

A Dickey Problem

问题描述

图 1 中 3*3 的网格是一个骰子地图。一个标准的六面骰子需要在地图上移动（图 2 是一个标准六面骰子的展开图）。每张地图有一个规定的初始位置和一个初始的骰子放置方式。在图 1 中，初始位置是第一行第二列——写着“2”的位置。假设你从地图的底边上观察，骰子的初始放置方式是底面（贴在地图上的那一面）是“2”，顶面（与底面相对的那一面）是“5”，“1”面向你。

你可以通过沿着骰子的一条边转动来将它移动到地图上水平或是竖直方向上相邻的格子上。假设当前骰子顶面的数字为 x ，那么你只能将它移动到写着 x 的格子上，或者画着星星图案的格子上。我们的最终目标是找到一条路径，使得骰子能从起点出发，最后又回到起点。

举个例子，在图 1 所示的地图中，一开始你有两种可行的移动方式——向下或者向左。由于此时骰子顶面所示的数字为“5”，当前格子下方的格子上写的数字也是“5”，因此向下移动骰子是可行的；因为当前位置左边的格子画着星星图案，因此向左移动是可行的。如果第一步选择向下移动，那么骰子朝上一面的图案会变成“6”，此时你可以向右或者向下移动骰子。如果第一步选择向左移动，那么骰子顶面的数字会变成 3，这种情况下，任何移动方式都是不可行的。

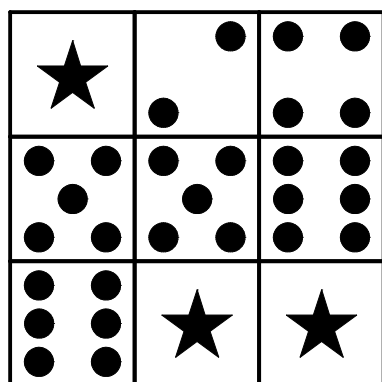


图 1: 一个骰子地图的例子

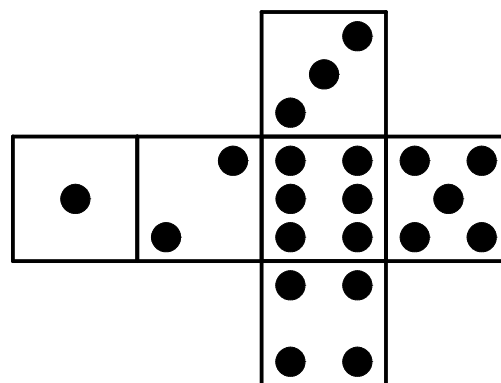


图 2: 一个六面骰子的展开图

我们用 (行号, 列号) 的方式来标记格子。行号从 1 开始，从上向下递增；列号从 1 开始，从左向右递增。这样，在上述例子中一个可行的解可以表示为: (1,2), (2,2), (2,3), (3,3), (3,2), (3,1), (2,1), (1,1), (1,2)。图 3 中展示了一个更加困难的例子。

在本题中，你需要编写程序对输入中给定的地图进行求解。每张地图可能存在一个唯一的解或者无解。也就是说，对于某一张地图，如果有解，你可以认为它只有一个解。对于每张地图，你需要输出一解或者一个信息表示无解。

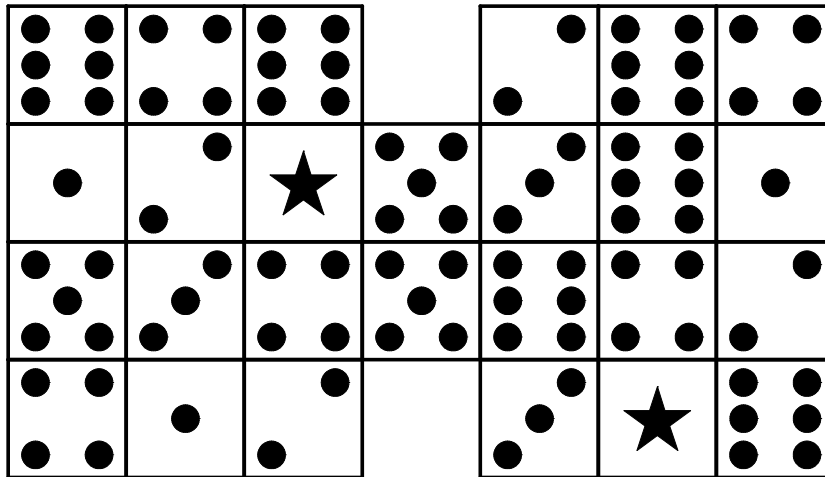


图 3: 初始位置(2,6), 骰子的顶面是数字 3, 数字 6 面向你。

输入格式

第一行包含 6 个用空格分隔的整数 R, C, x0, y0, T, F, 分别表示地图的行数、列数、初始位置行号、初始位置列号、初始状态下顶面的数字、初始状态下面向你的数字。

接下来的 R 行, 每行包含 C 个空格分隔的整数, 定义了这个地图。如果某个数是 0, 则表示对应的格子为空 (不可以被走到); 如果某个数是 -1, 则表示对应单元格是画着星星图案的单元格; 其余情况下, 这个数表示对应格子上写的数字。

输出格式

若该地图无解, 则输出一行 “No Solution Possible” (不包含引号)。否则, 你应该输出一个逗号分隔的位置序列, 表示骰子依次经过的位置。注意你输出的第一个位置和最后一个位置应该是相同的 (起始位置已在输入中规定), 位置的格式如上文所示。除了最后一行外, 每行应该包含九个位置。另外你的输出中不应包含空格。

样例输入 1

```
3 3 1 2 5 1
-1 2 4
5 5 6
6 -1 -1
```

样例输出 1

```
(1,2),(2,2),(2,3),(3,3),(3,2),(3,1),(2,1),(1,1),(1,2)
```

样例输入 2

```
4 7 2 6 3 6
6 4 6 0 2 6 4
```

```
1 2 -1 5 3 6 1
5 3 4 5 6 4 2
4 1 2 0 3 -1 6
```

样例输出 2

```
(2,6),(2,5),(2,4),(2,3),(2,2),(3,2),(4,2),(4,1),(3,1),
(2,1),(2,2),(2,3),(2,4),(2,5),(1,5),(1,6),(1,7),(2,7),
(3,7),(4,7),(4,6),(3,6),(2,6)
```

样例输入 3

```
3 3 1 1 2 4
2 2 3
4 5 6
-1 -1 -1
```

样例输出 3

No Solution Possible

数据规模和约定

20%的数据， $R \times C \leq 30$;

100%的数据， $R, C \leq 10$ ，保证初始状态合法。