IOI '01 P6 - Depot

Time limit: 1.0s   **Memory limit:** 32M

IOI '01 - Tampere, Finland

A Finnish high technology company has a big rectangular depot. The depot has a worker and a manager. The sides of the depot, in the order around it, are called left, top, right and bottom. The depot area is divided into equal-sized squares by dividing the area into rows and columns. The rows are numbered starting from the top with integers $1, 2, \ldots$ and the columns are numbered starting from the left with integers $1, 2, \ldots$.

The depot has containers, which are used to store invaluable technological devices. The containers have distinct identification numbers. Each container occupies one square. The depot is so big, that the number of containers ever to arrive is smaller than the number of rows and smaller than the number of columns. The containers are not removed from the depot, but sometimes a new container arrives. The entry to the depot is at the top left corner.

The worker has arranged the containers around the top left corner of the depot in such a way that he will be able to find them by their identification numbers. He uses the following method.

Suppose that the identification number of the next container to be inserted is k (container k, for short). The worker travels the first row starting from the left and looks for the first container with identification number larger than k. If no such container is found, then container k is placed immediately after the rightmost of the containers previously in the row. If such a container l is found, then container l is replaced by container k, and l is inserted to the following row using the same method. If the worker reaches a row having no containers, the container is placed in the leftmost square of that row. Suppose that containers $3, 4, 9, 2, 5, 1$ have arrived to the depot in this order. Then the placement of the containers at the depot is as follows.

```
1 4 5
2 9
3
```

The manager comes to the worker and they have the following dialogue:
Manager: Did container 5 arrive before container 4?
Worker: No, that is impossible.
Manager: Oh, so you can tell the arrival order of the containers by their placement.
Worker: Generally not. For instance, the containers now in the depot could have arrived in the order $3, 2, 1, 4, 9, 5$ or in the order $3, 2, 1, 9, 4, 5$ or in one of 14 other orders.

As the manager does not want to show that the worker seems much smarter, he goes away. You are to help the manager and write a program which, given a container placement, computes all possible orders in which they might have arrived.

Input Specification

The first line contains one integer R: the number of rows with containers in them. The following R lines contain information about rows $1, \ldots, R$ starting from the top as follows. First on each of those lines is an integer M: the number of containers in that row. Following that, there are M integers on the line: the identification numbers of the containers in

the row starting from the left. All container identification numbers I satisfy $1 \leq I \leq 50$. The total number of containers in the depot is at least 1 and no more than 13.

Output Specification

The output should contain as many lines as there are possible arrival orders. Each of these lines contains N integers: the identification numbers of the containers in the potential arrival order described by that line. All lines describe an arrival order not described in any other line.

Sample Input 1

```
3
3 1 4 5
2 2 9
1 3
```

Sample Output 1

```
3 2 1 4 9 5
3 2 1 9 4 5
3 4 2 1 9 5
3 2 4 1 9 5
3 2 9 1 4 5
3 9 2 1 4 5
3 4 2 9 1 5
3 4 9 2 1 5
3 2 4 9 1 5
3 2 9 4 1 5
3 9 2 4 1 5
3 4 2 9 5 1
3 4 9 2 5 1
3 2 4 9 5 1
3 2 9 4 5 1
3 9 2 4 5 1
```

Sample Input 2

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

Sample Output 2

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

Scoring

If the output file contains impossible orders or no orders at all, your score is 0 for that test case. Otherwise the score for a test case is computed as follows. If the output file contains all possible orders exactly once, your score is 4. If the output file contains at least half of the possible orders and each of them exactly once, your score is 2. If the output file contains less than half of the possible orders or some of them appear more than once, your score is 1.