

附录：

电梯运行过程详解（LOOK 算法）

输入：

1	楼层最小值：1
20	楼层最大值：20
<3,UP>	电梯外：3 楼向上请求
<5>	电梯内：去 5 楼
<6,DOWN>	电梯外：6 楼向下请求
<7>	电梯内：去 7 楼
<3>	电梯内：去 3 楼
end	输入结束标志

输出：

Current Floor: 1 Direction: UP	当前楼层：1，方向：向上
Current Floor: 2 Direction: UP	当前楼层：2，方向：向上
Current Floor: 3 Direction: UP	当前楼层：3，方向：向上
Open Door # Floor 3	当前楼层：3，开门
Close Door	关门
Current Floor: 4 Direction: UP	当前楼层：4，方向：向上
Current Floor: 5 Direction: UP	当前楼层：5，方向：向上
Open Door # Floor 5	当前楼层：5，开门
Close Door	关门
Current Floor: 6 Direction: UP	当前楼层：6，方向：向上
Current Floor: 7 Direction: UP	当前楼层：7，方向：向上
Open Door # Floor 7	当前楼层：7，开门
Close Door	关门
Current Floor: 6 Direction: DOWN	当前楼层：6，方向：向下
Open Door # Floor 6	当前楼层：6，开门
Close Door	关门
Current Floor: 5 Direction: DOWN	当前楼层：5，方向：向下
Current Floor: 4 Direction: DOWN	当前楼层：4，方向：向下
Current Floor: 3 Direction: DOWN	当前楼层：3，方向：向下
Open Door # Floor 3	当前楼当前楼层：3，开门
Close Door	关门

按照题目说明，乘客请求为一次性输入，乘客请求分为电梯内和电梯外两个队列，分别为：

电梯内：

InnerQueue	<5> <7> <3>
------------	-------------

电梯外：

OuterQueue	<3,UP> <6,DOWN>
------------	-----------------

电梯运行过程如下：

1. 电梯默认值停留在 1 层，状态为 Stopped（停止），方向为 IDLE（无方向）
2. 程序开始运行，检测 IQ（InnerQueue）和 OQ（OutQueue）两个队列排在最前面的请求，如果任意不为空，则代表有乘客请求，应该运行，检测结果为队列不为空，开始运行

3. 首先判断运行方向,离电梯当前层(1层)最近的为 OQ 内的第一个元素的请求楼层 3,3>1,则向上然后电梯开始向上运行,状态为 Moving (移动),方向为 UP (向上),输出:“当前楼层: 1, 方向: 向上”,
4. 运行到第 2 层,判断该楼层是否应该停下来处理请求,判断后应继续移动(请求是 3),输出“当前楼层: 2, 方向: 向上”
5. 运行到第 3 层,判断该楼层是否应该停下来处理请求,判断后应停止处理请求(请求是 3,当前楼层是 3),状态变为 Stopped,输出“Open Door # Floor 3 回车 Close Door”,此时将 IQ 和 OQ 两个队列中的队头的 3 层请求全部出队(删除)
6. 关门后电梯状态为“Stopped”,继续检测两个队列的队头请求,分别是<5>和<6,DOWN>,取同向且离得近的请求,即<5>,得出电梯运行方向应维持 UP
7. 继续向上运行,运行到第 4 层,判断该楼层是否应该停下来处理请求,判断后应继续移动(请求是 5),输出“当前楼层: 4, 方向: 向上”
8. 继续向上运行,运行到第 5 层,判断该楼层是否应该停下来处理请求,判断后应处理请求(请求是 5,且方向为 UP,与电梯同向),状态变为 Stopped,输出“Open Door # Floor 5 回车 Close Door”,此时将 IQ 和 OQ 两个队列中的队头的 5 层请求全部出队(删除)
9. 关门后电梯状态为“Stopped”,继续检测两个队列的队头请求,分别是<7>和<6,DOWN>,取同向且离得近的请求,即<7>,得出电梯运行方向应维持 UP
10. 继续向上运行,运行到第 6 层,判断该楼层是否应该停下来处理请求,判断后应继续移动(请求是 7),输出“当前楼层: 6, 方向: 向上”
11. 继续向上运行,运行到第 7 层,判断该楼层是否应该停下来处理请求,判断后应处理请求(请求是 7,且方向为 UP,与电梯同向),状态变为 Stopped,输出“Open Door # Floor 7 回车 Close Door”,此时将 IQ 和 OQ 两个队列中的队头的 7 层请求全部出队(删除)
12. 关门后电梯状态为“Stopped”,继续检测两个队列的队头请求,分别是<3>和<6,DOWN>,取同向且离得近的请求,结果为空,此时可判定当前电梯向上方向的请求已经全部处理完毕,此时方向应转向,即从 UP 到 DOWN,此时取同向离得最近的请求,即<6,DOWN>,得出电梯运行方向应向下
13. 继续向下运行,运行到第 6 层,判断该楼层是否应该停下来处理请求,判断后应处理请求(请求是 6,且方向为 DOWN,与电梯同向),状态变为 Stopped,输出“Open Door # Floor 6 回车 Close Door”,此时将 IQ 和 OQ 两个队列中的队头的 6 层请求全部出队(删除)
14. 关门后电梯状态为“Stopped”,继续检测两个队列的队头请求,只剩下是<3>,取同向且离得近的请求,即<3>,得出电梯运行方向应维持 DOWN
15. 运行到第 5 层,判断该楼层是否应该停下来处理请求,判断后应继续移动(请求是 3),输出“当前楼层: 5, 方向: 向下”
16. 运行到第 4 层,判断该楼层是否应该停下来处理请求,判断后应继续移动(请求是 3),输出“当前楼层: 5, 方向: 向下”
17. 运行到第 3 层,判断该楼层是否应该停下来处理请求,判断后应处理请求(请求是 3),状态变为 Stopped,输出“Open Door # Floor 3 回车 Close Door”,此时将 IQ 和 OQ 两个队列中的队头的 3 层请求全部出队(删除)
18. 关门后电梯状态为“Stopped”,继续检测两个队列的队头请求,结果为空,此时电梯运行结束,状态为“Stopped”,方向变为“IDEL”,电梯停留在 3 层