In this problem we will consider Boolean formulas written according to the following BNF grammar:

```
<formula> ::= <clause> | <formula> "|" <clause>
<clause> ::= "(" <conjunction-of-literals> ")"
<conjunction-of-literals> ::= <literal> | <conjunction-of-literals> "&" <literal>
<literal> ::= <variable> | "~" <variable>
<variable> ::= "a" | "b" | "c" | ... | "z"
```

Each formula can contain up to 26 different Boolean variables, which are denoted by lowercase English letters. We use the ampersand sign ("&") to denote conjunction, vertical bar ("|") for disjunction, and tilde ("~") for inversion. The truth tables of these operators are shown below for your reference.

x	y	x & y
false	false	false
false	true	false
true	false	false
true	true	true

x	y	x | y
false	false	false
false	true	true
true	false	true
true	true	true

x	~x
false	true
true	false

A formula is called satisfiable if it is possible to assign values to its variables in such a way as to make the formula evaluate to true.

Input

The first line of the input file contains an integer T ($1 \leq T \leq 5000$). Each of the next T lines contains a Boolean formula. You can assume that the formulas will strictly follow the grammar specified above, and will not be longer than 250 characters.

Output

For each formula, you should determine whether it is satisfiable, and output a line 'YES' if yes, it is, and 'NO' otherwise.

Sample Input

```
2
(a&b&c)|(a&b)|(a)
(x&~x)
```

Sample Output

```
YES
NO
```