A. Login Verification

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

When registering in a social network, users are allowed to create their own convenient login to make it easier to share contacts, print it on business cards, etc.

Login is an arbitrary sequence of lower and uppercase latin letters, digits and underline symbols («_»). However, in order to decrease the number of frauds and user-inattention related issues, it is prohibited to register a login if it is *similar* with an already existing login. More precisely, two logins s and t are considered similar if we can transform s to t via a sequence of operations of the following types:

- transform lowercase letters to uppercase and vice versa;
- change letter «`O`» (uppercase latin letter) to digit «`0`» and vice versa;
- change digit «`1`» (one) to any letter among «`l`» (lowercase latin «`L`»), «`I`» (uppercase latin «`i`») and vice versa, or change one of these letters to other.

For example, logins «`Codeforces`» and «`codef0rces`» as well as «`OOOOOOOOOOOOOOOOOOOOOO_lol`» and «`OOOOOOOOOOOOOOOOOOOOOO_1oI`» are considered similar whereas «`Codeforces`» and «`Code_forces`» are not.

You're given a list of existing logins with no two *similar* amonst and a newly created user login. Check whether this new login is *similar* with any of the existing ones.

Input

The first line contains a non-empty string s consisting of lower and uppercase latin letters, digits and underline symbols («_») with length not exceeding 50 — the login itself.

The second line contains a single integer n ($1 \le n \le 1\,000$) — the number of existing logins.

The next n lines describe the existing logins, following the same constraints as the user login (refer to the first line of the input). It's guaranteed that no two existing logins are similar.

Output

Print «`Yes`» (without quotes), if user can register via this login, i.e. none of the existing logins is *similar* with it.

Otherwise print «`No`» (without quotes).

Examples

input
1_wat 2 2_wat wat_1

output
Yes

input
000 3 00 ooA oOo

output

```
No
```

input

```
_i_
3
__i_
_1_
I
```

output

```
No
```

input

```
La0
3
2a0
La1
1a0
```

output

```
No
```

input

```
abc
1
aBc
```

output

```
No
```

input

```
0Lil
2
LIL0
0Ril
```

output

```
Yes
```

Note

In the second sample case the user wants to create a login consisting of three zeros. It's impossible due to collision with the third among the existing.

In the third sample case the new login is similar with the second one.