American Computer Science League

2020-2021 ● Contest 3: Multiple Arrays ● Intermediate Division

问题：给定 3 个形状相同的二维数组，从左上角开始在数组中移动。每一次移动，都要注意右侧和下侧的相邻单元格。在 3 个数组中找出最大的值然后移动到这个值所在的单元格位置。但是，如果最大的值有两个或两个以上，则移动到对角线上向下靠右的位置。当移动到最后一行或最后一列上的单元格位置时，则停止移动。打印输出经过的每个单元格位置上最小的值之和。

数组 A			
1	**2**	**3**	**4**
7	**7**	**8**	**9**
5	**6**	**7**	**8**

数组 B			
6	**8**	**6**	**4**
4	**5**	**3**	**2**
8	**3**	**1**	**9**

数组 C			
3	**6**	**7**	**3**
4	**6**	**2**	**1**
3	**2**	**5**	**5**

在上面例子中, 移动过程如下:

步骤 1: 从左上角的单元格 (0, 0) 开始。数组 A 中与之相邻的是 2 和 7 , 数组 B 中是 8 和 4 , 数组 C 中是 6 和 4，最大的是 8 ，所以移动到右边，至单元格位置(0, 1) 处；

步骤 2: 在数组 A 中，与 (0, 1) 相邻的单元格是 3 和 7 ; 数组 B 中为 6 和 5 ; 数组 C 中为 7 和 6。从数组 A 到数组 C 中最大的是 7，沿对角线上向下靠右进行移动, 至单元格位置 (1, 2) 处；

步骤 3: 在 (1, 2)处, 集合 {9, 7, 2, 1, 1, 5} 中最大值为 9，所以移动到位置 (1, 3) 时停止移动，因为 (1, 3) 在最后一列。

因此, 该例子中经过的单元格位置有 (0, 0), (0, 1), (1, 2), 和 (1, 3)。在 3 组数组中，每一组中最小的值为 1, 2, 2, 和 1, 加起来和为 6。

输入: 你将会接收到一组数据，每组包含 4 行整数值。第一行将包含两个正整数，表示每个数组的行数和列数。接下来的三行都将按行进行排列，包含 3 个数组的值，从(0, 0)开始.

输出: 打印输出所经过位置的最小值的总和.

American Computer Science League

2020-2021 ● Contest 3: Multiple Arrays ● Intermediate Division

样本输入:

```
3 4
1 2 3 4 7 7 8 9 5 6 7 8
6 8 6 4 4 5 3 2 8 3 1 9
3 6 7 3 4 6 2 1 3 2 5 5
4 2
31 17 24 19 15 29 22 26
25 13 25 18 19 27 19 13
12 15 17 18 29 16 25 20
4 5
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4
6 2 6 4 3 3 8 3 2 7 7 2 3 8 3 3 4 6 2 6
5 8 3 2 3 9 7 9 8 5 3 5 6 2 9 5 1 4 1 3
```

预期输出:

```
6

60

16
```

American Computer Science League

2020-2021 ● Contest 3: Multiple Arrays ● Intermediate Division

TEST INPUT (http://xxxx):

```
4 3
3 1 4 1 5 2 2 6 5 3 5 8
9 7 1 3 2 6 8 4 6 2 6 4
3 2 1 3 2 1 1 2 3 1 2 3
4 6
4 8 3 7 1 6 7 6 2 4 3 3 7 5 1 0 5 8 2 0 9 5 3 2
4 4 5 9 2 3 0 2 1 9 6 4 0 6 2 8 4 2 0 8 9 3 5 2
6 9 5 0 3 4 8 7 5 3 4 2 1 1 7 10 6 7 9 3 1 2 3 2
5 3
31 41 59 26 53 58 97 93 23 84 62 64 33 83 27
95 28 84 19 71 69 39 93 75 10 85 20 97 49 44
59 23 78 61 40 62 97 20 49 98 62 80 34 83 53
4 5
-3 -1 -4 -1 -5 -9 -2 -6 -5 -3 -5 -8 -9 -7 -4 -3 -5 -3 -8 -4
-6 -2 -6 -4 -3 -3 -8 -3 -2 -7 -9 -5 -2 -8 -8 -4 -4 -9 -7 -1
-6 -9 -3 -9 -9 -3 -7 -5 -1 -5 -8 -2 -9 -7 -4 -9 -4 -4 -5 -9
5 5
1 2 3 4 5 6 7 8 9 10 11 12 13 12 11 10 9 8 7 8 5 4 3 2 1
2 4 6 8 10 12 14 16 18 20 22 20 18 16 14 12 10 8 26 4 2 4 6 8 10
1 3 5 7 9 11 13 15 17 19 21 23 25 23 21 19 17 15 13 1 9 7 5 3 1
```

预期输出 (http://xxx):

```
9
11
159
-49
63
```