Expression Evaluation

Time limit: *8000 ms*
Memory limit: *256 MB*

In this challenge, you will evaluate an arithmetic expression. However the given expression may not be well formed, and you also want to check if this happens. If the expression has mismatched brackets, or it is missing operator or operand, then it is invalid.

Formally, an expression E is defined as:

$$E \to E + E \mid E - E \mid E * E \mid (E) \mid integer$$

Note that E must be non-empty, and a pair of brackets must not contain empty content.

Standard input

The input has a single integer T on the first line, the number of expressions to evaluate.

Each expression to evaluate is given on a single line. An expression is a non-empty string without spaces. It consists of digits giving integer operands, round brackets, and arithmetic operators `+`, `-`, and `*`.

Standard output

For each expression, if it is valid, output the result of its evaluation. Since the result may be very large, output the result modulo $1\,000\,000\,007$ ($10^9 + 7$). If the expression is invalid, output `invalid`.

Constraints and notes

- $1 \leq T \leq 25$
- The length of an expression does not exceed 10^5.
- All the integers in the expression are smaller than $1\,000$ and do not have leading zeroes, even when the expression is invalid.
- Note that all arithmetic operators are binary. That is, there are no unary negations and `-3`, `-1*2`, `1--2` are invalid.
- In modular arithmetic the result is always non-negative. If $a - b < 0 (0 \leq a, b < P)$, then the modular result $(a - b)\ mod\ P = (P + a - b)\ mod\ P$.

- For 20% of the test files, there are no brackets in the given expression.

Input	Output	Explanation
4 1+2*3+4 1*2-3 123 +-*	11 1000000006 123 invalid	There are 4 test cases without round brackets. • Case 1: It is straightforward to evaluate that $1 + 2 \times 3 + 4 = 11$. • Case 2: $1 \times 2 - 3 = -1$. In modular arithmetic, -1 is $1\,000\,000\,006$ (mod $1\,000\,000\,007$). • Case 3: By definition $E \to integer$. Therefore 123 evaluates to 123. • Case 4: Each operator must have operands. This expression is invalid.
4 (1+2)*3 () ()100 (((1)))	9 invalid invalid 1	There are 4 test cases with round brackets. • Case 1: It is straightforward to evaluate that $(1 + 2) \times 3 = 9$. • Case 2: Empty brackets are not allowed by definition. • Case 3: Not only empty brackets are disallowed, but also there misses an operator between $()$ and 100. • Case 4: By definition $E \to (E)$. Therefore $E \to (E) \to ((E)) \to (((E))) \to ((((1))))$, which evaluates to 1.