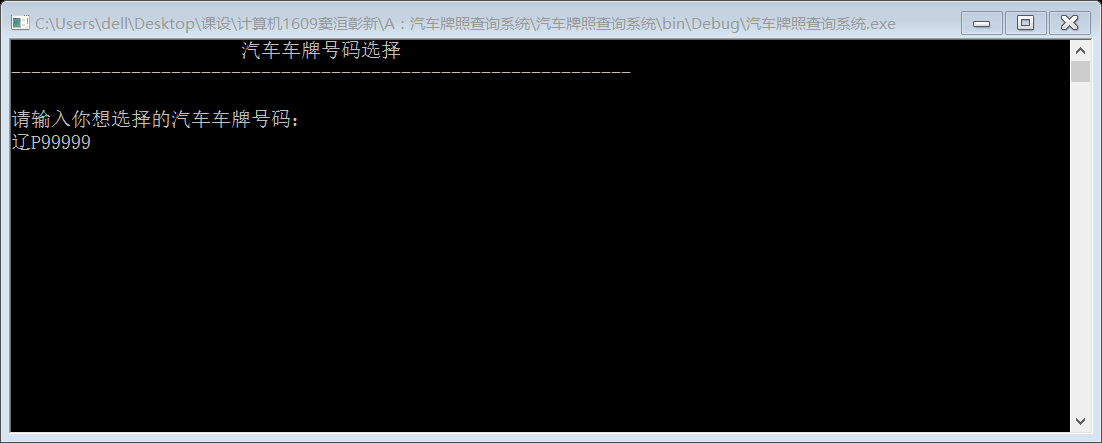


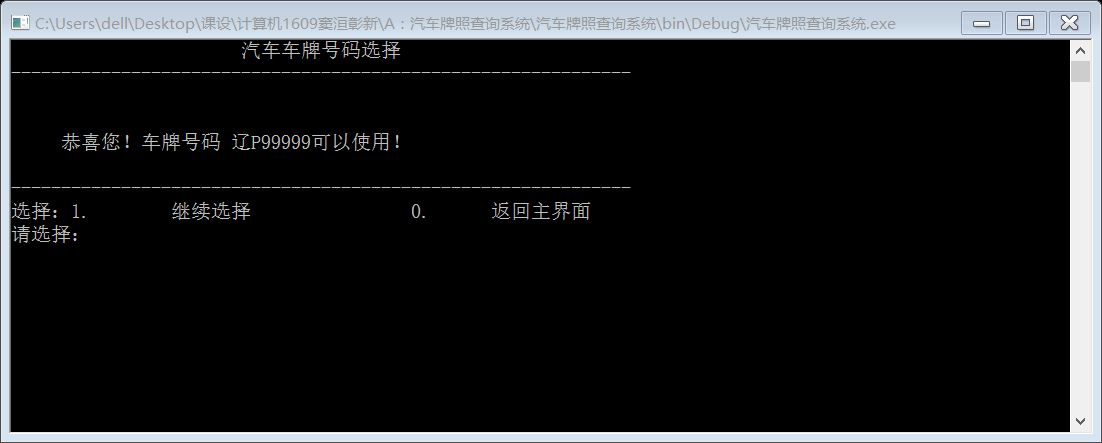
回到主界面，如果我们选2：汽车车牌号码选择

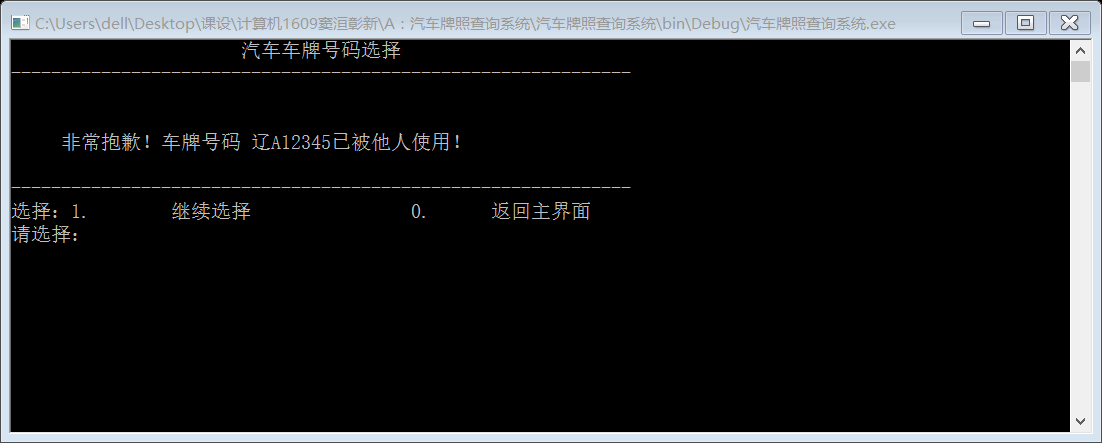
选择择汽车所在城市：1、沈阳市2、大连市3、 鞍山市4、抚顺市5、本溪市6、丹东市7、锦州市8、营口市9、阜新市10、 辽阳市11、盘锦市12、铁岭市13、朝阳市14、葫芦岛市



如果是葫芦岛市想要一个辽P99999的车牌号

那么我们先选14然后输入99999

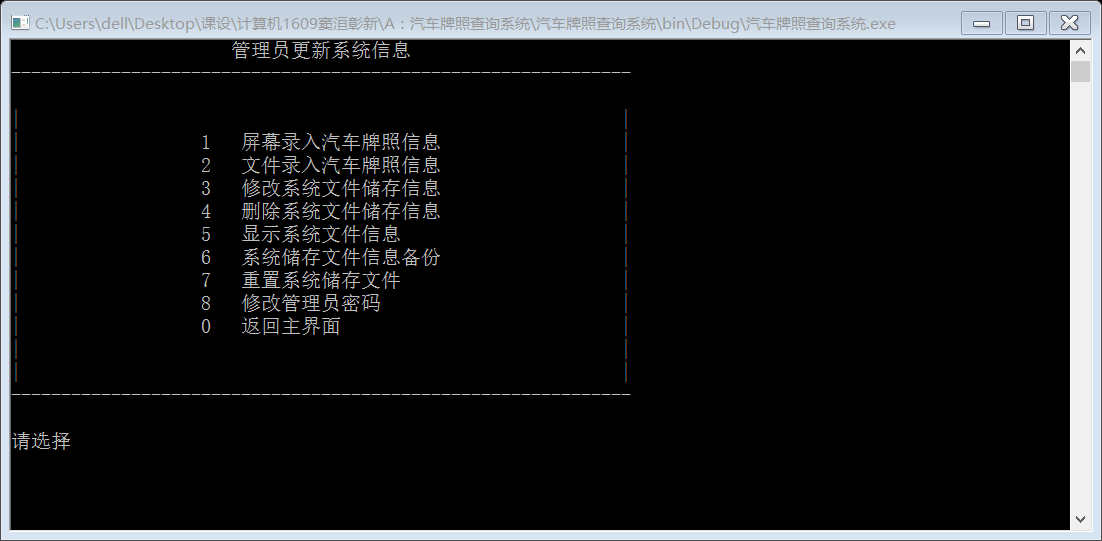


如果我们选辽A12345

返回主界面，现在我们来选9管理员更新系统信息

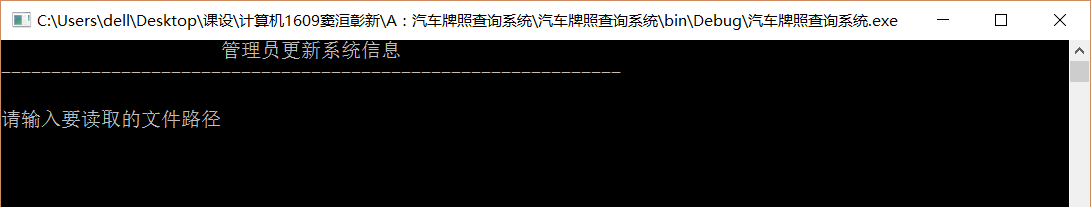
初始密码为000000

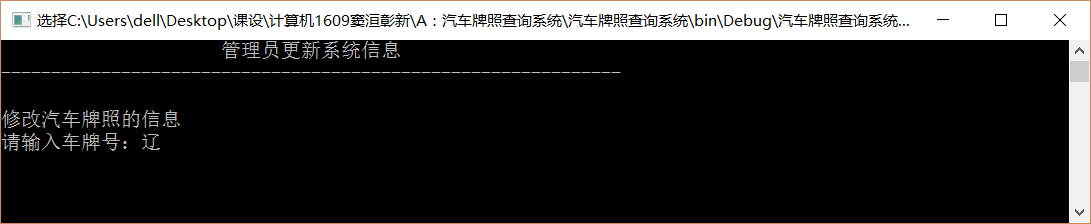
（密码可以修改，如果忘了也没关系，可以到工程文件中找password.txt文件，里面储存的就是密码）

密码验证成功后，进入到管理员更新系统信息界面

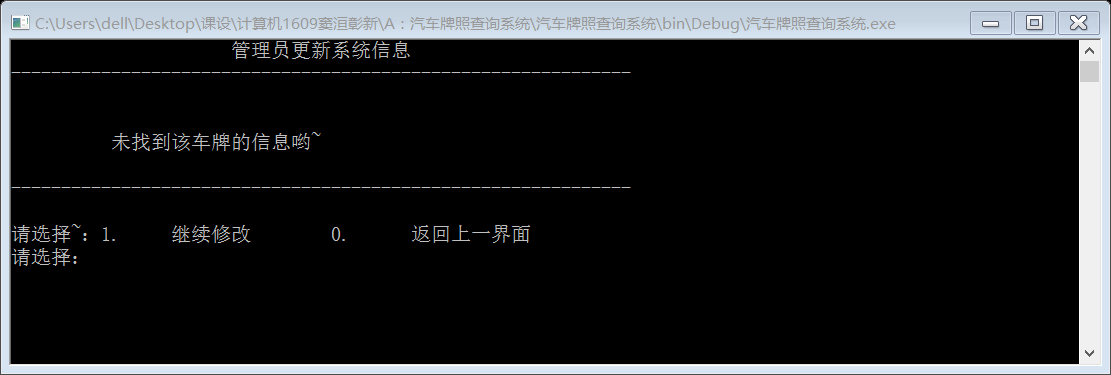
选择1之后按提示逐一录入信息，信息输入完之后会提示选择是否此信息输入正确，选择1则储存到系统文件，选择2则会修改信息

修改之后还是会显示修改后新的信息，然后继续提示选择1或2

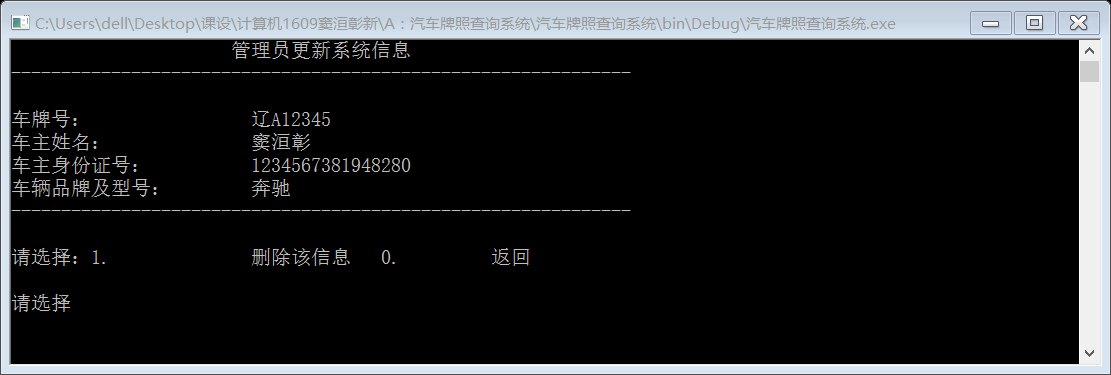
回到管理员更新系统信息界面，然后选择2

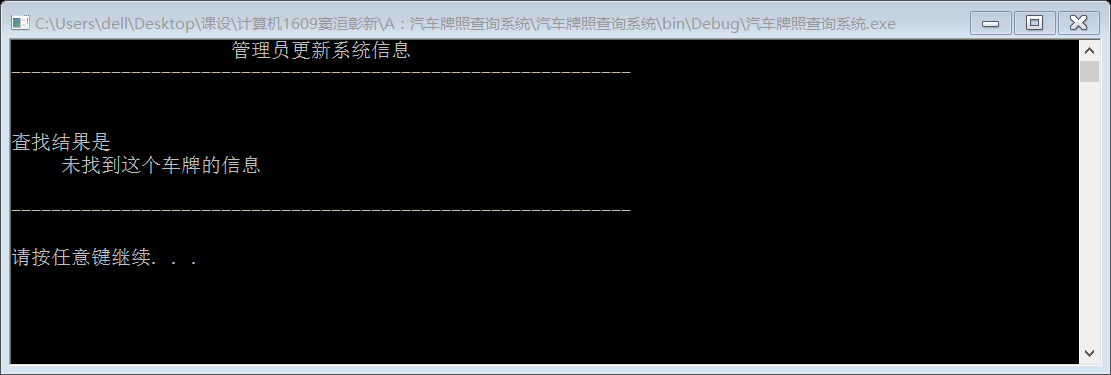
回到管理员更新系统信息界面，然后选择3

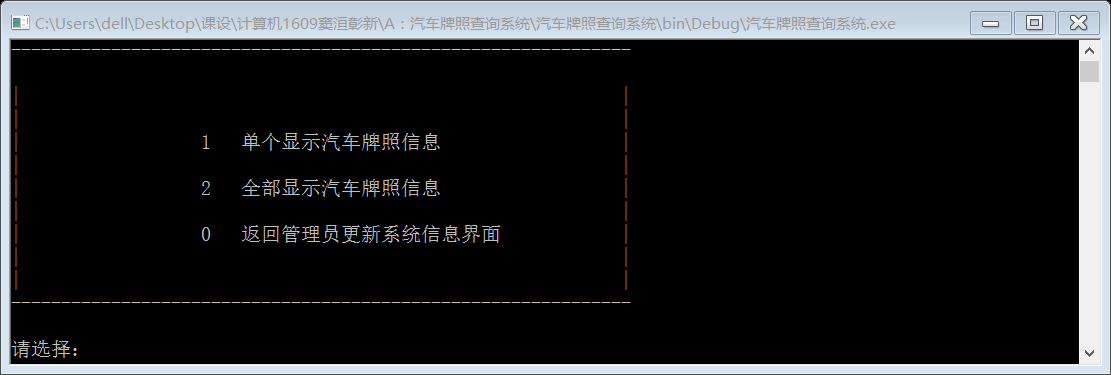
如果输入A12345

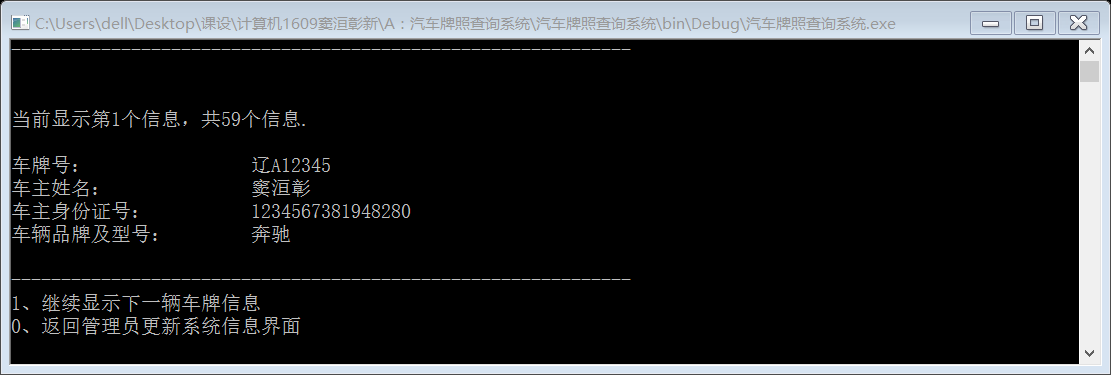
如果输入A11111

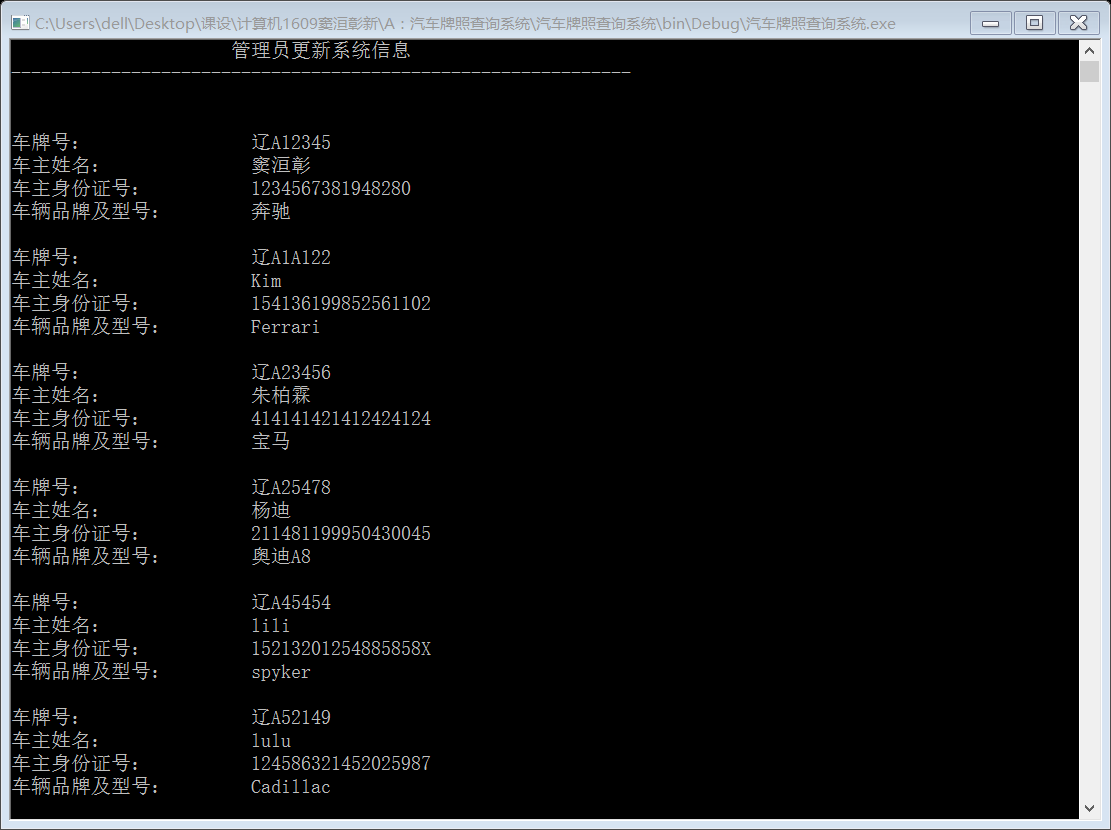
回到管理员更新系统信息界面，然后选择4

如果输入A12114

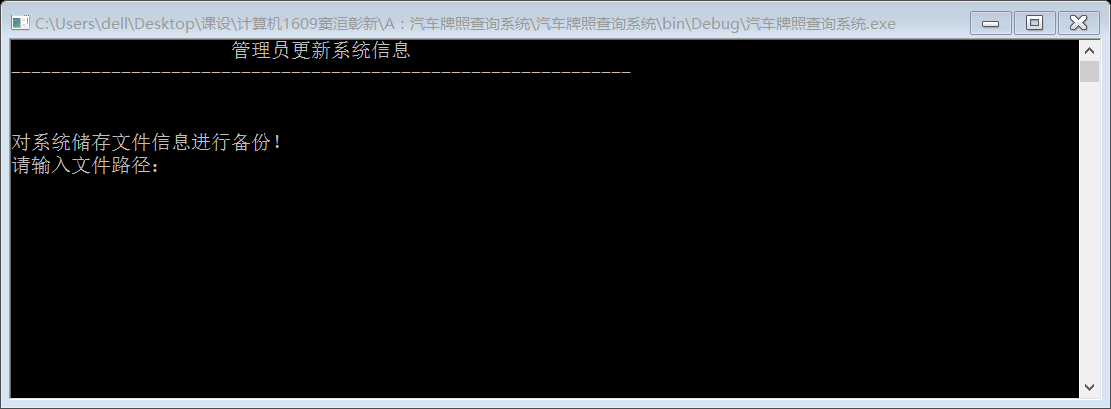
如果输入A11111

回到管理员更新系统信息界面，然后选择5

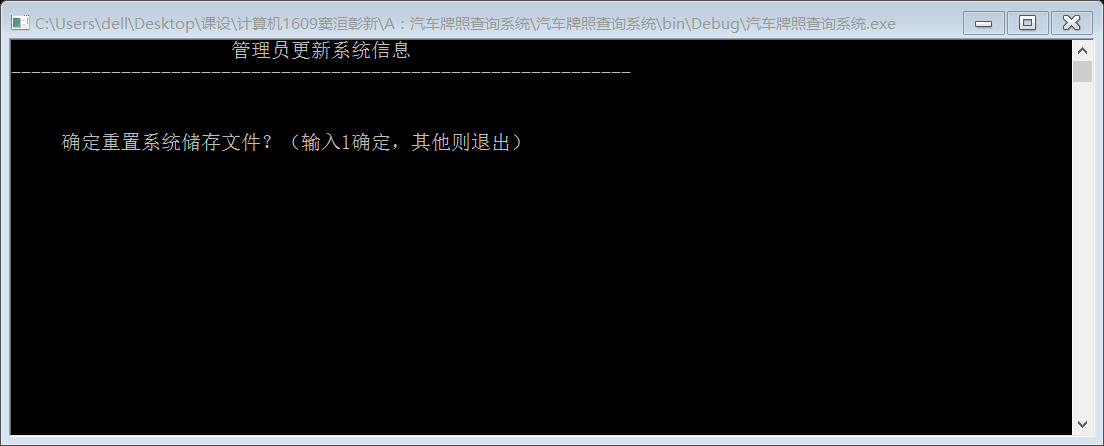
选1：

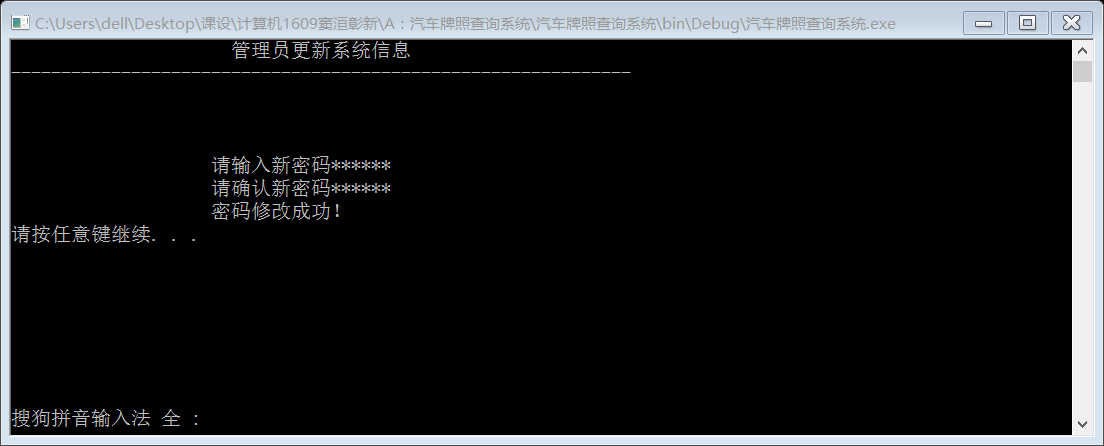
选2：

回到管理员更新系统信息界面，然后选择6

（注意路径输入不正确，就会存到系统文件中的哟！）

回到管理员更新系统信息界面，然后选择7

（小心哟！选1之后的数据就全没了！）

回到管理员更新系统信息界面，然后选择8

最后，就退出了！