Problem C. Keyboard

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 mebibytes

On the keyboard of Bytherine, the famous Bithuanian programmer, *Backspace* has been broken. This button is of great importance to her, as due to her sloppiness, she often makes mistakes in names of variables and has to correct them using this ill-fated key. On the other hand, she considers *CapsLock* pretty much useless. Indeed, to type a capital letter, you can obviously press a *Shift* button. This is why she swapped functionality of *CapsLock* and *Backspace*. Ever since, she has been using *CapsLock* button to delete recently typed characters.

But this fact doesn't spell the end of Bytherine's problems. A cunning hacker has been trying to steal Bytherine's password. He managed to intercept the signal emitted by the keyboard. Bytherine, unaware of danger, typed her password on the keyboard. This is something the hacker has been waiting for. Now he has everything that Bytherine typed on her keyboard.

Bytherine, to type her password, used only small and capital letters of English alphabet and *CapsLock* button. To enter her password, she pressed the *CapsLock* key every time she wanted to delete the last entered character. In particular, she did not press it when there had not been any entered character yet.

On the other hand, no character was removed in the hacker's editor. Every time Bytherine pressed the *CapsLock* button, only the writing mode changed. After every odd pressing of the *CapsLock* button, every lowercase letter entered by Bytherine was changed to a capital letter and vice versa. After every even pressing of the *CapsLock* button, the behaviour of the keyboard returned to normal.

For example, if Bytherine pressed the following keys: *P*, *CapsLock*, *t*, *A*, *CapsLock*, *a*, *k*, she would type the word `tak`, but the hacker would see the word `PTaak`.

Hacker's editor displays the word s. Your task is to write a program, which for every n popular passwords $z_1, z_2, \ldots, z_n$ will say if it can be Bytherine's password.

Input

The first line of the input contains s ($1 \le |s| \le 1\,000\,000$) – the word which is displayed in the hacker's editor. The second line contains one integer n ($1 \le n \le 1\,000\,000$) – the number of popular passwords to check. The i-th of the following n lines contains exactly one non-empty password z_i. The sum of lengths of words z_i does not exceed $1\,000\,000$. All words in the input consist of only capital and small letters of the English alphabet.

Output

You should write n lines; the i-th of them should contain the word `YES` or `NO`, depending on whether it is possible that the password of Bytherine is z_i.

Example

standard input	standard output
PTaak 4 PA tak ptak nie	YES YES NO NO