CCSC:MW Programming Competition

Password Validation

Web sites often require user passwords to meet certain criteria. For this problem you will validate passwords against rules which specify the number of characters required from the following character classes:

- Lowercase letters a through z (denoted with L)
- Uppercase letters A through Z (denoted with U)
- Digits 0 through 9 (denoted with D)
- Special characters (denoted with S)

A special character is any printable non-whitespace ASCII character that is not in the L, U, or D classes.

Rules

Each rule is a string that specifies minimum required counts of characters from 1 or more of the 4 classes. For each specified count c, $1 \leq c \leq 99$. If a character class is missing from a rule, there may be any number of characters present from that class (including 0). The classes may be specified in any order, but no class will appear more than once. Here are some example rules:

- `2L2U1D1S` - requires at least 2 lowercase, 2 uppercase, 1 digit, 1 special
- `2S4D` - requires at least 2 special, 4 digits

Input

The first line of input is an integer n which specifies the number of test cases, where $1 \leq n \leq 99$.

The next n lines each contain a rule and a password, separated by a space. Each password will contain at least 1 and at most 48 characters, and all of the characters in each password will come from the 4 defined character classes.

Output

For each rule/password pair, print `YES` if the password is valid according to the rule and print `NO` if the password is not valid according to the rule.

Example

Input:

```
4
1U1L1D1S hello
1L1U1S1D H4llo!
1L1U hello
10S a!@#$%^&*()z
```

Output:

```
NO
YES
NO
YES
```