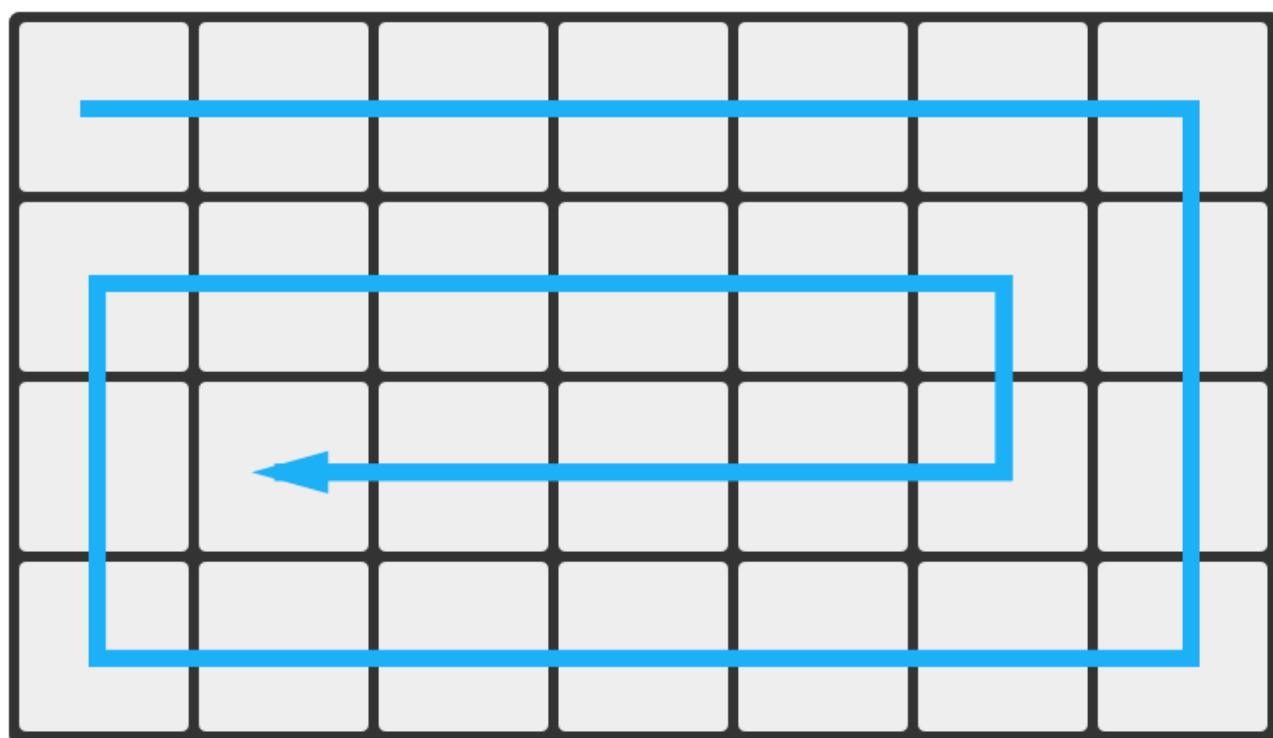


# 矩阵螺旋输出

## 限制

- 1000 ms
- 32768 KB

给定一个  $m$  行、 $n$  列的矩阵，请按照顺时针螺旋的顺序输出矩阵中所有的元素（从 `[0][0]` 位置开始，具体请参见下图）。



## 输入格式

测评机会反复运行你写的程序。每次程序运行时，首先在第一行输入 2 个整数，分别对应题目描述中的  $m$  和  $n$  ( $1 \leq m, n \leq 100$ )，之间用一个空格分隔。接下来输入  $m$  行，每行包含  $n$  个整数 ( $-10000 \leq a, b, c \leq 10000$ )，每两个整数之间用一个空格分隔。

## 输出格式

输出为一行，包括  $m \times n$  个整数，按照题目要求的顺序依次输出所有矩阵元素，任意两个整数之间用一个空格分隔，最后一个整数后面没有空格。

---

## 样例输入 1

```
2 3
1 2 3
3 4 6
```

## 样例输出 1

```
1 2 3 6 4 3
```

---

## 样例输入 2

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

## 样例输出 2

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

---

## 提示信息

每次运行后你可以使用二维数组存储读入的矩阵。请在输出时思考：题目描述的“顺时针螺旋”对应到二维数组元素的索引变化上会有什么规律，你是否可以推出其中的规律并且正确将其输出。

顺时针螺旋输出的过程可以看作一系列“向右-向下-向左-向上”进行输出的过程，只不过在这个过程中，要确保输出过的元素不被重复地访问和输出。

你需要考虑到以下一些情况：

- 只有一行
- 只有一列
- 奇数行、偶数列
- 偶数行、奇数列
- 偶数行、偶数列
- 奇数行、奇数列

如果你会在矩阵边界判断上很纠结。针对这个问题，在这里给大家两种不同的思路：

- 你可以用另一个等大的二维数组来标记对应位置的数字是否已经被输出了
- 你可以把每次沿着一个方向的输出看成一个独立过程，考虑到每沿着一个方向输出后剩余的待输出部分还是一个矩阵，你也可以通过四个变量分别记录现在还没被输出的矩阵的行数和列数可能的最小值、最大值。

请注意，如果你希望输出的行末没有多余的空格，在输出最后一个元素时，你将有可能需要用到 `if` 语句（就像我们在前面的课程中已经做过的一样）。

请注意，不要让你的程序输出任何多余的内容，否则测评机都会给出“运行结果错误”的提示。