

DSA Lab 6 Set 1 | Expression

Given a mathematical expression in infix notation where the operands are integers and operators are represented as follows:

1. + represents multiplication
2. - represents addition
3. * represents division
4. / represents subtraction or negative unary operator (/ 3 means negative 3).

Please note the difference in definition from the usual meanings.

Your task is to convert the expression to postfix notation. The precedence order is as below:

- '(' when it is in the expression
- Unary negation operator ' / '
- Multiplication(+), division(*)
- Addition(-), subtraction(/)
- '(' when it is inside the stack

For the unary minus, you assume that the unary minus will immediately follow a left parenthesis. i.e of the form ". . . (/ operand . . . " .

In the postfix notation denotes every unary negation with a ~ (tilde) to differentiate from the binary /

Input

The first line contains an integer N , indicating the number of tokens.

The second line contains N space-separated tokens representing the infix notation of the expression.

Constraints:

Basic:

$$1 \leq N \leq 50$$

$$\text{tokens} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0, (,), +, -, *, /\}$$

Advanced:

$$1 \leq N \leq 100000$$

$$\text{tokens} \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0, (,), +, -, *, /\}$$

Output

N single space-separated tokens representing the postfix notation of the expression.

Sample Test Case

Input:

16

7 + (4 + (/ 3) * (1 - 5))

Output:

7 4 3 ~ + 1 5 - * +