1. 개요

스택을 이용하여 간단한 계산기를 만들어 봅니다.

2. 뼈대 코드

이 코드를 기본으로 하여 내용을 추가하도록 합니다.
이 코드에는 제출을 위한 입출력과 파일이름만이 정의되어 있습니다.

3. 계산기 명세

0 자바의 long 범위에서 정수 연산을 지원하도록 합니다.

1 지원하는 연산자 : +, -, *, /, %, ^(지수 연산), 소괄호 : '(' ')'

2 괄호는 소괄호 ()만 사용합니다.

3 우선순위는 일반 수식 연산과 같습니다. 구체적으로 아래와 같으며 위쪽이 아래쪽보다 우선 순위가 높습니다. ^ 거듭제곱과 unary - 는 right-associative, 나머지는 left-associative 연산자입니다.

- ⑩ ( )
- ⑩ ^
- ⑩ - (unary)
- ⑩ * / %
- ⑩ + - (binary)

각 기호와 기호 또는 숫자와 기호 사이에는 공백(tab, space)이 여러 개 들어갈 수 있습니다. 그러나 숫자들은 붙여 쓰도록 합니다.

4. 실행 예

프로그램을 실행하면 infix expression을 입력받습니다. 이것을 postfix expression으로 변환한 다음 postfix expression과 계산 결과를 출력하도록 합니다. 수식이 postfix expression으로 변환되면, unary '-' 와 binary '-'는 구분이 되지 않습니다. 따라서 postfix expression으로 변환할 때 unary '-' 는 '~'로 변환해 주십시오. postfix expression출력시에 unary '-'의 경우는 '~'로 출력되야 합니다. 수식 입력 대신 q를 입력하면 프로그램을 종료합니다.

```
% java CalculatorTest                    ← 프로그램 실행
(3 + 4)^2 - 3 * (2 + 3)                 ← 이렇게 입력하면
3 4 + 2 ^ 3 2 3 + * -                     ← 이렇게 출력
34                                      ← 이렇게 출력한다.
3 * 4 - 5 * 6                           ← 이렇게 입력하면
3 4 * 5 6 * -                             ← 이렇게 출력
-18                                     ← 이렇게 출력한다.
100 / 2 ^ 2                             ← 이렇게 입력하면
100 2 2 ^ /                               ← 이렇게 출력
25                                      ← 이렇게 출력한다.
2^2^3                                   ← 이렇게 입력하면
2 2 3 ^ ^                                 ← 이렇게 출력
256                                     ← 이렇게 출력한다.
2^^3                                    ← 이렇게 입력하면
ERROR                                   ← 이렇게 출력한다.
- 51313                                 ← 이렇게 입력하면
51313 ~                                   ← 이렇게 출력
-51313                                  ← 이렇게 출력한다.
Q                                       ← 이렇게 입력하면
%                                         ← 종료한다.
```

입력은 공백이 아무리 많이 있더라도 정상적으로 인식하여야 합니다. (물론 공백 없이 사용해도 인식이 되어야 합니다)
postfix expression을 출력시에는, 각 숫자들이 구분이 되도록 띄어 쓰도록 합니다. 기호와 기호 사이, 기호와 숫자 사이도 띄어 써야 합니다.

5. 유의사항

1. 주어진 CalculatorTest 클래스 내에서 구현하여 **$ java CalculatorTest 같은 식으로 실행할 수 있도록 합니다.** 클래스 내에서 함수를 더 추가하거나 private 클래스를 더 만드는 건 상관없습니다.
2. 에러 처리를 잘 해야 합니다. 여기서 에러는 짝이 맞지 않는 괄호, 잘못된 연산자, 정의되지 않은 기호 등을 포함합니다.
   에러 메시지는 'ERROR' 라고 출력하도록 합니다.
   수식을 파싱하는 과정, 혹은 postfix 로 변환하는 과정에서 에러가 발생할 경우가 이에 해당합니다.
3. **제출 전, 반드시 컴파일 가능 유무와 프로그램의 수행시간을 확인**합니다.
   컴파일시 옵티마이제이션 옵션은 쓰지 않습니다.

⓾ 아래와 같은 명령어를 입력하면 컴파일이 이루어져야 하며, CalculatorTest 라는 이름의 클래스가 생성되어야 채점이 이루어집니다.

```
$ javac CalculatorTest.java
```

⓾ 컴파일 후, 일정 시간 초과시 프로그램을 강제 종료시키기 위해 다음과 같이 timeout 명령어를 반드시 사용합니다.

```
$ timeout [수행시간(초)] java CalculatorTest
        timeout 0.5 java CalculatorTest  // 0.5초 수행
        timeout 1 java CalculatorTest  // 1초 수행
```

⓾ 수행시간 측정을 위해 다음과 같이 time 명령어를 사용할 수 있습니다.

```
$ time java CalculatorTest
```

6. 자주 묻는 질문(필독!!!)

1)API 사용 가능 여부
Java의 Stack<E> 클래스를 사용하셔도 됩니다.

2)overflow 처리
overflow에 대한 처리를 따로 하지 않고 계산하시기 바랍니다.
예를 들어, 4비트 정수를 이용하여 계산한다고 가정한다면 3+(7*5) = 3+3 = 6입니다.
(실제로 long 타입은 64비트 정수이며 이해를 돕기 위해 위와 같이 설명하였습니다.)

3)입력되는 수식
다음의 기호로 이루어진 문자열이 입력됩니다. (q 제외)
'0', '1', …, '9', '+', '-', '*', '/', '%', '^', ' ', '₩t', '(', ')'
종료를 의미하는 문자열은 "q"로만 입력됩니다. 뼈대 코드의 처리를 수정하실 필요가 없습니다.

4)올바른 수식
다음의 경우에 해당하면 올바른 수식입니다. 이외의 경우는 올바른 수식이 아니므로 오류 처리해야 합니다.

정수
(0개 이상의 공백)E(0개 이상의 공백)
-E
E+E
E-E
E*E
E/E
E%E
E^E
(E)

5) 오류 처리
위에서 설명한 "올바른 수식에 해당하지 않는 경우" 또는 "연산이 성립하지 않는

경우" 오류 처리를 합니다.
연산이 성립하지 않는 경우는 x/0, x%0, (y<0일 때) 0^y 의 세 가지 경우만 고려하시면 됩니다.
이 경우 다른 출력을 하지 않고 ERROR만 출력합니다. 특히, 연산이 성립하지 않는 오류를 처리할 때 변환된 수식을 출력하면 틀린 출력이 됩니다.

6)연산자의 우선순위 충돌
문제 설명에 따르면 '^'와 'unary -'의 우선순위로 인해 "2^-3"과 같은 식을 처리할 수 없는 문제점이 있는데, 이는 테스트 케이스에 넣지 않을 예정이니 고려하지 않으셔도 됩니다.

7)연산의 정의
+, -, *, /, %는 정수 연산자를 적용하시면 됩니다.
a^b는 (long) Math.pow(a, b)의 결과를 적용하시면 됩니다. (0^0 포함)

8)주석(comment)
과제1, 과제2는 뼈대 코드가 구체적으로 주어져서 그 안을 잘 채우면 되었는데, 과제3부터는 그렇지 않으므로 코드에 주석을 달아 코드의 구조와 자신의 의도를 잘 표현하는 것을 권장합니다. 너무 상세한 내용까지 다 설명하지말고, 설명이 필요한 부분에만 적절하게 주석을 달아주세요.

9) 기타
입력은 올바른 수식, 올바르지 않은 수식, 연산이 불가능한 수식 모두 입력으로 들어갈 수 있습니다. 과제 1과 2는 올바른 입력만 들어간다고 간주한다고 했지만 과제3은 그렇지 않습니다. 위의 "입력되는 수식" 항목에 나열된 문자로 이루어진 문자열이 입력으로 들어갑니다. 올바르지 않은 수식, 연산이 성립하지 않는 경우에 대해서는 변환된 수식을 출력하지 않고 ERROR만 출력해야 합니다.
0을 제외한 모든 숫자는 0으로 시작하지 않습니다. 0은 0으로만 입력되고 123과 같은 숫자가 0123과 같이 0으로 시작되도록 들어가는 입력은 없습니다.