

Cloud Computing

By Lucas Hermann Negri, UDESC  Brazil

Timelimit: 1

The Cloud Computing is coming to Joinville's region. They're known for providing apps on the internet, more specifically a business model based on cloud computing.

In order to select the new employers of the company, they contacted UDESC marathon committee, so they could pass a problem to our marathoners. Those who solve it, beside de balloon, can fill a personnel file with more stars.

Basically, Gloud Computing has apps spread on their servers on several of places of the world. Those servers are specialized on a list of apps to be used by the users connected on the clouds.

For example, the server in Joinville can provide the application **A**, while the one in Pasadena, California provides the applications **A**, **B**, **C** and the server in Pomerode provides the application **C**.

We have a set of servers and each one comes with a set of applications to be provided to a set of users. Each user can be connected to one or more servers depending on its demand, as illustrated on the Figure 1.

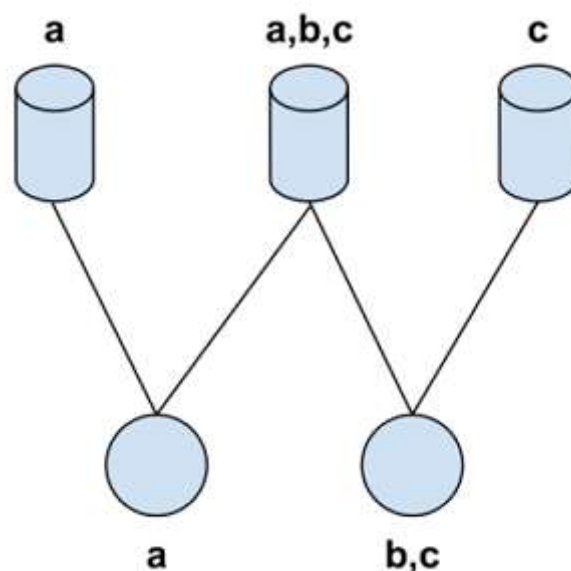


Figure 1: 3 servers, 2 users and 4 connections.

It will be given to you data about these two sets, servers and users' demand, and you should return the total amount of connections between clients and servers. The connections are made so that the redundancy is maximized. For instance, if a client wants to utilize the applications **B** and **C**, he will connect to all servers that provide at least the application **B** and all that provides at least **C**. Multiple connections between a pair of clients and servers count as only one. It can happen that a client requires an inexistent application, just like a server provide a non required application.

Input

The input is composed by several test cases. Each test case starts by two Integers **N** and **M** ($0 \leq N, M \leq 200$), that correspond to the number of servers and clients. Each one of next **N** lines, contains a value **Qi** ($0 \leq Qi \leq 100$) correspondent to the number of applications provided by the i-th server, followed by a word **Qi** (separated by blank spaces) related to the name of the provided app. After the server's description, follow **M** lines, each one containing a value **Pj** ($0 \leq Pj \leq 100$) corresponding to the number of requested applications by

the j -th client, followed by P_j word (separated by blank spaces) related to the name of the required applications. The input ends with $N = M = 0$. All application's name have between 1 and 20 characters.

Output

For each test case your program must print the total sum of connections between clients and servers in a line, disregarding multiple connections between the same pair of client and server.

Sample Input	Sample Output
3 2 1 a 3 a b c 1 c 1 a 2 b c 5 2 2 s1 s2 2 s3 s4 2 s5 s6 2 s7 s8 2 s1 s2 3 s1 s2 s20 3 s1 s2 s21 0 0	4 4