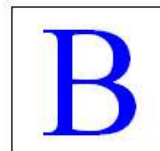




acm International Collegiate Programming Contest

IBM

event sponsor



Problem B

Propositional Satisfiability

A **Minimal Propositional Formula** (MPF) is a string of symbols made from propositional atoms using the boolean connectives $\&$ (“and”), $+$ (“or”), $!$ (“not”), and brackets, in the appropriate way. More accurately:

1. Any propositional atom (a, b, c, \dots, z) is an MPF.
2. If A is an MPF then so is $(!A)$.
3. If A, B are MPFs then so are $(A \& B)$ and $(A + B)$.
4. That’s it: nothing is an MPF unless built by these rules.

Brackets can be omitted. To get rid of brackets, we order the boolean connectives according to decreasing binding strength:

(strongest) $!$, $\&$, $+$ (weakest)

This is like in arithmetic, where \times is stronger than $+$. This means that $2 + 3 \times 4$ is read as $2 + (3 \times 4)$, not as $(2 + 3) \times 4$. So:

The semantics (meaning) of an MPF is simply given by the following truth tables:

$p + q \& r$ **is read as** $p + (q \& r)$, not as $(p + q) \& r$.

$!p \& q$ **is read as** $(!p) \& q$, not as $!(p \& q)$.

p	q	p & q	p + q	!p
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

An MPF is *satisfiable* if there is a *situation* that the MPF is evaluated to be TRUE. If there is no such situation, the MPF is *unsatisfiable*. For examples:

$p \& q$ is satisfiable on the situation that both p and q are TRUE.

$p \& !p$ is unsatisfiable as there is no situation that can make it to be TRUE.

Your job is to write a program to take an **MPF as an input** (from standard input) and determine whether is satisfiable or not, by **outputting YES or NO** (to standard output) accordingly. The maximum length of the input string is 100 characters including spaces and brackets.

Sample Input	Sample Output
a q + !q p & q + r & (s + !t) a & (b + !c) & (d + f + g) & !(d + f + g) p & !p	YES YES YES NO NO