## Balanced Brackets

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 considered to be *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, printYES on a new line; otherwise, print NO on a new line.

**Input Format**

The first line contains a single integer, , denoting the number of strings.   
Each line  of the  subsequent lines consists of a single string, , denoting a sequence of brackets.

**Output Format**

For each string, print whether or not the string of brackets is balanced on a new line. If the brackets are *balanced*, print YES; otherwise, print NO.

**Sample Input**

3

{[()]}

{[(])}

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

**Sample Output**

YES

NO

YES

Solution :

import java.io.\*;

import java.util.\*;

public class Solution {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int t = in.nextInt();

for(int a0 = 0; a0 < t; a0++) {

String s = in.next();

Stack stk = new Stack();

boolean flag = true;

if(s.length()%2!=0){ System.out.println("NO"); }

else if(s.charAt(0)=='}' || s.charAt(0)==']' || s.charAt(0)==')'){ System.out.println("NO"); }

else{

for(int i=0;i<s.length();i++){

if(s.charAt(i)=='{' || s.charAt(i)=='[' || s.charAt(i)=='('){ stk.push((Character)s.charAt(i)); }

else{

if(stk.empty()){ flag=false;break; }

char top = (char)stk.peek();

if(top=='{' && s.charAt(i)=='}'){ stk.pop(); }

else if(top=='[' && s.charAt(i)==']'){ stk.pop(); }

else if(top=='(' && s.charAt(i)==')'){ stk.pop(); }

else{ flag=false;break; }

}

}

if(flag==true && stk.empty()){ System.out.println("YES"); }

else{ System.out.println("NO"); }

}

}

}

}