Arranging Flowers

Bratislava has a long tradition in organizing an international flower exhibition – Flora. Professor Andrew as a true flower lover visited the Flora exhibition also this year. He was pleased by the most beautiful flowers and other plants of the world – roses, orchids, magnolias, cactuses. All flowers were nicely arranged.

The flower arrangement that was the most appealing to him (at least mathematically) was composed of many kinds of flowers arranged in a rectangular grid, such that each row of the grid contained each kind of flowers exactly once and each column of the grid contained each kind of flowers at most once.

Professor Andrew is a good mathematician and soon he realized that the number of columns of the grid has to be the same as the number of different kinds of flowers in the arrangement. But soon he encountered a problem he was unable to solve: He would like to add more rows of flowers to the arrangement without violating the rules stated above. (Note that he may not modify the existing rows and therefore he may not use any new kinds of flowers in the new rows.) Help him add as many rows as possible!

Input file specification

The input data set describes several flower arrangements; on the first line their number is given. Each flower arrangement description begins with two positive integers $N (1 <= N <= 220)$ and M, representing the number of columns and rows of the grid. The different kinds of flowers are represented by numbers 1,2,...,N. The following M lines contain N integers each representing the kinds of flowers in one row of the arrangement.

Output file specification

For each flower arrangement, output K in the first line, then print K lines containing N numbers each, where K is the maximum number of lines that can be added to the arrangement, describing the added rows. The numbers in the output file may be separated by **exactly one** space. If there are mutiple number of solutions, you must output the **lexicographically first** solution.

Print a blank line after each test case.

There is no special judge program for this problem.

Example

Input:
```
2

3 2
3 2 1
1 3 2

4 2
1 4 3 2
2 1 4 3
```

Output:
1
2 1 3

2
3 2 1 4
4 3 2 1