

NOI '98 P3 - Software Installation Disk

Time limit: 0.6s **Memory limit:** 16M

National Olympiad in Informatics, China, 1998

Software installation is often a headache-inducing task. Software usually consists of many parts (known as "components"). During the time of installation, the user is prompted to decide which components they would like to install. In addition, some components may be dependent on other components and thus they must sometimes be installed in a certain order.

Due to the modern prevalence of floppy drives coming with personal computers, the most widespread form of software installation media is the floppy disk. However, the limited capacity of floppy disks makes it so that slightly larger software may not fit on a single disk. At this point, software will need to be stored on many floppy disks. Every floppy disk contains one or more components. These disks are called software installation disks.

Since the components of a software are dispersed on different disks, and a particular installation order of the components is required, it is easy for the scenario of switching back and forth between disks to occur. When computer users are installing software, the most contemptible task has to be switching back and forth between installation disks (which involves finding the disk, inserting the disk, ejecting the disk, finding the disk, inserting the disk, ...). Everything about this seems tedious and disorderly, so it is important to design software installation disks to meet the following criteria:

- The user should never have to insert the same disk twice. Even better, the disks should be labeled starting from 1 so that at the time of installation, the user can follow the order to insert the disks.
- For economic reasons, the number of total installation disks should be minimized.

Write a program that, given information about the software components, finds the optimal software installation disk arrangement.

Input Specification

The first line of input contains the integer M ($1 \leq M \leq 10^9$), the capacity of each floppy disk in bytes.

The second line of input contains the integer N ($1 \leq N \leq 100$), the number of components.

The next N lines each provide information about a single component, including:

1. the number of bytes the component takes up, and
2. the ID(s) of the component(s), between 1 to N , that must be installed prior to installing the current component. If there are multiple prerequisite components, all of their IDs will be listed. If there are no dependencies, only the number of bytes will be listed on the line.

Adjacent values from any line in the input may be separated by one or more spaces.

Output Specification

The first line of output should contain the minimum number of floppy disks required to store all of the components under the requirements described above. If one does not exist, output $\emptyset$. Otherwise, the remaining lines should each

describe a floppy disk and the IDs of the component(s) that should be stored on it. The discs should be in the order that they are inserted by the user. For each floppy disk, the components on it may be listed in any order, but should be separated by spaces.

Sample Input

```
1457664
3
512665
912345 1
832542 1
```

Sample Output

```
2
1 3
2
```

Problem translated to English by [Alex](#).