



## Problem D Problemsetting

Input File: D.IN

Output File: standard output

Program Source File: D.C, D.CPP, D.JAVA

It's well-known that different programming contests require different kind of problems. For example, maximal array size for TopCoder problem is only 50, and you definitely can not give a Suffix Tree problem to IOI because children will not have a chance to solve it (except of some touristic-inclined ones). Thus not every problem is acceptable for every contest.

You are preparing problemsets for  $N$  different contests. These contests require different number of problems, depending of type. For example, ACM ICPC style problemset usually has 10 problems, TopCoder SRM – 5 and so on.

Luckily you have already prepared  $M$  different problems. For each problem you have determined a set of contests you can give that problem to. Also you know the required number of problems for each contest.

Find out the maximal number of different contests for which you can simultaneously compose *complete* problemsets from the given set of problems. All problems in the problemsets must be unique, i.e. no problem can be used twice in different problemsets.

The input file contains several test cases.

The first line of each test case contains 2 integers  $N$  and  $M$  ( $1 \leq N \leq 15$ ,  $0 \leq M \leq 50$ ) – the number of different contests and the number of prepared problems. Each of the following  $N$  lines contains the name of contest, followed by the required number of problems for that contest. The name of a contest consists of lower- and uppercase Latin letters and/or digits, is not empty and does not exceed 100 characters. Contest names are case-sensitive. It's guaranteed that all contest names will be pairwise different. The required number of problems does not exceed 100.

Each of the following  $M$  lines contains a (possibly empty) list of acceptable contest names for each problem, separated by a single space. It's guaranteed that all contest names will be correct (i.e., noted in the previous section of the current test case) and unique.

The line containing two zeroes indicates the end of the input file.

For each test case print an answer for that case on a new line, as shown in the sample output.

Sample input	Sample output
<pre>4 5 IOI 3 IPSC 2 TopCoder 2 SEERC 10 IOI IPSC TopCoder IOI IPSC IOI IPSC TopCoder SEERC 1 1 SampleContest 1 SampleContest 0 0</pre>	<pre>Case #1: 2 Case #2: 1</pre>