CCSC:MW Programming Competition

Wine Catalog Selection

Bob was visiting the beautiful Tuscany region of Italy and decided to take a wine testing tour. But before testing the wines in the restaurant, he wanted to make sure about the price and asked for the price card. Unfortunately, the card contained all the fancy names of the wines without the price tag. Bob, being smart, searched with the restaurant name online to get an idea about the price. Interestingly he found **N** different versions (old and new) of the price card. Now he decided to apply the following algorithm to find the most recent price card:

Rule 1: The price card which has the maximum number of wines with the maximum price for that item among the price cards will be chosen.
Rule 2: In case of multiple price cards with that condition (rule 1), the one with the maximum average price will be chosen.

Input format

- The first line contains two integers **x** and **y** that denote the number of price cards and the number of items (wines) on each price card respectively.
- The next **x** line each contains **y** integers represented as $\mathbf{P_{ij}}$, the j^{th} price on the i^{th} price card.

Output format

The number of the most recent price card.

It is guaranteed that the answer is unique.

Constraints

$1 \leq x,y \leq 10^3$
$1 \leq \mathbf{P_{ij}} \leq 10^9$

Example 1

Input:

3 4
1 2 1 10
3 2 3 4
1 3 3 2

Output:

2

Explanation

There are 4 items (types of wines) in this example. The maximum price for the first three types of wine is 3 and for the fourth wine is 10.

First price card has only one maximum price which is the last one (4^{th} wine).

Second price card has 2 maximum prices on it, which are first and third wines.

Third price card has 2 maximum prices on it, which are second and third wines.

Now according to rule 1, Bob can choose both the second and third price cards. Now by applying rule 2, Bob will choose the second price card (average of second price card is (3+2+3+4) / 4 = 3 and average of third price card is (1+3+3+2) / 4 = 2.25).

Example 2

Input:

17 8

768592562 80658517 135834655 63130656 678748721 400870645 365973099 309299056

55785735 695114644 122914940 737713426 812888851 91184743 75531013 455368561

745010016 850447933 976246878 705957296 333429982 656970725 530022083 548038943

649286529 583468339 940914825 978546802 78003653 537591070 776732611 747102879

630578674 419356778 279710456 521015895 107439298 45843066 505344438 499659327

603507660 649552779 550365415 922060433 496317326 665676714 518258593 318421278

561296860 681098977 914143864 571709766 880708450 913356293 230293459 608967024

236437595 881290520 583939123 154226826 614788012 838110455 455382451 71101659

978748061 273784116 65212058 82308568 809289220 569343306 378738567 283832647

302529051 812007914 846340931 321538217 262701167 292522171 873300483 789686742

630207942 219279368 337455188 614838328 122868381 377120475 807525733 816613047

768142670 5193838 716604462 20703082 551704576 3217613 635915377 643315125

49568620 118186606 160415198 905362546 961576576 17383422 345386016 710199082

856643567 712448108 630795339 778205884 394397748 903010796 555132643 178269036

686005982 638243499 83436537 806287543 340792693 839338326 130110689 662591591

624657804 102334818 622514062 734948202 497738238 481320058 341702638 194357085

227208193 612892252 845915410 650436179 288112426 857903034 486610979 815150560

Output:

7