

Problem LISP: Lisp Expressions

Mini-Lisp is the new programming language of the year. Mini-Lisp has a really easy syntax. Your task is to write a recognizer for Mini-Lisp expressions. A recognizer only checks the syntax of a program and replies `Yes` if the syntax is correct and `No` otherwise. The syntax of Mini-Lisp expressions is given as Extended Backus Naur Form (EBNF):

```
expression ::= identifier
             | (lambda (identifier) expression)
             | (expression expression)
```

Additionally take care of the following:

- An identifier can only consist of the lowercase letters a, b, \dots, z and nothing else.
- `lambda` is a reserved word and not an identifier.
- Each Mini-Lisp expression is written in one line with at most 100 characters, no linebreaks or tabs are allowed.
- But there can be as many spaces as the programmer wants between `(,)`, `lambda` and the identifiers. Between two parentheses and parentheses and identifiers no spaces are necessary.

Input

First the number of Mini-Lisp programs is given. Each Mini-Lisp program is given in one line. The next program starts in the next line.

Output

The output is either `Yes` or `No` depending on the correctness of the given Mini-Lisp program. Each answer is written in a new line.

Sample Input 1

10
brimborium
lambdam
(write this)
(lambda (test) case)
(lambda (test) (lambda (test) fertig))
lambda
(everything wrong here)
(alsowrong)
(lamda (test) case)
(lambda (test) case

Sample Output 1

Yes
Yes
Yes
Yes
Yes
No
No
No
No
No
No