

移球游戏 (ball)

【题目描述】

小 C 正在玩一个移球游戏，他面前有 $n + 1$ 根柱子，柱子从 $1 \sim n + 1$ 编号，其中 1 号柱子、2 号柱子、 \dots 、 n 号柱子上各有 m 个球，它们自底向上放置在柱子上， $n + 1$ 号柱子上初始时没有球。这 $n \times m$ 个球共有 n 种颜色，每种颜色的球各 m 个。

初始时一根柱子上的球可能是五颜六色的，而小 C 的任务是将所有同种颜色的球移到同一根柱子上，这是唯一的目标，而每种颜色的球最后放置在哪个柱子则没有限制。

小 C 可以通过若干次操作完成这个目标，一次操作能将一个球从一根柱子移到另一根柱子上。更具体地，将 x 号柱子上的球移动到 y 号柱子上的要求为：

1. x 号柱子上至少有一个球；
2. y 号柱子上至多有 $m - 1$ 个球；
3. 只能将 x 号柱子最上方的球移到 y 号柱子的最上方。

小 C 的目标并不难完成，因此他决定给自己加加难度：在完成目标的基础上，使用的操作次数不能超过 820000。换句话说，小 C 需要使用至多 820000 次操作完成目标。

小 C 被难住了，但他相信难不倒你，请你给出一个操作方案完成小 C 的目标。合法的方案可能有多种，你只需要给出任意一种，题目保证一定存在一个合法方案。

【输入格式】

从文件 *ball.in* 中读入数据。

第一行两个用空格分隔的整数 n, m 。分别表示球的颜色数、每种颜色球的个数。

接下来 n 行每行 m 个用单个空格分隔的整数，第 i 行的整数按自底向上的顺序依次给出了 i 号柱子上的球的颜色。

【输出格式】

输出到文件 *ball.out* 中。

本题采用自定义校验器 (special judge) 评测。

你的输出的第一行应该仅包含单个整数 k ，表示你的方案的操作次数。你应保证 $0 \leq k \leq 820000$ 。

接下来 k 行每行你应输出两个用单个空格分隔的正整数 x, y ，表示这次操作将 x 号柱子最上方的球移动到 y 号柱子最上方。你应保证 $1 \leq x, y \leq n + 1$ 且 $x \neq y$ 。

【样例 1 输入】

```
1 2 3
2 1 1 2
3 2 1 2
```

【样例 1 输出】

```
1 6
2 1 3
3 2 3
4 2 3
5 3 1
6 3 2
7 3 2
```

柱子中的内容为：按自底向上的顺序依次给出柱子上每个球的颜色。

【样例 1 解释】

操作	1 号柱子	2 号柱子	3 号柱子
初始	1 1 2	2 1 2	
1 3	1 1	2 1 2	2
2 3	1 1	2 1	2 2
2 3	1 1	2	2 2 1
3 1	1 1 1	2	2 2
3 2	1 1 1	2 2	2
3 2	1 1 1	2 2 2	

【样例 2】

见选手目录下的 *ball/ball2.in* 与 *ball/ball2.ans*。

【样例 3】

见选手目录下的 *ball/ball3.in* 与 *ball/ball3.ans*。

【数据范围】

操作	1 号柱子	2 号柱子	3 号柱子
初始	1 1 2	2 1 2	
1 3	1 1	2 1 2	2
2 3	1 1	2 1	2 2
2 3	1 1	2	2 2 1
3 1	1 1 1	2	2 2
3 2	1 1 1	2 2	2
3 2	1 1 1	2 2 2	

对于所有测试点，保证 $2 \leq n \leq 50$ ， $2 \leq m \leq 400$ 。

【校验器】

为了方便选手测试，在ball目录下我们下发了checker.cpp文件，选手可以编译该程序，并使用它校验自己的输出文件。但请注意它与最终评测时所使用的校验器并不完全一致。你也不需要关心其代码的具体内容。

编译命令为：`g++ checker.cpp -o checker`。

checker的使用方式为：`checker <iuputfile> <outputfile>`，参数依次表示输入文件与你的输出文件。

若你输出的数字大小范围不合法，则校验器会给出相应提示。若你的输出数字大小范围正确，但方案错误，则校验器会给出简要的错误信息：

1. **A x**，表示进行到第 x 个操作时不合法。
2. **B x**，表示操作执行完毕后第 x 个柱子上的球不合法。

若你的方案正确，校验器会给出**OK**。