10860   Many a Little makes a Mickle

A long string does not look so long if we can identify a few short substrings that were used (possibly more than once) in some permutation to construct the longer string. Your task is to find if a given (long) string can be made up by choosing some (shorter) strings from a given collection.

You should note that:

a. All the strings are composed of ASCII characters in the range 33 to 127.

b. Any of the short strings or their reversed forms can be used any number of times to construct the long string

c. Each use of a short string or its reverse would be counted as one occurance of that short string

When you construct the longer string from these short strings you should ensure that it is done by keeping the total occurances of the short strings minimum.

For example, if we want to construct the string "aabbabbabbbb" from the set {"a","bb","abb"}, there can be many ways to achieve the goal. "a-abb-abb-abb-bb" and "a-abb-a-bba-bb-bb" are two such valid constructions. However, we would prefer "a-abb-abb-abb-bb" (5 substrings) over "a-abb-a-bba-bb-bb" (6 substrings) because it uses lesser number of substrings. You would only need to find the minimum number of substrings that could be used to construct the given string.

Input

The first line of the the input contains S ($S < 51$), the number of data set. Then S number of data set follows. First line of each data set contains the long string, P ($0 < length(P) < 10001$). The next line contains the number of short strings, N ($0 < N < 51$) to choose from. Each of the next N lines contain the short string P_i ($0 < length(P_i) < 101$) [$i \geq 1, 2, 3 \ldots N$]. You can safely assume that there is no blank/empty line in the input file.

Output

For each data set print exactly one line of output.
    Either 'Set S: C'.
    Or 'Set S: Not possible.'
    If it is possible to construct the string using the given strings then print the first line otherwise print the second line. Here S is the serial of data set (sequentially from 1 to S) and C is the minimum number of times the substrings were used to construct P. For clarification see sample output below.

Sample Input

```
2
aabbabbabbbb
3
a
bb
abb
ewu**bbacsecsc
4
ewu
```

```
bba
cse
csc
```

Sample Output

```
Set 1: 5.
Set 2: Not possible.
```