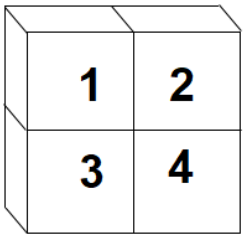
SameGame

2023/2024 SEASON – FINAL ROUND



Deni and Pepi were wandering around the office when they came across an amazing treasure – a demo of the computer game SameGame. The game is played on a rectangular table with *N* rows and *M* columns, which is placed vertically on its bottom side. The table's cells are numbered from left to right and from top to bottom consecutively with the integers from 1 to *N*M (see the figure).*

A ball is placed inside every cell. Every ball has a color which is denoted by the integer a_{ij}. We define *component* as a group of cells where every cell can be visited from every other cell by traversing the adjacent cells (where they share a wall) from the component. When the balls in the cells of the component are colored the same, the player can remove them from the field.

For gravitational concerns the removal of a ball can lead to the fall of the balls sitting above it (if they exist), until there is a ball just below them or they hit the bottom side of the table. When a component with x balls is removed, the player adds x^2 points to their result. When a component with one ball is removed, the player loses their right to remove components with more than one ball.

After a lengthy game with multiple record breaks, the clicking mania of Deni and Pepi has decreased. What turned out to be more interesting for them is how to extend the game with a bigger table and more colors for the balls. Write a program which follows the aforementioned rules and creates a sequence of valid moves aiming at a maximum result.

Input

The first line of the file **samegame.in** consists of two integers N and M – the number of rows and columns of the table. Each of the next N rows contains M numbers separated by a space, describing the colors of the balls in the given row.

Output

On the first line of the file **samegame.out** print K – the count of moves you have done. The next K lines describe the moves. The description of every move starts with the number of cells T_i of the removed component, followed by T_i numbers: $ind_1, ind_2, \ldots, ind_{T_i}$ – the indices of the cells which contained the removed balls.

SameGame

2023/2024 SEASON – FINAL ROUND

Scoring

If 1) the chosen cells do not form a component, 2) the chosen balls are not in the same color, 3) any of the chosen cells does not contain a ball, 4) after the removal of a component with size 1, you remove a component with size bigger than 1, 5) the index of a chosen cell is not inside the constraints of the table, your move will be invalid and you will receive "Error" message and 0 points for the test. Otherwise:

$$yourScore = \sum_{i=1}^{K} T_i^2$$

For each test, let *maxScore* be the biggest score among all participants' scores and *yourScore* be your score. You will be awarded $(\frac{yourScore+1}{maxScore+1})^{1.5}$ multiplied by the amount of points for the test.

Constraints

$15 \leq N \leq 10^5$

$1 \leq M \leq 200$

$1 \leq a_{ij} \leq cnt$, *where cnt is the count of different colors of the balls*

Time limit: 5 sec.

Memory limit: 256 MB.

The tests are distributed as follows:

Percentage	N	M	cnt
20%	$N = 5000$	$M = 1$	$2 \leq cnt \leq 100$
20%	$N = 100000$	$M = 1$	$2 \leq cnt \leq 200$
20%	$N = 15$	$M = 15$	$2 \leq cnt \leq 11$
20%	$N = 50$	$M = 50$	$2 \leq cnt \leq 20$
20%	$N = 200$	$M = 200$	$2 \leq cnt \leq 40$

SameGame

2023/2024 SEASON – FINAL ROUND

Sample test

Input (samegame.in)	Output (samegame.out)
4 5 1 2 2 1 1 1 1 1 2 2 1 2 1 1 1 2 1 2 2 2	6 5 7 8 13 14 15 8 7 12 13 14 15 18 19 20 3 1 6 11 2 19 20 1 16 1 17

Sample test explanation

In the table below you can see the sequence of moves. Notice that during the first move the biggest possible component with color 1 was not picked. The rule for components with size one is followed.

A game like this results in *yourScore* = $5^2 + 8^2 + 3^2 + 2^2 + 1^2 + 1^2 = 104$.

First move

1	2	2	1	1
1	1	1	2	2
1	2	1	1	1
2	1	2	2	2

Second move

1				
1	2		1	1
1	2	2	2	2
2	1	2	2	2

Third move

1				
1				
1				
2	1		1	1

Fourth move

2	1		1	1

Fifth move

2	1			

Sixth move

	1			