3100 String Scanner

Algorithms for searching a string or a substring have many applications, such as searching files for virus signatures or spyware.

Scanning for substrings is straightforward when the string is available in its original form. However, it is often necessary to search for substrings in compressed files.

For this problem, we will consider run-length-encoded segments. A run-length-encoded segment is a string where repeated substrings are encoded in pairs called "count substrings." For example, the count substring "`3 ACM`" (where 3 refers to the number of repetitions and ACM is referred to as the encoding substring) is equivalent to the segment "`ACMACMACM`".

You are asked to write a program that will search a main string composed of run-length-encoded segments for multiple substrings and print a frequency table of match results.

Search results may overlap. For example, a search for a substring "`AAAA`" in the main string "`AAAAAA`" will result in three matches, matching the first, second and third position.

Input

The input file consists of several test cases.

For each test case, input starts with n ($1 \leq n \leq 1000$) where n is the number of patterns to search for (or search patterns) from the main string. This is followed by the n patterns, one per line. Each pattern will be 1 to 7 characters long. The next line contains m ($1 \leq m \leq 1000$), the number of run-length-encoded segments composing the main string. This is followed by m "count substrings" providing the run-length encoding of the m segments. Each encoding substring will be from 1 to 80 characters and can be repeated at most 100 times in one run-length encoded segment. Therefore, the largest possible main string is 8000000 (8×10^6) characters long — 80-character substring $\times$ 100 repetitions $\times$ 1000 segments.

Both search patterns and encoding substrings are composed solely of the uppercase letters, '`A`'..'`Z`'.

The end of the input file is indicated by $n = 0$.

Output

For each test case, print an alphabetically sorted list of search patterns and their frequency in the run-length encoded string. The output should be in the form "pattern frequency". Patterns that did not match shall still be printed out with frequency 0.

Separate output for test cases with a single blank line.

Sample Input

```
5
AB
ABCD
EFGH
CDEF
EFAB
4
5 AB
1000000 CD
```

```
5 EF
1 GH
2
ACM
ICPC
3
1 HELLOAC
1 MICP
2 C
0
```

Sample Output

```
AB 5
ABCD 1
CDEF 1
EFAB 0
EFGH 1

ACM 1
ICPC 1
```