西安電子科技大學

XIDIAN UNIVERSITY

计算训练项目

姓名：王梦祥

学号：20009100716

指导老师：苏博

实验日期：2021.10

1、实验题目

本实验题目为车辆管理系统，该软件主要是使用 C 语言设计开发一个简单的车辆管理系统，实现租赁信息的浏览、查询、修改、删除及录入等功能。该预期实现如下系统功能：浏览所有车辆信息、查询车辆具体信息、录入车辆全部信息、修改车辆具体信息、删除车辆部分信息和退出系统结束任务。

2、实验环境

电脑操作系统：windows11
编译环境：visual studio 2019
CPU：AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz
内存：16GB

3、实验步骤

具体实验思路可以如下图所示

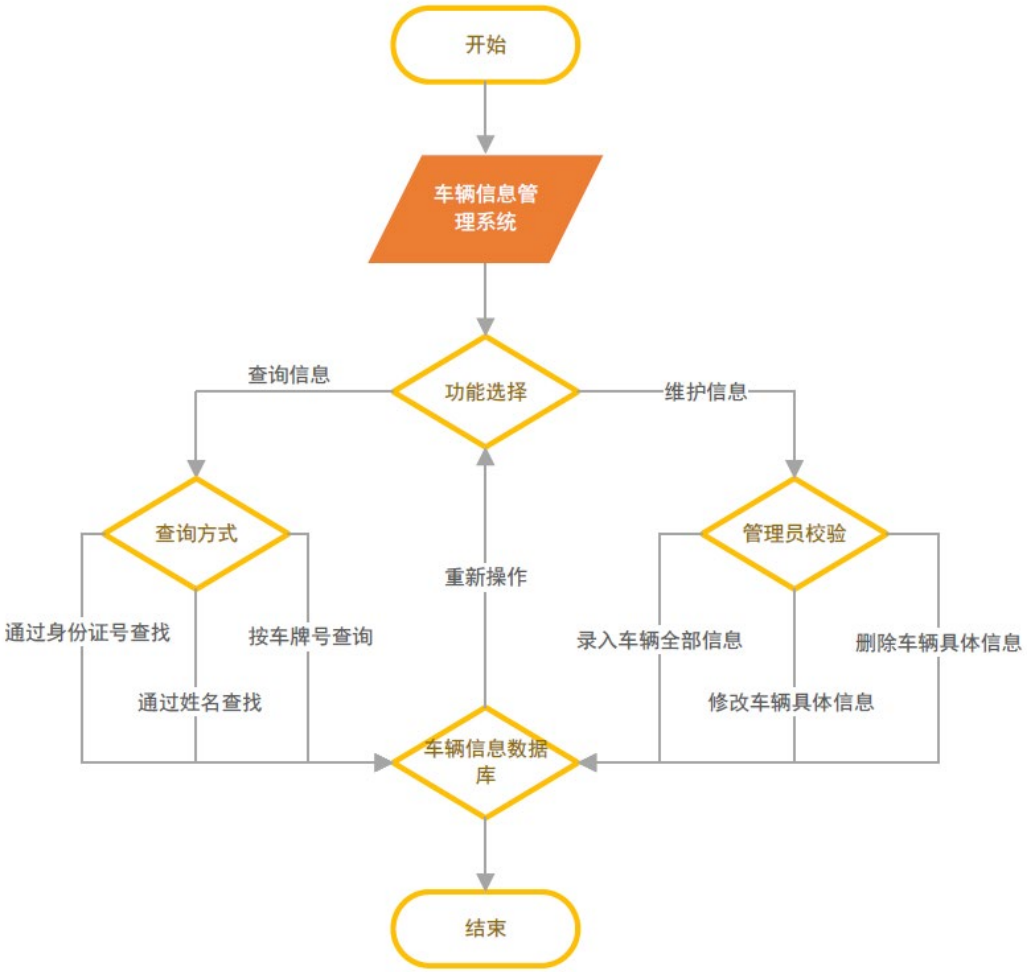

图 1 实验逻辑

4、实验代码

实验中主要涉及一个主函数，通过其他函数调用实现查询车辆具体信息、维护（录入、修改、删除）车辆信息。关键功能函数在本部分说明，详细代码见附录1。

```
//主函数
int main(int argc, char* argv[]) {
    int initFlag, quitFlag, maintainFlag, findFlag;
    char ch;
    quitFlag = 1;
    initFlag = InitStuInfo();
    if (initFlag)
        while (quitFlag) {
            printf("\n\t\t**********车辆信息查询系统**********\n");
            printf("\t\t              1.查询            *\n");
            printf("\t\t              2.维护            *\n");
            printf("\t\t              0.退出            *\n");
            printf("\t\t************************************\n");
            printf("请选择(1/2/0):");
            ch = _getche();
            switch (ch) {
            case'1':findFlag = Find();
                //调用查询模块
                break;
            case'2':maintainFlag = Maintain();
                //调用维护模块
                break;
            case'0':quitFlag = 0;
                break;
            default:printf("\n输入错误，请重新选择\n\n\n");
                break;
            }
        }
    else {
        printf("\n初始化车辆信息失败！");
        return 1;
    }
    printf("\n再见！请按任意键退出.");
    _getche();
    return 0;
}
```

程序首先会进行初始化，浏览所有的车辆信息，即：

```
int InitStuInfo(void)
```

当在交互界面输入“1”时，主函数会调用查询模块，分别涉及如下函数：

```
int Find(void)
int FindNumber(void)
int FindName(void)
int FindAddrApart(void)
```

这些分别表示查询主模块，以及分别查询模块（按学号、姓名、身份证号查询的模块）。

当在交互界面输入“2”时，主函数会调用维护模块，分别涉及如下函数：

```
int Maintain(void)
int VerificationIdentity(void)
int Repair(void)
int Save(void)
int Add(void)
```

这些分别表示维护主模块，以及维护人员校验模块、修改车辆信息模块、修改的车辆信息写入文件模块和增加车辆信息模块。

当在交互界面输入“3”时，主函数会调用删除模块，分别涉及如下函数：

```
int DeleteStu(void)
```

这个就表示将车辆信息删除模块。

4、实验结果

4.1 浏览所有车辆信息

当程序成功运行时，会显示出所有的车辆信息，分别表示为学号、姓名、身份证号。

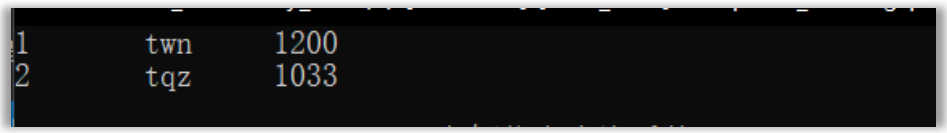

图 2 车辆信息概览

4.2 查询车辆具体信息

调出查询模块

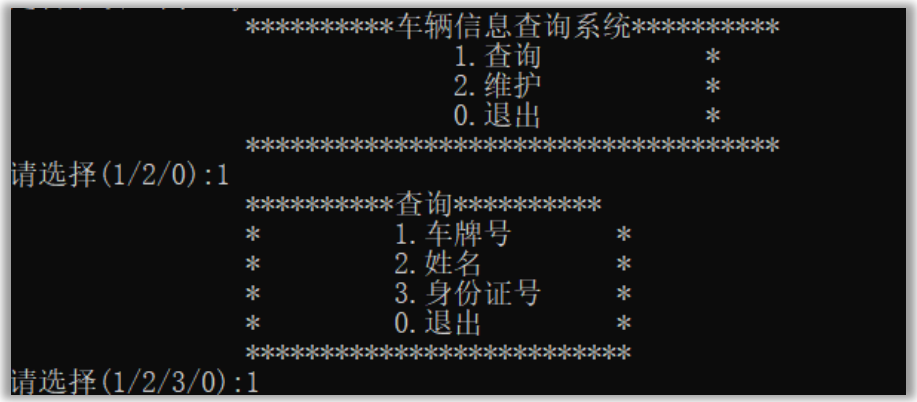

图 3 查询模块选择界面

4.2.1 通过车牌号查找

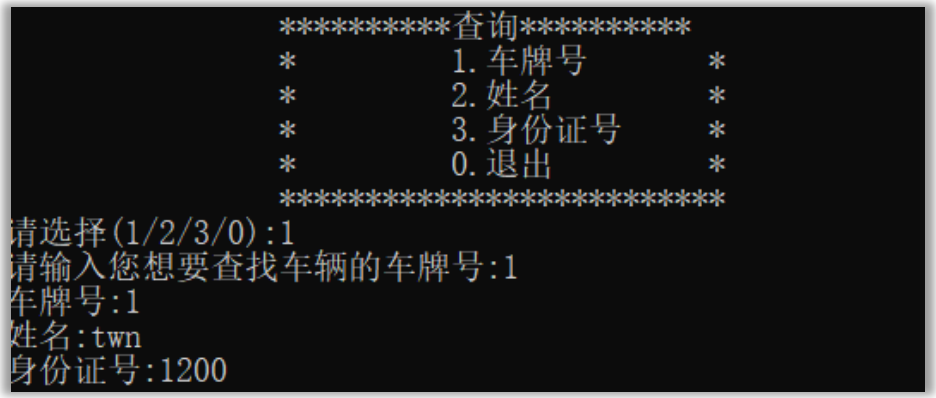

图 4 按车牌号查找的结果

4.2.2 通过姓名查找

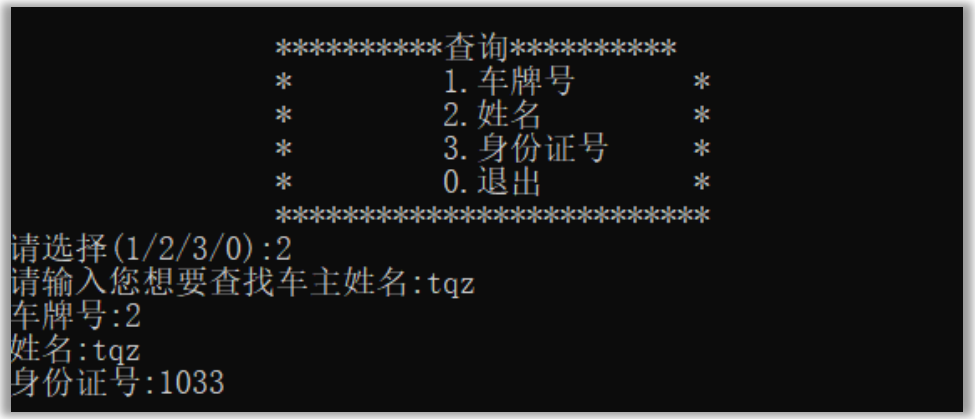

图 5 按姓名查找的结果

4.2.3 通过身份证号查找

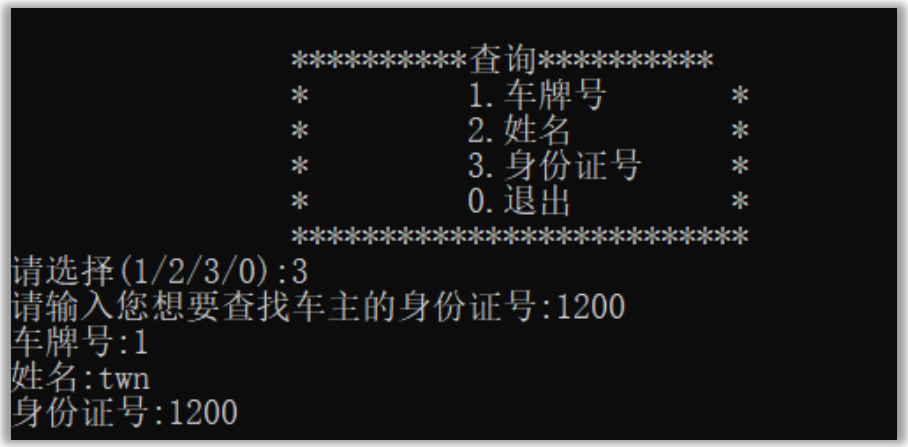

图 6 按身份证号查找的结果

4.3 维护车辆信息

首先要输入用户名和密码，进入管理员维护状态。

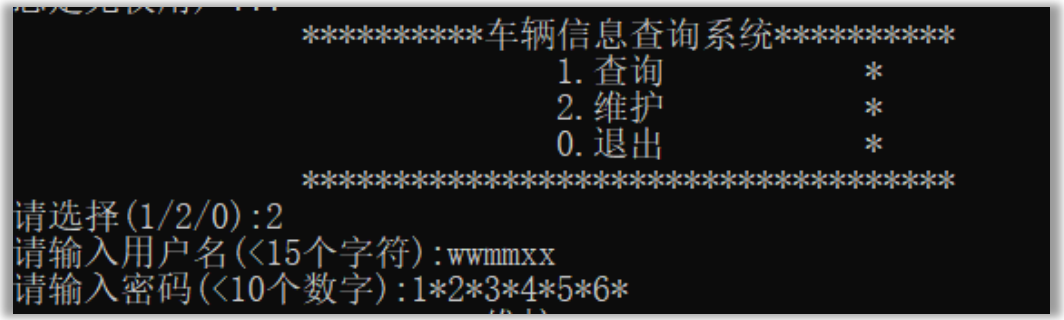

图 7 维护时管理员登录界面

4.3.1 录入车辆全部信息

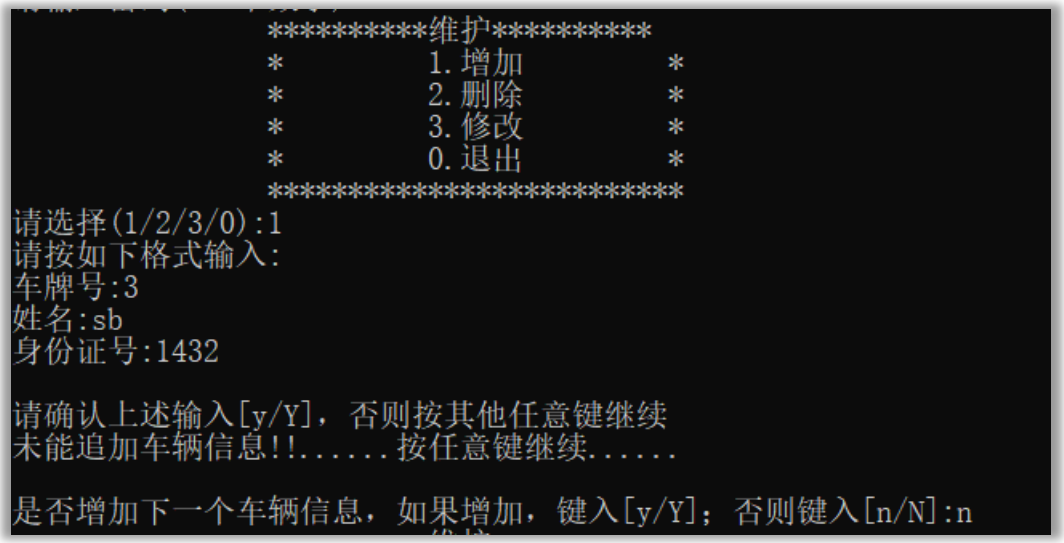

图 8 录入车辆信息过程

4.3.2 修改车辆具体信息

下图显示内容分别是车牌号、姓名和身份证号具体信息的修改过程。

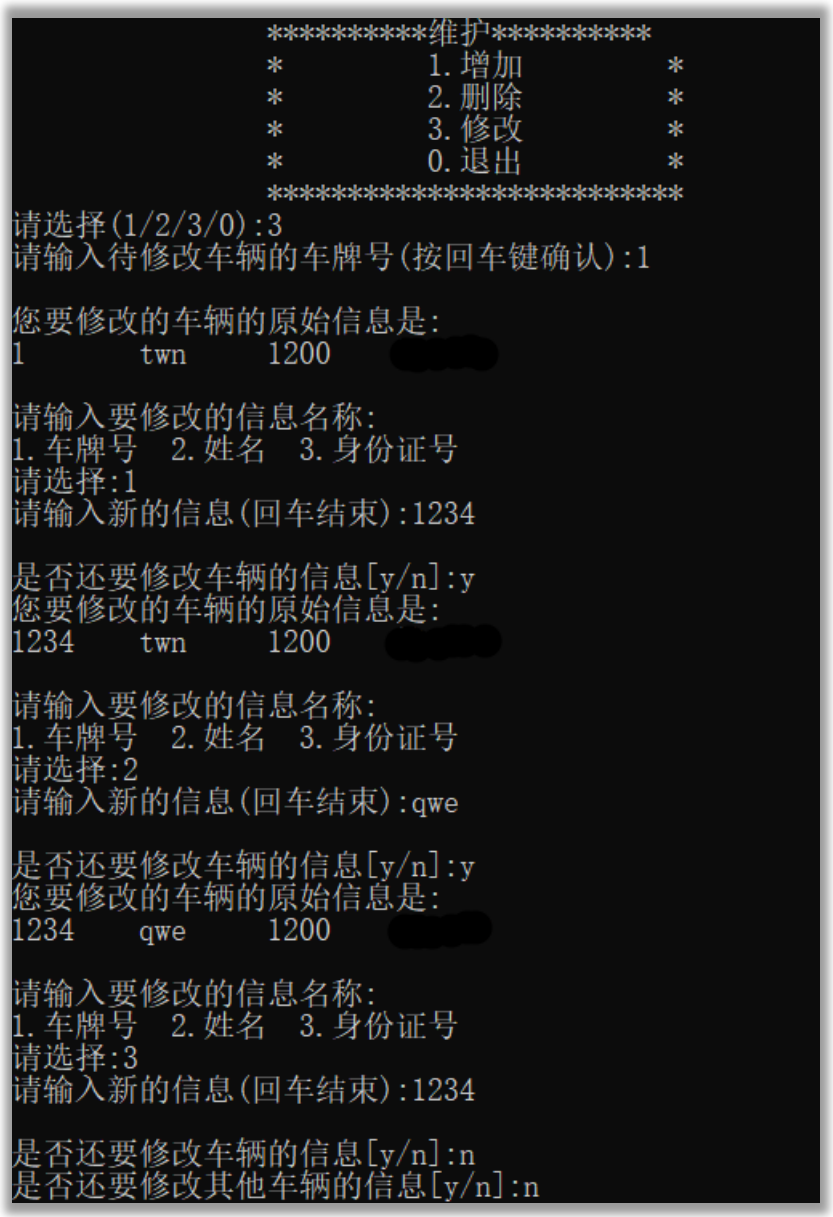

图 9 修改车辆信息过程（三种方式）

4.3.3 删除车辆部分信息

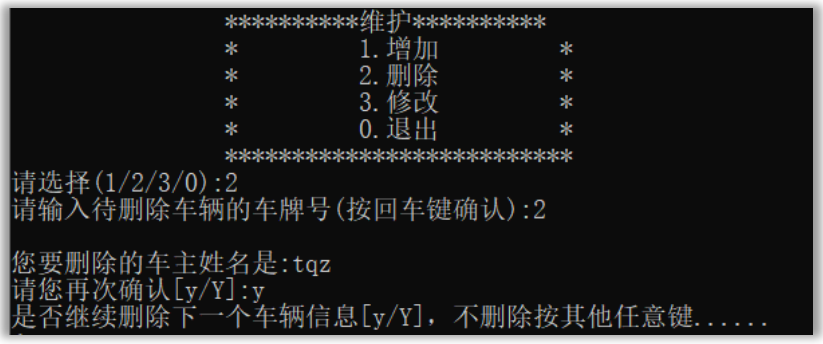

图 10 删除车辆全部信息

4.4 退出系统结束任务

逐步退出模块，并保存结果。

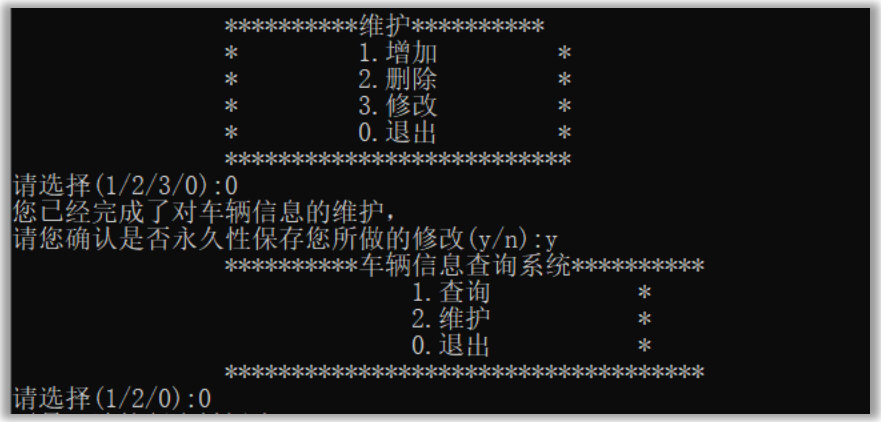

图 11 退出模块

本次操作之后，在文件中显示结果如下，可见程序成功执行。

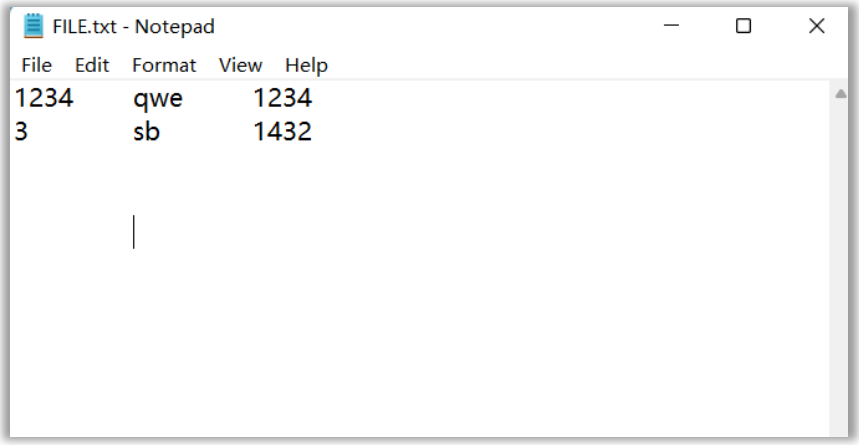

图 12 文件结果

5、感受与收获

总体来说，通过一个比较全面的程序编写经历，对一个有实践性系统的代码有了比较全面的把握，为之后多场景代码编写奠定了一个初步的基础。

这是一次比较全面的基础程序设计项目，提高了计算机项目的系统能力。车辆管理系统设计是一个基础的系统性设计，在设计过程中，会遇到许多问题，比如整个系统逻辑的设计、子系统实现的具体代码和协调运行、文档数据的读写解决方法等等。在整个系统逻辑的编写过程中，提高了我们的问题解决能力。

同时，这次项目也给予我们更大的信心，为未来设计更宏大软件工程项目奠定基础。

附录一：详细代码

```
#include <iostream>
#include<stdio.h>
#include<malloc.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#define LEN sizeof(struct car)

struct car {
    char No[10], name[10], ID[20];
    struct car* next;
};
struct car* head;

//函数声明
int Add(void);
int DeleteStu(void);
int Repair(void);
int VerificationIdentity(void);
int Maintain(void);
int FindName(void);
int FindNumber(void);
int Find(void);
int FindAddrApart(void);
int InitStuInfo(void);

//主函数
int main(int argc, char* argv[]) {
    int initFlag, quitFlag, maintainFlag, findFlag;
    char ch;
    quitFlag = 1;
    initFlag = InitStuInfo();
    if (initFlag)
        while (quitFlag) {
            printf("\n\t\t**********车辆信息查询系统**********\n");
            printf("\t\t              1.查询            *\n");
            printf("\t\t              2.维护            *\n");
            printf("\t\t              0.退出            *\n");
            printf("\t\t************************************\n");
            printf("请选择(1/2/0):");
```

```
            ch = _getche();
            switch (ch) {
            case'1':findFlag = Find();
                //调用查询模块
                break;
            case'2':maintainFlag = Maintain();
                //调用维护模块
                break;
            case'0':quitFlag = 0;
                break;
            default:printf("\n输入错误，请重新选择\n\n\n");
                break;
            }
        }
    else {
        printf("\n初始化车辆信息失败！");
        return 1;
    }
    printf("\n再见！请按任意键退出.");
    _getche();
    return 0;
}

//追加新的车辆信息
int Add(void) {
    char ch;
    int addFlag;
    struct car* p3, * p2;
    p2 = head;
    while (p2->next != NULL)
        p2 = p2->next;
    ch = 'y';
    while (tolower(ch) == 'y') {
        p3 = (struct car*)malloc(LEN);
        printf("\n请按如下格式输入:\n");
        printf("车牌号:");
        gets_s(p3->No);
        printf("姓名:");
        gets_s(p3->name);
        printf("身份证号:");
        gets_s(p3->ID);
        p3->next = NULL;
        printf("\n请确认上述输入[y/Y]，否则按其他任意键继续");
        ch = _getche();
```

```
        if (tolower(ch) == 'y') {
            p2->next = p3;
            p2 = p3;
            p3 = NULL;
            printf("\n追加车辆信息成功!!.....按任意键继续.....");
            _getche();
            _putch('\n');
            addFlag = 1;
        }
        else {
            printf("\n未能追加车辆信息!!.....按任意键继续.....");
            _getche();
            _putch('\n');
            addFlag = 2;
        }
        printf("\n是否增加下一个车辆信息，如果增加，键入[y/Y]；否则键入[n/N]:");
        ch = _getche();
    }
    p2 = NULL;
    return addFlag;
}   //Add

//删除车辆信息
int DeleteStu(void) {
    char ch, stuNo[10];
    int deleteFlag;
    struct car* p, * q;
    ch = 'y';
    while (tolower(ch) == 'y') {
        printf("\n请输入待删除车辆的车牌号(按回车键确认):");
        gets_s(stuNo);
        p = head;
        q = NULL;
        while (p != NULL) {
            if (strcmp(p->No, stuNo)) {
                q = p;
                p = p->next;
            }
            else {
                printf("\n您要删除的车主姓名是:%s\n", p->name);
                printf("请您再次确认[y/Y]:");
                ch = _getche();
                if (tolower(ch) == 'y') {
                    q->next = p->next;
```

```
                    deleteFlag = 3;
                    break;
                }
                else {
                    deleteFlag = 4;
                    break;
                }
            }
        }
        if (p == NULL) {
            printf("\n不存在您要删除的车辆信息，按任意键继续.....\n");
            deleteFlag = 5;
            _getche();
        }
        printf("\n是否继续删除下一个车辆信息[y/Y]，不删除按其他任意键.....\n");
        ch = _getche();
    }
    return deleteFlag;
}     //DeleteStu

//修改车辆信息
int Repair(void) {
    int repairFlag;
    char ch, stuNo[10];
    struct car* p, * q;
    do {
        printf("\n请输入待修改车辆的车牌号(按回车键确认):");
        gets_s(stuNo);
        p = head;
        q = NULL;
        while (p != NULL) {
            if (strcmp(p->No, stuNo))
                p = p->next;
            else break;
        }
        while (1) {
            printf("\n您要修改的车辆的原始信息是:\n");
            printf("%s\t%s\t%s\t%s\t%s\t\n", p->No, p->name, p->ID);
            printf("\n请输入要修改的信息名称:\n");
            printf("1.车牌号  2.姓名  3.身份证号\n");
            printf("请选择:");
            ch = _getche();
            switch (ch) {
            case'1':printf("\n请输入新的信息(回车结束):");
```

```
                gets_s(p->No); repairFlag = 6; break;
            case'2':printf("\n请输入新的信息(回车结束):");
                gets_s(p->name); repairFlag = 6; break;
            case'3':printf("\n请输入新的信息(回车结束):");
                gets_s(p->ID); repairFlag = 6; break;
            default:printf("\n输入错误，请重新输入\n");
                repairFlag = 5; break;
            }
            printf("\n是否还要修改车辆的信息[y/n]:");
            ch = _getche();
            if (tolower(ch) == 'y')
                continue;
            else
                break;
        }
        printf("\n是否还要修改其他车辆的信息[y/n]:");
        ch = _getche();
    } while (tolower(ch) == 'y');
    return repairFlag;
}    //Repair

//维护人员权限校检
int VerificationIdentity(void) {
    char userID[20], passWord[20];//存放由键盘输入的用户名和口令
    char superUID[20], passWD[20];//存放文件中读取的用户名和口令
    int i, legalUser;
    char ch;
    FILE* fp;
    legalUser = 0;
    fp = fopen("F:\\superUser.txt", "r");
    if (fp == NULL) {
        printf("\n权限文件不存在!按任意键继续.....\n");
        _getche();
    }           //superUserFlag=0;
    else {
        do {
            printf("\n请输入用户名(<15个字符):");
            i = 0;
            while (isalpha(ch = _getche()) && (i < 15)) {
                putchar(ch);
                userID[i] = ch;
                i++;
            }
            userID[i] = '\0';
```

```
            printf("\n请输入密码(<10个数字):");
            i = 0;
            while (isdigit(ch = _getche()) && (i < 10)) {
                putchar('*');
                passWord[i] = ch;
                i++;
            }
            passWord[i] = '\0';
            rewind(fp);
            while (!feof(fp)) {
                fscanf(fp, "%s\t%s\t\n", superUID, passWD);
                //从权限文件中读取用户名及密码
                if ((strcmp(userID, superUID) == 0) && (strcmp(passWord, passWD) ==
0)) {
                    //判断从键盘输入的用户名和密码是否与权限规定相符
                    legalUser = 1;
                    break;
                }
            }
            if (legalUser)
                break;
            else {
                printf("\n是否重新输入用户名和密码?(y/n)");
                ch = _getche();
            }
        } while ((ch == 'y') || (ch == 'Y'));
    }
    return legalUser;
}       //VerificationIdentity

//将修改的车辆信息写入文件
int Save(void) {
    FILE* fp;
    struct car* p;
    p = head;
    if ((fp = fopen("F:\\FILE.txt", "w")) == NULL) {     //以“写”方式打开车辆信息文件
        printf("\n保存文件不正常，请核对文件名!\n");
        fclose(fp);
        return 1;
    }
    else {
        rewind(fp);
        while (p != NULL) {
            fprintf(fp, "%s\t%s\t%s\t", p->No, p->name, p->ID);
```

```
            p = p->next;
        }
        fclose(fp);
        return 0;
    }
}   //Save

//维护模块
int Maintain(void) {
    char ch;
    int success, saveFlag, maintainR = 4;
    success = VerificationIdentity();    //用户权限校检
    if (success == 0) {
        printf("\n您是无权用户!!!");
        maintainR = 5;                   //maintainR=5:无权维护
    }
    else {
        do {
            printf("\n\t\t**********维护**********\n");
            printf("\t\t*         1.增加         *\n");
            printf("\t\t*         2.删除         *\n");
            printf("\t\t*         3.修改         *\n");
            printf("\t\t*         0.退出         *\n");
            printf("\t\t**************************\n");
            printf("请选择(1/2/3/0):");
            ch = _getche();
            switch (ch) {
            case'1':maintainR = Add();//maintainR=1:增加车辆的信息
                break;
            case'2':maintainR = DeleteStu();//maintainR=2:删除车辆的信息
                break;
            case'3':maintainR = Repair();//maintainR=3:修改车辆信息
                break;
            case'0':maintainR = 0;
                break;
            default:printf("\n输入错误，请重新输入您的选择:");
                break;
            }
        } while (maintainR != 0);
        printf("\n您已经完成了对车辆信息的维护，\n");
        printf("请您确认是否永久性保存您所做的修改(y/n):");
        ch = _getche();
        if (tolower(ch) == 'y') {
            saveFlag = Save();               //保存修改的车辆信息
```

```
            if (saveFlag)maintainR = 4;        //maintain=4:成功保存
        }
    }
    return maintainR;
}   //Maintain

//按身份证号查询
int FindAddrApart(void) {
    char saddress[20];
    struct car* p;
    int flag;
    flag = 5;                                  //查找失败标志
    p = head;
    printf("\n请输入您想要查找车主的身份证号:");
    gets_s(saddress);
    while (p != NULL) {
        if (strcmp(saddress, p->ID) == 0) {
            printf("车牌号:%s\n", p->No);
            printf("姓名:%s\n", p->name);
            printf("身份证号:%s\n", p->ID);
            flag = 1;                          //查找成功
        }
        p = p->next;
    }
    return flag;
}   //FindAddrApart

//按姓名查询
int FindName(void) {
    char sname[20];
    struct car* p;
    int flag;
    flag = 5;                                  //查找失败的标志
    p = head;
    printf("\n请输入您想要查找车主姓名:");
    gets_s(sname);
    while (p != NULL) {
        if (strcmp(sname, p->name) == 0) {
            printf("车牌号:%s\n", p->No);
            printf("姓名:%s\n", p->name);
            printf("身份证号:%s\n", p->ID);
            flag = 1;                          //查找成功
        }
        p = p->next;
```

```
    }
    return flag;
}     //FindName

//按学号查询
int FindNumber(void) {
    char sno[10];
    struct car* p;
    int flag;
    flag = 5;                                   //查找失败的标志
    p = head;
    printf("\n请输入您想要查找车辆的车牌号:");
    gets_s(sno);
    while (p != NULL) {
        if (strcmp(sno, p->No) == 0) {
            printf("车牌号:%s\n", p->No);
            printf("姓名:%s\n", p->name);
            printf("身份证号:%s\n", p->ID);
            flag = 1;                           //查找成功
        }
        p = p->next;
    }
    return flag;
}    //FindNumber

//查询模块
int Find(void) {
    char ch;
    int flag;
    flag = 4;
    while (flag) {
        printf("\n\t\t**********查询**********\n");
        printf("\t\t*         1.车牌号        *\n");
        printf("\t\t*         2.姓名          *\n");
        printf("\t\t*         3.身份证号      *\n");
        printf("\t\t*         0.退出          *\n");
        printf("\t\t**************************\n");
        printf("请选择(1/2/3/0):");
        ch = _getche();
        switch (ch) {
        case'1':flag = FindNumber(); break;
        case'2':flag = FindName(); break;
        case'3':flag = FindAddrApart(); break;
        case'0':flag = 0; break;
```

```
            default:printf("\n输入错误，请重新输入:"); flag = 4; break;
            }
            if (flag == 5) {
                printf("您所查找的车辆不存在!\n");
                printf("是否继续查找？[y/n]");
                ch = _getche();
                if (tolower(ch) == 'y')
                    continue;
                else flag = 0;
            }
        }
        return flag;
}       //Find

//初始化车辆信息。从文件中读取车辆信息，用链表存放
int InitStuInfo(void) {
    FILE* fp;
    struct car* p1;
    struct car* p2 = NULL;
    fp = fopen("F:\\FILE.txt", "r");          //以“读”方式打开车辆信息文件
    if (fp == NULL) {
        printf("未能初始化车辆信息\n");
        fclose(fp);
        return 0;
    }
    else {
        p1 = (struct car*)malloc(LEN);
        head = p1;
        while (!feof(fp)) {
            fscanf(fp, "%s\t%s\t%s\t", p1->No, p1->name, p1->ID);
            p2 = p1;
            p1 = (struct car*)malloc(LEN);
            p2->next = p1;
        }
        p2->next = NULL;
        p2 = NULL;
        free(p1);
        p1 = NULL;
        p1 = head;
        while (p1 != NULL) {                          //显示链表中的车辆信息
            printf("%s\t%s\t%s\t\n", p1->No, p1->name, p1->ID);
            p1 = p1->next;
        }
        fclose(fp);
```

```
            return 1;
        }
    }     //InitStuInfo
```