

Southeastern European Regional Programming Contest
Bucharest, Romania
October 15, 2011

Problem J
Painting

Input File: J.IN
Output File: standard output
Program Source File: J.C, J.CPP, J.JAVA

There is a wall consisting of $n \times n$ tiles. Long ago those tiles were painted in k different colors. Now the paint wears out and the wall has to be repainted. This time all the tiles of the wall should be painted in just one of the k colors. In one move we can paint one horizontal or one vertical row of tiles. Also we can paint a row in a given color only if at least two tiles in this row (horizontal or vertical) are already in this color (either old or new paint). You need to find out the minimal number of moves needed to paint all tiles.

The first line of input contains the number of test cases. The first line of each test case contains two integers: n $(1 < n \leq 500)$ – the number of tiles in one row and k $(1 \leq k < n)$ – the number of different colors available. In each of the following n lines there are n natural numbers giving the numbers of the colors in which the corresponding tile was painted before. The colors have the numbers from 1 to k.

The first line of the output for each test case should contain the number q – the minimal number of moves needed to paint all the tiles. In the second line list the numbers of all the colors in which we can paint the wall according to the given rules with the same number of moves. The numbers should be listed in increasing order. If it's impossible to paint all the tiles following the rules in the statement then print just the number 0.

Sample input	Sample output
2 3 2 1 2 1 2 1 1 1 2 2 2 1 1 1 1 1	4 1 2 1