ACM-ICPC Live Archive

4163 Search of Concatenated Strings

The amount of information on the World Wide Web is growing quite rapidly. In this information explosion age, we must survive by accessing only the Web pages containing information relevant to our own needs. One of the key technologies for this purpose is keyword search. By using well-known search engines, we can easily access those pages containing useful information about the topic we want to know.

There are many variations in keyword search problems. If a single string is searched in a given text, the problem is quite easy. If the pattern to be searched consists of multiple strings, or is given by some powerful notation such as regular expressions, the task requires elaborate algorithms to accomplish efficiently.

In our problem, a number of strings (*element strings*) are given, but they are not directly searched for. Concatenations of all the element strings in any order are the targets of the search here.

For example, consider three element strings "`aa`", "`b`" and "`ccc`" are given. In this case, the following six concatenated strings are the targets of the search, i.e. they should be searched in the text.

```
aabccc
aacccb
baaccc
bcccaa
cccaab
cccbaa
```

The text may contain several occurrences of these strings. You are requested to count the number of occurrences of these strings, or speaking more precisely, the number of positions of occurrences in the text.

Two or more concatenated strings may be identical. In such cases, it is necessary to consider subtle aspects of the above problem statement. For example, if two element strings are x and xx, the string xxx is an occurrence of both the concatenation of x and xx and that of xx and x. Since the number of positions of occurrences should be counted, this case is counted as one, not two.

Two occurrences may overlap. For example, the string $xxxx$ has occurrences of the concatenation xxx in two different positions. This case is counted as two.

Input

The input consists of a number of datasets, each giving a set of element strings and a text. The format of a dataset is as follows.

$n \quad m$
e_1
e_2
$\vdots$
e_n
t_1
t_2
$\vdots$
t_m

The first line contains two integers separated by a space. n is the number of element strings. m is the number of lines used to represent the text. n is between 1 and 12, inclusive.

Each of the following n lines gives an element string. The length (number of characters) of an element string is between 1 and 20, inclusive.

The last m lines as a whole give the text. Since it is not desirable to have a very long line, the text is separated into m lines by newlines, but these newlines should be ignored. They are not parts of the text. The length of each of these lines (not including the newline) is between 1 and 100, inclusive. The length of the text is between 1 and 5000, inclusive.

The element strings and the text do not contain characters other than lowercase letters.

The end of the input is indicated by a line containing two zeros separated by a space.

Output

For each dataset in the input, one line containing the number of matched positions should be output. An output line should not contain extra characters.

CAUTION! Although the sample input contains only small datasets, note that $12! \times 5000$ is far larger than 2^{31}.

Sample Input

```
3 1
aa
b
ccc
aabccczbaacccbaazaabbcccaa
3 1
a
b
c
cbbcbcbabaacabccaccbaacbccbcaaaccccbcbcbbcacbaacccaccbbcaacbbabbabaccc
3 4
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
0 0
```

Sample Output

```
5
12
197
```