Problem E. Expression

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

Dwarfs' legendary machinery construction skills allowed them to build mechanical computing machines predating Babbage's analytical engine. Recently though, a trend of minimalism has gained popularity, leading some to suggest that these elaborate machines might be swapped for simpler ones with reduced sets of instructions, which should be much smaller, cheaper, and perhaps also significantly faster in operation. So far, the dwarfs were unable to determine what is the minimal set of operations that suffices to perform all computations they deem important, and to resolve this issue, they have decided to perform numerous tests. You are to solve some of them, and your goal is to rewrite, if possible, given expressions into equivalent ones using only a specified subset of operators.

You are given an expression consisting of binary operators `min`, `max`, `<=`, `<` and variables (the first 10 letters of the English alphabet). A valuation of the variables from the set S is defined as assigning each element of S a value of 0 or 1. We say that two expressions are equivalent if both of the following conditions are satisfied:

- the sets of variables occurring in the expressions are equal – let us denote this set by S,
- for every valuation of S, the expressions with substituted values evaluate to the same value.

Your task is to create an expression consisting solely of the operators `min` and `<=` that is equivalent to the input expression, or to state that such an expression does not exist.

Input

The first and only line of standard input contains an expression. It is an `expr` in the following grammar.

- `var := a | b | ... | j`
- `op := expr <= expr | expr < expr | min expr expr | max expr expr`
- `expr := var | (op)`

Limits Given expression consists of at most 200 binary operations.

Output

If a solution exists, print `YES` on the first line of output, and on the second line any equivalent expression consisting of at most 40 000 binary operations. Otherwise, print a single line containing `NO`.

Note: It can be proven that if a solution of any size exists, there also exists one consisting of at most 40 000 operations.

Examples

standard input	standard output
`h`	`YES` `h`
`((max a a) <= b)`	`YES` `(a <= b)`
`((max a b) < (min a b))`	`NO`