**考试说明**

1. **考试时间：合计3小时。选择题部分60分钟交卷，允许提前交卷。**
2. **考试过程中，不能连接未经指定网站或服务器。**
3. **闭卷考试部分，不能查阅任何类型的参考资料。**
4. **开卷考试部分，可以查阅纸质文档，不能查阅任何类型的电子文档。**
5. **考试过程中，不得使用任何形式的电子存储设备，不可使用手机。**
6. **违反上述2-5条者，视为考试作弊。**

**选择题答题方式（25分，闭卷，严禁使用vs编程环境进行尝试）**

1. **打开浏览器，在地址栏中输入**[**http://192.1****68****.125.3**](http://192.168.125.3)**，点击相应链接进入登录页面。**
2. **按要求输入两遍自己的学号。**
3. **点击“登录”按钮即可进入答题页面。如考试尚未开始，系统会进入等待页面并倒计时。考试开始时间到，系统会自动进入答题页面。**
4. **在页面左侧选择题号，页面右侧即会显示相应的题目。考生只需点击选择相应的选项。**
5. **答题过程中如关闭浏览器或出现系统故障导致计算机重新启动，系统不会丢失之前已经完成的题目的答案。考生可以打开浏览器重新登录并继续考试。**
6. **答题完成后，点击“交卷”按钮即可完成交卷。交卷后不能再次登录系统继续考试。**
7. **考试结束时间到，系统会自动收卷。**

**编程题提交方式（75分，开卷）**

1. **提交前务必关闭PyCharm编程环境。**
2. **所有源程序内容必须仅包含在一个源程序文件（py文件）中。**
3. **在浏览器的地址栏中输入**[**http://192.168.125.3**](http://192.168.125.3)**，点击相应链接进入提交页面。**
4. **按要求输入两遍自己的学号。**
5. **点击“选择文件”按钮，选择自己的源程序文件。点击“提交”按钮提交。**
6. **如提交成功，系统会显示相关信息。如果提交不成功，请重复步骤16-18。**
7. **提交成功后，可点击“查看内容”按钮检查提交的内容。**

**按以下要求编写程序**

现有一个文本文件data.txt，其中记录了车辆进出校园时在自动收费系统ETC中记录下的ETC编号和时间信息。每个ETC编号唯一地对应于一辆机动车。试用python语言编写程序，按要求从该数据文件中提取所需的信息。

说明：

1. 一个ETC编号由5部分构成，本别是：两个大写字母、一个短横线、三位数字、一个短横线、三位数字。如：“FG-102-934”、“BA-724-433”等都是合法的ETC编号。
2. 时间信息的记录格式为：2016-01-08#07:21:31。
3. 整条ETC记录的格式由5部分构成，分别是：ETC编号、字符“|”、入校时间、字符“|”，离校时间。如：“BA-724-433|2016-01-08#07:21:31|2016-01-08#17:01:09”就是一条结构完整的ETC记录。
4. 数据文件中每一行为一个ETC记录。
5. 车辆进入校园后都是在当天离开校园的，即每条ETC记录中出入校园的日期是相同的。
6. ETC记录中出现的字符都是英文符号，无汉字和中文标点符号。

要求：

1. 从文件中识别ETC记录，计算总的ETC记录的条数。提示：读取文件，并将ETC记录放入列表中。
2. 计算ETC记录中共有多少辆不同的车。提示：通过正则表达式（或字符串分片）识别每条ETC记录中的ETC编号，并将ETC编号放入集合中。
3. 找出进出校园次数最多的5辆车。提示：构建ETC编号和出现次数的字典。
4. 找出在校园中累计停留时间最长的5辆车。提示：构建ETC编号和累计停留时间的字典。可通过正则表达式（或字符串分片）识别时间字符串及其中的时、分、秒信息。为简化计算，可将时间转化成以零点开始计算的秒数。可以把计算时间差作为一个独立的函数。
5. 将上述计算结果按“report.txt文件内容示例”（见最后一页）所示的要求输出到report.txt文件中。

下面是给定的main函数和对main函数的调用。请将下列内容复制到你的源程序中，然后按照上述要求完成程序。**对给定的内容不允许改变。每修改一处，扣3分；**

def main():

vehicle\_lst = get\_record("data.txt") *# 读文件，获取全部ETC记录，构成列表*

vehicle\_set = get\_v(vehicle\_lst) *# 获取全部不同的ETC编号，构成集合*

fre\_dict = count\_v(vehicle\_lst, vehicle\_set) *# 构造车辆进出校园次数的字典*

inter\_dict = count\_t(vehicle\_lst, vehicle\_set) *# 构造车辆累计停留时间的字典*

write\_to\_file(vehicle\_lst, fre\_dict, inter\_dict, "report.txt") *# 输出结果到文件中*

return

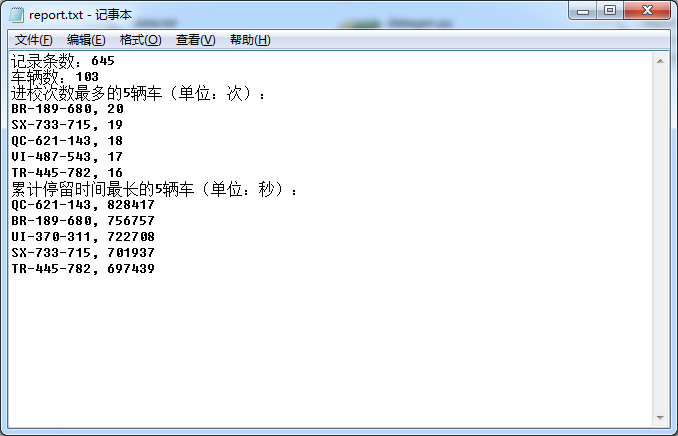
main() # 调用main函数

**评分标准**

**（编程题满分为75分）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **大项** | **子项** | **评分项** | **应得分** | **实得分** |
| 正  确  性  65分 | 各项结果（65分）  （如程序无法运行则此大项不得分） | 读取文件信息并用文件内容构成列表 | 10 |  |
| 计算出ETC记录总条数 | 10 |  |
| 计算出ETC记录中不同车的数量 | 10 |  |
| 找出进出校园次数最多的5辆车 | 10 |  |
| 正确计算时间差 | 10 |  |
| 找出停留时间最长的5辆车 | 5 |  |
| 输出report.txt | 10 |  |
| 修改给定的内容 | -3× |  |
| 可  读  性  10分 | 注释（5分） | 有详细且正确的注释 | 5 |  |
| 有注释，但不够详细 | 3 |  |
| 完全没有注释 | 0 |  |
| 变量命名（5分） | 变量命名有规则 | 5 |  |
| 变量命名有规则、但规则使用不一致 | 3 |  |
| 变量命名无规则 | 0 |  |
| 总分（满分75分） | | | |  |

**report.txt文件内容示例：**



修改说明：

1. 仔细考虑后，我还是放弃了html文件的做法，改用纯文本文件做输入。
2. 基于第1条修改，文件编码的问题也不存在了。
3. 题目第4项要求中，明确了“累计停留时间最长”。
4. 关于时间计算的问题，其实不需要用mktime函数，用分片就可以做。我给出了更明确的提示。