

OO 第三次作业

感谢同学能在周末抽空测我的程序~~~~

输入格式:

(FR, n, UP/DOWN, t) 或者 (ER, n, t)

测试时请跟我一起念: 无忌坐电梯啦! `elevator.exploded()`!

- 输入中允许包含空格 (space), 数字可带前导零, 不允许小数, 负数会报错。
- 第一条指令时间必须从0开始。
- 其中n 为1-10 的整数, t 为非负整数, t 除去前导零不超过int最大数, 前导零个数10个以内, 否则可能会报错。第一个请求中t 必须为0。
- UP/DOWN 表示输入UP 或者DOWN;
- 因为FR 中n 表示请求所在楼层数, 所以当n 为1 时不能为DOWN, n 为10 时不能为UP;
- 输入请求时尽量不要太多, 程序设定最多requests为200 (刷数量没意思对吧~);
- 控制台输入, 逐行输入, 最后一行由run 结束;
- 输入示例:
(FR, 3, DOWN, 0)
(FR, 1, UP, 1)
(ER, 1, 2)
(ER, 6, 4)

输出格式:

(n, UP/DOWN, t)

- 其中n 为1-10 的整数, t 为非负浮点数;
- 输出的t 只表示电梯由运动改为静止时的相对时间, 并不包括开关门的时间;
- 输出信息分为两部分, 电梯停靠部分, 主请求请求以及捎带部分 (无捎带时空)。
- 输出时间和方向说明。输出时间t不包括停止时的开关门时间。
方向为电梯从运动到静止的方向 (对应两种不同的请求), 如果两次楼层不变, 即电梯不动, 则不输出停靠信息。
- 样例输出 (以输入示例为例)
(FR,3,DOWN,0)()
(3, UP, 1.0)
(FR,1,UP,1)((ER,1,2))
(1, DOWN, 3.0)

```
(ER,1,2)()  
(ER,6,4)()  
(6, UP, 6.5)  
Program Stopped.
```

细节说明:

输入为逐行输入请求，一行一个请求，最后以run 表示输入结束，且输入请求序列t 按照顺序非严格递增；

当输入请求格式有问题时，程序自动略过，然后可以继续输入下一个请求，如果第一个请求有问题，那么第一个有效的请求的t 仍然要求为0；

同一时间多个不同输入请求顺序处理。

关于顺路捎带，严格满足教主表达式。

TIPS:

1. 空间上不会错过。对一个捎带请求，该请求产生时，它的目的楼层大于电梯所在的位置，考虑到电梯如果没有停靠计划时不能突然停下来。
2. 如果被捎带的请求在主请求完成后未完成，按照时间排序最早的变为主请求。
3. 一个主请求的响应过程中，每一层一次开关门能满足所有捎带请求。
4. 如果有多个完全相同的请求，会被合并成为一个请求。
5. 一次开关门会处理所有开关门前在这层楼的请求，但是开关门时到来的请求需要第二次开关门。

如：

```
(ER, 8, 0)  
(FR, 8, DOWN, 4)  
(ER, 3, 5)  
run
```

```
(ER, 8, 0) ( )  
(8, UP, 3.5)  
(FR, 8, DOWN, 4) ( ) //此请求到来时正在开关门，所以要重新开关门  
(ER, 3, 5) ( )  
(3, DOWN, 8.0) //3.5+2(两次开关门)+0.5*5 = 8
```

FUCK THE 00!

FUCK THE STUPID GUIDE BOOK!

FUCK THE INCOMPREHENSIBLE LOGIC EXPRESSION!

FUCK ALL!