

# Calculator

The input is an arithmetic expression in infix notation, for example,  $24 * (3 + 5) / 7 \% 3$ . The problem is to write a program that reads the expression given over multiple lines, parses it, evaluates it and gives the output value as a floating point integer or prints the string "Malformed expression" if the input is malformed. The numbers in the expression are integers but the output can be floating point. The operators come from the set  $\{+, -, *, /, \%, (, )\}$ . The operators act as per their usual definitions. We assume the usual precedence among operators, that is, from lowest to highest,  $\{+, -\}$ , then  $*$ , followed by  $\{/ , \%\}$  and parenthesis, where,  $+$  and binary  $-$  are at the same level of precedence and have left to right associativity, so do  $\{/ , \%\}$ .

## Note

The three problems given in the assignment have been merged. 75% of the test cases will check if your code works for problem 1 and 2 and 25% of the test cases will check for problem 3. The result that you will get will be between 0 and 100, denoting the number of testcases that your code passed. We will consider your maximum score while grading.

## Update 1: (7 Sep, 2017)

For negative numbers, the '-' and the number will be in separate lines. It is your job to figure out whether it is a unary minus or binary.

## Update 2: (13 Sep, 2017 3:15PM)

Earlier you were supposed to print string "Malformed expression" without new line at the end. Sorry, for not being explicit about this fact. But from now onwards, new line at the end of "Malformed expression" won't matter.

## Input

The first line contains 'n', the number of tokens in the expression. 'n' lines follow each containing either a number or an operator.

## Output

This should be a single real number. Errors less than  $10^{-2}$  will be ignored.

## Constraints

$$1 \leq n \leq 20$$

Inputs will be between -1000 to 1000

## Example

INPUT 1:

4

4

-

-

3

OUTPUT 1:

7

INPUT 2:

7

3

/

(

2

+

1

)

OUTPUT 2:

1

INPUT 3:

3

1

+

(

OUTPUT 3:

Malformed expression