Problem A. Assignment Problem

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

There are m open positions in our company and $n \geq m$ candidates for these positions. We want to hire the best candidates, obviously. We can't hire the same candidate for two or more different positions, so we have to hire exactly m candidates. Let's call the way to choose different candidates for each position *an assignment*. Two assignments are different if there exists a position for which we hire different candidates in these assignments.

There is a matrix A of profits: $A_{ij} \geq 0$ denotes what profit we will gain from hiring j-th candidate for i-th position. We want to maximize the sum of profits we will gain from all hires. An assignment is optimal if it maximizes the sum of profits.

It would be easy to choose the best candidates given the matrix A. Unfortunately, HR world is not so simple, and they can't provide the matrix A for you. Even after interviewing all the candidates we can only compare how two candidates will behave in the same position. More precisely, we know m permutations P_i of length n. For all $1 \leq i \leq m$, $1 \leq x < y \leq n$: $A_{iP_{ix}} > A_{iP_{iy}}$. In human words, for each position we know the ranking of all candidates.

A candidate is *promising* if and only if there exists a matrix A which is consistent with all the given rankings, such that for this matrix there is only one optimal assignment and this particular candidate is hired.

You are to find all promising candidates so that we can conduct more thorough tests with them.

Input

The first line contains two integers n and m ($1 \leq m \leq 11$, $m \leq n \leq 1000$) — the number of candidates and the number of positions.

Next m lines contain rankings for each position. The i-th line contains a permutation $P_{i1}, P_{i2}, \ldots, P_{in}$ of numbers from 1 to n.

Output

In the first line print the number of promising candidates, in the second line print indices of promising candidates **in increasing order**.

Examples

standard input	standard output
4 2 1 2 4 3 1 3 4 2	3 1 2 3
4 2 1 4 3 2 2 3 4 1	2 1 2