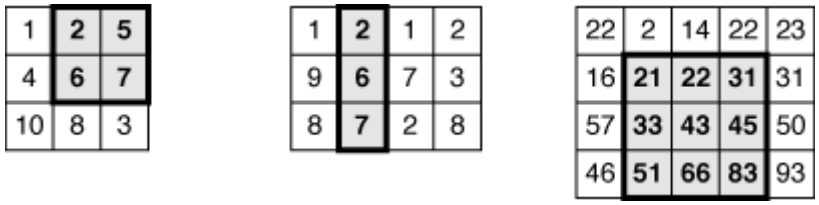
UVa Online Judge

11499 Longest Increasing Sub-sequence

The problem of determining the longest increasing sub-sequence in a list of numbers is already a classic in programming competitions. Despite this fact, that is the problem you must solve here. But, in order to avoid having you yawning while solving the problem, we introduced a small change: the list of numbers will be given as a bidimensional matrix, and the increasing sequence must be "embedded" in a submatrix of the original matrix.

Let's define the problem more precisely. The linearization of a bidimensional matrix is the concatenation of its lines, from the first to the last. A submatrix is a rectangular region of a bidimensional matrix (with sides paralel to the sides of the matrix). The size of a submatrix is its number of elements. You must write a program that, given a matrix of integers, determines the largest submatrix such that, when linearized, results in an increasing sequence.

The figure below shows some examples of submatrices of largest size which contain increasing sequences. Notice that more than one such a submatrix may exist in a given matrix.



Input

The input contains several test cases. The first line of a test case contains two integers N and M indicating the matrix dimensions ($1 \leq N, M \leq 600$). Each of the next N lines contains M integers, separated by a space, describing the elements of the matrix. Element $X_{i,j}$ of the matrix is the j-th integer of the i-th line in the input ($-10^6 \leq X_{i,j} \leq 10^6$).

The end of input is indicated by a line containing only two zeros, separated by a space.

Output

For each test case in the input your program must print one single line, containing the size of the largest sub-matrix that, when linearized, results in an increasing sequence.

Sample Input

```
3 3
1 2 5
4 6 7
10 8 3
3 4
1 2 1 2
9 6 7 3
8 7 2 8
4 2
-23 -12
0 2
```

```
16 15
57 33
4 4
2 2 2 2
2 2 2 2
2 2 2 2
2 2 2 2
0 0
```

Sample Output

```
4
3
4
1
```