

# Problem A. *String Calculator*

*Write a java program for the below problem statement. Your Java program should be as per the standards:*

1. Your solution should be a maven/gradle/ant project
2. Follow TDD( Test Driven Development ) principles
3. Make necessary assumptions where required, mention any specific assumptions
4. Solution will be judged basis extensibility ,scalability, exception handling and use of proper design and OOPS principles.
5. Use Java 8 features where possible.
6. Please do not google, we appreciate originality of thoughts

*After you are done with the assignment, please checkin your code to GIT repository under your user profile by creating a new repository and share the link with your recruitment contact.*

## Problem

John is a software programmer and he is trying to develop a calculator program. He feels that normal calculators are boring as they dont allow to have input as a full mathematical expression, rather it allows is a sequence of consecutive operands and operators. John is aware of scientific calculators which have support for complex mathematical expressions. However he is much interested to write his own program which can accept a simple to complex mathematical expression and give the output of it.

The mathematical expression can have operators and operands. Operators can be `+` `-` `\` `*` `^` `(` `)` and operands can be number between `0-9`. A sample mathematical expression will look as `7 + (6 * 5^2 + 3-4/2)`. A mathematical expression cannot start or end with a operator with only exception for `(` (expression can start but cannot end) and `)` (expression can end but cannot start). Every opening brace `(` must have a corresponding closing brace `)`. An operand cannot have its adjacent as a operator except for `(` i.e `((` and `)` i.e `)`

## Input

The first line of the input gives the number of test cases, T. T test cases follow. Each consists of one line with a string S. S represents the mathematical expression which will be the input for the string calculator

## Output

For each test case, output one line containing Case #x: y, where x is the test case number (starting from 1) and y is either INVALID EXPRESSION if input string is invalid, or an integer representing the value of the mathematical expression after calculation.

## Limits

$1 \leq T \leq 100$ .

In case of output as decimals, print values only upto 2 decimals

## Sample

## Input

4

$7 + (6 * 5^2 + 3 - 4/2)$

$7 + (67(56 * 2))$

$8 * + 7$

$(8 * 5/8) - (3/1) - 5$

## Output

Case #1: 158

Case #2: INVALID EXPRESSION

Case #3: INVALID EXPRESSION

Case #4: -3

Note: In case of any doubts or confusions please get in touch with your recruitment contact