

Las Vegas

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

Las Vegas is an easy-to-learn dice-rolling board game. Players roll dice and commit them to one of the casinos, and the player with the most dice on a casino wins its money, but only if they're not tied! The game perfectly combines luck and strategy, and was nominated for the *Spiel des Jahres* prize in 2012.



Photo by @denislageron on BoardGameGeek

Consider a game with n casinos and $(m + 1)$ players. You're the $(m + 1)$ -th player, and all other players have already placed (or decided not to place) their dice on the casino. It is known that on the i -th casino, the j -th player has placed $a_{i,j}$ dice on it.

Your task is to decide on the number of dice b_i you want to place on the i -th casino for each $1 \leq i \leq n$. After you place your dice, we'll determine the winner of each casino separately according to the following rule:

1. First, if two or more players have placed the same number of dice on this casino, all their dice on this casino will be removed.
2. After that, the player with the most number of dice on this casino is determined to be the winner. If there are no dice on this casino, there is no winner.

For example:

- Consider a casino on which 7 players have placed 1, 2, 3, 3, 4, 4, 4 dice respectively. First, as there are two or more players with 3 or 4 dice, all their dice are removed, and the remaining number of dice are 1, 2, 0, 0, 0, 0, 0 respectively. The winner of this casino is the player with 2 dice.
- Consider a casino on which 4 players have placed 2, 3, 2, 3 dice respectively. First, as there are two or more players with 2 or 3 dice, all their dice are removed, and the remaining number of dice are 0, 0, 0, 0 respectively. As there are no dice on this casino, there is no winner.

Your goal is to win in the most number of casinos among all players. More precisely, let w_i be the number of casinos where player i is the winner, there must be $w_{m+1} \geq w_i$ for all $1 \leq i \leq m$. Minimize the total number of dice you place in all casinos while achieving your goal. That is, minimize $\sum_{i=1}^n b_i$.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 100$) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n, m \leq 50$), indicating the number of casinos and the number of players (except you).

For the following n lines, the i -th line contains m integers $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($0 \leq a_{i,j} \leq 10^8$), where $a_{i,j}$ is the number of dice on casino i placed by player j .

It's guaranteed that there are at most 3 test cases satisfying $n > 20$ or $m > 20$.

Output

For each test case, output one line containing n integers b_1, b_2, \dots, b_n ($0 \leq b_i \leq 10^9$) separated by a space, where b_i is the number of dice you place on casino i . It can be proven that an optimal answer always exists under such constraints of b_i . If there are multiple optimal answers, you can output any of them.

Example

standard input	standard output
3	4 0 6 0
4 3	1 0 2
3 3 2	0
2 7 5	
3 5 2	
1 6 4	
3 4	
100 100 100 1	
0 0 0 1	
100 100 100 1	
1 4	
20 0 20 0	

Note

For the first sample test case, you are the winner of casino 1 and 3, while player 2 is the winner of casino 2 and 4.

For the second sample test case, you are the winner of casino 3, while player 4 is the winner of casino 2. There is no winner for casino 1.