Brackets

Time limit: 2 sec.
Memory limit: 64MB

Description

Junhee Cho is a computer science graduate student at KAIST. Recently, he finished writing a paper and he decided to polish it. As he started to read it from the beginning, he realized that some of the formulas have problems: some of the brackets are mismatched! Since there are so many formulas in his paper, he decided to check their validity with a computer program.

The following kinds of brackets are included in Junhee's paper.

- Round brackets are opened by a '(' and closed by a ')'.
- Curly brackets are opened by a '{' and closed by a '}'.
- Square brackets are opened by a '[' and closed by a ']'.

A formula is said to be well-matched when the following conditions are met:

1. Every bracket has a corresponding pair. '(' corresponds to ')', '[' corresponds to ']', et cetera.
2. Every bracket pair is opened first, and closed later.
3. No two pairs "**cross**" each other. For example, [(]) is not well-matched.

Write a program to help Junhee by checking if each of his formulas is well-matched. To make the problem easier, everything other than brackets are removed from the formulas.

Input

The input is given in a single line as a character string. The strings will only include characters in "[](){}" (quotes for clarity). The length of the string will not exceed 10,000.

Output

For each test case, print a single line "YES" when the formula is well-matched; print "NO" otherwise. (quotes for clarity)

Sample I/O

Input(s)	Output(s)
()()	YES
({[}])	NO
({}[(){}])	YES