

Readme

一、 运行环境及指令规范说明

运行环境: JDK 1.8 Eclipse

指令规范: Eclipse 内运行程序后直接在 console 内进行输入即可。

二、 输入说明

标准输入格式:

楼层请求格式为: (FR, m, UP/DOWN, T), 其中 FR 为标识, 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。

电梯内请求格式为: (ER, n, T), 其中 ER 为标识, n 为请求去往的目标楼层号, T 为发出时刻。

括号内的逗号应采用 ASCII 字符集中的逗号 “,” , 而不是中文字符逗号 “,” 。请求之间使用换行分隔。以 run 为结束。(同一楼层可同时发出上行和下行请求)

m, n, T 均支持前导 0 和正号, 但不支持 -0 。

第一条请求只能是 (FR, 1, UP, 0) (1 和 0 均可以有前导 0 和正号);

输入实例: (FR,1,DOWN,0)

(FR,1,UP,1)

(ER,1,2)

(ER,6,4)

run

输入错误时响应信息:

INVALID[request]

并且当输入格式正确但出现类似第一条指令不是 (FR, 1, UP, 0) 或第一层 DOWN 第十层 UP 或时间超过整数范围的情况, 若输入相应位置的数字有前导 0 或正号时, 作为报错输出, 数字将不再会有前导 0 或正号。这种情况 [] 中不会再有小括号。而如果输入格式不合法 [] 里将会直接输出完整的不合法输入内容。

若输入 ctrl z, 不会有任何反应, 还可进行后续输入。

三、 输出说明

输出规范:

程序的输出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)

每个输出间换行分隔

其中 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）。

同层请求时输出为：[request] / (n, STILL, t) , 此处 t 应考虑开关门时间。

[] 中不会再有小括号，且若输入相应位置的数字有前导 0 或正号时，作为报错输出，数字将不再会有前导 0 或正号。

如果一次停靠了多条请求那么需要分行输出

例：

[FR, 1, UP, 0] / (1, STILL, 1.0)

[ER, 4, 1] / (4, UP, 2.5)

[FR, 8, DOWN, 0] / (8, UP, 5.5)

[FR, 8, UP, 1] / (8, UP, 5.5)

[ER, 8, 1] / (8, UP, 5.5)

[FR, 5, DOWN, 6] / (5, DOWN, 8.0)

[ER, 10, 6] / (10, UP, 11.5)

对于乱序请求应输出：INVALID[request]

[] 中不会再有小括号，且若输入相应位置的数字有前导 0 或正号时，作为报错输出，数字将不再会有前导 0 或正号。

乱序请求报错输出的位置在所有输入结束后程序开始运行时，且在所有正确输出前。

对于同质请求应输出：SAME[request]

[] 中不会再有小括号，且若输入相应位置的数字有前导 0 或正号时，作为报错输出，数字将不再会有前导 0 或正号。

同质请求提示输出的位置穿插在正常输入之间。但因为判断位置不同出现位置不固定。

当输出时间较大时，会采用科学记数法的形式输出