Many Many Heads

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

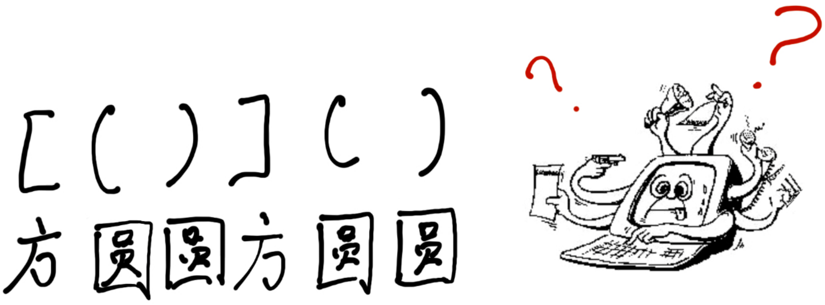
Multi-Heads Cup, or MHC for short, is a worldwide programming contest made for the participants with **many many heads**. The Chief Judge of the competition, Little Cyan Fish, is considering to design an identification number for each participant.

“And that’s it,” Little Cyan Fish thought, “Let’s use some bracket sequences!” He assigned a unique balanced bracket sequence to each participant, with two kinds of the brackets - the round brackets (also known as parentheses), and the square brackets. To make sure you understand the concepts of balanced bracket sequence, Little Cyan Fish prepared a formal definition of the balanced bracket sequence:

- ε (an empty string) is a balanced bracket sequence.
- If A is a balacned bracket sequence, then (A) and $[A]$ are both balanced bracket sequences.
- If A and B are balanced bracket sequences, then AB is also a balanced bracket sequence.

For example, “`()`”, “`[()]`” and “`[()]()`” are balanced bracket sequences, but “`)(`”, “`[(])`“ and “`[)`“ are not.

For our multi-headed participants, memorizing a bracket sequence is not a difficult task. However, the challenge lies in their unique ability: because they have too much heads, they can’t discern the direction of each bracket! Consequently, compared to the original balanced bracket sequence, they might recall the sequence with some brackets changing their directions. For example, the bracket sequence “`[()]()`” might be memorized as “`])))]))`” or “`]()]))`”. Fortunately, the type of brackets are guaranteed to remain unchanged.



On the contest day, as Little Cyan Fish receives each participant’s bracket sequence, a question arises: Can the original bracket sequence be deduced uniquely? In other words, Little Cyan Fish needs to determine if the provided bracket sequence maps to exactly one balanced bracket sequence.

Please, help Little Cyan Fish to finish this task, so our multi-heads friends will be able to participate in the contest!

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains a string S consisting of ‘`(`’, ‘`)`’, ‘`[`’, and ‘`]`’ ($1 \le |S| \le 10^5$), indicating the bracket sequence.

It is guaranteed that:

- The sum of $|S|$ over all test cases does not exceed 10^6.
- Each bracket sequence is obtained by changing the directions of some brackets in a balanced bracket sequence.

Output

For each test case:

- If the provided bracket sequence maps to more than one balanced bracket sequence, output a single line `No`.
- Otherwise, output a single line `Yes`.

Example

standard input	standard output
`6`	`Yes`
`))`	`No`
`(()`	`Yes`
`[()]`	`No`
`()[()]()`	`Yes`
`([()])`	`No`
`([])([])`	

Note

In the first test case, the bracket sequence maps to exactly one balanced bracket sequence: `()`. So the answer is `Yes`.

In the second test case, the bracket sequence corresponds to two distinct balanced bracket sequences: `(())` and `()()`. So the answer is `No`.

In the third test case, the bracket sequence maps to exactly one balanced bracket sequence: `[()]`. So the answer is `Yes`.

In the fourth test case, the bracket sequence corresponds to two distinct balanced bracket sequences: `(([()]))` and `()[()]()`. So the answer is `No`.

In the fifth test case, the bracket sequence maps to exactly one balanced bracket sequence: `([()])`. So the answer is `Yes`.

In the sixth test case, the bracket sequence corresponds to three distinct balanced bracket sequences: `([])([])`, `([]()[])` and `([[()]])`. So the answer is `No`.