Two brackets are considered to be a matched pair if the an opening bracket (i.e., (, [, or {) occurs to the left of a closing bracket (i.e.,),], or }) of the exact same type. There are three types of matched pairs of brackets: [], {},and ().

A matching pair of brackets is not balanced if the set of brackets it encloses are not matched. For example, {[(])} is not balanced because the contents in between { and } are not balanced. The pair of square brackets encloses a single, unbalanced opening bracket, (, and the pair of parentheses encloses a single, unbalanced closing square bracket,].

By this logic, we say a sequence of brackets is balanced if the following conditions are met:

It contains no unmatched brackets.

The subset of brackets enclosed within the confines of a matched pair of brackets is also a matched pair of brackets.

Given n strings of brackets, determine whether each sequence of

brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.

INPUT:

The first line contains a single integer n, the number of strings.

Each of the next n lines contains a single string s, a sequence of brackets.

OUTPUT:

For each string, return YES or NO.

SAMPLE INPUT

```
3
{[()]}
{[(])}
```

```
{{[[(())]]}}
```

SAMPLE OUTPUT

```
YES
NO
YES
```

Explanation

1.The string {[()]} meets both criteria for being a balanced string, so we print YES on a new line.

2.The string {[(])} is not balanced because the brackets enclosed by the matched pair { and } are not balanced: [(]).

3.The string {{[[(())]]}} meets both criteria for being a balanced string, so we print YES on a new line.