

Problem C: Problems

Advanced Algorithms for Programming Contests

Restrictions

Time: 2 seconds

Memory: 512 MB

Problem description

Arguably the hardest task in the organization of a programming contest is the creation of a good problem set. To make for an interesting competition, it is necessary to have problems of varying complexities. And also the topics should differ, shouldn't they? Fortunately, contest organizers usually have a large pool of problems to choose from, but they will still need your help.

There is a pool of n problems, each of them associated with a complexity level and a list of the topics it involves. You are to choose a valid problem set, i.e. a subset of those problems that fulfills the following requirements:

- Exactly one problem of each complexity level is chosen.
- Each topic is covered by exactly one of the chosen problems.

Input

The input consists of

- one line containing n ($1 \leq n \leq 300$) – the number of problems, d ($1 \leq d \leq 50$) – the amount of complexity levels and m ($1 \leq m \leq 18$) – the amount of different topics
- n lines containing problem descriptions in the format " $c_i k_i t_{i,1} t_{i,2} \dots t_{i,k_i}$ ", where c_i is the complexity level of the i -th problem ($1 \leq c_i \leq d$), k_i is the number of topics the problem covers ($0 \leq k_i \leq m$) and $t_{i,1}, \dots, t_{i,k_i}$ is the list of said topics ($1 \leq t_{i,j} \leq m$). It is guaranteed that $t_{i,j} \neq t_{i,j'}$ for $j \neq j'$.

Output

Output "OK" if it's possible to create a valid problem set from the given pool and "Impossible" otherwise.

Sample input and output

Input	Output
4 2 3 1 1 1 1 2 1 2 2 2 1 2 2 2 2 3	OK
1 2 1 1 1 1	Impossible