TLE '16 Contest 8 P1 - Paper Hole Punching

Time limit: 1.0s **Memory limit:** 256M

Nathan is a competitor in the Canadian Computing Olympiad (CCO). To prepare for this difficult competition, he will bring a thick binder full of code samples (even though the CCO disallows paper notes).

Hole punched paper in a 3-ring binder! An upgrade over stapled paper!

Nathan begins by categorizing the pages according to the holes that appear. The rules are stated below:

- A category is a string with a length between 1 and 7.
- The first letter of the string is `-`.
- The possible hole locations on the page are labelled from `A` to `F`. When a page has a hole, its corresponding letter is in the string.
- The letters in the string will appear in sorted order.

For example, the category `-ABF` represents a page with holes at `A`, `B`, and `F`. The category `-ABCDEF` represents a page with holes at all 6 locations.

Nathan's binder is slightly unusual. Since he has a *massive* number of pages, he needs certain holes in certain locations. The page template T contains all of the holes needed for his pages. Nathan **absolutely cannot** rotate/flip over any page and put it into the binder, since he needs the extra few seconds to boost his ranking.

Nathan will put N pages into his binder. Can you tell him whether each page will fit or not?

Input Specification

The first line contains T, the page template.

The second line contains N $(1 \leq N \leq 1000)$.

The next N lines contain a category. Nathan has a page in this category, and he wants to put it into the binder.

Output Specification

For each of the N pages, print `yes` or `no` on its own line. Print `yes` if the page can fit into the binder. Print `no` if additional holes need to be punched.

Sample Input

```
-ABC
4
-AB
-ABC
-ABCDEF
-DEF
```

Sample Output

```
no
yes
yes
no
```

Explanation for Sample Output

From the template T, there must be holes at `A`, `B`, and `C` for a page to fit into a binder.

The first page is missing hole `C`, so the output is `no`.
The second and third pages have the necessary 3 holes, so the output is `yes`.
The fourth page does not have any of the holes, so the output is `no`.