

G. Suggested Friends

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Polycarpus works as a programmer in a start-up social network. His boss gave him a task to develop a mechanism for determining suggested friends. Polycarpus thought much about the task and came to the following conclusion.

Let's say that all friendship relationships in a social network are given as m username pairs a_i, b_i ($a_i \neq b_i$). Each pair a_i, b_i means that users a_i and b_i are friends. Friendship is symmetric, that is, if a_i is friends with b_i , then b_i is also friends with a_i . User y is a *suggested friend* for user x , if the following conditions are met:

1. $x \neq y$;
2. x and y aren't friends;
3. among all network users who meet the first two conditions, user y has most of all common friends with user x . User z is a common friend of user x and user y ($z \neq x, z \neq y$), if x and z are friends, and y and z are also friends.

Your task is to help Polycarpus to implement a mechanism for determining suggested friends.

Input

The first line contains a single integer m ($1 \leq m \leq 5000$) — the number of pairs of friends in the social network. Next m lines contain pairs of names of the users who are friends with each other. The i -th line contains two space-separated names a_i and b_i ($a_i \neq b_i$). The users' names are non-empty and consist of at most 20 uppercase and lowercase English letters.

It is guaranteed that each pair of friends occurs only once in the input. For example, the input can't contain x, y and y, x at the same time. It is guaranteed that distinct users have distinct names. It is guaranteed that each social network user has at least one friend. The last thing guarantees that each username occurs at least once in the input.

Output

In the first line print a single integer n — the number of network users. In next n lines print the number of suggested friends for each user. In the i -th line print the name of the user c_i and the number of his suggested friends d_i after a space.

You can print information about the users in any order.

Examples

input
5 Mike Gerald Kate Mike Kate Tank Gerald Tank Gerald David
output
5 Mike 1 Gerald 1 Kate 1 Tank 1 David 2

input
4 valera vanya

valera edik pasha valera igor valera
output
5 valera 0 vanya 3 edik 3 pasha 3 igor 3

Note

In the first test case consider user David. Users Mike and Tank have one common friend (Gerald) with David. User Kate has no common friends with David. That's why David's suggested friends are users Mike and Tank.