

果冻消除（服务器端）

有一个果冻消除游戏，这个消除游戏的基本规则是：

1. 有一个8x8的棋盘，棋盘每个格子可以用一个二维坐标表示[row, col]。其中从上到下依次是第0, 1,7行，从左到右依次是第0, 1,7列；
2. 棋盘的每个格子中有一个果冻，可以是以下四种类型中的一种：普通果冻、横炸弹果冻、竖炸弹果冻和方炸弹果冻；
3. 每次用户操作时，会在棋盘上选择一个矩形区域(AABB)。该矩形可用其左上角和右下角的坐标来确定，比如[0, 1] [2, 2]；
4. 矩形内的所有元素都会被消除，对于[0, 1] [2, 2]这个矩形来说，这6个元素都会被消除：[0, 1], [0, 2], [1, 1], [1, 2], [2, 1], [2, 2]；
5. 炸弹果冻在消除后会有特殊的效果：横炸弹会消除掉所在行的所有果冻；竖炸弹会消除掉所在列的所有果冻；方炸弹会消除掉周围一圈8个果冻；
6. 被炸弹消除的果冻也按照第5条规则处理；
7. 消除结束后，被消掉的地方会由上方的果冻掉落进行填补。第0行的元素被消掉（或下落到下方）时，会由系统随机生成一个果冻进行填补（4种果冻的生成概率均等）。

你的任务是开发一个服务器端程序（使用HTTP协议），实现如下的2个接口：

1. 开始某一关卡

`/start-level?level=<关卡编号>`

返回：

第一行为一个字符串(sessionId)，用于唯一标识一个进行中的关卡，sessionId的格式可以自行确定；

后续8行表示关卡初始布局，对应棋盘中的8行，每行包括8个字符。每个字符表示一种果冻，对应关系为：（B普通果冻，H横炸弹果冻，V竖炸弹果冻，S方炸弹果冻）

2. 一次消除操作

`/move?`

`sessionId=<sessionId>&row0=<row0>&col0=<col0>&row1=<row1>&col1=<col1>`

参数：

sessionId: 第2个接口返回的sessionId

row0, col0, row1, col1表示一次操作的左上角和右下角的果冻坐标，比如：

`row0=0&col1=1&row1=2&row2=2`

返回：

8行文本，表示消除结束且完成下落后的局面，格式与接口1的返回中后8行相同。

举例：

请求： `/start-level?level=1`

响应：

```
40ab3f2j
HBBBSBBB
BBBBBBBB
BSBBBBBB
BBBVB BBB
BHBBBBBB
SBBBBBBS
BBBBBBBB
BBBBBBBB
```

请求： `/move?sessionId=40ab3f2j&row0=1&col0=0&row1=1&col1=2`

响应：

```
VSBBBSBB
HBBBBBBB
BSBBBBBB
BBBVB BBB
BHBBBBBB
SBBBBBBS
BBBBBBBB
BBBBBBBB
```

说明：消除了第1行的前3个普通果冻后，第0行的前3个果冻掉落到第1行，第0行由系统补充了3个果冻，分别是V、S、B

请求： `/move?sessionId=40ab3f2j&row0=5&col0=0&row1=6&col1=2`

响应

```
BBSBBBBH
SBBBSBBB
BBHBBBBB
VSBBBBBB
HBBVB BBB
BSBBBSBB
BBBBBBBB
BBBBBBBB
```

说明：初始消除了5个普通果冻和一个方炸弹，方炸弹爆炸时，炸到了[4,0]的普通果冻和[4,1]的横炸弹，炸弹将第4行全部消除

其他要求：

1. 使用关系型数据库做持久化存储
2. 每一关的初始布局保存在数据库中
3. 支持多个用户并行玩游戏，游戏服务器重启或切换（如负载均衡）后不影响用户游戏进程
4. 所有接口均以HTTP GET请求的方式由客户端向服务器发送，服务器以HTTP响应的方式返回给客户端，返回格式为纯文本（不带任何HTML标签）
5. 如果用户给出的参数非法（无论是类型还是数值），仅返回一行文本：“INVALID PARAMS”
6. 最后提交的项目需要包含完整的项目文件和源代码，以及创建数据库的相关脚本