Problem J
Miniature Golf
Time limit: 6 seconds

A group of friends has just played a round of miniature golf. Miniature golf courses consist of a number of holes. Each player takes a turn to play each hole by hitting a ball repeatedly until it drops into the hole. A player's score on that hole is the number of times they hit the ball. To prevent incompetent players slowing down the game too much, there is also an upper limit ℓ (a positive integer) on the score: if a player has hit the ball ℓ times without the ball dropping into the hole, the score for that hole is recorded as ℓ and that player's turn is over. The total score of each player is simply the sum of their scores on all the holes. Naturally, a lower score is considered better.

There is only one problem: none of the players can remember the value of the integer ℓ. They decide that they will not apply any upper limit while playing, allowing each player to keep playing until the ball drops into the hole. After the game they intend to look up the value of ℓ and adjust the scores, replacing any score on a hole that is larger than ℓ with ℓ.

The game has just finished, but the players have not yet looked up ℓ. They wonder what their best possible ranks are. For this problem, the *rank* of a player is the number of players who achieved an equal or lower total score after the scores are adjusted with ℓ. For example, if the adjusted scores of the players are 3, 5, 5, 4, and 3, then their ranks are 2, 5, 5, 3 and 2 respectively.

Given the scores of the players on each hole, determine the smallest possible rank for each player.

Input

The first line of input contains two integers p and h, where p ($2 \leq p \leq 500$) is the number of players and h ($1 \leq h \leq 50$) is the number of holes. The next p lines each contain h positive integers. The j^{th} number on the i^{th} of these lines is the score for player i on hole j, and does not exceed 10^9.

Output

Output a line with the minimum possible rank for each player, in the same order as players are listed in the input.

Sample Input 1	Sample Output 1
3 3 2 2 2 4 2 1 4 4 1	1 2 2

Sample Input 2	Sample Output 2
6 4 3 1 2 2 4 3 2 2 6 6 3 2 7 3 4 3 3 4 2 4 2 3 3 5	1 2 5 5 4 3