Bytetown has a long tradition in organizing an international flower exhibition. Professor Feuerbach, a true flower lover, visited the exhibition this year. He was pleased by the most beautiful flowers and other plants around the world roses, orchids, magnolias, cactuses. All flowers were nicely placed.

The flower placement that was the most appealing to him was composed of many kinds of flowers placed in a rectangular grid, such that each row of the grid contained each kind of flowers exactly once and each column of the grid contained each kind of flowers at most once.

Professor Feuerbach is a good mathematician and soon he realized that the number of columns of the grid has to be the same as the number of different kinds of flowers in the placement. The different kinds of flowers are represented by numbers $1, 2 \ldots, N$, where N is the number of different kinds of flowers. Soon he encountered a new problem. He would like to add one row of flowers to the placement without violating the rules stated above. (Note that he may not modify the existing rows and therefore he may not use any new kinds of flowers in the new rows.) It's quite easy, but he wants to know the **K-th lexicographically valid placement.**

Input

The input data set describes about 100 flower placements. In the first line the number of placements is given.

Each flower placement description begins with three positive integers N ($1 \leq N \leq 200$), M ($0 \leq M \leq N$), K ($1 \leq K \leq 200$) representing the number of columns and rows of the grid. The following M lines contain N integers each representing the kinds of flowers in one row of the placement.

Output

For each flower placement output one line containing N numbers, describing the added row. The numbers in the output file should be separated by exactly one space. If there are less than K valid placements, output '-1' instead.

Sample Input

```
2
3 2 2
3 2 1
1 3 2
4 2 2
1 4 3 2
2 1 4 3
```

Sample Output

```
Case #1: -1
Case #2: 4 3 2 1
```