**东北大学信息科学与工程学院**

**数据结构课程设计报告**

题目A **汽车牌照的快速查询**

课题组长 闻亮

课题组成员 吕传昊、梁浩纯

专业名称 计算机科学与技术

班级 计1604

指导教师 高岩

2018 年 1月

目录

[1 课题概述 2](#_Toc503278177)

[1.1 课题任务 2](#_Toc503278178)

[1.2 课题原理 2](#_Toc503278179)

[1.3 课题知识 2](#_Toc503278180)

[2 需求分析 2](#_Toc503278181)

[2.1 课题调研 2](#_Toc503278182)

[2.2 用户需求 3](#_Toc503278183)

[3 方案设计 3](#_Toc503278184)

[3.1 总体功能设计 3](#_Toc503278185)

[3.1.1 界面设计 3](#_Toc503278186)

[3.1.2 数据维护 5](#_Toc503278187)

[3.1.3 数据查询 5](#_Toc503278188)

[3.1.4 数据统计 6](#_Toc503278189)

[3.2 数据结构设计 6](#_Toc503278190)

[3.3 函数原型设计 7](#_Toc503278191)

[3.4 主算法设计 8](#_Toc503278192)

[3.5 用户界面设计 8](#_Toc503278193)

[3.6 输入输出设计 9](#_Toc503278194)

[4 方案实现 9](#_Toc503278195)

[4.1 开发环境与工具 9](#_Toc503278196)

[4.2 个人程序实现 10](#_Toc503278197)

[4.2.1 闻亮设计实现 10](#_Toc503278198)

[4.2.2 吕传昊设计实现 12](#_Toc503278199)

[4.2.3 梁浩纯设计实现 14](#_Toc503278200)

[5 测试与调试 17](#_Toc503278201)

[5.1 个人测试 17](#_Toc503278202)

[5.1.1 闻亮测试 17](#_Toc503278203)

[5.1.2 吕传昊测试 19](#_Toc503278204)

[5.1.3 梁浩纯测试 21](#_Toc503278205)

[6 课题总结 22](#_Toc503278206)

[6.1 课题评价 22](#_Toc503278207)

[6.2 团队协作 22](#_Toc503278208)

[6.3 个人总结 22](#_Toc503278209)

[6.3.1 闻亮设计小结 22](#_Toc503278210)

[6.3.2 吕传昊设计小结 23](#_Toc503278211)

[6.3.3 梁浩纯设计小结 23](#_Toc503278212)

[7 附录 23](#_Toc503278213)

[7.1 课题任务分工 23](#_Toc503278214)

[7.1.1 课题程序设计分工 23](#_Toc503278215)

[7.1.2 课题报告分工 25](#_Toc503278216)

[7.2 使用手册 26](#_Toc503278217)

# 课题概述

## 课题任务

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。

## 课题原理

根据题意，首先采用线性链表数据结构来储存车牌号。录入方面，录入部分分为随机录入和人工录入，同时在人工录入方面设定判错。查找方面，采用二分法查找，同时也采用了另一种分块查找，加快查找速度。排序方面，运用了链式基数排序，可对车牌大小和地区分布多重排序。

## 课题知识

李开主编. C语言实验与课程设计. 科学出版社,2011.3.

曹计昌主编. C语言与程序设计. 电子工业出版社，2013.1.

陈越主编 数据结构 高等教育出版社 2016.6

# 需求分析

## 课题调研

根据我们的研究和查阅相关资料，首先牌照总的来讲分为三类，摩托车车牌、货车车牌、还有就是常见的中小型车车牌。这三类车牌是可以存在重复的，不过极少见，同类车牌不能有重复，且在目前车牌最多只有三位字母。当然，还有其他种类的车牌比如军用等，但是肯定不会有这种权限去查询的。其次是，目前车牌采用的是摇号制，所以肯定要有随机数据机制。再然后辽宁一共有14个城市，分为A-沈阳市 B-大连市 C-鞍山市 D-抚顺市 E-本溪市 F丹东市 G-锦州市 H-营口市 J-阜新市 K-辽阳市 L-盘锦市 M-铁岭市 N-朝阳市 P-葫芦岛市 V-省直系统。

## 用户需求

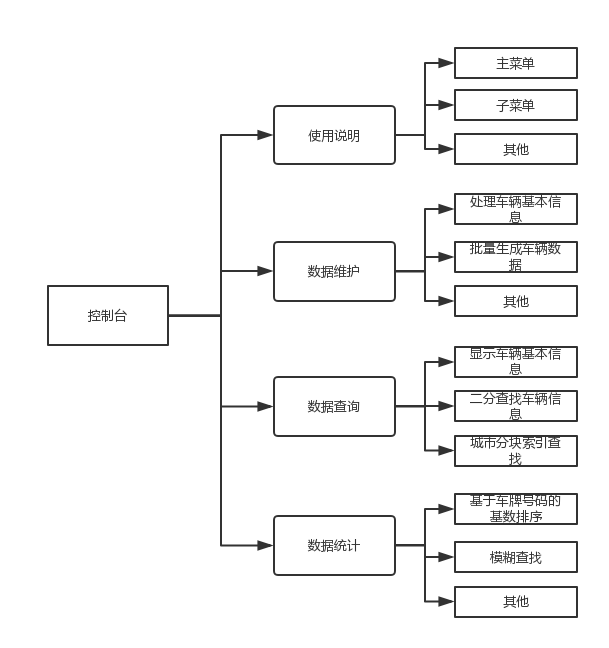
用户需求基本跟题目要求一样，随机数据，折半查询，车排排序，界面可视化。这些上文中有就不做过多解释。除此之外我们还加入了模糊查找，在查询时只要输入关键字就能把含有关键字的所有信息显示出来，极大方便了查找，而且还加入了统计，最大限度考虑到了用户使用时的需求。

# 方案设计

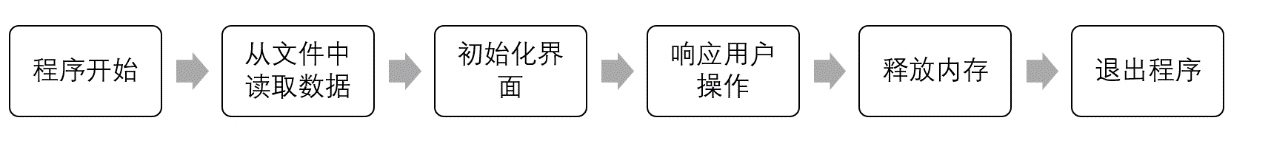
## 总体功能设计

### 界面设计

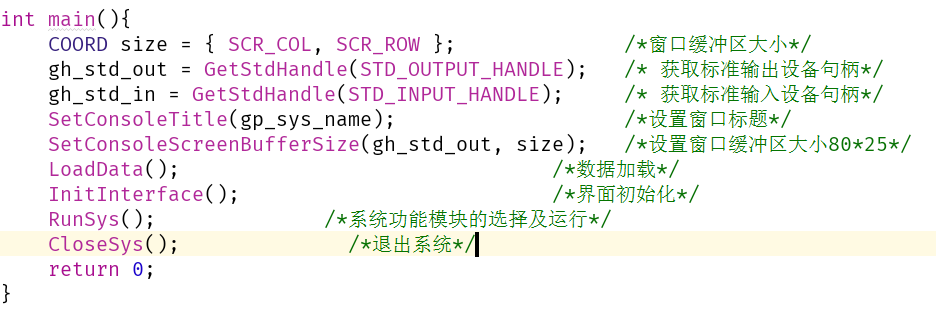
运用控制台操作界面，界面主菜单及子菜单设置如下：



对于程序刚开始运行，我们采用的流程是：



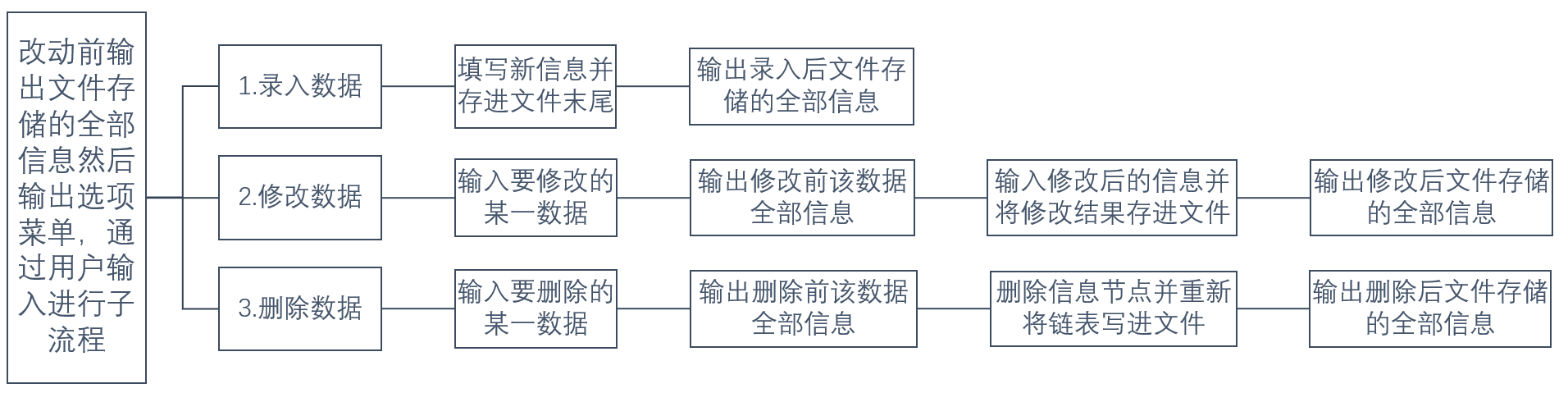
程序主流程设置



界面操作流程代码

### 数据维护

三种数据的录入，修改和删除采用字符界面的形式进行操作，即用户输入1,2,3进行选择选项，然后按照提示进行信息改动。在录入，修改或者删除信息过程中，遇到异常输入，则执行输入循环直到用户输入符合条件的信息为止。改动后的信息通过先写进链表的方式全部写进文件。具体实现流程如下图所示：



### 数据查询

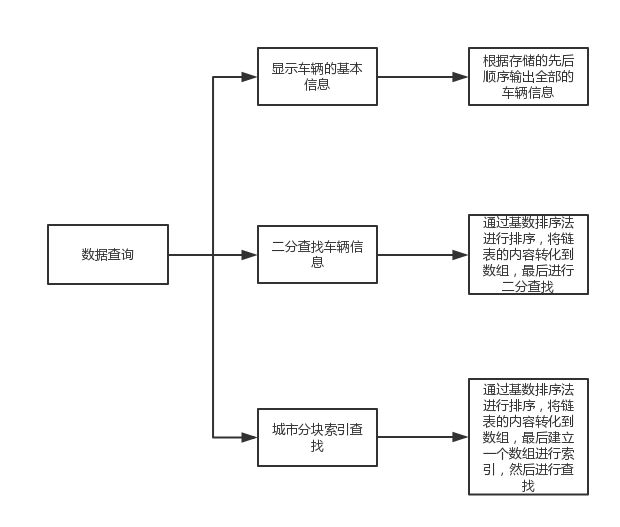
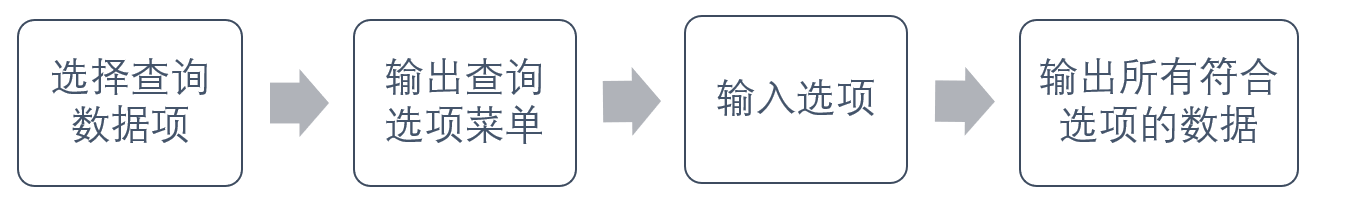


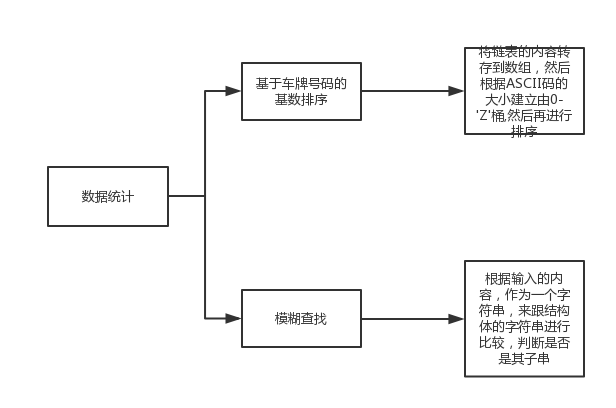
图4-5 数据查询菜单项

进入界面后执行流程如图所示（如果输入选项有误，一直循环输入，直到输入正确的选项为止）：



数据查询流程

### 数据统计



数据统计流程

## 数据结构设计

|  |  |  |
| --- | --- | --- |
| **中文名** | **标识符** | **类型** |
| 车主姓名 | Name | char[15] |
| 车辆类型 | Type | char[15] |
| 车牌号 | Num | char[15] |
| 车主号码 | Phone | char[15] |
| 指向下一节点的指针 | next | Car \* |

车辆的链表（类型：Car ）成员

|  |  |  |
| --- | --- | --- |
| **中文名** | **标识符** | **类型** |
| 弹出窗口层数 | LayerNo | char |
| 弹出窗口区域坐标 | rcArea | SMALL\_RECT |
| 弹出窗口区域字符单元原信息存储缓冲区 | pContent | CHAR\_INFO\* |
| 弹出窗口区域字符单元原属性值存储缓冲区 | pScrAtt | char \* |
| 指向下一结点的指针 | next | LAYER\_NODE\* |

屏幕窗口信息链结点结构链表（类型：LAYER\_NODE）成员

|  |  |  |
| --- | --- | --- |
| **中文名** | **标识符** | **类型** |
| 标签字符串数组首地址 | ppLabel | char \*\* |
| 标签定位数组首地址 | pLoc | COORD \* |
| 标签个数 | num | int |

标签束结构链表（类型：LABEL\_BUNDLE）成员

|  |  |  |
| --- | --- | --- |
| **中文名** | **标识符** | **类型** |
| 热区定位数组首地址 | pArea | SMALL\_RECT \* |
| 热区类别(按键、文本框、选项框)数组首地址 | pSort | char \* |
| 热区序号数组首地址 | pTag | char \* |
| 热区个数 | num | int |

热区结构链表（类型：HOT\_AREA）成员

|  |  |
| --- | --- |
| **全局头指针标识符** | **类型** |
| gp\_top\_layer | LAYER\_NODE \* |
| gp\_head1 | CAR\* |

全局头指针成员

## 函数原型设计

LoadData(); /\*数据加载\*/

InitInterface(); /\*界面初始化\*/

RunSys(); /\*系统功能模块的选择及运行\*/

CloseSys(); /\*退出系统\*/

List MakeEmpty(); //创建一个空的线性表,将链表数据读入数组

BOOL ShowAllInfor(); //显示数组的所有信息

void ShowInfor(List L);

BOOL MaintainCarInfo(void); //维护车辆基本信息

BOOL ShowRadixSorting(void); //显示车辆信息基数排序

void RadixSorting(List L);

void freedom(void); //释放所有的内存

//随机数据的生成

BOOL ShowGenrnd(void); //随机生产数据的界面

void Genrnd(); //随机结构体的生成

void GenrndNum(char \* buff); //随机车牌的生成

void GenrndPhone(char \* buff); //随机号码的生成

void GenrndName(char \* buff); // 随机车主名字的生成

void GenrndType(char \* buff); //随机汽车类型的生成

char \*wholename(int a); //返回对应的项目类型全称

//模糊查找

BOOL ShowFuzzySearch(void); //显示模糊查找界面

void FuzzySearch(List l,char k[]); //模糊查找函数

int intain(char \*a,char \*b); //判断a是不是b的子串

//

int panduan(char str[]);

## 主算法设计

基于基数排序的车牌号码查找

基于判断子字符串的模糊查找

二分查找

## 用户界面设计

（1）采用控制台界面，实现通过键盘输入能够控制菜单选择的功能（如图2-1）。

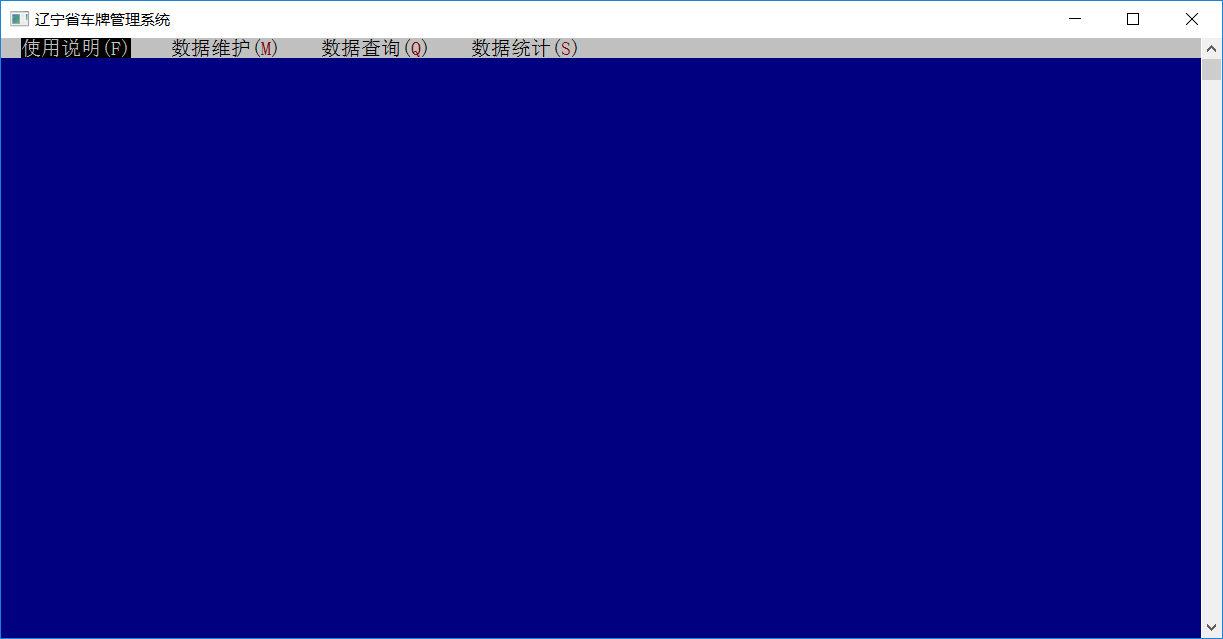


图2-1 控制台界面示例

（2）通过在标准输入输出界面中输入选择选项来达到数据变动，查询和统计的功能，并输出相应结果。

（3）在使用说明中介绍该系统的一些快捷键使用方法。

## 输入输出设计

通过读取文件的内容进行输入

通过手动输入的方式进行设计

通过随机生成的方式进行生成数据

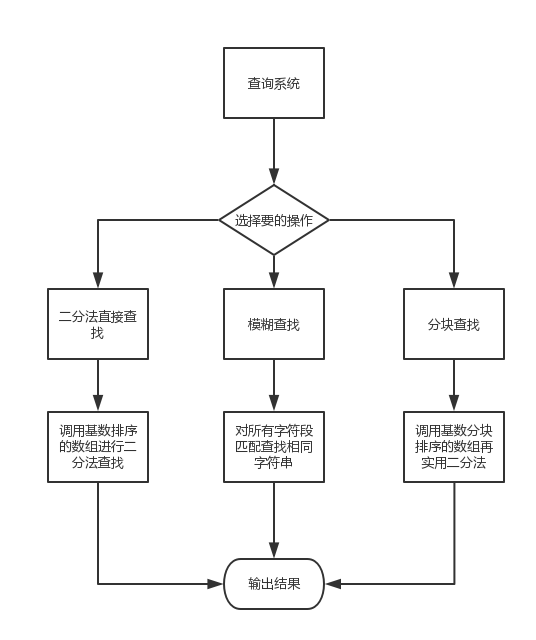
# 方案实现

## 开发环境与工具

我们组采用的是在windows下的c-free系统，包括可视化的实现也是采用了windows的系统函数。

## 个人程序实现

### 闻亮设计实现



负责的是查询系统，依题意，首要采用的是二分法，再分为分块查询和直接查询。因为二分法的局限性，为此的对策是，因为是同一地区，把基数排序好的录入为数组再进行二分法查找。因为基数排序恰好有两种，所以不同的查询方式调用不同的基数排序后的结果。

首先是二分法的核心算法，从文件中将数据读到数组中，在采用传统二分法，如下：

Position BinarySearch**(**List Tb1**,**char k**[]){**

Position left**,**right**,**mid**;**

left**=**0**;**

right**=**Tb1**->**Last**;**

**while(**left**<=**right**)**

**{**

mid**=(**left**+**right**)/**2 **;**

**if(**strcmp**(**k**,**Tb1**->**Data**[**mid**].**Num**)<**0**)**

**{**

right**=**mid**-**1**;**

**}else** **if(**strcmp**(**k**,**Tb1**->**Data**[**mid**].**Num**)>**0**)**

**{**

left**=**mid**+**1**;**

**}else**

**{**

**return** mid**;**

**}**

**}**

**return** NotFind**;**

**}**

然后扩充增加了模糊查找的功能，匹配相应的字符串，这里采用的是字符串的匹配查找，其中intain为字符串比较函数：

void FuzzySearch**(**List L**,**char k**[])**

**{**

int i**;**

int numnum**=**0**;**

int temp**=**1**;**

**for(**i**=**0**;**i**<=**L**->**Last**;**i**++)**

**{**

**if((**intain**(**k**,**L**->**Data**[**i**].**Name**)||**intain**(**k**,**L**->**Data**[**i**].**Type**)||**intain**(**k**,**L**->**Data**[**i**].**Num**)||**intain**(**k**,**L**->**Data**[**i**].**Phone**))==**1**)**

**{**

printf**(**"%-17s%-10s%-16s%-14s\n"**,** L**->**Data**[**i**].**Num**,** L**->**Data**[**i**].**Name**,** L**->**Data**[**i**].**Phone**,** L**->**Data**[**i**].**Type**);**

temp**=**0**;**

numnum**++;**

**}**

**}**

**if(**temp**!=**1**)**

**{**

printf**(**"符合条件的有%d条记录"**,**numnum**);**

**}**

**if(**temp**==**1**)**

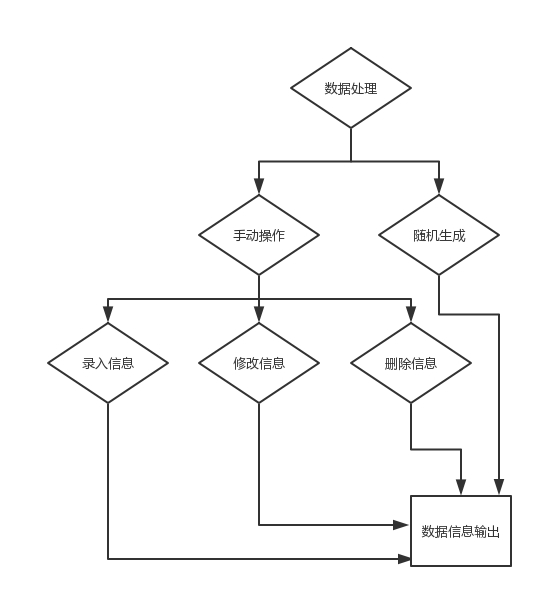
**{**

printf**(**"没有相互匹配的内容。\n\n"**);**

**}**

**}**

### 吕传昊设计实现



负责的是数据处理，在数据输入函数中设计了两种输入方式，随机和手动，在输入的随机生产的过程中，设置了非法输入判断且信息均以链表的形式存入文件中。在数据信息输出的过程中从文件中读取信息再显示到界面上。

**while(**strlen**(**newcol**->**Num**)!=**6**||(**int**)**newcol**->**Num**[**0**]<**48**||(**int**)**newcol**->**Num**[**0**]>**57**||**panduan**(**newcol**->**Num**))**//非法输入判断，1.号码长度为62.号码第一位均为数字3.号码最多有2个大写字母，且不能出现小写字母

//显示所有车牌信息

BOOL ShowAllInfor**()**

**{**

CAR **\***c1**=**gp\_head1**;**

**if(**c1**==NULL)**

**{**

printf**(**"当前未有车辆信息\n\n"**);**

**return;**

**}**

List L**=**MakeEmpty**();**

ShowInfor**(**L**);**

**}**

//显示静态链表的所有信息

void ShowInfor**(**List L**)**

**{**

int i**;**

printf**(**"%-17s%-10s%-16s%-14s\n"**,** "车牌号码"**,** "车主姓名"**,** "车主联系方式"**,** "车辆类型"**);**

**for(**i**=**0**;**i**<=**L**->**Last**;**i**++)**

**{**

printf**(**"%-17s%-10s%-16s%-14s\n"**,** L**->**Data**[**i**].**Num**,** L**->**Data**[**i**].**Name**,** L**->**Data**[**i**].**Phone**,** L**->**Data**[**i**].**Type**);**

**}**

**if(**i**==**0**)**

**{**

printf**(**"当前未有车辆信息\n\n"**);**

**}**

**}**

//随机车牌的生成

void GenrndNum**(**char **\*** buff**)**

**{**

int n **=** 7**;**

int count**=**0**;**

char First**[]** **=** "01234567890123456789012345678901234567890123456789"**;**

char Second**[]** **=** "ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890123456789"**;**

srand**(**time**(NULL));**

int i **=** 0**;**

buff**[**i**]** **=** First**[**rand**()** **%** 50**];**

**for** **(**i **=** 1**;** i **<** 6**;** i**++)**

**{**

buff**[**i**]** **=** Second**[**rand**()** **%** 46**];**

**if(**'A'**<=**buff**[**i**]&&**buff**[**i**]<=**'Z'**)**

count**++;**

**if(**count**>**2**)**

buff**[**i**]=**First**[**rand**()** **%** 50**];**

**}**

buff**[**n **-** 1**]** **=** '\0'**;**

**}**

//对号码字母个数和字母大小写的判断函数

int panduan**(**char str**[])**

**{**

int i**;**

int temp**=**0**;**

int count**=**0**;**

int flage**=**0**;**

**for(**i**=**1**;**i**<**6**;**i**++)**

**{**

**if(**'a'**<=**str**[**i**]&&**str**[**i**]<=**'z'**)**

**{**

flage**=**1**;**

**}**

**if(**'A'**<=**str**[**i**]&&**str**[**i**]<=**'Z'**)**

count**++;**

**if(**count**>**2**||**flage**==**1**)**

temp**=**1**;**

**}**

**return** temp**;**

**}**

### 梁浩纯设计实现

负责的部分是界面设计以及算法部分的基数排序和索引查找。其中，界面设计是听取别人的建议，以及结合Github的一些代码修改而成。代码主要是调用windows系统的API函数，最后形成有下拉菜单的界面，可以用鼠标进行点击操作。

算法部分中的基数排序根据ASCII 码的大小进行比较，建立从 0-‘Z’的通，然后进行排序，代码如下：

//基数排序

void RadixSorting**(**List L**){**

int N**=**L**->**Last**+**1**;**

intD**,**Di**,**i**;**

Bucket B**;**

PtrToNode1 tmp**,**p**,**List1**=NULL;**

**for(**i**=**0**;**i**<**Radix**;**i**++)**

**{**

B**[**i**].**head**=**B**[**i**].**tail**=NULL;**

**}**

//初始化每个通为空链表

**for(**i**=**0**;**i**<**N**;**i**++)**

**{**

tmp**=(**PtrToNode1**)**malloc**(sizeof(**struct Node**));**

strcpy**(**tmp**->**key**.**Num**,**L**->**Data**[**i**].**Num**);**

strcpy**(**tmp**->**key**.**Type**,**L**->**Data**[**i**].**Type**);**

strcpy**(**tmp**->**key**.**Name**,**L**->**Data**[**i**].**Name**);**

strcpy**(**tmp**->**key**.**Phone**,**L**->**Data**[**i**].**Phone**);**

tmp**->**next**=**List1**;**

List1**=**tmp**;**

**}**

//开始排序

**for(**D**=**MaxDigit**;**D**>=**1**;**D**--)**

**{**

p**=**List1**;**

**while(**p**)**

**{**

Di**=**GetChar**(**p**->**key**.**Num**,**D**);**

tmp**=**p**;**

p**=**p**->**next**;**

tmp**->**next**=NULL;**

**if(**B**[**Di**].**head**==NULL)**

**{**

B**[**Di**].**head**=**B**[**Di**].**tail**=**tmp**;**

**}**

**else**

**{**

B**[**Di**].**tail**->**next**=**tmp**;**

B**[**Di**].**tail**=**tmp**;**

**}**

**}**

List1**=NULL;**

**for(**Di**=**Radix**-**1**;**Di**>=**0**;**Di**--)**

**{**

**if(**B**[**Di**].**head**)**

**{**

B**[**Di**].**tail**->**next**=**List1**;**

List1**=**B**[**Di**].**head**;**

B**[**Di**].**head**=**B**[**Di**].**tail**=NULL;**

**}**

**}**

**}**

**for(**i**=**0**;**i**<**N**;**i**++)**

**{**

tmp**=**List1**;**

List1**=**List1**->**next**;**

strcpy**(**L**->**Data**[**i**].**Num**,**tmp**->**key**.**Num**);**

strcpy**(**L**->**Data**[**i**].**Type**,**tmp**->**key**.**Type**);**

strcpy**(**L**->**Data**[**i**].**Name**,**tmp**->**key**.**Name**);**

strcpy**(**L**->**Data**[**i**].**Phone**,**tmp**->**key**.**Phone**);**

free**(**tmp**);**

**}**

还有索引查找的部分，将链表的内容复制到数组中，以每个字的第一个字母进行索引，输入查找时，判断第一个字符，然后跳到根据第一个字符所对应的序号，然后进行顺序查找。

//分块索引查找函数

Position BlockIndex**(**List L**,**char k**[])**

**{**

//初始化索引数组

int number**[**10**];**

int xx**;**

**for(**xx**=**0**;**xx**<**10**;**xx**++)**

**{**

number**[**xx**]=**10000**;**

**}**

inti**;**

**for(**i**=**0**;**i**<=**L**->**Last**;**i**++)**

**{**

**if(**number**[(**int**)**L**->**Data**[**i**].**Num**[**0**]-**48**]>**i**)**

**{**

number**[(**int**)**L**->**Data**[**i**].**Num**[**0**]-**48**]=**i**;**

**}**

**}**

**for(**i**=**number**[(**int**)**k**[**0**]-**48**];**i**<=**L**->**Last**;**i**++)**

**{**

**if(**strcmp**(**L**->**Data**[**i**].**Num**,**k**)==**0**)**

**{**

**return** i**;**

**}**

**}**

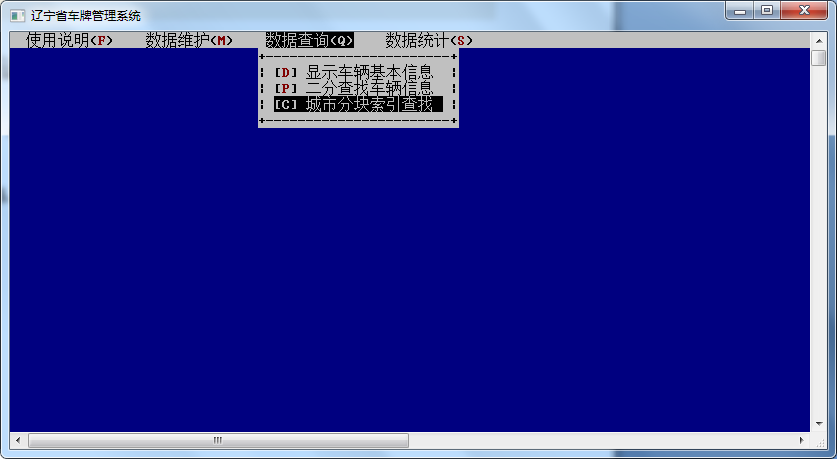
**return** **-**1**;**

**}**

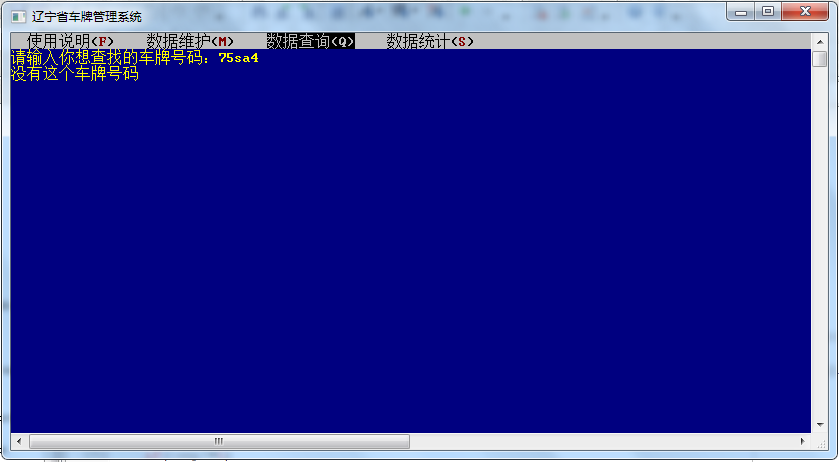
# 测试与调试

## 个人测试

### 闻亮测试



5-1界面测试



5-2二分法查找失败

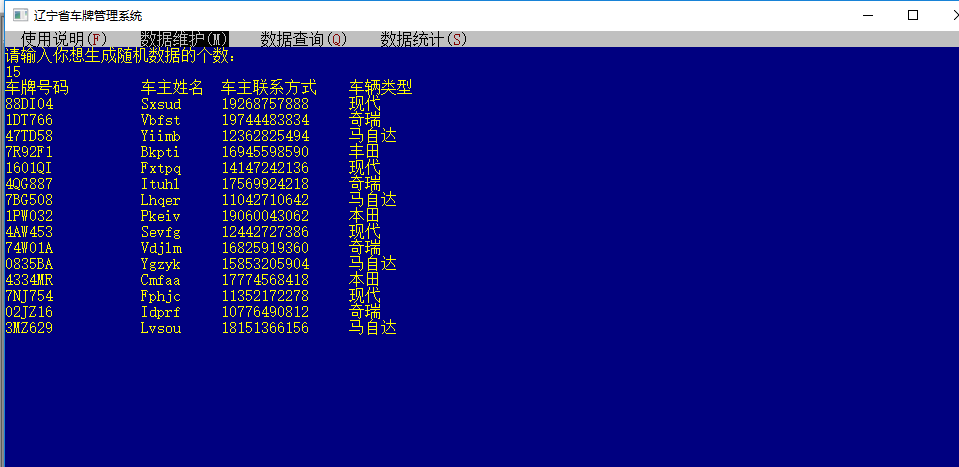


5-3分块查找

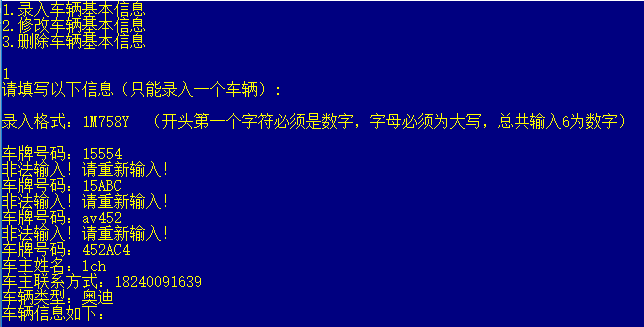


5-4模糊查找

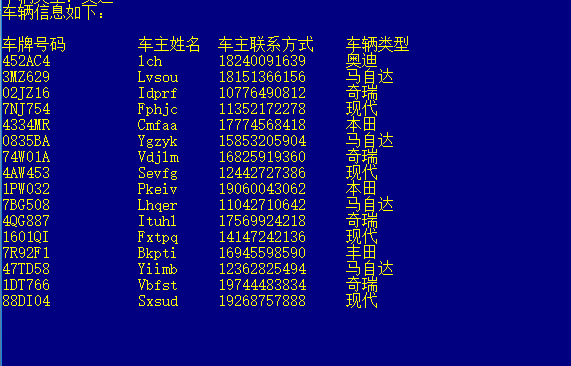
### 吕传昊测试



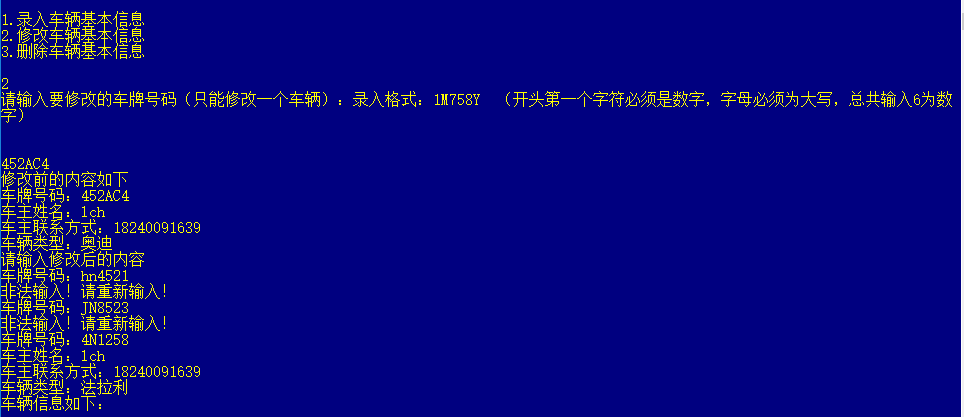
5-5随机生成15个信息，并录入文件中



5-6手动录入车辆信息，并进行非法判断



5-7信息显示

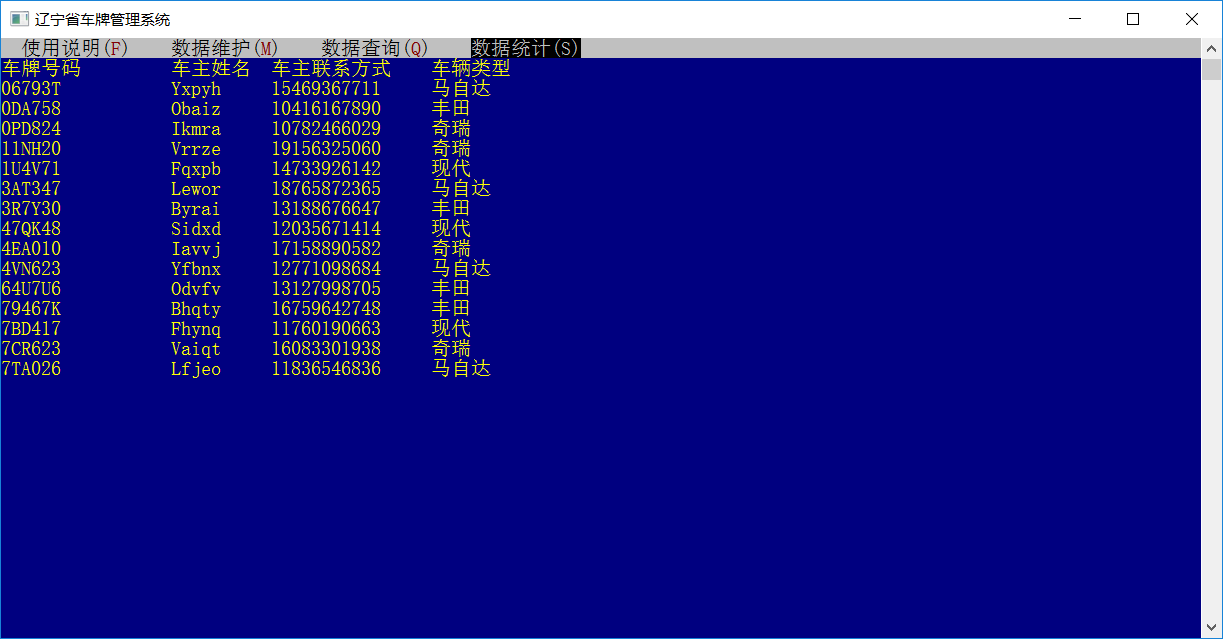


5-8修改车牌

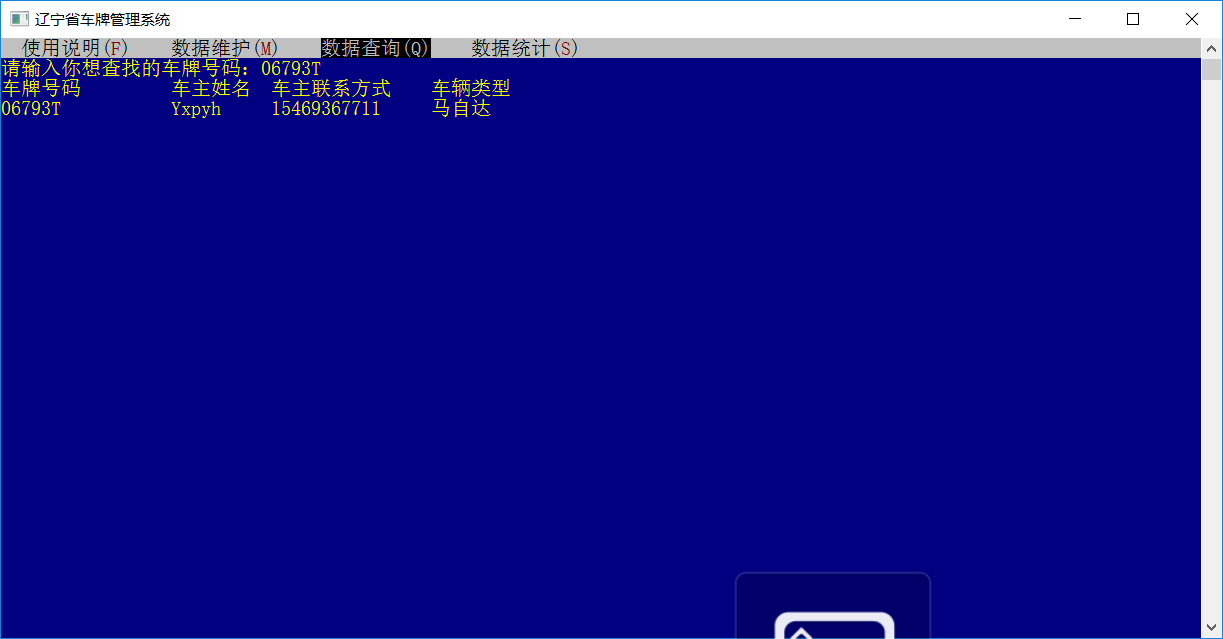


5-9删除信息

### 梁浩纯测试



5-10



5-11

# 课题总结

## 课题评价

汽车车牌录入是一个一直存在的社会问题，随着人口的增长和国民购买力的提高，录入的信息增加了数倍，如何找到一个既省空间，又快的系统十分重要。而大数据时代为这个问题提供了很好的思路，相信以后可以运用云数据库来解决。

## 团队协作

Teamwork自古以来都不可丢弃，一个人总是无法走很远。尤其是IT行业，巨大的工作量，复杂的逻辑思考，显然如果个人来做实在是力不从心，而且还浪费时间。通过teamwork可以快速有效达到这个目标，无论是对彼此的成长还是对社会的进步都有着举足轻重的作用。

## 个人总结

### 闻亮设计小结

通过这次学习让我对数据的处理有了更深刻的理解，以前的程序只考虑结果，不考虑维护和优化，有点顾头不顾尾。当面临这次较大的数据处理时，出现了很多以前没有见过的问题，比如文件读取的速率，反复读出会严重降低速率。而且代码的规范性同样也警示了我，如何写出便于别人理解，方便大家阅读的代码至关重要。非常感谢能和其他两个兄弟一起完成这次的任务，也非常感谢老师能给我们这次机会去锻炼自己。

### 吕传昊设计小结

在数据结构的课程设计中，对于我有3点学习和深刻意识到的事情。1.团队合作中大家的交流是不可缺少的，因为团队协作是三个人共同来完成的，只有大家交流到位才能有效的提高代码的质量，每个人的优缺点，擅长和不擅长的方向。2.数据结构中对于线性表和图的理解加深了一步，从课堂的偏理论学习到实际的代码实现有很大的不同。3.对于编程的认识，从大一到大二在专业的学习中，也更加明确自己的方向，更多的是在实际行动，代码编程上了解了更多的知识和内容。

### 梁浩纯设计小结

通过这一次的课程设计，我明白了界面设计在用户的体验中占了很大的一部分，好的界面更能满足用户的需求。同时，算法和数据结构是一个程序的核心，对数据结构和算法的掌握也是我们计算机专业的学生所必须掌握的。同时，努力做到在白板上写代码写成bug-free是一项极具挑战性的工作。

# 附录

## 课题任务分工

### 课题程序设计分工

**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 |
| 20164493 | 闻亮 | 二分法查找  分块查找  模糊插找 | 匹配相应的车牌号并把信息输出或者匹配相应的字段，并把所有有关信息输出 |
| 20164580 | 梁浩纯 | 基数排序  界面设计 | 将存储的车辆信息的车牌号码根据ASCII码的大小进行排序。 |
| 2014531 | 吕传昊 | 随机生成数据信息  手动录入、修改、删除数据信息 | 通过随机函数生成车牌号，车主信息等。设置非法输入检测，正确输入信息 |
|  |  |  |  |

### 课题报告分工

**课题报告分工**

|  |  |  |
| --- | --- | --- |
| 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 | 闻亮，梁浩纯 |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 | 闻亮 |
| 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 输入输出设计  3.5 主算法设计  3.6 用户界面设计 | 闻亮，梁浩纯，吕传昊 |
| 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（按组员分工）  4.3.1闻亮设计实现  4.3.2吕传昊设计实现  4.3.3梁浩纯设计实现 | 闻亮，吕传昊，梁浩纯 |
| 5 测试与调试 | 5.1 个人测试（按组员分工）  5.1.1闻亮测试  5.1.2吕传昊测试  5.1.3梁浩纯测试  5.2 组装与系统测试  5.3 系统运行 | 闻亮，吕传昊，梁浩纯 |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 下一步工作  6.4 个人设计心得（按组员分工）  6.4.1闻亮设计心得  6.4.2吕传昊设计心得  6.4.3梁浩纯设计心得 | 闻亮，吕传昊，梁浩纯 |

## 使用手册

主菜单快捷键使用说明：

Alt+F:使用说明

Alt+M:数据维护

Alt+Q:数据查询

Alt+S:数据统计

子菜单快捷键使用说明：

S/s:主菜单快捷键说明

B/b:子菜单快捷键说明

R/r:其他快捷键说明

D/d:车辆基本信息

P/p:随机生成数据

P/p:二分查找

C/c:城市分块索引查找

I/i:基于车牌号码的基数排序

T/t:模糊查找

其他快捷键使用说明：

Alt+X:退出系统

F1:显示部分快捷键使用方式

方向键:弹出菜单/选项移动

Enter:弹出菜单/点击确定

Esc:关闭菜单