

上机报告-2

数算B 谢胡睿 2400014151

题目

1.题目背景

某单位最多100辆车辆，每辆车的基本信息包括：车号、车型、购买日期、购买价格、驾驶员名称、违章记录等。其中，各辆车的违章次数可能差别很大（有的一次没有，有的可能有数十次）。每次违章记录中包括：违章时间、违章地点、违章代码（如闯红灯、违规停车等）、违章处理等。实现上述车辆管理的程序，包括车辆信息的输入/输出和显示等。

2.题目描述

对于给定的类型，输出对应的所有未被处理的违章信息。

3.输入格式

首先进行车辆信息的输入，以car为车辆输入开始的信号，后输入车辆数目，仅记录车牌号（5位数字，不含字母）。具体格式如下：

```
car 3
10000
66666
99999
```

然后进行违章信息的输入，以violation为违章信息输入开始的信号，后输入违章数目。每行输入顺序为车牌号（5位数字，不含字母），违章代码（3位数字），违章地点代号（3位大写字母），违章时间（XXXX-XXXX），违章是否被处理（0为未处理，1为已处理）。具体格式如下：

```
violation 4
10000 100 PEK 2000-01-01 0
99999 100 AVA 2012-03-04 0
10000 101 KMG 2010-10-12 0
10000 100 PEK 2000-01-01 1
```

最后进行查询，以query为查询开始的信号，可能会对车牌、违章代码、违章地点代号、违章时间进行查询（仅查询其中一项）。具体格式如下：

Solution

总体描述

考虑到系统需要处理的数据特点，我选择采用以下数据结构：

1. 违章记录：使用链表存储每辆车的违章记录，便于动态添加
2. 车辆信息：使用数组存储所有车辆
3. 查询结果：使用向量存储查询结果，方便统计和输出

通过封装不同的输出，输入，使得代码更加清晰

设计与实现

准备

车辆信息读取：read_car函数，按指定格式读取车辆信息

违章记录添加：add_violation函数，使用头插法将违章记录添加到相应车辆的链表

违章信息读取：read_violation函数，读取违章信息并调用add_violation函数

查询功能：根据输入特征判断查询类型，调用相应的打印函数

print_car：按车牌号查询

print_code：按违章代码查询

print_place：按地点查询

print_time：按时间查询

结果

输出

遇到的问题

链表操作错误

问题：初始实现时，在add_violation函数中尝试直接访问record指针的成员，但这可能导致空指针访问。

解决方案：正确使用头插法，先创建新节点，设置其各个字段值，然后将其连接到链表头部。

字符串转整数问题

问题：需要将字符串形式的违章代码转换为整数进行比较，不熟悉相关函数。

解决方案：使用C++标准库中的stoi()函数将字符串转换为整数。

输出格式问题

问题：需要先输出符合条件的未处理违章记录数量，再输出详细信息。

解决方案：使用vector收集所有符合条件的记录，统一输出。

总结：

未能够从头完整设计代码流程，导致调整了很长时间统计结果数量。

需要顶层设计完善。