滑雪

（http://noi.openjudge.cn/ch0206/90/）

**描述**

Michael喜欢滑雪百这并不奇怪， 因为滑雪的确很刺激。可是为了获得速度，滑的区域必须向下倾斜，而且当你滑到坡底，你不得不再次走上坡或者等待升降机来载你。Michael想知道载一个区域中最长的滑坡。区域由一个二维数组给出。数组的每个数字代表点的高度。下面是一个例子

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

一个人可以从某个点滑向上下左右相邻四个点之一，当且仅当高度减小。在上面的例子中，一条可滑行的滑坡为24-17-16-1。当然25-24-23-...-3-2-1更长。事实上，这是最长的一条。

**输入**

输入的第一行表示区域的行数R和列数C(1 <= R,C <= 100)。下面是R行，每行有C个整数，代表高度h，0<=h<=10000。

**输出**

输出最长区域的长度。

**样例输入**

5 5

1 2 3 4 5

16 17 18 19 6

15 24 25 20 7

14 23 22 21 8

13 12 11 10 9

**样例输出**

25

Two Ends

<http://poj.org/problem?id=2738>

**描述：**

这里有一个游戏，将偶数张纸牌摆成一列，每张纸牌上都有一个正整数，玩家有两个，他们都轮流选一张牌，这张牌只能是最左边的或者最右边的牌，最后看谁的牌上的数字总和最大，谁就是胜利者。如果每次都选两端最大的牌，被称为是贪心策略，但是贪心策略并不一定总是最佳的，比如以下数据：

3 2 10 4

第一个玩家第一次选择3会比选择4更好。

现在如果第二个玩家总是做贪心选择，第一个玩家最多能赢第二个玩家多少分？

**输入：**

有多组测试数据，每组测试数据一行：

第一个数为一个偶数n（n<=1000）

接下来会有n个正整数，保证正整数都不超过1000000

当n为0表示输入结束；

**输出**

对于每组数据都有单独的一行输出；

输出内容格式如下：

In game m, the greedy strategy might lose by as many as p points.

其中m为第几组游戏；

p为最多能赢的分数;

**样例输入：**

4 3 2 10 4

8 1 2 3 4 5 6 7 8

8 2 2 1 5 3 8 7 3

0

**样例输出：**

In game 1, the greedy strategy might lose by as many as 7 points.

In game 2, the greedy strategy might lose by as many as 4 points.

In game 3, the greedy strategy might lose by as many as 5 points.