

# 三重镇

## 【问题描述】

小西同学最近喜欢上了 iOS 游戏《三重镇 Triple Town》。游戏之余，小西也在思考如何才能在这个游戏中获得更高的分数。

如图 1 所示，游戏在一个  $n * m$  的地图中进行。游戏给定一个建造序列，玩家按照此建造序列依次选择空白位置建造相应的建筑单位。建筑有九个不同的等级，由低到高分别为 Grass, Bush, Tree, Hut, ... 等（为了方便描述，我们称之为  $L_1, L_2, L_3, \dots, L_9$ ）。



图 1

当玩家在一个空白位置建造单位之后，有可能引起反应。反应的构成条件是：从这个格子出发，与该建筑单位等级相同的格子所构成的连通块大小大于等于 3，则这个连通块将被合并为一个下一等级的建筑，此建筑的位置为最后建造的建筑单位位置，连通块中其他位置将变回空格。这里的连通块是指直接或者间接相邻的位置集合。另外需要注意的是， $L_9$  为建筑的最高等级，所以多个  $L_9$  的连通块并不会合并。例如在图 2 中，当建造了中间的  $L_1$  之后，与该位置相连的  $L_1$  就被合并成了一个  $L_2$ 。

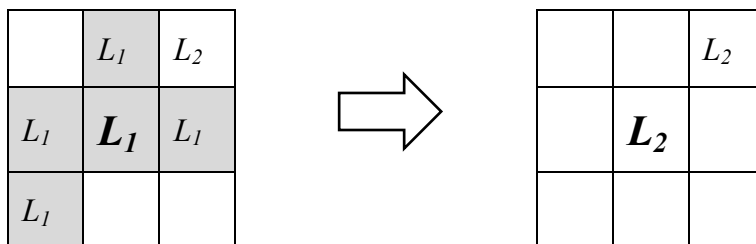


图 2

注意，在合并的过程中，可能会引起连环反应。如下图所示。

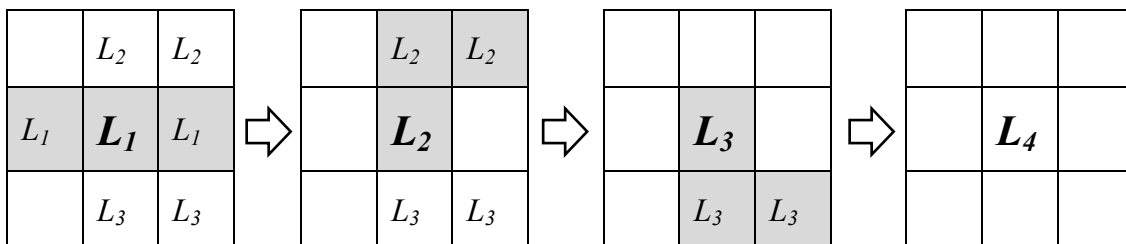


图 3

游戏的得分取决于玩家建造和反应生成的单位，建造或者反应生成建筑单位就可以获得相应的分数。不同等级建筑的得分表如下：

建筑	$L_1$	$L_2$	$L_3$	$L_4$	$L_5$	$L_6$	$L_7$	$L_8$	$L_9$
得分	4	20	100	500	1,500	5,000	20,000	100,000	500,000

以刚才的两个游戏过程为例。图 2 中，首先建造了  $L_1$ ，将得到 4 分，随后， $L_1$  进行了反应生成了  $L_2$ ，此时再得到 20 分。总共得分为 24。而在图 3 中，这一步操作得分为  $4+20+100+500=624$  分。

为了降低游戏的难度，游戏中还设有两种道具，分别为“星星”和“炸弹”。在游戏开始时，玩家被给定  $p$  个星星道具和  $q$  个炸弹道具，玩家可以在任意时刻使用。两者功能如下：

“星星”道具	可以放置在一个空格位置。当星星被放置时，星星会自动变为 <b>能引起反应的最高等级建筑</b> 。当在该位置不能引起任何反应时，星星变为 $L_1$ 。例如，在图 3 正中间位置放置星星，星星自动变为 $L_3$ 。星星的得分按照变化后的建筑计算得分。
“炸弹”道具	炸弹道具可以放在在一个有建筑的位置上，作用为炸掉这个建筑并将该位置恢复为空格。当使用炸弹时，得分将扣除被炸掉的建筑的一半分数（即，得分为负数）。

在游戏的进行过程中，玩家必须按照给定的顺序进行建造，但可以随时穿插使用两种道具。**游戏的目标是，通过合理的操作，取得最高的分数。**

为了更好的帮助同学们理解这个游戏，小西还编写了一个简单的游戏 Demo 放在选手目录中，大家可以进行简单的尝试以帮助理解。

### 【输入格式】

该题为提交答案型试题，所有输入数据 tritown1.in~tritown10.in 已在试题目录下。

对于每个数据，输入文件中第一行为两个整数  $n, m$ ，表示地图一共包含  $n$  行  $m$  列。

接下来一行包含两个整数  $p, q$ ，分别表示道具星星和道具炸弹的数目。

接下来  $n$  行包含一个  $n * m$  的初始地图。其中字符“.”表示空地，数字 1~9 分别表示相应等级的建筑。

再接下来一行包含一个数字  $k$ ，表示建造序列的长度。

最后一行包含  $k$  个空格隔开的 1~9 之间的数字，表示建造序列的内容。

**【输出格式】**

针对给定的 10 个输入文件 tritown1.in~tritown10.in, 你需要分别提交你的输出文件 tritown 1.out~ tritown 10.out。

输出文件包含玩家进行游戏的指令, 共 4 种指令:

指令	含义
PUT $x$ $y$	将建造序列中的下一个单位放置到第 $x$ 行第 $y$ 列的空格中。
STAR $x$ $y$	放置星星道具到第 $x$ 行第 $y$ 列的空格中。
BOMBER $x$ $y$	在第 $x$ 行第 $y$ 列放置炸弹, 此位置必须非空。
END	游戏结束, 结算当前得分。

输出必须以 END 指令结尾, 玩家可以在任意时刻结束游戏。

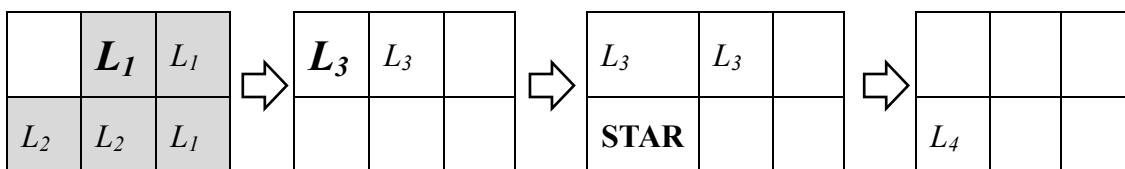
**【样例输入】**

```
2 3
1 1
..1
221
2
1 3
```

**【样例输出】**

```
PUT 1 2
PUT 1 1
STAR 2 1
END
```

**【样例说明】**



第一步得分为  $4+20+100=124$ ;  
 第二步得分为 100;  
 第三步得分为  $100+500=600$ ;  
 游戏总得分  $124+100+600=824$  分。

**【评分方法】**

对于每组数据，我们设置了 9 个评分参数  $a_{10}, a_9, \dots, a_2$ 。如果选手的输出不合法，则得零分。否则，设在你的方案中，游戏得分为  $w_{\text{user}}$ ，你的分数将会由下表给出：

得分	条件	得分	条件
10	$w_{\text{user}} \geq a_{10}$	5	$w_{\text{user}} \geq a_5$
9	$w_{\text{user}} \geq a_9$	4	$w_{\text{user}} \geq a_4$
8	$w_{\text{user}} \geq a_8$	3	$w_{\text{user}} \geq a_3$
7	$w_{\text{user}} \geq a_7$	2	$w_{\text{user}} \geq a_2$
6	$w_{\text{user}} \geq a_6$	1	$w_{\text{user}} > 0$

如果有多项满足，则取满足条件中的最高得分。

**【如何测试你的输出】**

我们提供 checker 这个工具来测试你的输出文件是否是可接受的。使用这个工具的方法是，在终端中运行：

```
./checker <case_no>
```

在你调用这个程序后，checker 将根据你给出的输出文件给出测试的结果，其中包括：

- 非法退出：未知错误；
- Output File Do Not Exist：找不到输出文件；
- Output Invalid!：输出文件有误，此时可能包含具体错误信息；
- Correct! Your score is x：输出可接受，你的得分  $x$ 。

**【特别提示】**

请妥善保存输入文件 tritown\*.in 和你的输出 tritown\*.out，及时备份，以免误删。