为了便于测试，把等待的时间缩短，跟实际情况有差异

电梯类：

[初始化]

[开门函数]：开门--如果传入参数为1，则调用【语音提示函数】--

--输入重量的变化（调用【计算载重函数】）--关门（调用【关门函数】）

[关门函数]：关门--等待两秒

[语音提示函数]：语音提示--到达楼层自动取消按下--没有楼层按下，暂时停止该楼层

[按铃求助函数]：求助--等待3秒

[电梯外上行键函数]：传入上行健按下的楼层--调用【按键函数】

[电梯外下行键函数]：传入下行键按下的楼层--调用【按键函数】

[按键函数]：

传入参数为按下的楼层（电梯内外）--

--原来停止--按下同楼层--开门（调用【开门函数】，传入开门函数的参数为0）

--最先按下的楼层高于目前楼层--确定方向为上行--目标楼层按键按下

--最先按下的楼层低于目前楼层--确定方向为下行--目标楼层按键按下

--原来上行--如果电梯外按下下行键，忽略

--按下的楼层高于已经到达的楼层--目标楼层按键按下

--原来下行--如果电梯外按下上行健，忽略

--按下的楼层低于已经到达的楼层--目标楼层按键按下

[手动取消楼层按键函数]：

判断原来该按键是否按下

按键原来是按下状态--

--原来上行--判断原来是否是原来要到达的最高层--

--是--判断已经到达的和要取消的楼层之间有没有楼层后来按下

--无--比现在到达的高一层的楼层改为按下状态--取消目标楼层按下

--有--直接取消目标楼层按下

--否--直接取消目标楼层按下

--判断原来是否是原来要到达的最低层--

--是--判断已经到达的和要取消的楼层之间有没有楼层后来按下

--无--比现在到达的低一层的楼层改为按下状态--取消目标楼层按下

--有--直接取消目标楼层按下

--否--直接取消目标楼层按下

[计算载重函数]：

运算输入的重量总和--大于等于630kg--超重--开门（开门函数）

--大于530kg小于630kg--准备超重状态--忽略电梯外后来按下的上行键 和下行键

--小于530kg--继续运行

[监听输入函数]：

监听键盘的输入，按下回车键停止，超过3秒没有开始输入则自动跳过

[转换键盘输入，控制按键函数]：

将键盘的输入转换成字符串：

只输入数字--表示电梯内按下的楼层--先调用【取消函数】，再调用【按键函数】

输入’o’--表示电梯内按下开门键--调用【开门函数】

输入’c’-- 表示电梯内按下关门键--调用【关门函数】

先输入’u’，再输入数字--表示电梯外按下上行键--调用【电梯外上行键函数】

先输入’d’，再输入数字--表示电梯外按下下行键--调用【电梯外下行键函数】

输入’h’--表示按下求助按键--调用【按铃求助函数】

主函数：

自动回停：如果超过1分钟按键无变化，则自动回停到1楼

判断电梯状态为上（下）行--电梯向上（下）运行，运行途中可接收键盘输入--当到达按下的楼层时，调用电梯类的【开门函数】，每次关门前需输入重量的变化