Problem C: Copycat Catcher

Time limit: 3 seconds



A Plagiarism Keyboard

Your university recently established the Graduate Code Plagiarism Control (GCPC) initiative to get hold of the ever-increasing load on the graders for the computer science assignments. Currently, the graders need to check the code of assignments manually for plagiarism. The GCPC aims to simplify this part of the graders' jobs by performing the plagiarism checks automatically.

Code consists of tokens separated by spaces. Tokens are strings of alphabetical letters, numerals, and brackets. If a token consists of only a single alphabetical letter (upper or lowercase), it is a variable in the code.

The GCPC wants the plagiarism checker to compare query pieces of code to a reference code. Specifically, it should check whether each query could have been obtained by selecting a contiguous string of tokens from the reference and consistently renaming variables. Variables are consistently renamed if no two occurrences of the same variable are renamed to different variables, and if no two different variables are renamed to the same variable.

The GCPC has asked you to develop the plagiarism checker.

Input

The input consists of:

- A description of the reference, consisting of:
  - One line containing an integer n ($1 \leq n \leq 2\,000$), the number of tokens in the reference.
  - One line containing n tokens, each consisting only of the characters '`a`'-'`z`', '`A`'-'`Z`', '`0`'-'`9`', '`(`' and '`)`'.
- An integer q ($1 \leq q \leq 2\,000$), the number of queries.
- $2 \cdot q$ lines, each two lines in the same format as the reference.

It is guaranteed that each query as well as the reference consist of at most $2\,000$ characters (excluding spaces). Tokens are separated by single spaces.

Output

For each query, output "`yes`" if the query could have been obtained from the reference, and "`no`" otherwise.

Sample Input 1

```
9
for i in range(10) do print i j end
4
3
print j i
2
do print
6
k in range(10) do print k
6
k in range(10) do print j
```

Sample Output 1

```
yes
yes
yes
no
```

Sample Input 2

```
5
i is i times j
7
5
i is i times j
5
a is a times b
5
j is j times c
5
a is i times j
5
j is i times j
5
0 is 0 times j
5
i is i times i
```

Sample Output 2

```
yes
yes
yes
no
no
no
no
```

Sample Input 3

```
5
A 1 ( ) b
4
2
b 2
2
b 1
3
1 ) (
5
a 1 ( ) F
```

Sample Output 3

```
no
yes
no
yes
```