Problem M
Smart Password Validation
Time Limit: 1 second, Memory limit: 2G

People often make errors when entering passwords for one of their many accounts. This is becoming more of a problem with the rather lengthy passwords that many online portals require (the more characters in the password, the more chance there is of making a typing mistake). Practitioners of the "touch typing" technique are prone to make serious errors if one of their hands is not in the proper home row position when typing, since touch typists do not need to look at the keyboard as they type. This can cause all characters typed by a misplaced hand to be "one key off". E.g., on a QWERTY keyboard, "lion" might become "kuib" if the right hand is placed one key to the left of normal home position.

The Frobozz Magic Security Company has decided to implement a new password validation algorithm that takes into account some common mistakes (but not all) made by folks when entering passwords on a QWERTY-style keyboard (see Figure M.1).

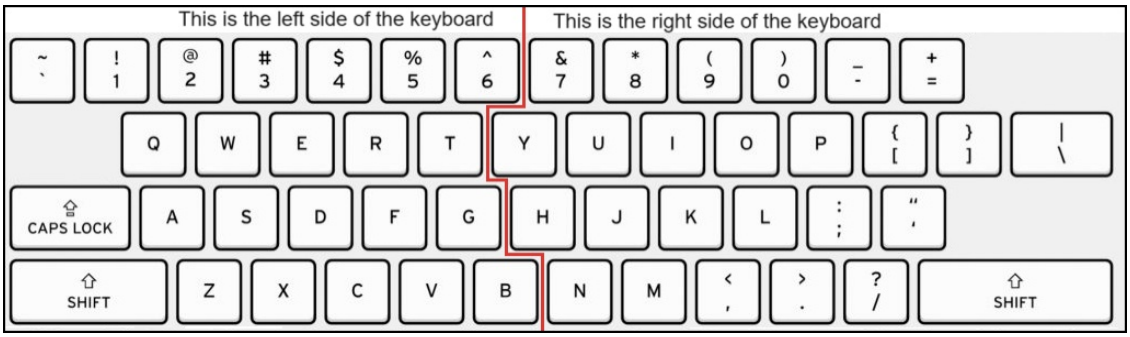


Figure M.1: QWERTY keyboard

The effect of pressing CAPS LOCK is limited to alphabetic characters and toggles the effect of the shift key for those characters. In others words, when CAPS LOCK is not in effect, all letters are typed as lowercase unless one or both shift keys are pressed to produce an uppercase letter. The opposite behavior occurs when CAPS LOCK is on—uppercase is the default unless one or both shift keys are pressed, producing lowercase letters. The algorithm designers assume that the state of the CAPS LOCK key will not change in the middle of entering a password.

Their algorithm looks for deviations from a correct sequence of keystrokes. For example, if the correct password is "`ALg*`", the correct sequence, using the keyboard above, is "SHIFT-A, SHIFT-L, G, SHIFT-8". If the user were to accidentally hit the CAPS LOCK key before entering the password, but then used the correct keystroke sequence, they would end up typing "`alG*`". Likewise, if a touch typist accidentally placed the fingers of the left hand one space to the right of the correct position, made no other errors, and attempted to follow the correct sequence (unaware of the shift in finger positions), they would type "`SLh*`". And if, in addition, they also accidentally activated the CAPS LOCK key before entering the password, they would type "`slH*`". Finally, a user might accidentally insert or delete a character. For example, the password "ALg*" might be mistyped as "AL*" or "ALLg*".

The Frobozz Magic Security Company has decided to permit certain deviations, or combinations of deviations, from the correct sequence, as described in Figure M.2. At most one of the LS, RS, EC and MC errors

Name	Acronym	Description	Examples
Left side off by one	LS	All characters on the left side of the keyboard are shifted right by one key. The left-shift key shifted right by one key is still the left-shift key.	`FlatHead` ⇒ `GlsyHrsf`
Right side off by one	RS	All characters on the right side of the keyboard are shifted left by one key. The right-shift key shifted left by one key is still the right-shift key.	`FlatHead` ⇒ `FkatGead`
Single extra character	EC	The entered password has an extra character	`Zorkmid` ⇒ `Zorkmiid` (extra `i`)
Single missing character	MC	The entered password is missing a character.	`FCD#3` ⇒ `FCD3` (missing the `#`)
CAPS LOCK was in effect by mistake	CL	All lowercase letters appear as uppercase and all uppercase letters appear as lowercase	`WE do it!` ⇒ `we DO IT!`

Figure M.2: Permitted Error Types

is permitted, but any of these may occur in conjunction with a CL error.

Input

There are multiple lines of input. The first line contains a string, p, which is the correct password ($2 \leq \text{length}(p) \leq 24$). The second line contains a single positive integer n ($1 \leq n \leq 1\,000$), which is the number of lines that follow. Each of the following n lines contains a single password to test, t_i ($\text{length}(p) - 1 \leq \text{length}(t_i) \leq \text{length}(p) + 1$). The characters for all passwords come only from those that appear on the QWERTY keyboard picture above, as well as the lowercase letters.

Output

Output consists of n lines. For each t_i the line should consist of the word `YES` if t_i matches the password p using Frobozz's algorithm, `NO` if it does not.

Sample Input 1	Sample Output 1

`Password='A:'`
`9`
`Password='A:'`
`pASSWORD='a:'`
`Psddeotf=1S:'`
`oASSWIRD-'al;`
`Password='A:'x`
`pASWORD='a:'`
`pASSWORD='a:'X`
`Osddeitf-1SL;`
`Psdeotf=1S:'`
	`YES`
	`YES`
	`YES`
	`YES`
	`YES`
	`YES`
	`YES`
	`NO`
	`NO`