Question 9: T, Tu, Tur, Turk, Turke, Turkey

Description

Turkeys, as it turns out, have an unusual system for getting food at a turkey sanctuary. A set of turkey feeding stations are set up, and each day, turkeys travel to them to feed a little bit before moving on to another one (possibly one they have visited before, or even they just walk away from the one they are at for a little bit before coming back).

The turkey guardians at the sanctuary are interested in which sequences of feeding stations are the most popular for the turkeys. In particular, they wish to know, for each k less than the length of the longest sequence, which length k sequence, starting from the beginning of the day, is the most popular of all of the possible sequences. If there is a tie, the lexicographic minimum sequence among the ties should be output.

For example, if there are 3 different stations and 5 different turkeys who visited stations:

- 1 1 1 2
- 1 1 2 2 2
- 1 1 3 1 1 3
- 3 3 3 2
- 3 3 3 1 2

One would output:

- 1
- 1 1
- 3 3 3
- 1 1 1 2
- 1 1 2 2 2
- 1 1 3 1 1 3

since 1 is the most common prefix (sequence starting from the beginning of the day) of length 1, 1 1 is the most common prefix of length 2, 3 3 3 is the most common length 3 prefix, 1 1 1 2 is the lexicographic minimal sequence between all of the prefixes of length 4 (since each of these appear only once, we break the tie), etc.

Input Description

The input will be given as follows:

The first line will contain two integers, separated by a space, $1 <= N <= 50,000$ and $1 <= M <= 30$, the number of turkeys and the number of stations (labeled 1 to M), respectively. The next N lines will contain an integer $1 <= K <= 250$ followed by K integers (each between 1 and M) separated by spaces. These correspond to the sequence of feeding stations that a particular turkey takes.

Output Description

The output should have `max(K)` lines. Each line `i` should contain the sequence of length `i` that appears most frequently (taking the lexicographic minimum if there are ties) among all of length `i` sequences starting from the beginning of the day. The elements in the sequence should be separated by a space with no trailing spaces.

Input Example

```
5 3
4 1 1 1 2
5 1 1 2 2 2
6 1 1 3 1 1 3
4 3 3 3 2
5 3 3 3 1 2
```

Output example

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