A - Grandpa is Famous

Latin America - South America - 2004/2005

The whole family was excited by the news. Everyone knew grandpa had been an extremely good bridge player for decades, but when it was announced he would be in the Guinness Book of World Records as the most successful bridge player ever, whow, that was astonishing!

The International Bridge Association (IBA) has maintained, for several years, a weekly rank-ing of the best players in the world. Considering that each appearance in a weekly ranking constitutes a point for the player, grandpa was nominated the best player ever because he got the highest number of points.

Having many friends who were also competing against him, grandpa is extremely curious to know which player(s) took the second place. Since the IBA rankings are now available in the internet he turned to you for help. He needs a program which, when given a list of weekly rankings, finds out which player(s) got the second place according to the number of points.

Input

The input contains several test cases. Players are identified by integers from 1 to 10000. The first line of a test case contains two integers N and M indicating respectively the number of rankings available ($2 \leq N \leq 500$) and the number of players in each ranking ($2 \leq M \leq 500$). Each of the next N lines contains the description of one weekly ranking. Each description is composed by a sequence of M integers, separated by a blank space, identifying the players who figured in that weekly ranking. You can assume that:

- in each test case there is exactly one best player and at least one second best player,
- each weekly ranking consists of M distinct player identifiers.

The end of input is indicated by N = M = 0.

Output

For each test case in the input your program must produce one line of output, containing the identification number of the player who is second best in number of appearances in the rankings. If there is a tie for second best, print the identification numbers of all second best players in increasing order. Each identification number produced must be followed by a blank space.

Sample Input

```
4 5
20 33 25 32 99
32 86 99 25 10
20 99 10 33 86
19 33 74 99 32
3 6
2 34 67 36 79 93
100 38 21 76 91 85
32 23 85 31 88 1
0 0
```

Sample Output

```
32 33
1 2 21 23 31 32 34 36 38 67 76 79 88 91 93 100
```