

Problem M

C'mon K-mers!

Problem ID: kmers

A useful operation on strings, specially when analyzing certain types of data, like genomic data, is to find all the substrings of length k in a string. These substrings are called the string's k -mers and we are typically interested in finding out how many times each k -mer appears within the string. For example, consider the following string:

aaabaab

This string has three unique 2-mers:

aaabaab

aa
 aa
 ab
 ba
 aa
 ab

More specifically, the 2-mers are aa (appearing three times), ab (appearing two times), and ba (appearing once). Notice how counting the k -mers considers *overlapping* occurrences of each substring. For example, the first two occurrences of aa above overlap with each other.

We could do a similar analysis for the string's 3-mers:

aaabaab

aaa
 aab
 aba
 baa
 aab

In this case, there are four unique 3-mers: aaa (appearing once), aab (appearing twice), aba (appearing once), and baa (appearing once).

In this problem, you will write a program that, given a potentially large string and some k -mers, will determine how many times each k -mer appears in the string. Unlike other problems on this exam, your solution must not only be correct but also *efficient* (we provide specific bounds below).

Input

The input contains a string and the k -mers we want to count in that string. This string and the k -mers will contain only lowercase letters (a through z). Because the string can be very long, it may be broken up into multiple lines. In particular, if the string has 40 or less characters, it will appear in a single line. If the string has 41 or more characters, it will be broken up into multiple lines, with every line (except possibly the last one) having 40 characters.

The first line of input contains four positive integers: L , N , k , Q (each separated by a single space). This first line is followed by L lines containing the input string, which will have N characters in total. As noted above, if $N \leq 40$, the input string will appear in a single line (and L will be 1). If $N > 40$, the input string will be broken up into multiple lines.

The L lines specifying the string are followed by Q lines containing the k -mers we want to count in the string. Note that an input file will use the same value of k for all the k -mers, with that value specified in the first line. So, if the first line specifies $k = 3$, then the Q lines will all contain 3-mers. Note that the k -mers are not guaranteed to be unique; the same k -mer may appear more than once in these Q lines.

You may assume the following bounds:

$$\begin{aligned} 1 < L &\leq 3,750 \\ 1 < N &\leq 150,000 \\ 1 < k &\leq 100 \\ 1 < Q &\leq 150,000 \end{aligned}$$

Output

For each k -mer specified in the input, print the k -mer and the number of times it appears in the string (separated by a single space). The k -mers must appear in the same order they appear in the input.

Complexity

Your solution must be able to produce the expected output by making at most one pass through the string (not counting the initial loading of the string from the input). More specifically, your solution should run in $O(k \cdot N + k \cdot Q)$. Solutions that run in $O(k \cdot N \cdot Q)$ will result in a *Time Limit Exceeded* when run on our secret test cases. Think carefully about what data structure(s) and algorithm(s) to use in your solution.

Sample Input 1

```
1 7 2 3
aaabaab
ab
ba
aa
```

Sample Output 1

```
ab 2
ba 1
aa 3
```

Sample Input 2

```
1 7 3 5
aaabaab
aaa
aab
aaa
baa
xyz
```

Sample Output 2

```
aaa 1
aab 2
aaa 1
baa 1
xyz 0
```

Sample Input 3

```
5 170 3 4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaab
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaa
aab
aba
baa
aaa
```

Sample Output 3

```
aab 1
aba 1
baa 1
aaa 165
```