H – CYCLIC ISOMORPHISM

Apparently, the query "How many times does a string x appear in a string s?" is one of the most basic problems of string manipulation. Tèo has learned about this problem, and he can solve it in a blink of an eye. Therefore, he wants to challenge himself by complicating the problem.

So, he tries solving another problem: given a string **s** and n strings x_i, for each string x_i, find the number of successive substrings of s that are cyclically isomorphic to x_i.

Two strings are called *cyclically isomorphic* strings if one string is a *rotation* of the other. In this case, *rotation* is defined as an operation of rearranging the characters in a string by moving some consecutive characters (maybe none) from the beginning to the end of the string in the same order, while shifting all other characters to the preceding position. For example, the string *"bcda"* is a rotation of *"abcd"*, but the string *"bdca"* is not a rotation of *"abcd"*.

This problem is kind of tricky to Tèo. Let's help him to solve it.

INPUT:

- The first line contains a non-empty string s. The length of s is not greater than 10^6 characters.
- The second line contains an integer n ($1 \le n \le 10^5$) - the number of queries.
- The following n lines: the i-th line contains the string x_i - the string for the i-th query. The total length of x_i is less than or equal to 10^6 characters.

In this problem, strings only consist of lowercase English letters.

OUTPUT:

For each query x_i print a single integer that is the number consecutive substrings of s are cyclical isomorphic to x_i. Print the answers to the queries in the order they are given in the input.

EXAMPLE:

Input	Output
babbabbabbabbabbabbabbabbabbab 10 bbab abbabbab abba abbab abbabba a babb abba aabaaaa babbab	18 16 9 18 8 10 18 9 0 25