ADVANCED CALCULATOR

CHALLENGE DESCRIPTION:

The goal of this challenge is to create an advanced calculator.
The following operations should be supported with their order (operator precedence):

```
    Pi       Pi number
    e        Euler's number
    sqrt()   Square root
    cos()    Cosine (using radians as an argument)
    sin()    Sine (using radians as an argument)
    tan()    Tangent (using radians as an argument)
    lg()     Decimal logarithm
    ln()     Natural logarithm
1   ()       Brackets
2   ||       Absolute value, e.g. |-5 - 10|
3   !        Factorial
4   -        Unary minus
5   ^        Exponent
6   mod      Modulus divide, e.g. 5 mod 2 = 1 (only integers will be supplied here)
7   *, /     Multiply, Divide (left-to-right precedence)
8   +, -     Add, Subtract (left-to-right precedence)
```

INPUT SAMPLE:

Your program should accept as its first argument a path to a filename. The input file contains several lines. Each line is one test case. Each line contains mathematical expression. E.g.

```
250*14.3
3^6 / 117
(2.16 - 48.34)^-1
(59 - 15 + 3*6)/21
lg(10) + ln(e)
15*5 mod 2
```

OUTPUT SAMPLE:

For each set of input produce a single line of output which is the result of calculation.

```
3575
6.23077
−0.02165
2.95238
2
15
```

Note: Don't use any kind of **eval** function.

Constraints:
Each number in input expression is greater than -20,000 and less than 20,000.
Each output number is greater than -20,000 and less than 20,000.
If output number is a float number it should be rounded to the 5th digit after the dot.
E.g 14.132646 gets 14.13265, 14.132644 gets 14.13264, 14.132645 gets 14.13265.

If output number has less than 5 digits after the dot you don't need to add zeros.
E.g. you need to print 16.34 (and not 16.34000) in case the answer is 16.34.
And you need to print 16 (and not 16.00000) in case the answer is 16.