# 多线程电梯readme

## 输入规范

1. 本次作业要求使用控制台进行输入符合规范的请求字符串。
2. 请求分为两类：一类是楼层请求，一类是电梯内请求。

# 输入格式: (FR, floor, UP/DOWN)，即楼层请求；或(ER, #Elevator, floor)， 电梯请求，注意需指明是发出请求的电梯编号。（注，电梯编号为 1,2,3， 如(ER,#1,5)） ，输入中的数字不得有‘+’且不支持前导0

1. 请求事件中的时间 t 自动从系统获得，所获得的时间（微秒）要转换为电梯系统时间(浮点数，单位为秒)（例：假设第一条请求发生的时间为 1234123ms，当前请求输入时间为 1236789ms，则当前时刻为 2.666s）。
2. 由于请求是从控制台输入，且请求时间自动取自系统时间，因此请求队列是按照时间先后进行排序的，不存在输入时间不按顺序的问题。
3. 一行可输入多条请求，请求之间用“;”分割。同一行的指令取相同的系统时间。当一次输入多条请求时，该次输入的所有请求都采用同一个系统时间。

本次作业规定**一次最多输入 10 条请求(即使包含无效请求也不超过 10 条)**

**入最多 10 个请求，一次测试最多输入 50 次。**

**为了确保测试中多线程调度的不确定性不会干扰测试判断，限制一次测试输**

1. 所有字符均为英文状态下的字符（要求测试者保证）。请求内部元素之间可以有空格，要求程序能够自动过滤。
2. 需要退出程序运行时，要求输入“END”，程序结束运行。

## 输出规范

1) 对于无效请求、实质上的相同请求一定要输出（报告），即使进行容错也要输出相应的字符串。

格式为：st:INVALID [request, T]，或

st:SAME [request, T]

其中st是输出时的系统时间，request为输入的请求字符串，T是请求产生时刻的电梯系统时间。

2) 每个有效请求执行完毕的输出请求内容和请求执行结果，分两种情况：

i. 任何一部电梯停靠时要立刻输出停靠信息，输出信息为输出时系统时间，请求信息，电梯执行效果：

格式为：st:[request, T] / (#电梯号, 楼层, UP/DOWN, 累积运动量，t)

其中st是输出时的系统时间，[request, T]为有效请求的字符串和输入时的电梯系统时间，用“[]”包含。中间使用“/”分割。后一部分是该请求的执行效果，其中t是停靠时刻，即电梯刚到达目标楼层由运动转为静止状态，尚未执行开关门的电梯时间。

ii. 同层请求时输出为：st:[request, T]/（#电梯, 楼层, STILL, 累积运动量, t）,其中STILL代表静止。（参考第三次作业指导书例11）

iii. 如果一次停靠执行了多条请求，那么需要分行输出(参考第三次作业例12)

3) 输出格式要求所有字符为英文符号。一个请求和执行效果占用一行。

4) 时间处理方面，st的单位为毫秒，直接取自系统时间；T和t的单位为秒，支持一位小数,取整方式自定，但要满足电梯运行时间和停靠时间要求（即处理后运行一层和停靠时间差要准确为3.0S和6.0S），不支持科学记数法。

5) 其他未规定的地方可由编程者自行决定。

## 思路说明

为了方便同学阅读代码，在此说明自己的思路。对于输入与调度器，有一个临时队列，在W\_legal类中，这个类的对象是输入的请求线程与调度器的线程之间共享的数据。所以，在W\_legal中，有三个同步方法，其中一个方法release只运行一次，为了唤醒程序一开始就处于阻塞状态的调度器线程。方法in\_put用于输入，当完成一次输入后，临时队列要送给调度器，所以此时阻塞输入线程，然后调度完成之后，临时队列为空，所以此时阻塞调度器线程。

对于调度器与电梯运动之间的问题。每个电梯都有自己的请求队列，还有自己电梯的按钮，我把楼层的按钮当作了电梯的按钮，这样处理的目的是统一按钮，便于处理捎带与同质。所以电梯运动与调度器的共享的资源就是电梯。所以我们在线程中给电梯上锁。不允许两个线程同时访问电梯。

关于处理请求与判断捎带与同质。临时队列一旦到达调度器，调度器就会立马处理临时队列。处理的方式是，先判断电梯1是否可以捎带，在判断电梯2是否可以捎带，再判断电梯3是否可以捎带，并且在判断的过程中取运动量最小的电梯。如果都不可以捎带那么就直接找出运动量最小的电梯。捎带的实现在Three\_elev中的bring方法中，不捎带在unbring中。处理一个请求的方式就是将这个请求加入到处理该请求的电梯的队列中去，同时这个电梯的这一楼层对应的按钮将会被按下。那么捎带的判断就是，根据电梯的运动方向与楼层的关系。如果目标请求的那个按钮已经被按下了，那么就是同质请求了。调度最终会产生三个请求队列，对应于三个电梯。

关于判断线程的结束，输入线程输入“END”结束，调度线程一旦输入结束也会结束，电梯运行线程，调度线程结束了，并且，电梯的执行队列为空，那么就结束。