Problem B
Shortest Missing Subsequences

Time Limit: 10 sec

Given a string s we say that string t is a *Subsequence* of s if t can be obtained from s by deleting zero or more characters of s. Note that t is not necessarily a substring of s—that is, t is not necessarily contiguous in s, but the characters of t appear in the same order as they do in s.

For a given subset, v, of the lowercase English alphabet characters from `'a'` to `'z'`, we say that string u is a *Missing Subsequence* of another string s if u is not a *Subsequence* of s, but all characters in u and all the characters of s are in the set v. A *Shortest Missing Subsequence* of s is a *Missing Subsequence* of s with the smallest length among all *Missing Subsequences* of s.

Given a set of English alphabetic characters, a target string made up of characters from that set, and a list of query strings made up of characters from that set, determine if each of the query strings is a *Shortest Missing Subsequence* of the target string.

Input

The first line of input contains a string v ($1 \leq |v| \leq 26$) of lowercase letters, in lexicographical order. Each letter appears at most once. This is the set of alphabetic characters.

The next line of input contains a string s ($1 \leq |s| \leq 10^6$, s only contains letters from v). This is the target string to be queried.

The next line contains an integer n ($1 \leq n \leq 10^6$). This is the number of queries.

Each of the next n lines contains a string q ($1 \leq |q| \leq 10^6$, q only contains letters from v). These are the query strings. The sum of the lengths of all query strings will not exceed 10^6.

Output

Output n lines, one for each query. On each line, output either 1 if the query string is a *Shortest Missing Subsequence* of the target string, or 0 if it is not. The outputs must be in the order of the input queries.

Sample Input 1	Sample Output 1
abc abcccabac 3 cbb cbba cba	1 0 0