Chamber of Secrets 2

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

You are playing the game Henry Spotter and the Chamber of Secrets 2.

You want to unlock the next level, the Chamber of Secrets. The entry door contains n panels, each displaying a sequence of m symbols. The product nm is even. The system generates these sequences from a **secret permutation** using the following four-step process:

- first, it starts from a secret permutation $[p_1, p_2, \ldots, p_{nm/2}]$;
- then, it repeats the secret permutation by concatenating it with itself, forming the array $[b_1, b_2, \ldots, b_{nm}]$;
- then, it splits this array into n consecutive blocks, i.e., disjoint subarrays of length m;
- then, it shuffles these blocks in arbitrary order across the panels.

You are given the final n panel sequences produced by the system. The i-th panel shows $[a_{i,1}, a_{i,2}, \ldots, a_{i,m}]$. Your task is to recover one possible original secret permutation $[p_1, p_2, \ldots, p_{nm/2}]$. For the given input, at least one solution exists. If multiple secret permutations are valid, output any one of them.

The concatenation of two arrays $[x_1, x_2, \ldots, x_{k_1}]$, $[y_1, y_2, \ldots, y_{k_2}]$ is the array $[x_1, x_2, \ldots, x_{k_1}, y_1, y_2, \ldots, y_{k_2}]$ of length $k_1 + k_2$.

A permutation of length l is an array consisting of l distinct integers from 1 to l in arbitrary order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (2 appears twice in the array), and $[1, 3, 4]$ is also not a permutation ($l = 3$ but there is 4 in the array).

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The first line of each test case contains two integers n, m ($1 \le n \le 70$, $1 \le m \le 70$) — the number of panels, and the length of each displayed sequence.

The i-th of the next n lines contains m integers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$ ($1 \le a_{i,j} \le nm/2$), representing the sequence shown on the i-th panel.

Note that there are no constraints on the sum of n and m over all test cases.

Output

For each test case, output a single line containing a secret permutation $[p_1, p_2, \ldots, p_{nm/2}]$ such that the process described above can produce the n panel sequences $[a_{i,1}, a_{i,2}, \ldots, a_{i,m}]$. For the given input, at least one solution exists.

Example

standard input	standard output
5 6 2 1 2 3 4 5 6 5 6 3 4 1 2 5 2 1 3 4 1 2 4 5 2 3 5 5 4 4 1 3 2 6 9 5 10 5 10 4 1 3 2 7 8 7 8 6 9 4 3 3 5 2 1 4 6 1 4 6 3 5 2 1 8 3 1 2 4 3 1 2 4	5 6 3 4 1 2 2 4 1 3 5 3 2 7 8 6 9 5 10 4 1 3 5 2 1 4 6 3 1 2 4

Note

Explanation of sample 1.

In the first test case, one valid secret permutation is $[p_1, p_2, \ldots, p_{nm/2}] = [5, 6, 3, 4, 1, 2]$:

- the array $[b_1, b_2, \ldots, b_{nm}]$ is the concatenation of two copies of $[p_1, p_2, \ldots, p_{nm/2}] = [5, 6, 3, 4, 1, 2]$;
- $[b_1, b_2, \ldots, b_{nm}]$ splits into the blocks $[5, 6]$, $[3, 4]$, $[1, 2]$, $[5, 6]$, $[3, 4]$, $[1, 2]$;
- after shuffling, the final panel sequences $[a_{i,1}, a_{i,2}, \ldots, a_{i,m}]$ can coincide with these blocks.