Library for Madrid

Library for Madrid

Good preparation is essential for winning programming contests. Thus, if you read this document, you're on the right track ;) Knowing your algorithms and the programming language you use is of prime importance. However, you don't have to learn everything by heart; each team is allowed to bring 25 pages of documentation to the contest.

Now the difficult question is: What should you put on those 25 pages? You know that you can fit 10 paragraphs of text on a page, but your stock of useful code snippets and handy texts is much larger than that. To make things more complicated, some topics depend on each other. You cannot include the line-circle intersection formula if you do not also include the code for lines, circles and points.

As a programmer, you decide to let your computer do the hard work for you. Given a set of topics, their space requirements and dependencies, write a program that prints the maximum number of topics that fit into the library.

Input

The input consists of several testcases. Each problem description starts with the numbers $1 \le M \le 100$ and $0 \le D \le 10$, the number of topics and the number of dependencies. The following M lines contain the name of a topic (one word) and the number L_t of paragraphs ($1 \le L_t \le 1000$) of the topic, separated by a space. The next D lines each contain two topic names separated by space, indicating that the first topic depends on the second.

The input file ends with a testcase having M=0, which should not be processed.

Output

For each test case, print a single line containing the maximum number of topics that you can fit in the library, followed by the number of free paragraphs that remain. If several solutions yield the same number of topics, choose the one that leaves as much empty space as possible.

Example

Input:

```
5 4
Dijkstra 50
Intersections 30
Lines 70
Circles 120
Points 40
Intersections Lines
Intersections Circles
Lines Points
Circles Points

0 0
```

Output:

```
3 90
```

Notes

The sample output corresponds to choosing Dijkstra's algorithm, lines and points.