A bracket is considered to be any one of the following characters: (,), {, }, [, or].

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 strings of brackets, determine whether each sequence of brackets is balanced. If a string is balanced, return YES. Otherwise, return NO.

Function Description

Complete the function *isBalanced* in the editor below.

isBalanced has the following parameter(s):

- *string s*: a string of brackets

Returns

- *string:* either YES or NO

Input Format

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

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

Constraints

-

- , where is the length of the sequence.

All chracters in the sequences ∈ { **{**, **}**, **(**, **)**, **[**, **]**

```
import java.io.*
import java.math.*
import java.security.*
import java.text.*
import java.util.*
import java.util.concurrent.*
import java.util.function.*
import java.util.regex.*
import java.util.stream.*
import kotlin.collections.*
import kotlin.comparisons.*
import kotlin.io.*
import kotlin.jvm.*
import kotlin.jvm.functions.*
import kotlin.jvm.internal.*
import kotlin.ranges.*
import kotlin.sequences.*
import kotlin.text.*

/*
 * Complete the 'isBalanced' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts STRING s as parameter.
 */

fun isBalanced(s: String): String {
    // Write your code here

}

fun main(args: Array<String>) {
    val t = readLine()!!.trim().toInt()

    for (tItr in 1..t) {
        val s = readLine()!!
```

```
        val result = isBalanced(s)

        println(result)
    }
}
```

- }.

Output Format

String YES or NO