

# 推箱子-3

注：本次lab为推箱子游戏系列中的第三个lab，本次实验的截止日期为4月14日23:59

## 题目描述

本次lab在第二次lab的基础上增加对推箱子的支持：

1. 地图中加入箱子，使用编号5，保持其他编号的含义和前两次推箱子lab的相同
2. 当角色和箱子相邻时，如果合法输入使得角色向箱子所在方向移动，称之为“推箱子”
3. 箱子不可以穿过墙和障碍物，也不会存在箱子和这两种元素重叠的情况；如果“推箱子”的方向上不存在这样的阻碍，则角色会和箱子一起向该方向移动1步，称之为“推动箱子”，同时算作一次“合法移动次数”，反之分别称为“箱子碰墙”和“箱子碰障碍物”
4. 游戏延续了之前两名玩家的设定，每位玩家在各自的地图实例上，获胜的条件变为：箱子到达终点（与终点重合），玩家到达终点不再影响游戏进度，玩家同样可以输入q主动退出
5. 更改单个地图的获胜条件之后，两名玩家的输入次序合法性、输入字符合法性要求与第二次lab保持一致
6. 两名玩家的胜负判定逻辑同第二次lab略加调整：如果两人都到达终点，合法移动次数、角色进入障碍物次数、角色碰墙次数、箱子碰墙次数、箱子碰障碍物次数之和较小的玩家判定为胜利，并输出胜利的玩家昵称；如果数值相同，则箱子先到达终点者胜；如果只有一个玩家的箱子到达终点则直接判定该玩家获胜，并输出昵称；如果没有玩家的箱子到达终点，则输出 `draw`

## 输入格式

本次lab采用读取命令行的形式进行数据输入。

1. 首先输入角色昵称，中间用空格分开
2. 然后输入地图，其中1表示墙壁，0表示空地，2表示角色的起点，3表示箱子的终点，4表示仅角色可穿越的障碍物，5表示箱子，地图元素之间使用空格进行分割；
3. 两个玩家在相同的地图，不同的实例上进行游戏；
4. 地图输入完成后的下一行开始输入指令序列，每条指令占据一行，采取 昵称 + 指令 的方式进行描述，昵称和指令中间使用空格分隔

## 输出格式

输出格式为固定格式：

1. 按照角色的数目，按照输入昵称的顺序，输出每位玩家的数据统计，格式如下，数据之间用空格分隔，不同玩家需要不同行输出，按照输入的角色昵称的顺序，其中，只要没有违反移动次序与输入正确性的指令都算成合法移动

角色昵称 角色的合法移动次数 错误移动次序数 错误输入指令数 角色碰墙次数 角色进入障碍物次数 箱子碰墙次数 箱子碰障碍物次数

2. 另起一行，按照题目描述中的规则判定胜负，并输出获胜玩家昵称或表示平局的 `draw`

## 样例

### 样例输入

```
1 rick morty
2 1 1 1 1
3 1 0 0 2 1
4 1 0 5 0 1
5 1 3 4 0 1
6 1 1 1 1
7 morty j
8 morty g
9 rick i
10 morty j
11 rick h
12 morty h
13 rick j
14 rick l
15 morty h
16 rick l
17 morty k
18 rick j
19 morty h
20 rick h
21 morty j
22 rick k
23 rick h
24 morty j
25 rick j
26
```

### 样例输出

```
1 rick 8 1 1 0 0 0 1
2 morty 6 3 0 0 0 1 0
```

## 样例解读

- 1 输入昵称为 rick morty
- 2 之后由morty先行移动，在第8行，由于rick上一轮已经进行了移动，此命令判定为非法移动次序，次序和字符同时出错时仅计算次序非法。
- 3 第9行 rick输入了一个无效移动指令，需要重新进行输入，因此地10行morty的输入指令是次序非法的
- 4 第13行 rick推的箱子与障碍物碰撞。
- 5 第15行 morty推的箱子与墙壁发生碰撞。
- 6 由于第21行 morty已经移动到终点，所以第24行的移动指令被判定为错误移动次序
- 7 由于统计数据中，两人都推箱子到达了终点，因此计算判定胜负的数值，rick为 $8+0+0+0+1=9$ ，morty为 $6+0+0+1+0=7$ ，因此morty取胜

## 约定

输入时保证地图中的元素不会重叠，角色位于起点是个例外

输入保证两人都能到达终点或者退出，使得游戏不会进入死循环

输入保证角色的名字中没有空白符，有且仅有两名角色参加游戏

指令中表示角色的昵称保证不会出现参与玩家之外的情况

指令中表示指令的部分为单个字符

第一条指令保证合法