Problem K. Keys and Locks Boolean Logic

Time limit:	2 seconds
Memory limit:	512 megabytes

Karen and her friends play in a music band in Karen's garage. She has complicated thoughts on which subsets of the band should be allowed to enter the garage. She designed a boolean formula F in which each variable denotes the presence of a certain band member. She wants a group of people to be able to enter the garage if and only if F is true.

You are asked to create a system of wires and locks that will fulfill the formula F. Each band member will receive a key that opens all the locks that are denoted by their letter.

For example, suppose that $F = a \vee (b \wedge c)$ and band members a and b are present. Then they should be able to enter the garage with their keys because $true \vee (true \wedge false) = true$.

A band member doesn't have to use their key to open all the locks possible. For example, suppose that $F = a \oplus b$. Then band member a alone should be able to enter the garage because $true \oplus false = true$. But if both a and b come to the garage, they can just disregard b's key and open the garage using a's one. However, $true \oplus true = false$, so this formula F is not fulfillable.

The system must be a rectangular grid of size at most 50×50 containing horizontal wires '`-`', vertical wires '`|`', wire connections '`+`', and locks. The upper leftmost and the upper rightmost cells of the grid must contain wire connections that will be attached to the doors of the garage. The garage doors stay closed as long as there is a path between these two connections via wires and locked locks.

You are to design such a system corresponding to the given formula F, or to state that there is no such system.

Input

The input contains a non-empty boolean formula F.

The formula can contain letters `a`, ..., `h` denoting the presence of band members, operators "`and`", "`or`", "`not`", and parentheses. The length of the formula does not exceed 2020. The "`not`" operator has the highest priority. The "`and`" operator has higher priority than the "`or`" operator. There are single spaces around each "`and`", around each "`or`", and after each "`not`". There are no spaces other than those.

Output

If it is impossible to design the desired system, print one line "`IMPOSSIBLE`".

Otherwise, output a rectangular grid of characters — the system corresponding to formula F. The system can only contain spaces, '`-`', '`|`', '`+`', and letters corresponding to the band members that are mentioned in the input.

The upper leftmost and the upper rightmost cells should be wire connections. The width of the grid should be between 2 and 50, inclusive, the height of the grid should be between 1 and 50, inclusive.

Character '`-`' must be used for a wire if and only if it connects something on the left and something on the right with empty cells below and above. Similarly, '`|`' must be used for a wire if and only if it connects something below and something above with empty cells on the left and on the right.

Examples

standard input	standard output
a or (b and c)	+-+ +b-+ | | | | +-a-+-c+
(a or f) and ((a and g) or (a and h))	+a-f+ | | +g+h+ a | a +-+-+
a and not b or not a and b	IMPOSSIBLE
b or not b	+ +
d and not d	+ +---+ +---+--+---+ | | | | | +--+ +---+ | | | | | + +------+ +---+