

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）

7. 打开浏览器，在地址栏中输入 <http://192.168.125.3>，点击相应链接进入登录页面。
8. 按要求输入两遍自己的学号。
9. 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。
10. 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。
11. 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。
12. 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。
13. 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）

14. 提交前务必关闭 PyCharm 编程环境。
15. 所有源程序内容必须仅包含在一个源程序文件（py 文件）中。
16. 在浏览器的地址栏中输入 <http://192.168.125.3>，点击相应链接进入提交页面。
17. 按要求输入两遍自己的学号。
18. 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。
19. 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。
20. 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；**

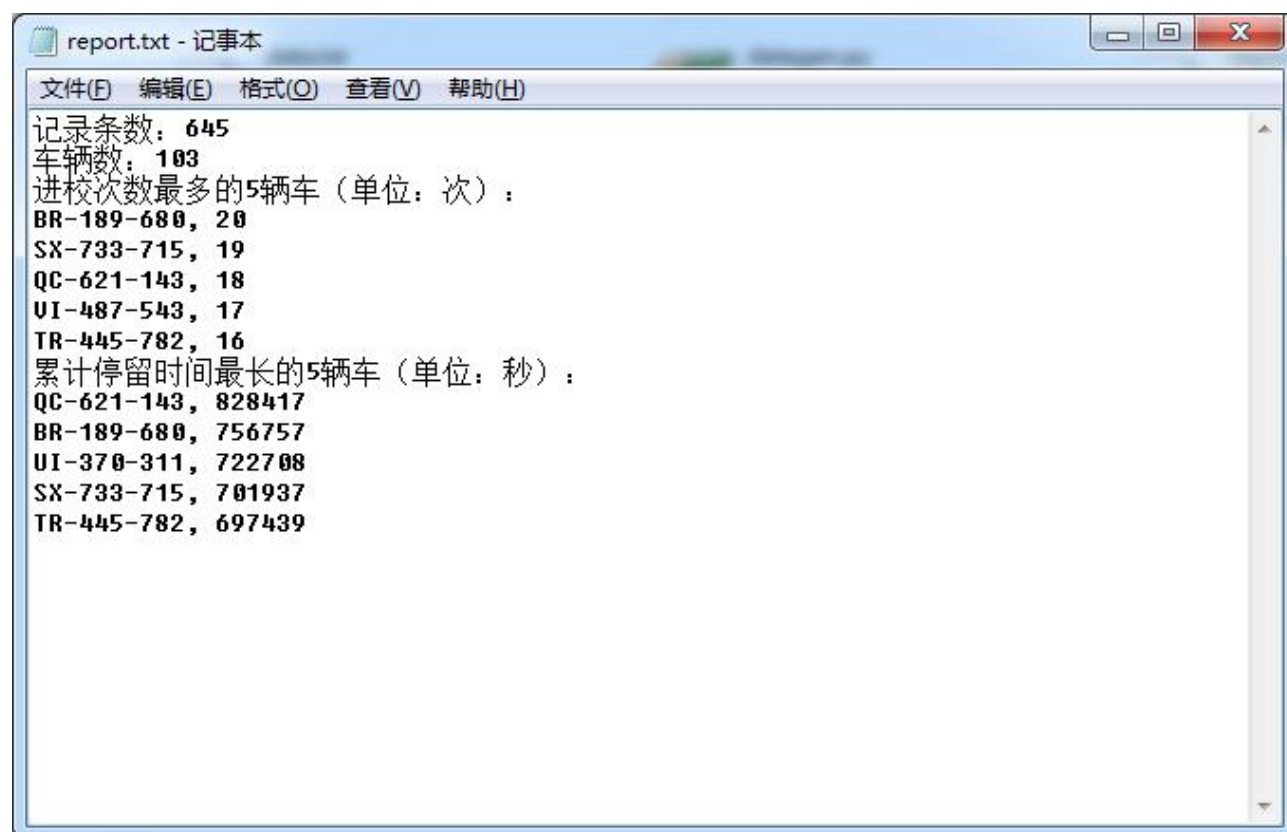
```
def main():  
    vehicle_lst = get_record("data.txt")          # 读文件，获取全部ETC记录，构成列表  
    vehicle_set = get_v(vehicle_lst)              # 获取全部不同的ETC编号，构成集合  
    fre_dict = count_v(vehicle_lst, vehicle_set)  # 构造车辆进出校园次数的字典  
    inter_dict = count_t(vehicle_lst, vehicle_set) # 构造车辆累计停留时间的字典  
    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。