

调度策略:

电梯采用 ASL(A little smart)策略调度电梯, 即在完成当前主请求的过程中, 尽可能捎带同向的其他请求。

程序功能说明:

本程序能够通过控制台输入的电梯请求输入, 调度电梯以完成响应这些请求, 并输出电梯的运动情况。

程序运行所需环境:

本程序需要在 java 虚拟机环境下运行

运行指令为:

```
java Scheduler.class
```

```
java Scheduler
```

程序的输入说明:

用户输入为按照请求产生时间排序的请求序列 (注意: 可以输入时间相同的两个请求, 排在前面请求被优先执行), 序列通过字符串表示;

请求分为两类: 一类是楼层请求, 一类是电梯内请求。

楼层请求格式为: (FR, m, UP/DOWN, T), 其中 FR 为楼层请求标识, m 为发出请求的楼层号, UP 为向上请求, DOWN 为向下请求, T 为发出时刻。(注释: 相当于请求者在楼道里的某楼层按“上行”或“下行”键)

电梯内请求格式为: (ER, n, T), 其中 ER 为电梯内请求标识, n 为请求前往的目标楼层号, T 为发出时刻。(注释: 相当于人在电梯里按一个目标楼层号)

所有的逗号应采用 ASCII 字符集中的逗号 “,” , 而不是中文字符逗号 “,” 。请求之间必须通过换行进行分隔, 两条请求之间不允许有空行。一条请求的内部元素之间可以有空格, 程序能够自动过滤。

T 为请求产生的相对时刻 (4 字节非负整数), 第一个请求必须为 (FR, 1, UP, 0)。

设电梯运行一个楼层距离消耗时间为 0.5; 达到楼层后停靠、开关门等一系列动作消耗时间为 1; 有效的 FR 请求会带来一个开关门动作。合法的请求产生时刻 T 为非负整数 (4 字节非负整数, 支持前导 0 和正号), $1 \leq n, m \leq 10$ (正整数, 支持前导 0 和正号)。

不正确的标识符, 不正确的方向, 不正确的数字范围, 多余的其他字符, 均认定为不合法输入, 即无效输入, 无效输入不被程序接受。

特别地, 对于 FR 标识符, 1 楼的 DOWN 和 10 楼的 UP 也认为是无效输入。

输入全部请求后, 必须再键入 RUN 表示输入结束, 回车后程序开始执行调度。程序等待 RUN 输入。

标准输入的请求是按照时间排序的, 如果遇到一个乱序的请求, 即请求产生时间小于前面一个请求产生时间, 则该请求在输出该请求存在时间为乱序情况的提示信息后, 将其从请求队列中删除, 继续处理下一个请求。

程序计算结果的输出规格:

本次作业有两种输出:

1) 对于无效请求, 要输出该请求为无效的信息, 即使进行容错处理也要输出相应的字符串。

格式为: INVALID [request]

request 为过滤空格后的原输入请求

2) 实质上相同的请求，要输出该请求为相同请求的信息

格式为：# SAME [request] request 为带()的输入请求

3) 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. 电梯停靠信息为按照时间排序的电梯运动停靠楼层、停靠前的运动方向及停靠时刻（即电梯刚到达目标楼层由运动转为静止状态，尚未执行开关门的时刻）：

格式为：[request]/(n, UP/DOWN, t) request 为带()的输入请求

本输出为一个对偶输出，前一部分是[request]，为有效请求的字符串，用“[]”包含。中间使用“/”分割。后一部分该请求的执行效果，包括： n 为楼层号，UP/DOWN 为电梯运行方向；t 为相对于第一个请求发生的时间（浮点数）。

ii. 同层请求时输出为：[request]/(n, STILL, t)，本输出为一个对偶输出，前一部分是[request]，为有效请求的字符串，用“[]”包含。中间使用“/”分割。后一部分包括： n 为楼层号，STILL 代表静止，t 为考虑开关门用时后的时刻。

iii. 如果一次停靠执行了多条请求，那么需要分行输出

4) 输出要求按照请求执行完的时间进行排序。无效请求或实质上相同的请求也许输出信息，可以插在中间。

5) 输出格式所有字符为英文符号。