

# Snakes on a Plane



Consider a 2-dimensional array of positive integers in which you must construct the longest possible snake made up of adjacent cells. The construction the snake is further constrained in that each cell in the snake's body must have a value in the array that is larger than the previous cell.

Notes: A cell is adjacent to at most four cells –the cell above, below, to the left, and the right. The matrix does not wrap around, i.e. in a matrix with more than two rows, the cells at the top of the matrix are not adjacent to the cells at the bottom of the matrix. Similarly, in a matrix with more than two columns, the cells in the first column are not adjacent to the cells in the last column.

## Input Format

The input begins with two positive integers,  $n$  and  $m$ , giving the dimensions of the array, where  $n$  is the number of rows in the array, and  $m$  is the number of columns. Following this are the values in the array, displayed in  $n$  lines, each containing  $m$  integers separated by spaces.

## Constraints

The values in the array will be greater than  $-10^9$  and less than  $10^9$ .

$$0 < n, m \leq 10^4$$

## Output Format

The output of the program is an integer containing the length of the longest possible snake.

## Sample Input 0

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

## Sample Output 0

```
6
```

## Explanation 0

The snake could appear as follows, where arrows point towards the head.

```
. ^ .
> ^ .
^ < <
```

## Sample Input 1

```
4 5
9 8 9 9 9
```

```
6 7 9 9 9
5 4 1 0 1
4 3 2 1 2
```

### Sample Output 1

```
10
```

### Explanation 1

The snake could appear as follows:

```
.v<..
v<...
v...^
>>>^.
```