

Problem D. Ones

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Let us define *1-expressions* to be the numeric expressions containing only ones, addition signs, multiplication signs and parentheses. In such expressions no two digits can be neighboring – every two ones must be separated by an operator. We follow the usual order of evaluating the expressions – for example, the multiplication has larger priority than the addition.

For example, each of the following 1-expressions evaluates to 6:

$(1+1)*(1+1+1)$, $(1+1+1)*(1+1)*1$, $((1+1)+1)*(1+1)$, $1+1+1+1+1+1$, $1+(1+(1+(1+(1+1))))$.

Formally, the following grammar describes all the correct 1-expressions E :

$$E ::= 1 \mid E+E \mid E*E \mid (E+E) \mid (E*E)$$

Write a program that, given an integer k ($k \leq 10^9$), outputs a 1-expression evaluating to k that contains at most 100 ones.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 100$) – the number of testcases.

Each of the following t lines describe a single testcase. The i -th of these lines describes the i -th test and contains a single integer k_i ($1 \leq k_i \leq 10^9$).

Output

You should output exactly t lines.

If there is no 1-expression evaluating to k_i and containing at most 100 ones, you should output NO in the i -th line. Otherwise, the line should contain the required solution. You should not print any spaces inside the expression. If there is more than one correct solution, print any.

Example

standard input	standard output
2	$(1+1)*(1+1+1)$
6	$1+1+1+1+1+1+1+1+1+1$
10	