

**高级语言程序设计**

**实训报告**

**实训题目： 汽车索赔管理程序**

**专 业：**  **计算机科学与技术**

**班 级: 计算机222**

**姓 名： 章崇文**

**成 绩：**

**完成日期: 2022-02-03**

**指导教师: 史旅华 齐心**

**目 录**

[**高级语言程序设计** 1](#_Toc127906007)

[**实训报告** 1](#_Toc127906008)

[一 实训题目与要求 3](#_Toc127906009)

[1.1题目描述 3](#_Toc127906010)

[1.2题目要求： 3](#_Toc127906011)

[1.3输入要求： 3](#_Toc127906012)

[1.4实现要求： 3](#_Toc127906013)

[1.5扩展功能： 4](#_Toc127906014)

[二 系统设计 5](#_Toc127906015)

[2.1系统函数划分及声明 5](#_Toc127906016)

[2.2系统模块划分 7](#_Toc127906017)

[2.2.1链表实现： 8](#_Toc127906018)

[2.2.2链表头插法 8](#_Toc127906019)

[2.3数据结构设计 8](#_Toc127906020)

[2.3.1 宏定义： 8](#_Toc127906021)

[2.3.2 单位链表定义： 8](#_Toc127906022)

[struct Node 8](#_Toc127906023)

[{ 8](#_Toc127906024)

[struct ClimeData data; 8](#_Toc127906025)

[struct Node\* next; 8](#_Toc127906026)

[}; 8](#_Toc127906027)

[2.3.3 索赔信息结构体定义： 8](#_Toc127906028)

[2.3.4 全局变量及结构体指针的定义：、 9](#_Toc127906029)

[2.4主函数的流程分析与设计： 9](#_Toc127906030)

[2.5主要函数模块流程分析与测试： 9](#_Toc127906031)

[三 程序测试和运行结果 10](#_Toc127906032)

[3.0文件txt操作函数 10](#_Toc127906033)

[3.1 welcome界面 11](#_Toc127906034)

[3.2用户交互界面 12](#_Toc127906035)

[3.3读取课设函数 13](#_Toc127906036)

[3.4 case 1 显示课设需求 13](#_Toc127906037)

[3.5 case 2 查询索赔信息 14](#_Toc127906038)

[3.6 case 3 查询车辆索赔历史 15](#_Toc127906039)

[3.7 case 4 修改信息函数 17](#_Toc127906040)

[3.8 case 5 删除索赔信息 19](#_Toc127906041)

[3.9 case 6 添加索赔信息 21](#_Toc127906042)

[3.10 case 7 查看统计信息 22](#_Toc127906043)

[3.11 case 8 退出程序 25](#_Toc127906044)

[3.12 检测时间合法函数 25](#_Toc127906045)

[四 课程设计归纳总结 27](#_Toc127906046)

[五 源码展示 28](#_Toc127906047)

# 一 实训题目与要求

## 1.1题目描述

已知在汽车销售后，服务站在对故障汽车维修后，可以向汽车生产厂家进行索赔，设计一个汽车索赔管理程序，以方便汽车生产厂家对汽车售后索赔进行管理，该程序完成如下功能：

(1)能从键盘输入索赔信息。

(2)指定索赔编号，显示索赔信息。

(3)指定底盘号，显示索赔历史。

(4)给定索赔编号，修改索赔信息。

(5)给定索赔编号，删除索赔信息。

## 1.2题目要求：

(1)按照分析、设计、编码、调试、测试的软件过程完成这个应用程序。

(2)索赔信息包括：索赔编号、服务站名称、底盘号、审核人、索赔金额、索赔日期。

(3)为各项操作功能设计一个菜单，应用程序运行后，先显示这个菜单，然后用户通过菜单项选择希望进行的操作项目。

## 1.3输入要求：

(1)应用程序运行后在屏幕上显示一个菜单。用户可以根据需求，选定相应的操作项目。进入每个操作后，根据应用程序的提示信息，从键盘输入相应的信息。程序根据用户输入的信息完成相应的处理，实现要求的功能。

(2)能对输入的数据进行简单的校验，例如，索赔日期必须是一个合法的日期格式，索赔编号是唯一的（一个索赔编号对应一次索赔的索赔信息）

输出要求：

(1)应用程序运行后，要在屏幕上显示一个菜单，多条记录分屏显示。

(2)要求用户输入数据时，给出清晰、明确的提示信息，包括输入的数据内容、格式以及结束方式等。

(3)在程序完成处理后，要清楚地给出程序的处理结果。例如，在给定索赔编号删除索赔信息时，如果该索赔不存在，要提示没能删除，如果删除成功要提示删除成功。

## 1.4实现要求：

(1)在程序中使用链表存储索赔信息。

(2)采用模块化程序设计的方法，将程序中的各项功能用函数实现。

## 1.5扩展功能：

(1)提供一些统计分析功能。例如按服务站、审核人、索赔日期对索赔金额进行统计分析。

(2)索赔信息从文件读入，可以多次导入且能够保证记录的唯一性，可将索赔信息保存到文件中。

# 二 系统设计

## 2.1系统函数划分及声明

2.1.1 欢迎模块 构造函数void welcome();起界面装饰作用，说明制作人和指导老师。

2.1.2 主界面模块 构造函数void makeMenu();实现系统主菜单的显示。

2.1.3 创造表头struct Node\* createHead();创造表头，动态申请内存，储存表信息。

2.1.4 创造节点struct Node\* createNode(struct ClimeData data);储存单位表索赔信息。

2.1.5 插入数据void insertNodeBYHead(struct Node\* headNode , struct ClimeData data) 把单位表信息用表头法插入。

2.1.6 插入数据void insertNodeBYTail(struct Node\* headNode , struct ClimeData data) 把单位表信息用表尾法插入。

2.1.7删除信息void deleteNodeBYClimeID(struct Node\* headNode , char \* DataId) 指定位置删除，给定索赔编号，删除索赔信息。

2.1.8 查找信息struct Node\* searchByCarVIN(struct Node\* headNode,char\* CarVIN) 通过车辆底盘号查找信息。

2.1.9 查找历史信息void searchAndprintfBYCarVIN(struct Node\* headNode,char\* CarVIN) 利用底盘号查找当前历史信息。

2.1.10 查找信息struct Node\* searchByCarID(struct Node\* headNode,char\* ID)

利用编号查找信息。

2.1.11 数据排序void bubbleSortList(struct Node\* headNode) 冒泡排序（价格小到大）

2.1.12 数据检测 int checkID(char \* ID)检测编号的合法性。

2.1.13 统计金额 int StatisticalAmountByName(struct Node\* headNode,char\* name)

通过检查站名统计金额。

2.1.14 统计金额 int StatisticalAmountByReviewer(struct Node\* headNode,char\* name) 通过审核员统计金额。

2.1.15统计金额int StatisticalAmountByTime(struct Node\* headNode,char\* Time)

通过日期统计金额。

2.1.16 文件操作 void saveInfoFile(const char\* fileName,struct Node\* headNode) 储存链表信息到txt文档里面。

2.1.17 文件操作 void  readInfoFile(const char\* fileName,struct Node\* headNode)

读取txt文件里面信息到链表里。

2.1.18打印信息 void printfList(struct Node\* headNode) 打印索赔信息。

2.1.19 用户交互函数 void keyDown()

2.1.20 主函数 int main()

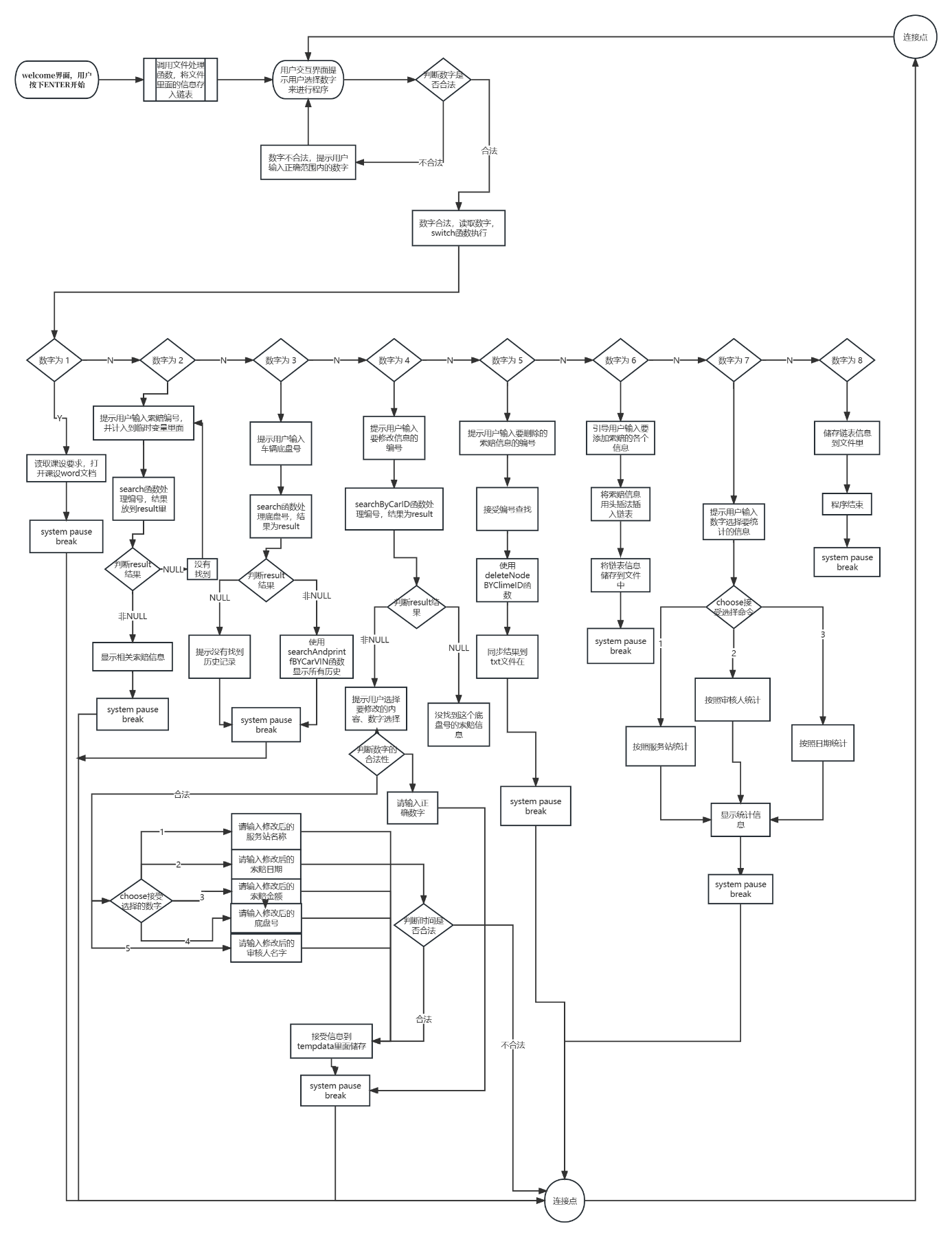
2.1.21 自己写的字符串函数void substr(char dest[], char src[], int pos, int length)

2.1.22 判断年份是否是闰年int IsLeapYear(int year)

2.1.23 提取当前时间 void Nowtime()

2.1.24 判断时间合法性 int  checkTrueTime(char \*time )

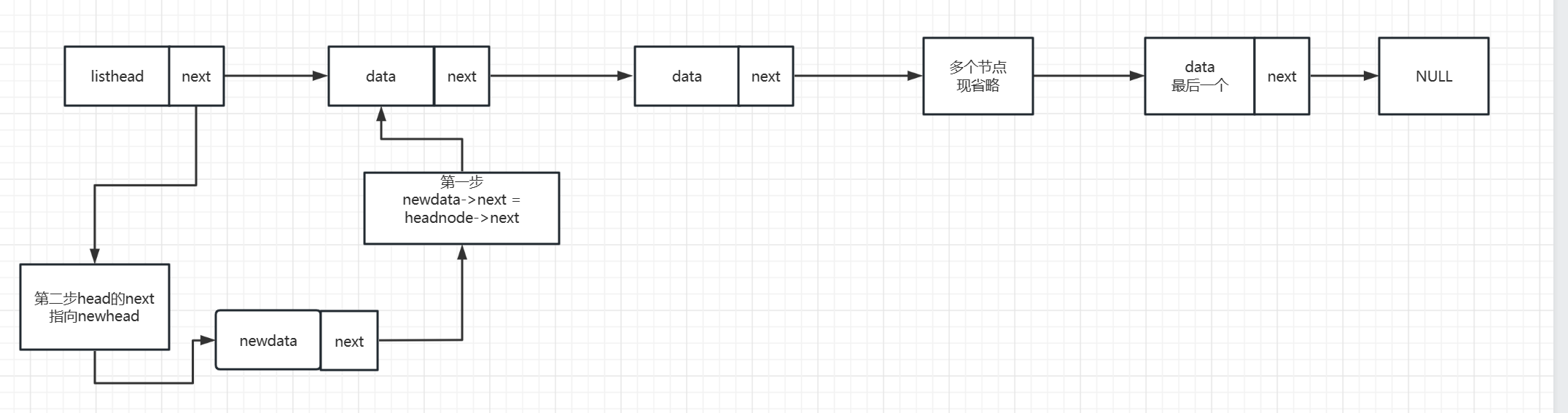
## 2.2系统模块划分



以上为系统模块划分图

### 2.2.1链表实现：

### 2.2.2链表头插法



## 2.3数据结构设计

2.3.1 宏定义：

#define MAXNAME 30

#define MAXDAY 20

#define MAXID 20

typedef long long ll;

int Nowyear,Nowmonth,Nowday;

int monthday[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

2.3.2 单位链表定义：

struct Node

{

struct ClimeData data;

struct Node\* next;

};

2.3.3 索赔信息结构体定义：

struct ClimeData

{

    char ServiceStationName[MAXNAME];//服务站名称 字符串类型

    char  id[MAXID];//索赔编号 字符串类型

    int ClaimAmount;//索赔金额 整数类型

    char ClaimTime[MAXDAY]; //索赔日期 字符串类型

    char CarVIN[MAXID];//底盘号 字符串类型

    char Reviewer[MAXNAME];//审核人 字符串类型

};

2.3.4 全局变量及结构体指针的定义：、

struct Node\* list = NULL;//使用全局链表

## 2.4主函数的流程分析与设计：

2.4.1主函数流程简述与设计

对于课设的各项要求，我准备使用分布函数来简化流程，去除重复代码，减少bug。提高程序的可维护，可读，可改的能力。程序以主函数 int main() 为入口，首先调用 void welcome() 函数，显示系统欢迎界面。然后显示主菜单界面，其次调用 readInfoFile("ClimeData.txt",list)函数，从文件读入索赔信息到链表。也可以用户自己加入文件，一样可以读入。在循环里使用makeMenu()函数和keyDown()函数和用户进行交互，交互中将储存信息到文件里，最后结束程序。

以下是设计源码：

//主函数

int main()

{

    welcome();

    list = createHead();

    readInfoFile("ClimeData.txt",list);//读入数据

    while(1)

    {

        makeMenu();

        keyDown();

        system("pause");

        system("cls");

    }

    system("pause");

    return 0;

}

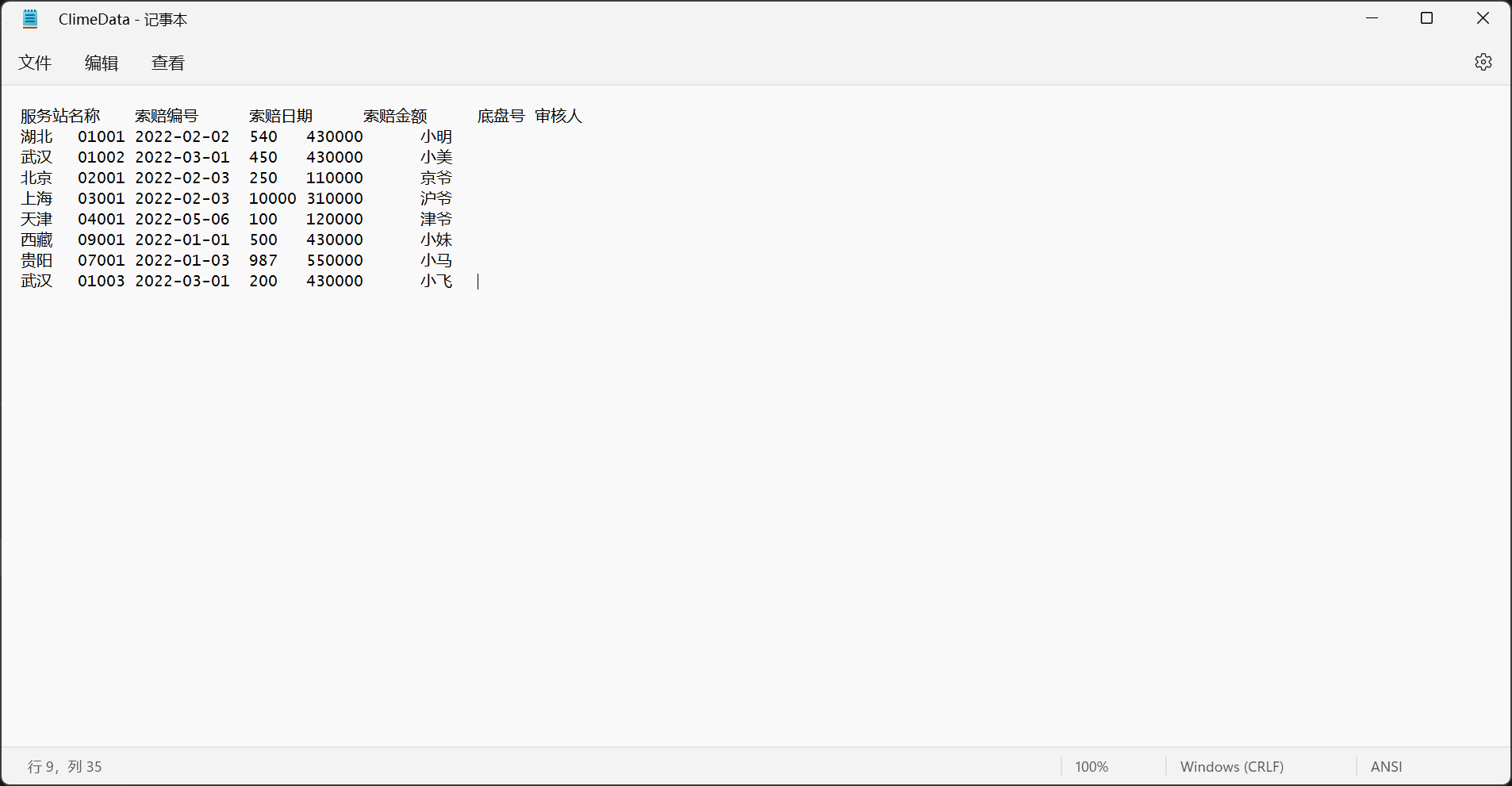
## 2.5主要函数模块流程分析与测试：

主要函数分析就是程序的各个需求测试，我将在第三部分程序测试和运行结果里面统一介绍。

# 三 程序测试和运行结果

## 3.0文件txt操作函数

3.0.1下面是txt文件：



3.0.2文件操作函数：

//文件操作 写入文件

//写入文件

void saveInfoFile(const char\* fileName,struct Node\* headNode)

{

FILE \*fp = fopen(fileName,"w+");

struct Node\* pMove = headNode->next;

fprintf(fp,"服务站名称\t索赔编号\t索赔日期\t索赔金额\t底盘号\t审核人\t\n");

while(pMove != NULL)

{

fprintf(fp,"%s\t%s\t%s\t%d\t%s\t%s\t\n",pMove->data.ServiceStationName,pMove->data.id,pMove->data.ClaimTime,pMove->data.ClaimAmount,

pMove->data.CarVIN,pMove->data.Reviewer);

pMove = pMove->next;

}

fclose(fp);

}

//文件读操作

void readInfoFile(const char\* fileName,struct Node\* headNode)

{

FILE \*fp = fopen(fileName,"r");

if(fp == NULL)//文件不纯在就创建文件

{

fp = fopen(fileName,"w+");

}

struct ClimeData tempData;

fscanf(fp,"%\*[^\n]");

while

(

(fscanf(fp,"%s\t",tempData.ServiceStationName) != EOF) &&(fscanf(fp,"%s\t",tempData.id) != EOF) &&

(fscanf(fp,"%s\t",tempData.ClaimTime) != EOF) && (fscanf(fp,"%d",&tempData.ClaimAmount)) &&

(fscanf(fp,"%s\t",tempData.CarVIN) != EOF) && (fscanf(fp,"%s\n",tempData.Reviewer))

)

{

insertNodeBYHead(list,tempData);

}

fclose(fp);

}

文件读入和写入函数，说实话，没什么好解释的运用好fprintf和fscanf函数即可实现这一切，使用fscanf(fp,"%\*[^\n]")跳过第一行的解释数据。其实文件操作里面的知识非常深，格式要求严格，还有很多的知识我没体现在程序里，在第一次写的时候遇到了fscanf函数的段错误Segment fault。详细解决的过程我放在了个人的博客里面。程序和人总要跑一个的。

## 3.1 welcome界面

void welcome()

{

system("color F0") ;

printf("\n\n\n\n\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* 欢 迎 使 用 汽 车 索 赔 管 理 系 统 ! \*\* \n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* \t\t\t\t\t\t\t\t \*\*\n") ;

printf("\t\t \*\* 制 作 人：章崇文 指 导 教 ；史旅华 齐心 \*\*\n") ;

printf("\t\t \*\* 学校 湖北汽车工业学院 班级 计算机 222 \*\*\n") ;

printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\* 请 按 ENTER 键 进 入 系 统\*\*\*\*\*\*\*\*\*\*\*\*\n") ;

printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

fflush(stdin) ;

getchar() ;

system("cls") ;

return ;

}

这是我程序中的欢迎模块，在一开始显示，主要是装饰和打上人工水印的作用。使用system函数发出一个DOS命令，用于执行系统命令，使背景和前景改变颜色，还有system("cls")和fflush(stdin) 函数，更合适观看，

下面是运行结果：



## 3.2用户交互界面

void makeMenu()//菜单界面

{

printf("\n\n\t\t--------------------成 功 进 入 系 统 !--------------------\n");

printf("\t\t--------------------【以 下 是 菜 单：】--------------------\n");

printf("\t\t--------------- 1.0读取课设要求 ---------------\n");

printf("\t\t--------------- 2.0查询索赔信息 ---------------\n");

printf("\t\t--------------- 3.0查询车辆索赔历史 ---------------\n");

printf("\t\t--------------- 4.0修改索赔信息 ---------------\n");

printf("\t\t--------------- 5.0删除索赔信息 ---------------\n");

printf("\t\t--------------- 6.0添加索赔信息 ---------------\n");

printf("\t\t--------------- 7.0查看统计信息 ---------------\n");

printf("\t\t--------------- 8.0退出该程序 ---------------\n");

printf("\t\t------------------------------------------------------------\n");

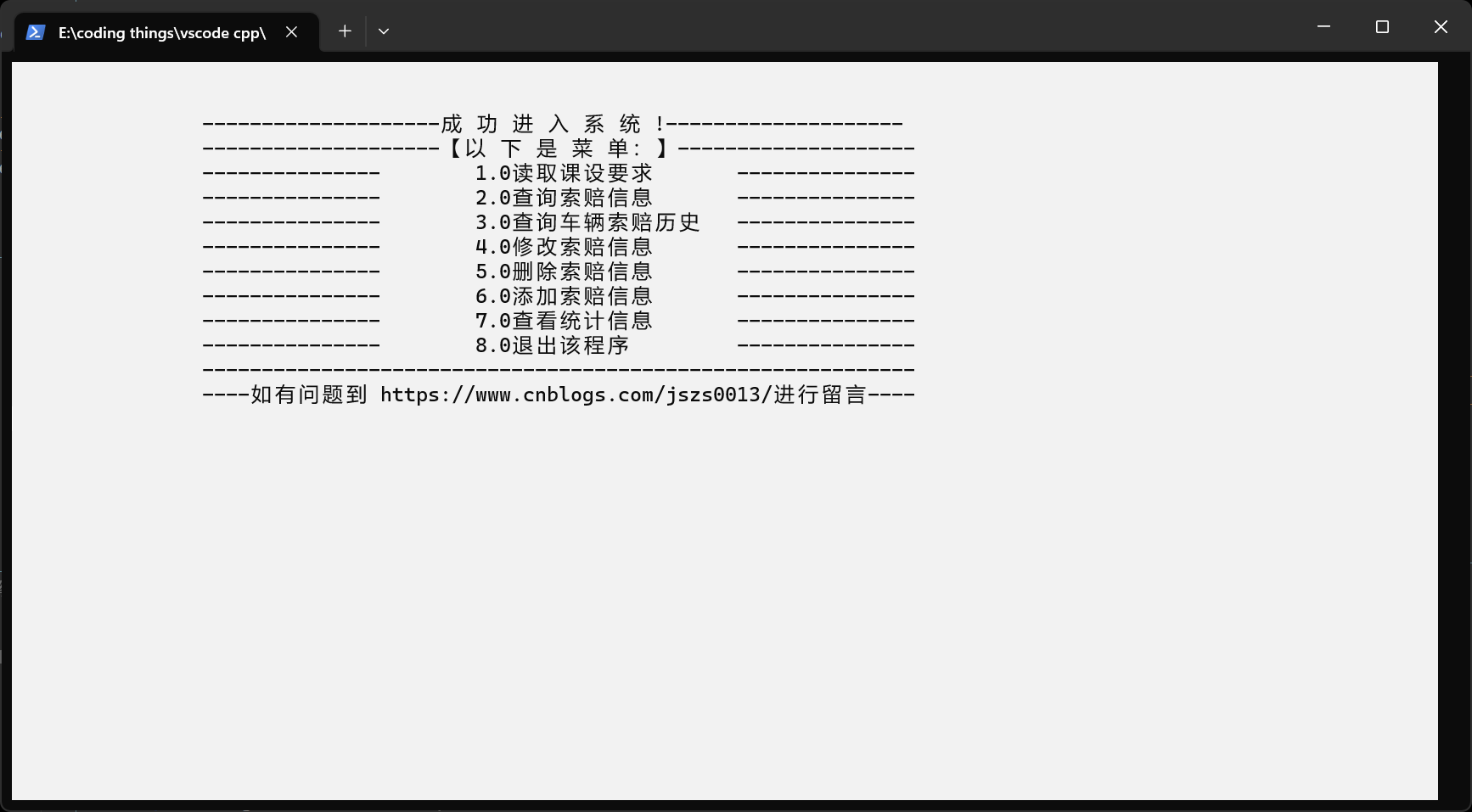
printf("\t\t----如有问题到 https://www.cnblogs.com/jszs0013/进行留言----\n");

return ;

}

也可以称作为菜单界面，主要是提供选择。用户交互界面和函数void KeyDown完美交互。这里提供了我的博客，遇到的大部分问题都解决并且放入博客方便观看。

下面是运行结果：



## 3.3读取课设函数

菜单交互界面互相融合，是本函数最重要的部分。里面每一个case函数在后面的选择函数中会体现，现在就先不复制源代码以节省空间，在最后的源代码阶段会体现。

无测试画面。

## 3.4 case 1 显示课设需求

case 1:

printf("----------读取课设要求----------\n");

//system("E:\\coding things\vscode cpp\\课设要求.docx");

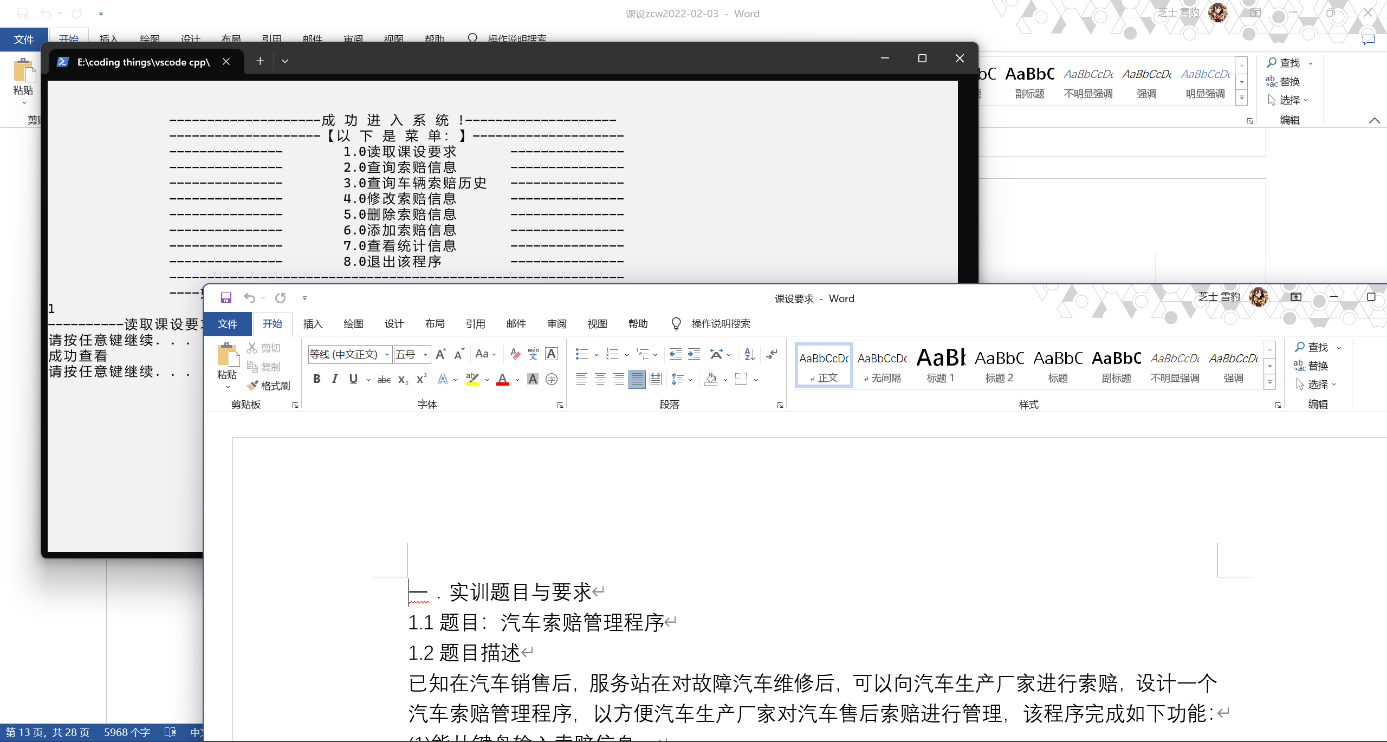
system("pause");

printf("成功查看\n");

break;

使用cmd指令打开同一文件夹下的word文档。

下面是运行结果：



## 3.5 case 2 查询索赔信息

 case 2:

        printf("----------查询索赔信息----------\n");

        printf("请输入索赔编号:\n");

        scanf("%s",tempData.id);

        result = searchByCarID(list,tempData.id);

        if(result == NULL)

        printf("未找到相关信息");

        else

        {

            printf("\n服务站名称: %s\n",result->data.ServiceStationName);

            printf("索赔编号: %s\n",result->data.id);

            printf("索赔日期: %s\n",result->data.ClaimTime);

            printf("索赔金额: %d\n",result->data.ClaimAmount);

            printf("底盘号: %s\n",result->data.CarVIN);

            printf("审核人: %s\n",result->data.Reviewer);

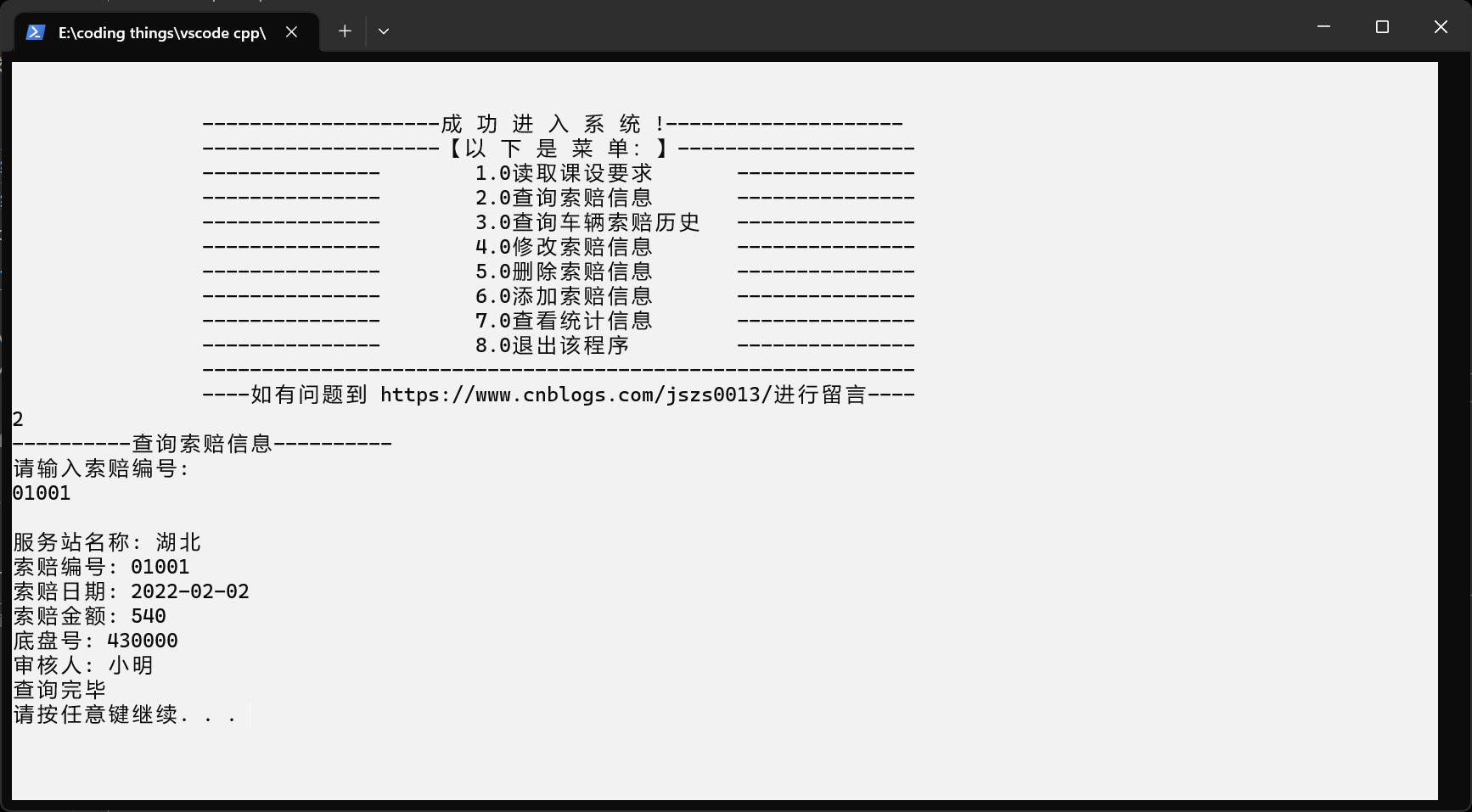
            printf("查询完毕\n");

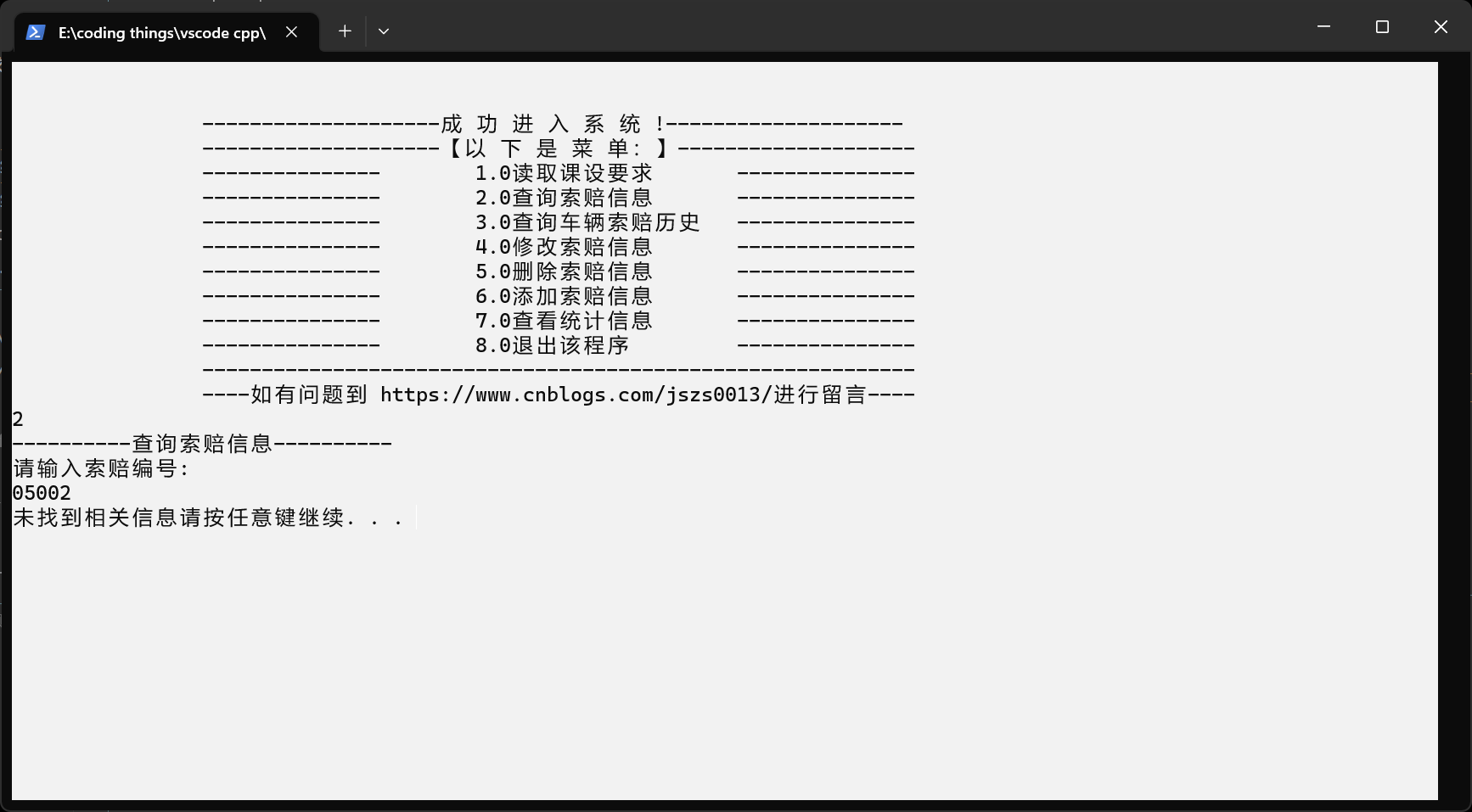
        }

        break;

使用全局变量的result接受函数的返回值，判断是否查找到信息。这个阶段没有判断用户ID的输入是否合法，不合法也是没有查找到，所以没有接入check函数。

1查找到时：



2没查到时：

## 3.6 case 3 查询车辆索赔历史

case 3:

        printf("----------查询车辆索赔历史----------\n");

        printf("请输入车底盘号\n");

        scanf("%s",tempData.CarVIN);

        result = searchByCarVIN(list,tempData.CarVIN);

        if(result != NULL )

        {

            printf("查询成功以下是历史信息\n");

searchAndprintfBYCarVIN(list,tempData.CarVIN);

        }

        else

        {

            printf("请输入正确的编号\n");

        }

        break;

在接受到用户的指令的后，根据是否查找到结果来分开处理函数。

这是searchByCarVIN函数：

struct Node\* searchByCarVIN(struct Node\* headNode,char\* CarVIN)

{

    struct Node\* posNode = headNode->next;

    while(posNode != NULL && strcmp(posNode->data.CarVIN,CarVIN))

    {

        posNode = posNode->next;

    }

    return posNode;

}

查找到后使用searchAndprintfBYCarVIN函数 显示相关信息

void searchAndprintfBYCarVIN(struct Node\* headNode,char\* CarVIN)

{

    struct Node\* posNode = headNode->next;

    for(;posNode != NULL;posNode = posNode->next)

    {

        if(strcmp(posNode->data.CarVIN,CarVIN) == 0)

        {

            printf("\n服务站名称: %s\n",posNode->data.ServiceStationName);

            printf("索赔编号: %s\n",posNode->data.id);

            printf("索赔日期: %s\n",posNode->data.ClaimTime);

            printf("索赔金额: %d\n",posNode->data.ClaimAmount);

            printf("底盘号: %s\n",posNode->data.CarVIN);

            printf("审核人: %s\n",posNode->data.Reviewer);

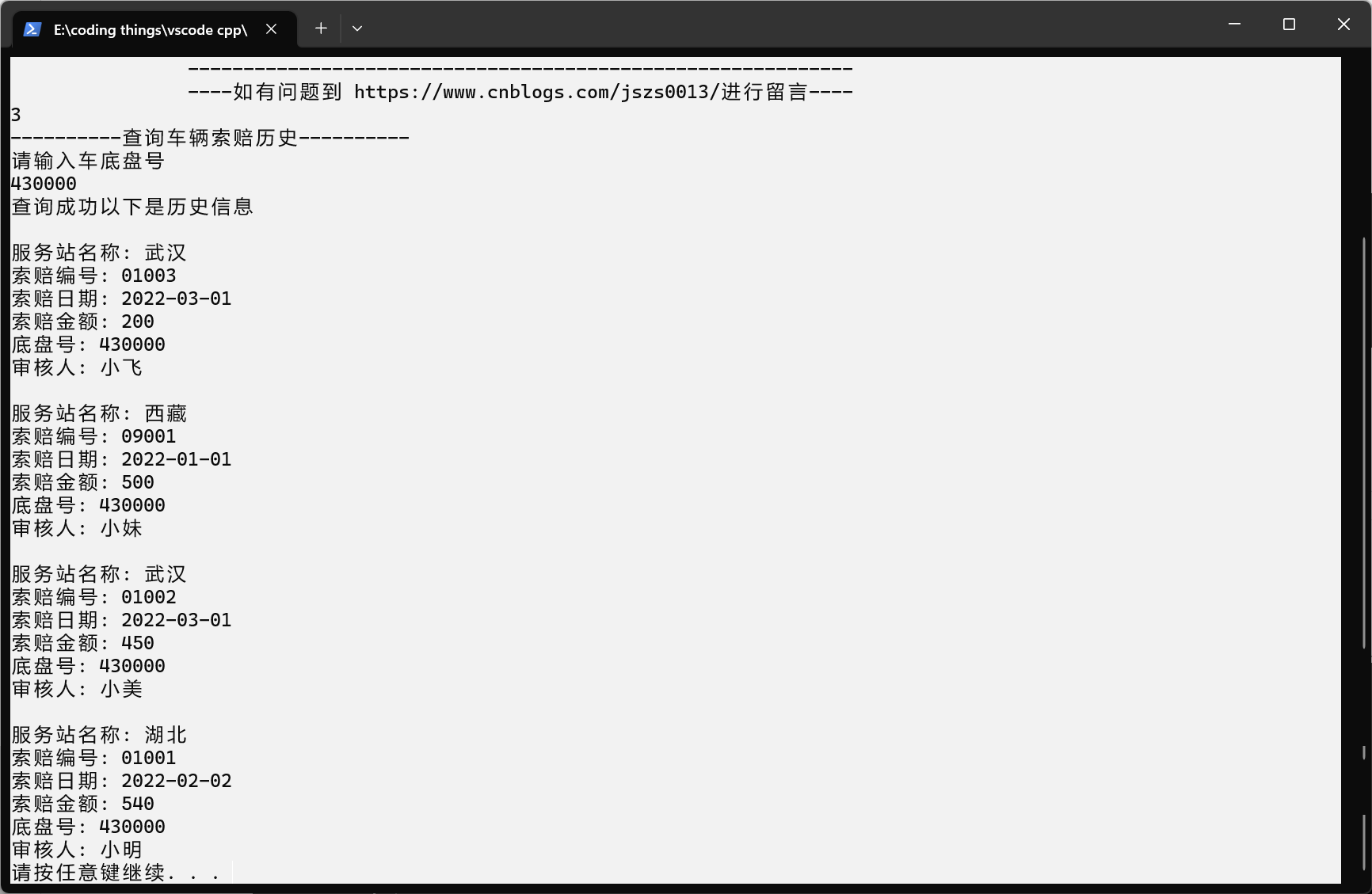
        }

    }

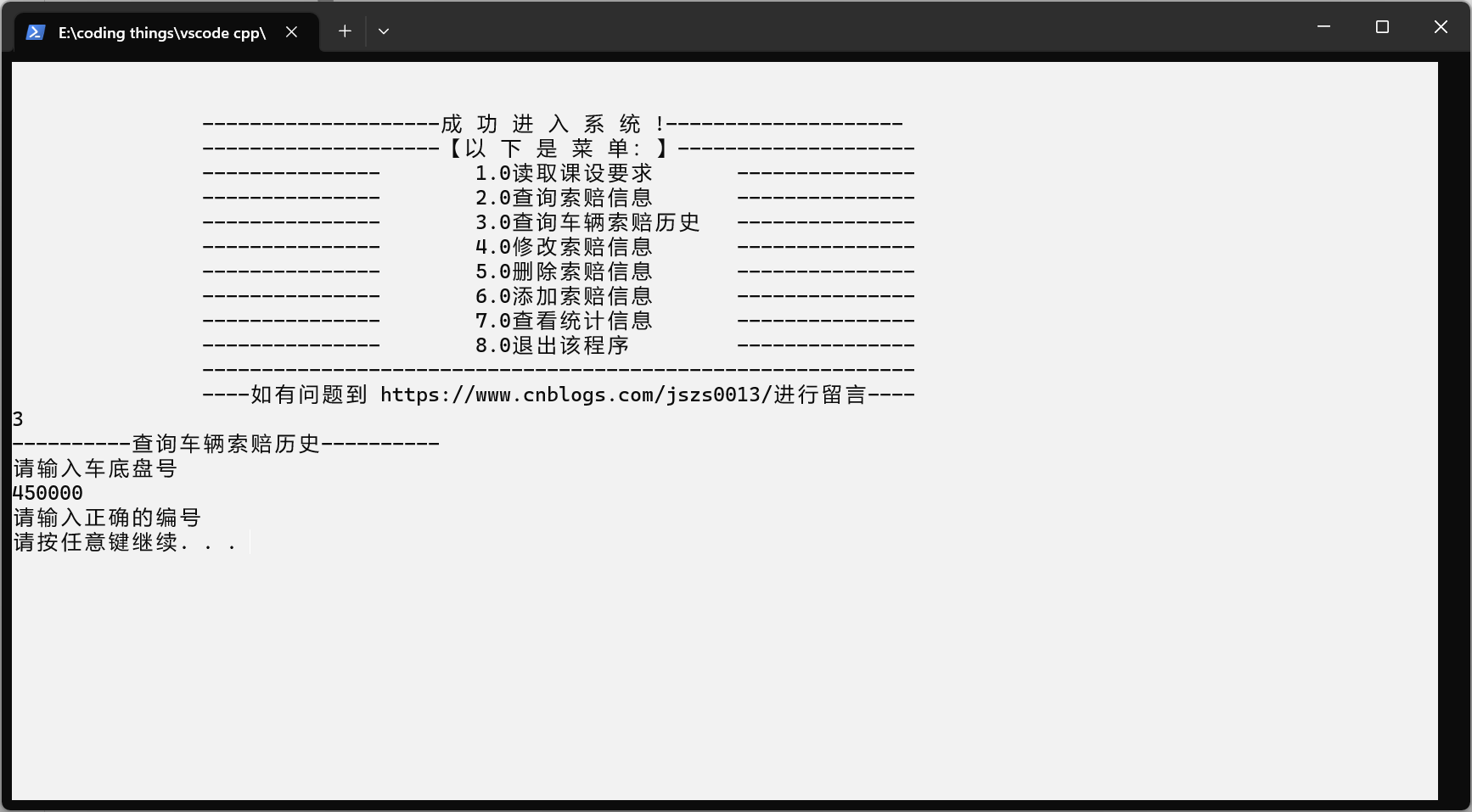
}

每个车的车盘好是唯一的，可以通过车盘好查询历史。这个函数可以检测链表里面存在的车盘号，如果没有相应的车盘好，会提示用户输入正确的车盘号

下面是找到车盘号运行结果：



下面是没有找到车盘号



## 3.7 case 4 修改信息函数

case 4:

        printf("----------修改索赔信息----------\n");

        printf("请输入要修改信息的编号\n");

        scanf("%s",tempData.id);

        result = searchByCarID(list,tempData.id);

        if(result == NULL)

        {

            printf("请输入正确的索赔编号");

        }

        else

        {

        printf("请选择要修改的内容\n");

        printf("1.服务站名称||2.索赔日期||3.索赔金额||4.底盘号||5.审核人\n");

        printf("注意!如要修改编号，删除编号重新添加即可\n");

        printf("-----请输入数字-----\n");

        scanf("%d",&choose);

        if(choose>0&&choose<6)

        {

        switch (choose)

        {

            case 1:

            printf("请输入修改后的服务站名称");

            scanf("%s",tempData.ServiceStationName);

            strcpy(result->data.ServiceStationName,tempData.ServiceStationName);

            printf("修改成功！\n");

            break;

            case 2:

            printf("请输入修改后的索赔日期");

            scanf("%s",tempData.ClaimTime);

            if(checkTrueTime(tempData.ClaimTime) == 0)

            {

                printf("输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

            }

            else

            {

                strcpy(result->data.ClaimTime,tempData.ClaimTime);

                printf("修改成功！\n");

            }

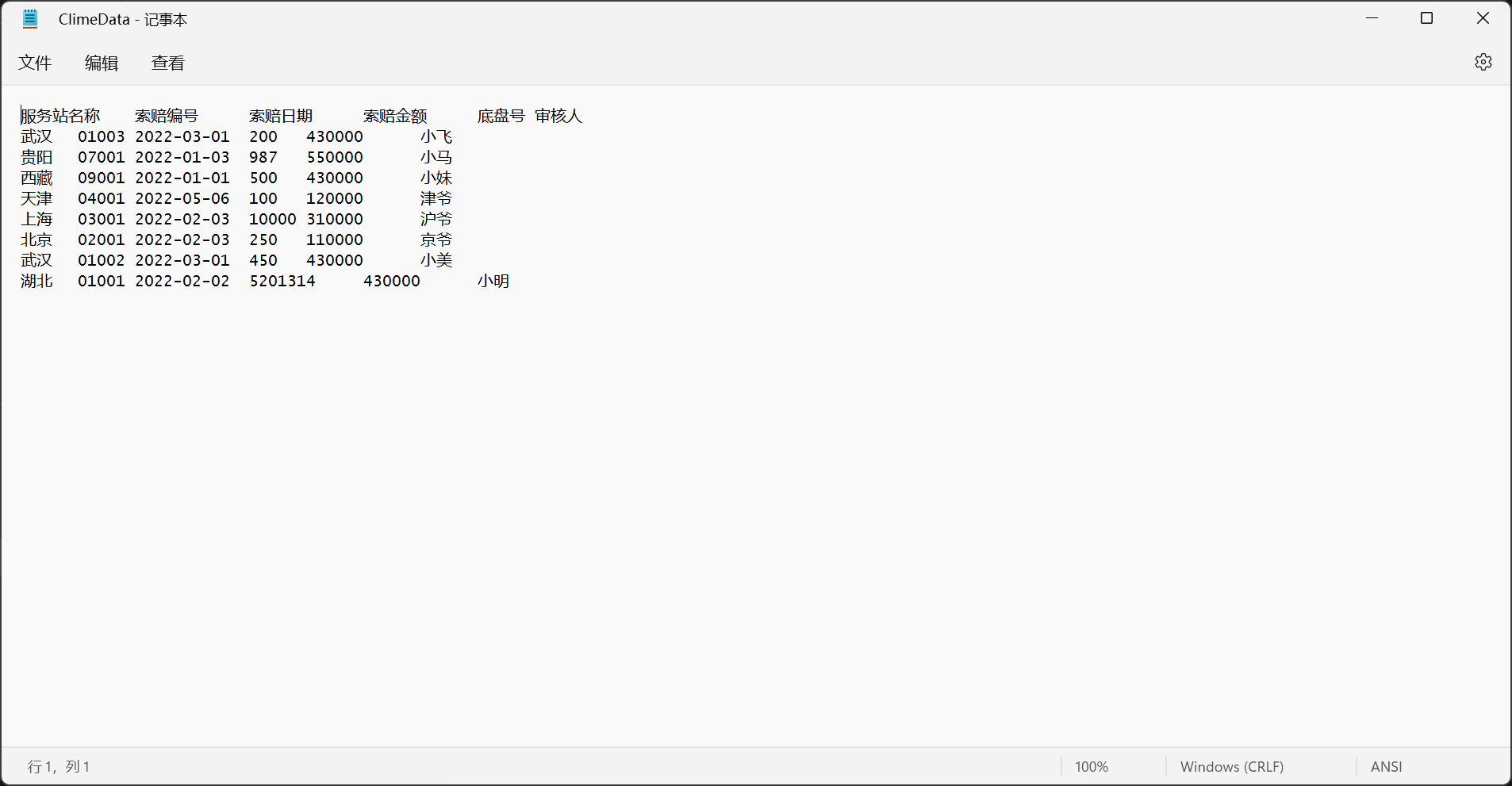
            break;

在引导用户输入需要修改的编号，在引导用户选择需要修改内容，接收用户修改内容到临时变量中去。使用函数进行信息的替换，再将信息整合到链表上最后整合到文件中。

下面是运行画面：



下面是同步文件结果：



## 3.8 case 5 删除索赔信息

case 5:

        printf("----------删除索赔信息----------\n");

        printf("请输入要删除的索赔信息的编号：\n");

        scanf("%s",tempData.id);

deleteNodeBYClimeID(list,tempData.id);

        saveInfoFile("ClimeData.txt",list);

        break;

引导用户删除信息，首先接收到用户输入要删除的索赔信息的编号，调用函数进行信息删除，在同步到文件中。

这是deleteNodeBYClimeID函数：

//指定位置删除，给定索赔编号，删除索赔信息

void deleteNodeBYClimeID(struct Node\* headNode , char \* DataId)

{

    struct Node\* posLeftNode = NULL;

    struct Node\* posNode = headNode->next;

    while(posNode!=NULL && strcmp(posNode->data.id,DataId))//比较id

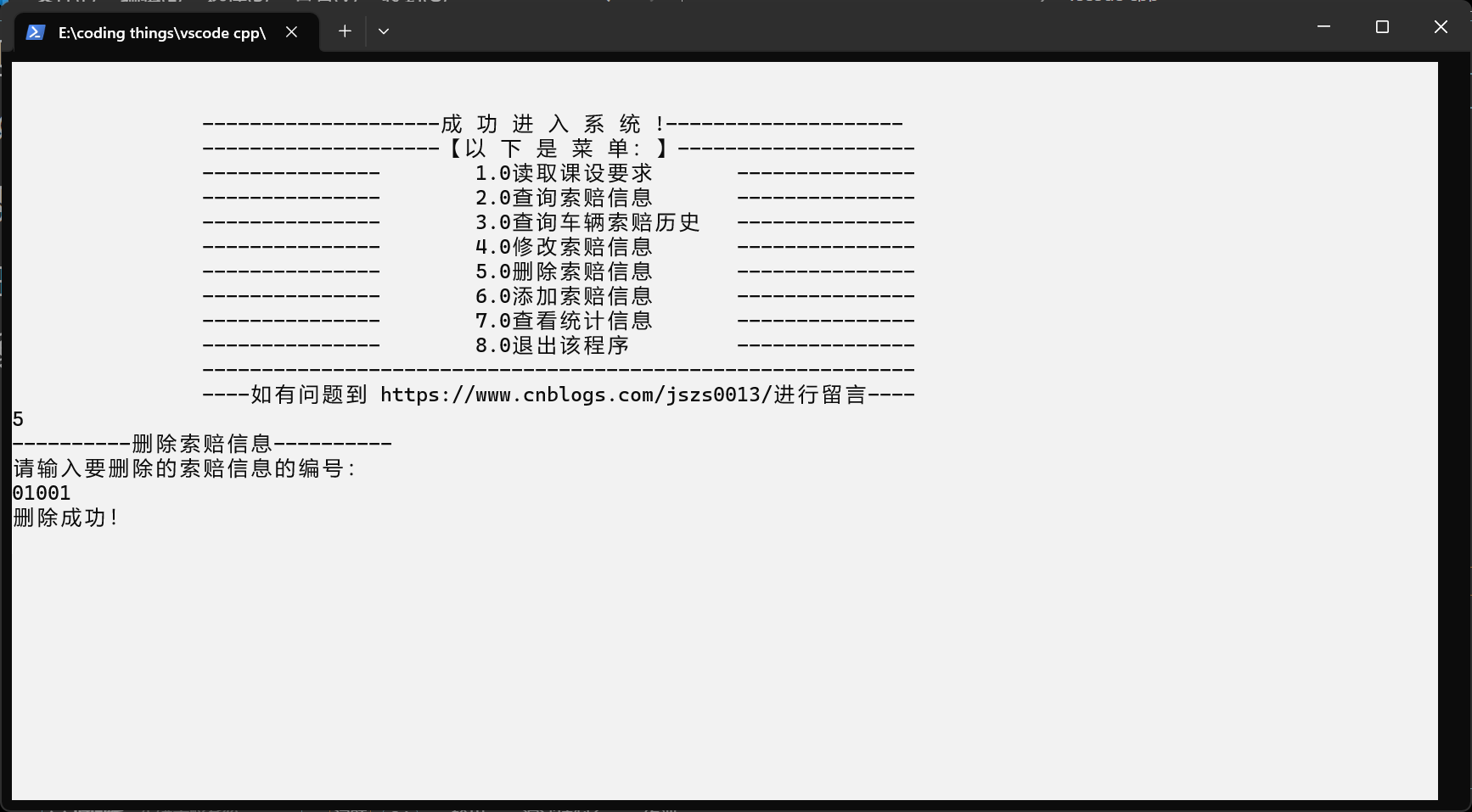
    {

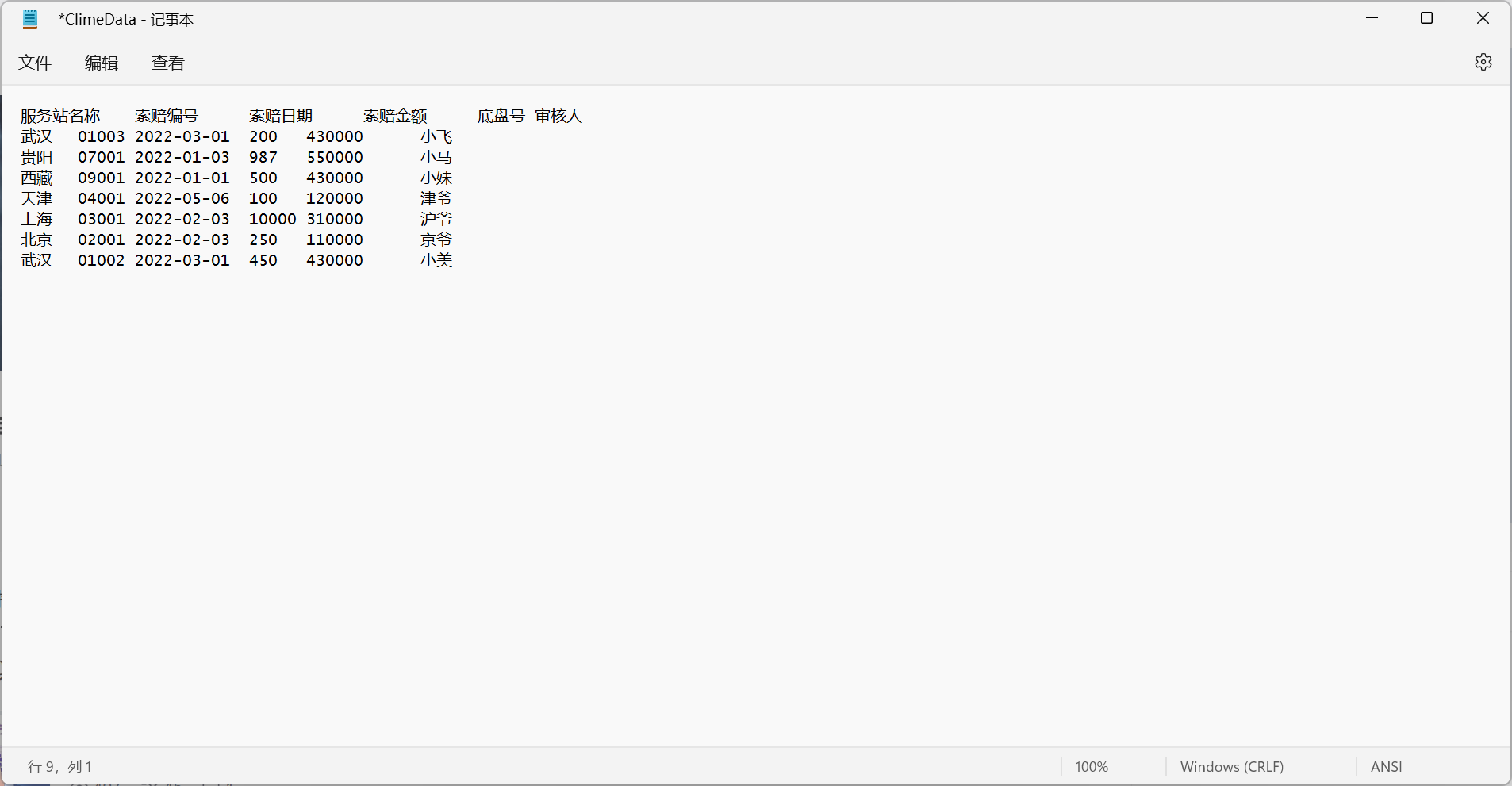
        posLeftNode = posNode;

        posNode = posLeftNode->next;

    }

下面是运行测试：



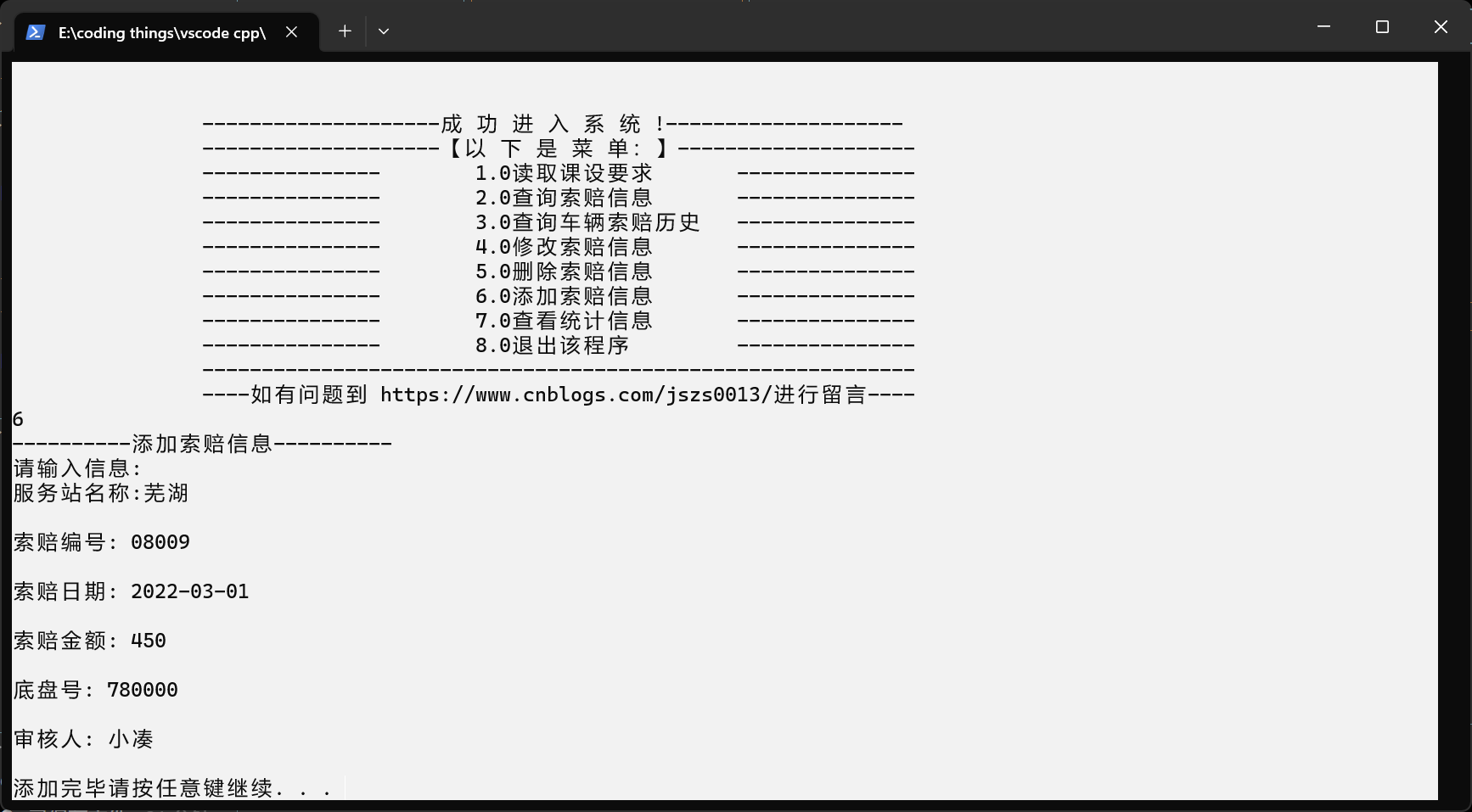


可以明显看到txt文件里面的编号01001的索赔信息被删除了。

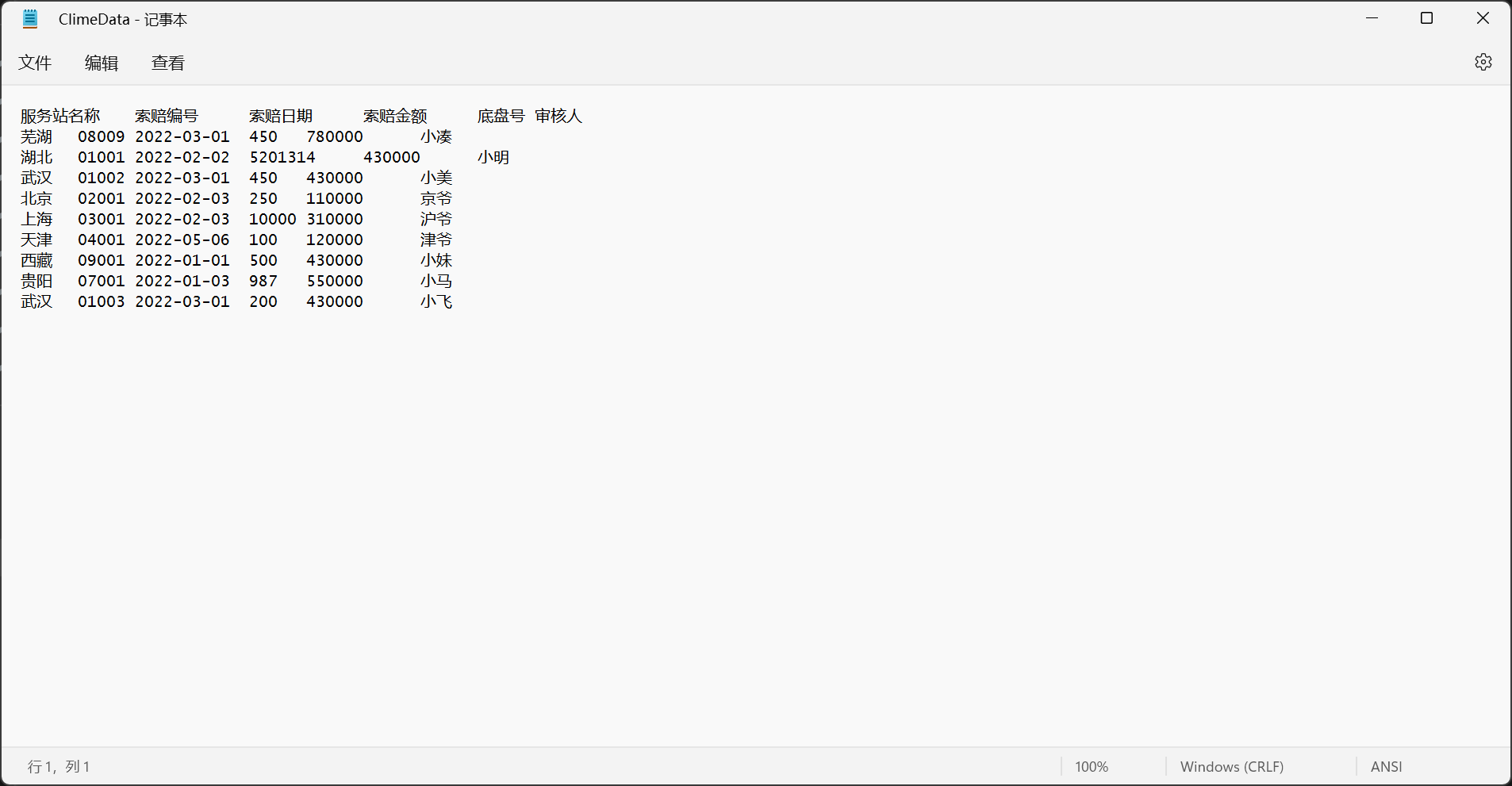
## 3.9 case 6 添加索赔信息

引导用户添加索赔信息一步一步的添加各项信息，可以防止信息混乱，也可以减少bug的产生，最后使用头插法将信息插入到链表中，再通过函数将链表储存到文件中。

下面是运行测试：



文件同步： 因为是头插法所以后进去的在上面。



如果已经有ID存在了，为了保证ID的唯一性，提醒工作人员：

struct Node\* checkRepeatID(struct Node\* headNode,char\* ID)//判断编号是否重复

{

    struct Node\* posNode = headNode->next;

    while(posNode != NULL && strcmp(posNode->data.id,ID))

    {

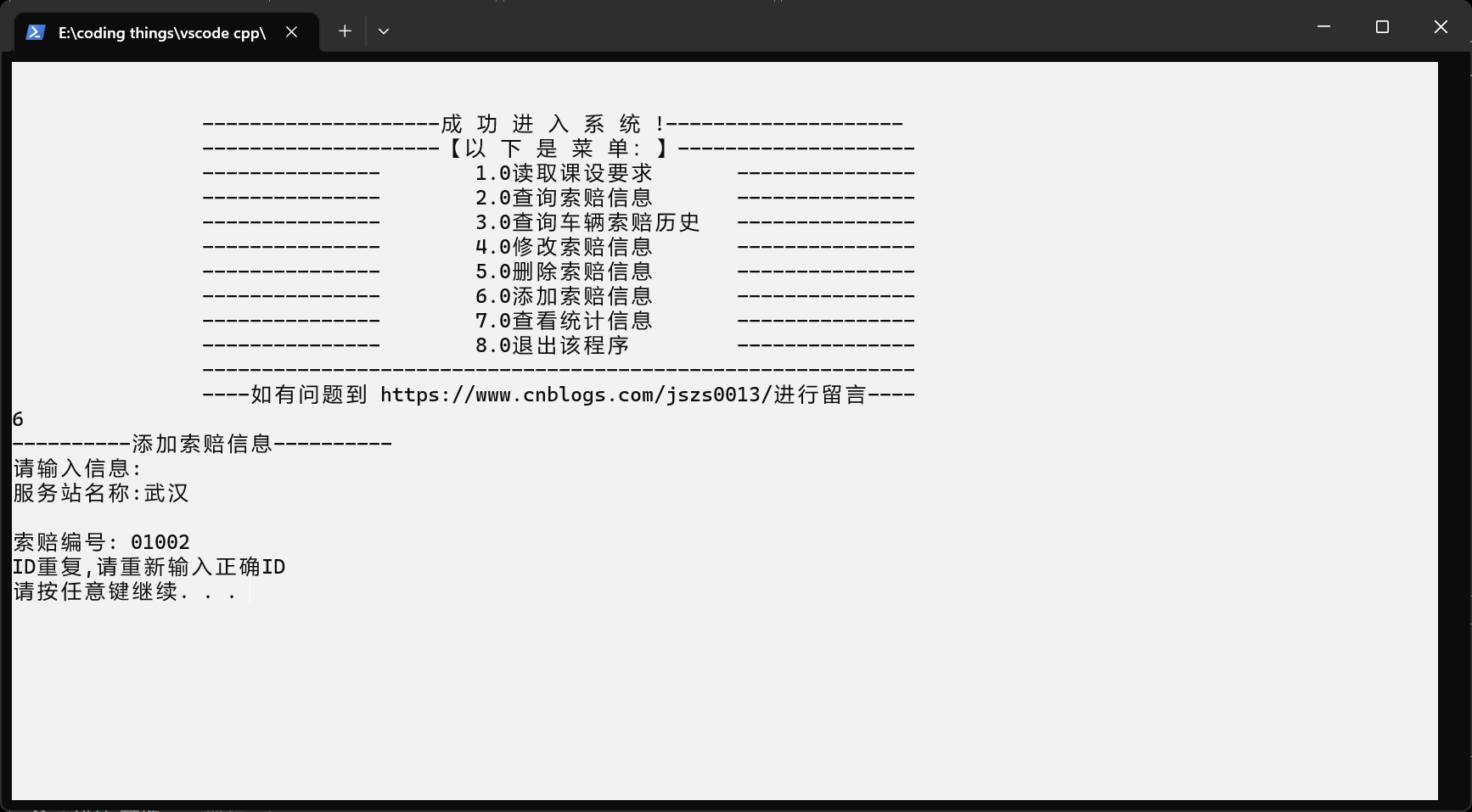
        posNode = posNode->next;

    }

    return posNode;

}

因为id的唯一性：



## 3.10 case 7 查看统计信息

case 7:

        printf("----------查看统计信息----------\n");

        printf("选择方式统计金额：\n");

        printf("1.按照服务站统计2.按照审核人统计3.按照日期统计\n");

        scanf("%d",&choose);

        switch(choose)\

        {

            case 1:

            printf("请输入服务站名\n");

            scanf("%s",tempData.ServiceStationName);

            money = StatisticalAmountByName(list,tempData.ServiceStationName);

            printf("这是统计后的数字：%d\n",money);

            break;

            case 2:

            printf("请输入审核员名字\n");

            scanf("%s",tempData.Reviewer);

            money = StatisticalAmountByReviewer(list,tempData.Reviewer);

            printf("这是统计后的数字：%d\n",money);

            break;

            case 3:

            printf("请输入日期 格式:yyyy-mm-dd\n");

            scanf("%s",tempData.ClaimTime);

            money = StatisticalAmountByTime(list,tempData.ClaimTime);

            printf("这是统计后的数字：%d\n",money);

            break;

        }

        break;

引导用户选择需要统计的内容，接收用户的命令，在调用相应的函数进行统计分析，最后显示统计后的结果。

下面是三个统计函数：

//通过服务站统计金额

int StatisticalAmountByName(struct Node\* headNode,char\* name)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.ServiceStationName,name) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

        posNode = posNode->next;

    }

    return amount;

}

//通过审核员统计金额

int StatisticalAmountByReviewer(struct Node\* headNode,char\* name)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.Reviewer,name) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

        posNode = posNode->next;

    }

    return amount;

}

//通过时间统计金额

int StatisticalAmountByTime(struct Node\* headNode,char\* Time)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.ClaimTime,Time) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

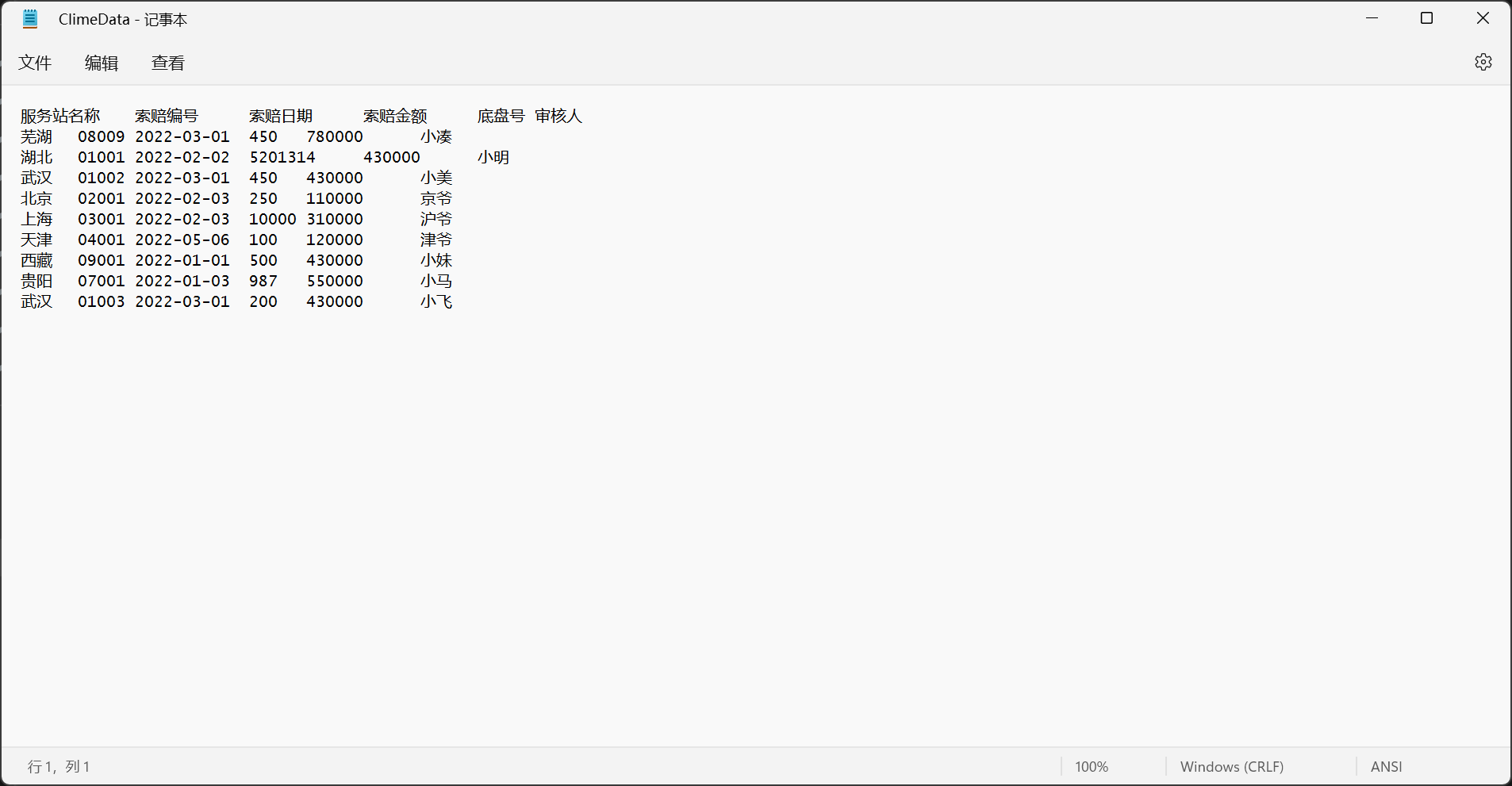
        posNode = posNode->next;

    }

    return amount;

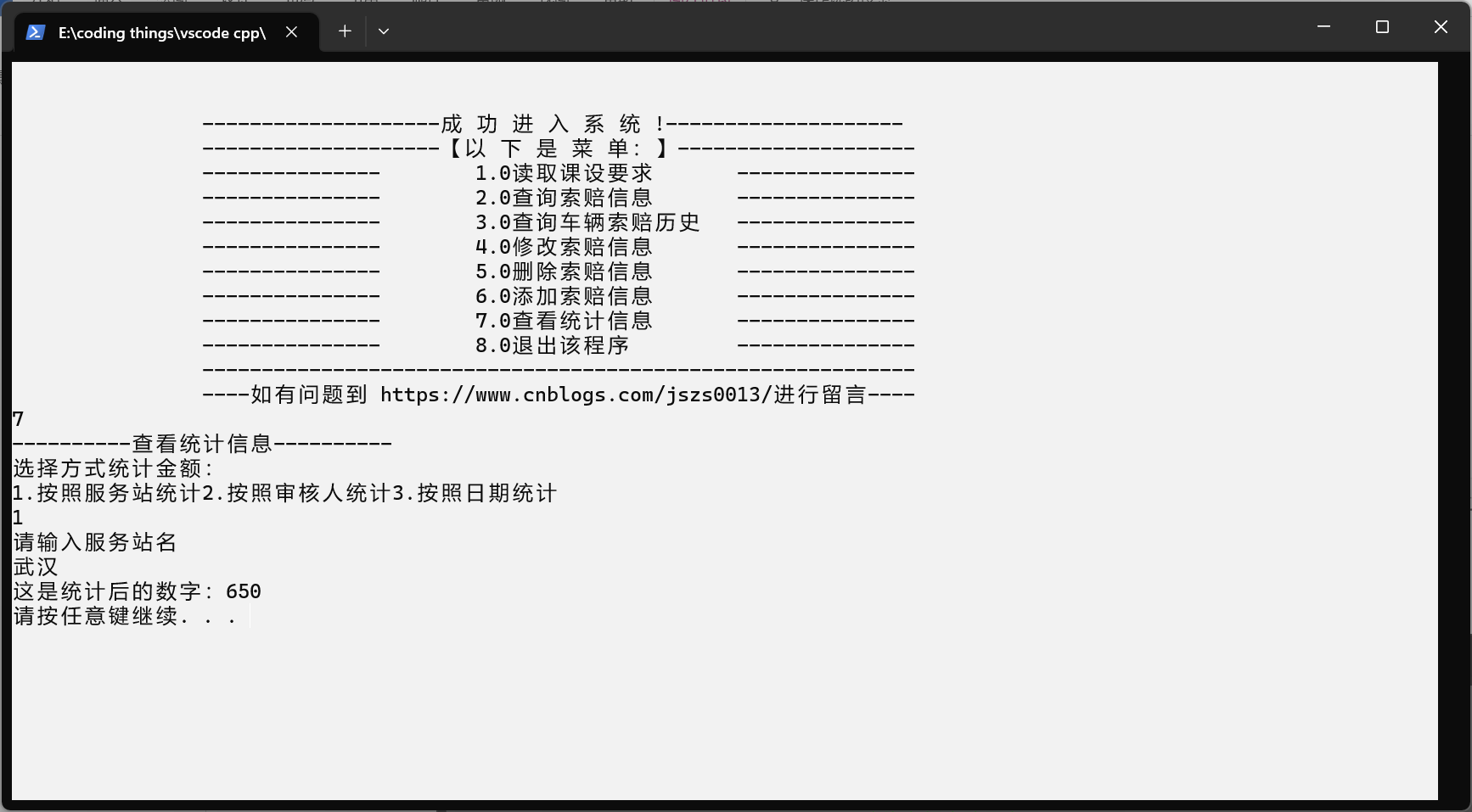
}

下面是运行测试 ps由于3个函数过于相似在这里只显示一种测试：

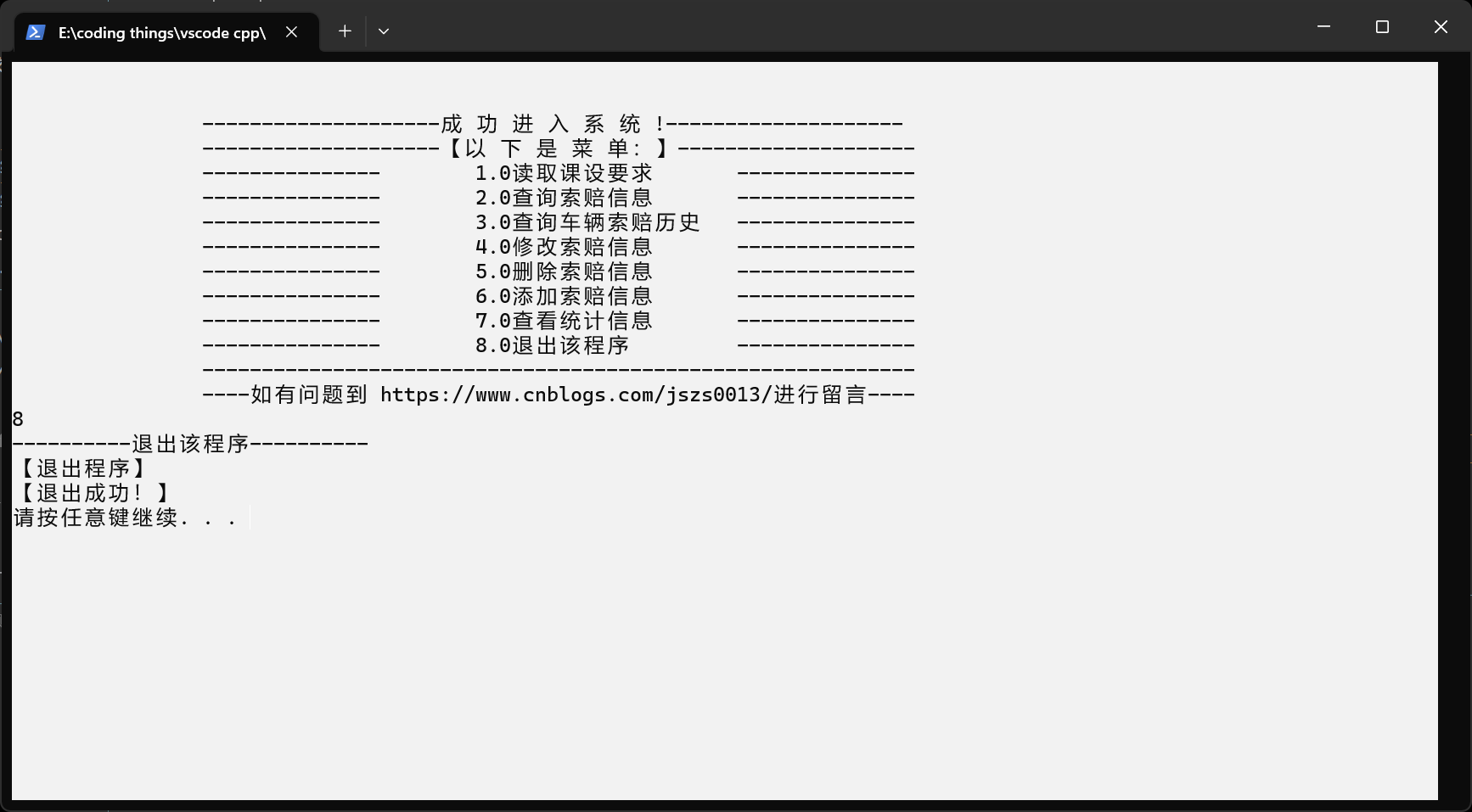


明显武汉的金额是200+450=650；

下面是结果显示：



## 3.11 case 8 退出程序



过于简单 不做解释。

## 3.12 检测时间合法函数

void substr(char dest[], char src[], int pos, int length)//判断合法日期的函数之一

{

    strncpy(dest, &src[pos], length);

    dest[length] = '\0';

}

int IsLeapYear(int year)//判断年份是否是润年

{

    if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))

    return 1;

    else

    return 0;

}

void Nowtime()//通过函数得到当前时间

{

    time\_t timep;

    time(&timep);

    struct tm \*p;

    p = gmtime(&timep);

    Nowyear = (1900 + p->tm\_year);

    Nowmonth = 1 + p->tm\_mon;

    Nowday = p->tm\_mday;

    //printf("%d-%d-%d",Nowyear,Nowmonth,Nowday);

}

int  checkTrueTime(char \*time )//判断时间合法性

{

    int flagyear = 0, flagmonth = 0, flagday = 0,trueflag = 0;

    char ID[15];

    strcpy(ID,time);

    printf("%s",ID);

    char syear[10], smonth[10], sday[10];

    substr(syear, ID, 0, 4);

    substr(smonth, ID, 5, 2);

    substr(sday, ID, 8, 2);

    int putyear = atoi(syear);

    int putmonth = atoi(smonth);

    int putday = atoi(sday);

    if(IsLeapYear(putyear))

    monthday[2] = 29;

    if(putyear<=Nowyear&&putyear>0)

    {

        flagyear = 1;

        if(putmonth<=Nowmonth&&putmonth>0)

        flagmonth = 1;

        if(putday<monthday[putmonth]&&putday>0)

        flagday = 1;

    }

    if(flagday&&flagmonth&&flagyear)

    trueflag = 1;

    /\*else

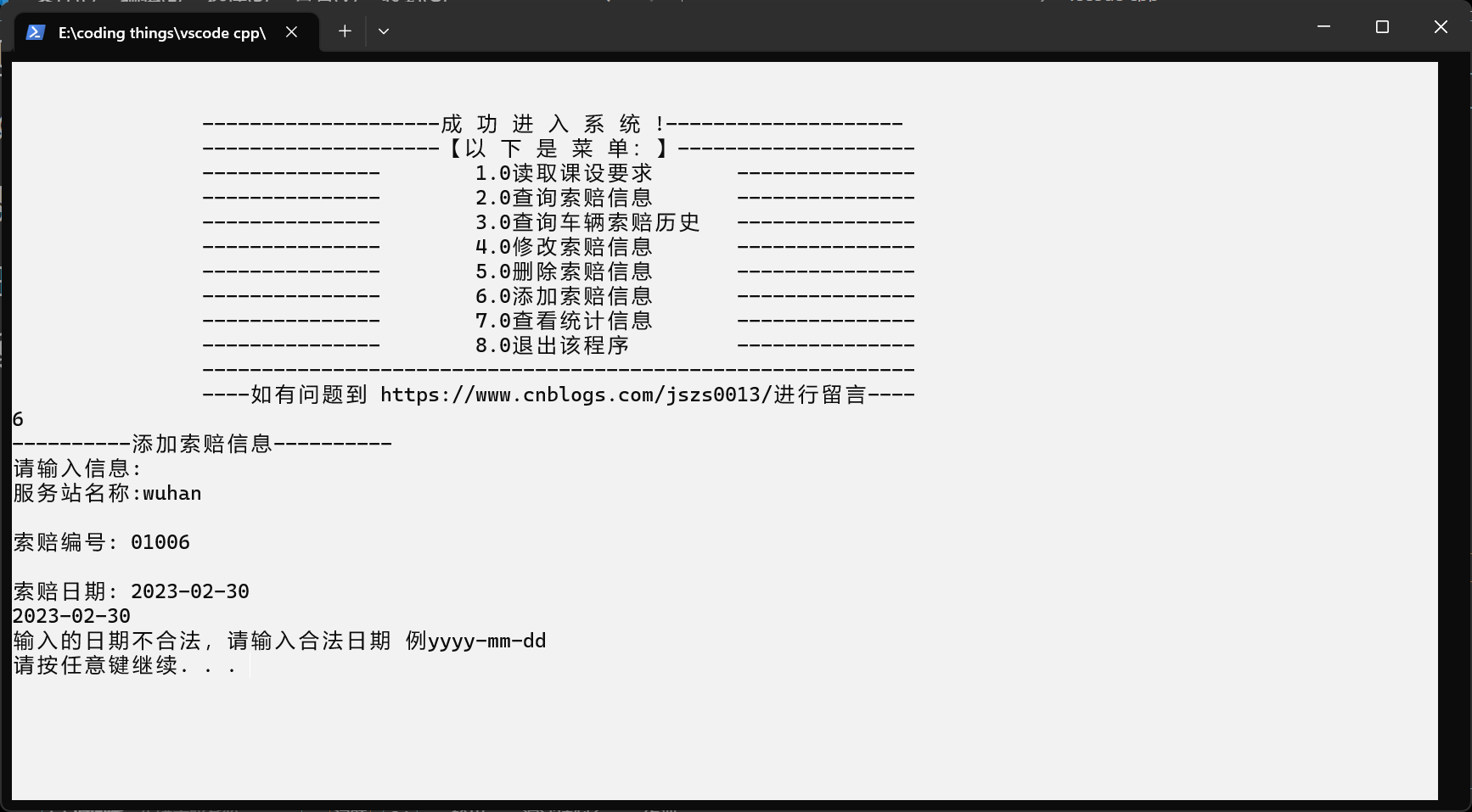
    {

        printf("输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

    }\*/

    return trueflag;

}  
通过自己创造的函数读取，仿造c++的库函数写一个可以定点取字符到特定的支付串里面。在通过time库里面获取到现在的年月日，判断年是否是闰年，在和现在的时间对比，合法就存储到链表里。



# 四 课程设计归纳总结

本次课程设计是一个C语言项目，题目要求我们开发一个简单的索赔管理系统。在完成这个项目的过程中，我们从需求分析、代码实现到测试等方面全面提高了自己的编程技能。

在需求分析阶段，我们根据题目要求制定了详细的项目计划，确定了项目的整体架构和功能模块。在代码实现阶段，我们认真遵循了代码可读性和可维护性的原则，并通过编写注释和代码分块等方式提高代码的可读性。

在测试阶段，我们对程序进行了大量的测试，包括正常数据测试和异常数据测试，确保程序能够正常运行并且符合题目要求。

通过这次课程设计，我对C语言有了更加深入的理解，同时也提高了自己的代码能力和项目开发经验。在整个项目开发过程中，我也意识到了自己在代码实现和测试方面的不足，并且在以后的学习和实践中努力改进。

官方话说完了，现在来谈谈我个人的想法。要我评价下这次的课设，我的评价是折磨。一开始我忘记了分组，写了一份图书管理系统，我的程序大部分bug都是图书管理系统里面出现的，我刚开始写的时候自信满满，我自认为c语言学习的不错，在使用VSCODE写程序的时候也在IDE给出警告后快速的解决问题，我花了一个下午写完了图书管理系统，当我满怀期待的按下F6后，却没有正常启动，IDE给了我一大推看不懂的报错，万恶的Process returned xxxxxxxx，当时我就不服气了，一个一个改正，越改正越感觉自己像个小丑，我自认为C语言学的还行，在更加深入的了解我错的位置后，才发现自己其实只是浮在水面上，很多底层的交互，逻辑的构成我都一知半解。我另一个收到领教的点，是凡是不可以急，我有一个Process returned -1073741819 (0xC0000005)的问题一直解决不了，我一度想放弃想去淘宝买一个课设直接抄，我告诫自己不可以急，去江边陪家人散步的时候。我领悟到了，没有什么问题是可以难死人的，也奇怪我散步完回家后那另一个ide运行代码，他告诉我了问题的说在，Segment fault ，我的fscanf函数爆显存了，就是指访问的内存超出了系统所给这个程序的内存空间。通常这个值是由gdtr来保存的，他是一个48位的寄存器，其中的32位是保存由它指向的gdt表， 后13位保存相应于gdt的下标，最后3位包括了程序是否在内存中以及程序的在cpu中的运行级别,指向的gdt是由以64位为一个单位的表，在这张表中 就保存着程序运行的代码段以及数据段的起始地址以及与此相应的段限和页面交换还有程序运行级别还有内存粒度等等的信息。一旦一个程序发生了越界访 问，cpu就会产生相应的异常保护，于是segmentation fault就出现了。越学习越感觉到自己的无知，通过学习前人的经验和开发的工具，不断的尝试和研究，找出更恰当的方法来避免、发现并处理它。对于一些常见的地方，我们可以避免，对于一些“隐藏”的地方，我们要发现它，发现以后就要及时处理，避免留下隐患。平稳心态后，慢慢的解决所以的问题，我记得在我运行程序成功时的那份喜悦，一看电脑时间已经快凌晨了。在第一次写过小项目以后，我获得了许多经验，可阅读性代码，可维护性代码，等等。在写本次程序的时候，有了上次的经验，酣畅淋漓的写完了索赔程序，在研究如何在程序里写入BMP格式图片时，发现了不一样的景色，也学习到了许多实用的知识，收获颇丰，有时候代码这东西，越研究越有味道。

2023年2月5日22：26

# 五 源码展示

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#include <process.h>

#include <conio.h>

#include <time.h>

#define MAXNAME 30

#define MAXDAY 20

#define MAXID 20

typedef long long ll;

struct Node\* list = NULL;//使用全局链表

int Nowyear,Nowmonth,Nowday;

int monthday[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

void welcome()

{

 system("color F0") ;

 printf("\n\n\n\n\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*   欢   迎   使   用   汽   车   索   赔   管   理   系   统 !  \*\* \n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*  \t\t\t\t\t\t\t\t   \*\*\n") ;

 printf("\t\t \*\*   制   作   人：章崇文       指   导   教   ；史旅华 齐心      \*\*\n") ;

 printf("\t\t \*\*   学校 湖北汽车工业学院      班级 计算机 222                   \*\*\n") ;

 printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*   请   按   ENTER   键   进   入   系   统\*\*\*\*\*\*\*\*\*\*\*\*\n") ;

 printf("\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

 fflush(stdin) ;

 getchar() ;

system("cls") ;

 return ;

}

void makeMenu()//菜单界面

{

    printf("\n\n\t\t--------------------成 功 进 入 系 统 !--------------------\n");

    printf("\t\t--------------------【以 下 是 菜 单：】--------------------\n");

    printf("\t\t---------------        1.0读取课设要求       ---------------\n");

    printf("\t\t---------------        2.0查询索赔信息       ---------------\n");

    printf("\t\t---------------        3.0查询车辆索赔历史   ---------------\n");

    printf("\t\t---------------        4.0修改索赔信息       ---------------\n");

    printf("\t\t---------------        5.0删除索赔信息       ---------------\n");

    printf("\t\t---------------        6.0添加索赔信息       ---------------\n");

    printf("\t\t---------------        7.0查看统计信息       ---------------\n");

    printf("\t\t---------------        8.0退出该程序         ---------------\n");

    printf("\t\t------------------------------------------------------------\n");

    printf("\t\t----如有问题到 https://www.cnblogs.com/jszs0013/进行留言----\n");

    return ;

}

//索赔信息的设定

struct ClimeData

{

    char ServiceStationName[MAXNAME];//服务站名称

    char  id[MAXID];//索赔编号

    int ClaimAmount;//索赔金额

    char ClaimTime[MAXDAY]; //索赔日期

    char CarVIN[MAXID];//底盘号

    char Reviewer[MAXNAME];//审核人

};

//主要数据链表的设定

struct Node

{

     struct ClimeData data;

    struct Node\* next;

};

struct Node\* createHead()

{

    struct Node\* headNode =(struct Node\*)malloc(sizeof(struct Node));//动态内存申请

    headNode->next = NULL;

    return headNode;

};

//创建节点为数据做准备

struct Node\* createNode(struct ClimeData data)

{

    struct Node\* newNode=(struct Node\*)malloc(sizeof( struct Node));

    newNode->data = data;

    newNode->next = NULL;

    return newNode;

}

//插入数据 使用表头法插入

void insertNodeBYHead(struct Node\* headNode , struct ClimeData data)

{

    struct Node\* newNode = createNode(data);

    newNode->next = headNode->next;

    headNode->next = newNode;

}

//插入数据 使用表尾法

void insertNodeBYTail(struct Node\* headNode , struct ClimeData data)

{

    struct Node\* pMove = headNode;

    while(pMove != NULL)

    {

        pMove = pMove->next;

    }

    struct Node\* newNode = createNode(data);

    pMove->next = newNode;

}

//指定位置删除，给定索赔编号，删除索赔信息

void deleteNodeBYClimeID(struct Node\* headNode , char \* DataId)

{

    struct Node\* posLeftNode = NULL;

    struct Node\* posNode = headNode->next;

    while(posNode!=NULL && strcmp(posNode->data.id,DataId))//比较id

    {

        posLeftNode = posNode;

        posNode = posLeftNode->next;

    }

//讨论查找结果

    if(posNode == NULL)

    return ;

    else

    {

        printf("删除成功！\n");

        posLeftNode->next = posNode->next;

        free(posNode);

        posNode=NULL;

    }

}

//用底盘号查找

struct Node\* searchByCarVIN(struct Node\* headNode,char\* CarVIN)

{

    struct Node\* posNode = headNode->next;

    while(posNode != NULL && strcmp(posNode->data.CarVIN,CarVIN))

    {

        posNode = posNode->next;

    }

    return posNode;

}

//通过底盘号查历史

void searchAndprintfBYCarVIN(struct Node\* headNode,char\* CarVIN)

{

    struct Node\* posNode = headNode->next;

    for(;posNode != NULL;posNode = posNode->next)

    {

        if(strcmp(posNode->data.CarVIN,CarVIN) == 0)

        {

            printf("\n服务站名称: %s\n",posNode->data.ServiceStationName);

            printf("索赔编号: %s\n",posNode->data.id);

            printf("索赔日期: %s\n",posNode->data.ClaimTime);

            printf("索赔金额: %d\n",posNode->data.ClaimAmount);

            printf("底盘号: %s\n",posNode->data.CarVIN);

            printf("审核人: %s\n",posNode->data.Reviewer);

        }

    }

}

//用编号查找

struct Node\* searchByCarID(struct Node\* headNode,char\* ID)

{

    struct Node\* posNode = headNode->next;

    while(posNode != NULL && strcmp(posNode->data.id,ID))

    {

        posNode = posNode->next;

    }

    return posNode;

}

struct Node\* checkRepeatID(struct Node\* headNode,char\* ID)//判断编号是否重复

{

    struct Node\* posNode = headNode->next;

    while(posNode != NULL && strcmp(posNode->data.id,ID))

    {

        posNode = posNode->next;

    }

    return posNode;

}

//数据排序按索赔价格(冒泡排序小到大)

void bubbleSortList(struct Node\* headNode)

{

    for(struct Node\* p = headNode->next;p != NULL;p = p->next)

    {

        for(struct Node\* q = headNode->next; q->next != NULL; q = q->next)

        {

            if(q->data.ClaimAmount>q->next->data.ClaimAmount)

            {

               struct ClimeData tempData = q->data;

                q->data = q->next->data;

                q->next->data = tempData;

            }

        }

    }

}

//检测编号是否符合要求

int checkID(char \* ID)

{

    int len = strlen(ID);

    if(len == 5)

    return 1;

    else

    return 0;

}

void substr(char dest[], char src[], int pos, int length)//判断合法日期的函数之一

{

    strncpy(dest, &src[pos], length);

    dest[length] = '\0';

}

int IsLeapYear(int year)//判断年份是否是润年

{

    if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))

    return 1;

    else

    return 0;

}

void Nowtime()//通过函数得到当前时间

{

    time\_t timep;

    time(&timep);

    struct tm \*p;

    p = gmtime(&timep);

    Nowyear = (1900 + p->tm\_year);

    Nowmonth = 1 + p->tm\_mon;

    Nowday = p->tm\_mday;

    //printf("%d-%d-%d",Nowyear,Nowmonth,Nowday);

}

int  checkTrueTime(char \*time )//判断时间合法性

{

    int flagyear = 0, flagmonth = 0, flagday = 0,trueflag = 0;

    char ID[15];

    strcpy(ID,time);

    printf("%s",ID);

    char syear[10], smonth[10], sday[10];

    substr(syear, ID, 0, 4);

    substr(smonth, ID, 5, 2);

    substr(sday, ID, 8, 2);

    int putyear = atoi(syear);

    int putmonth = atoi(smonth);

    int putday = atoi(sday);

    if(IsLeapYear(putyear))

    monthday[2] = 29;

    if(putyear<=Nowyear&&putyear>0)

    {

        flagyear = 1;

        if(putmonth<=Nowmonth&&putmonth>0)

        flagmonth = 1;

        if(putday<monthday[putmonth]&&putday>0)

        flagday = 1;

    }

    if(flagday&&flagmonth&&flagyear)

    trueflag = 1;

    /\*else

    {

        printf("输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

    }\*/

    return trueflag;

}

//通过服务站统计金额

int StatisticalAmountByName(struct Node\* headNode,char\* name)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.ServiceStationName,name) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

        posNode = posNode->next;

    }

    return amount;

}

//通过审核员统计金额

int StatisticalAmountByReviewer(struct Node\* headNode,char\* name)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.Reviewer,name) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

        posNode = posNode->next;

    }

    return amount;

}

//通过时间统计金额

int StatisticalAmountByTime(struct Node\* headNode,char\* Time)

{

    int amount = 0;

    struct Node\* posNode = headNode->next;

    while(posNode != NULL )

    {

        if(strcmp(posNode->data.ClaimTime,Time) == 0)

        {

            amount+=posNode->data.ClaimAmount;

        }

        posNode = posNode->next;

    }

    return amount;

}

//文件操作 写入文件

//写入文件

void saveInfoFile(const char\* fileName,struct Node\* headNode)

{

    FILE \*fp = fopen(fileName,"w+");

    struct Node\* pMove = headNode->next;

    fprintf(fp,"服务站名称\t索赔编号\t索赔日期\t索赔金额\t底盘号\t审核人\t\n");

    while(pMove != NULL)

    {

        fprintf(fp,"%s\t%s\t%s\t%d\t%s\t%s\t\n",pMove->data.ServiceStationName,pMove->data.id,pMove->data.ClaimTime,pMove->data.ClaimAmount,

        pMove->data.CarVIN,pMove->data.Reviewer);

        pMove = pMove->next;

    }

    fclose(fp);

}

//文件读操作

void  readInfoFile(const char\* fileName,struct Node\* headNode)

{

    FILE \*fp = fopen(fileName,"r");

    if(fp == NULL)//文件不纯在就创建文件

    {

        fp = fopen(fileName,"w+");

    }

    struct ClimeData tempData;

    fscanf(fp,"%\*[^\n]");

    while

    (

        (fscanf(fp,"%s\t",tempData.ServiceStationName) != EOF) &&(fscanf(fp,"%s\t",tempData.id) != EOF) &&

        (fscanf(fp,"%s\t",tempData.ClaimTime) != EOF) && (fscanf(fp,"%d",&tempData.ClaimAmount)) &&

        (fscanf(fp,"%s\t",tempData.CarVIN) != EOF) && (fscanf(fp,"%s\n",tempData.Reviewer))

    )

    {

        insertNodeBYHead(list,tempData);

    }

    fclose(fp);

}

//打印索赔信息

void printfList(struct Node\* headNode)

{

    struct Node\* pMove = headNode->next;\

    printf("以下是现在储存的信息");

    while(pMove != NULL)

    {

        printf("\n服务站名称: %s\n",pMove->data.ServiceStationName);

        printf("索赔编号: %s\n",pMove->data.id);

        printf("索赔日期: %s\n",pMove->data.ClaimTime);

        printf("索赔金额: %d\n",pMove->data.ClaimAmount);

        printf("底盘号: %s\n",pMove->data.CarVIN);

        printf("审核人: %s\n",pMove->data.Reviewer);

    }

}

//用户交互界面

void keyDown()

{

    int userKey = 0;

    int choose = 0;

    int money = 0;

    struct ClimeData tempData;//临时变量存储索赔信息

    struct Node\* result = NULL;

    scanf("%d",&userKey);

    switch(userKey)

    {

        case 1:

        printf("----------读取课设要求----------\n");

        //system("E:\\coding things\vscode cpp\\课设要求.docx");

        system("pause");

        printf("成功查看\n");

        break;

        case 2:

        printf("----------查询索赔信息----------\n");

        printf("请输入索赔编号:\n");

        scanf("%s",tempData.id);

        result = searchByCarID(list,tempData.id);

        if(result == NULL)

        printf("未找到相关信息");

        else

        {

            printf("\n服务站名称: %s\n",result->data.ServiceStationName);

            printf("索赔编号: %s\n",result->data.id);

            printf("索赔日期: %s\n",result->data.ClaimTime);

            printf("索赔金额: %d\n",result->data.ClaimAmount);

            printf("底盘号: %s\n",result->data.CarVIN);

            printf("审核人: %s\n",result->data.Reviewer);

            printf("查询完毕\n");

        }

        break;

        case 3:

        printf("----------查询车辆索赔历史----------\n");

        printf("请输入车底盘号\n");

        scanf("%s",tempData.CarVIN);

        result = searchByCarVIN(list,tempData.CarVIN);

        if(result != NULL )

        {

            printf("查询成功以下是历史信息\n");

            searchAndprintfBYCarVIN(list,tempData.CarVIN);

        }

        else

        {

            printf("请输入正确的编号\n");

        }

        break;

        case 4:

        printf("----------修改索赔信息----------\n");

        printf("请输入要修改信息的编号\n");

        scanf("%s",tempData.id);

        result = searchByCarID(list,tempData.id);

        if(result == NULL)

        {

            printf("请输入正确的索赔编号");

        }

        else

        {

        printf("请选择要修改的内容\n");

        printf("1.服务站名称||2.索赔日期||3.索赔金额||4.底盘号||5.审核人\n");

        printf("注意!如要修改编号，删除编号重新添加即可\n");

        printf("-----请输入数字-----\n");

        scanf("%d",&choose);

        if(choose>0&&choose<6)

        {

        switch (choose)

        {

            case 1:

            printf("请输入修改后的服务站名称");

            scanf("%s",tempData.ServiceStationName);

            strcpy(result->data.ServiceStationName,tempData.ServiceStationName);

            printf("修改成功！\n");

            break;

            case 2:

            printf("请输入修改后的索赔日期");

            scanf("%s",tempData.ClaimTime);

            if(checkTrueTime(tempData.ClaimTime) == 0)

            {

                printf("输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

            }

            else

            {

                strcpy(result->data.ClaimTime,tempData.ClaimTime);

                printf("修改成功！\n");

            }

            break;

            case 3:

            printf("请输入修改后的索赔金额");

            scanf("%d",&tempData.ClaimAmount);

            result->data.ClaimAmount = tempData.ClaimAmount;

            printf("修改成功！\n");

            break;

            case 4:

            printf("请输入修改后的底盘号");

            scanf("%s",tempData.CarVIN);

            strcpy(result->data.CarVIN,tempData.CarVIN);

            printf("修改成功！\n");

            break;

            case 5:

            printf("请输入修改后的审核人名字");

            scanf("%s",tempData.Reviewer);

            strcpy(result->data.Reviewer,tempData.Reviewer);

            printf("修改成功！\n");

            break;

        }

    }

    else

    {

        printf("请输入正确数字\n");

    }

        }

        break;

        case 5:

        printf("----------删除索赔信息----------\n");

        printf("请输入要删除的索赔信息的编号：\n");

        scanf("%s",tempData.id);

        deleteNodeBYClimeID(list,tempData.id);

        saveInfoFile("ClimeData.txt",list);

        break;

        case 6:

        printf("----------添加索赔信息----------\n");

        printf("请输入信息:\n");

        printf("服务站名称:");

        scanf("%s",tempData.ServiceStationName);

        printf("\n索赔编号: ");

        scanf("%s",tempData.id);

        result = checkRepeatID(list,tempData.id);

        if(result != NULL)

        {

            printf("ID重复,请重新输入正确ID\n");

            break;

        }

        printf("\n索赔日期: ");

        scanf("%s",tempData.ClaimTime);

        if(checkTrueTime(tempData.ClaimTime) == 0)

        {

            printf("\n输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

            break;

        }

        printf("\n索赔金额: ");

        scanf("%d",&tempData.ClaimAmount);

        printf("\n底盘号: ");

        scanf("%s",tempData.CarVIN);

        printf("\n审核人: ");

        scanf("%s",tempData.Reviewer);

        printf("\n添加完毕");

        insertNodeBYHead(list,tempData);

        saveInfoFile("ClimeData.txt",list);

        break;

        case 7:

        printf("----------查看统计信息----------\n");

        printf("选择方式统计金额：\n");

        printf("1.按照服务站统计2.按照审核人统计3.按照日期统计\n");

        scanf("%d",&choose);

        switch(choose)\

        {

            case 1:

            printf("请输入服务站名\n");

            scanf("%s",tempData.ServiceStationName);

            money = StatisticalAmountByName(list,tempData.ServiceStationName);

            printf("这是统计后的数字：%d\n",money);

            break;

            case 2:

            printf("请输入审核员名字\n");

            scanf("%s",tempData.Reviewer);

            money = StatisticalAmountByReviewer(list,tempData.Reviewer);

            printf("这是统计后的数字：%d\n",money);

            break;

            case 3:

            printf("请输入日期 格式:yyyy-mm-dd\n");

            scanf("%s",tempData.ClaimTime);

            if(checkTrueTime(tempData.ClaimTime) == 0)

            printf("输入的日期不合法，请输入合法日期 例yyyy-mm-dd\n");

            else

            {

                money = StatisticalAmountByTime(list,tempData.ClaimTime);

                printf("这是统计后的数字：%d\n",money);

            }

            break;

        }

        break;

        case 8:

        printf("----------退出该程序----------\n");

        saveInfoFile("ClimeData.txt",list);

        printf("【退出程序】\n");

        printf("【退出成功！】\n");

        system("pause");

        exit(0);

        break;

         default:

        printf("\terror!\n");

        printf("\t请输入数字 0~8\n");

        break;

    }

}

//主函数

int main()

{

    welcome();

    list = createHead();

    readInfoFile("ClimeData.txt",list);//读入数据

    while(1)

    {

        makeMenu();

        keyDown();

        system("pause");

        system("cls");

    }

    system("pause");

    return 0;

}