CS2810 (OOAIA) : Assignment #5

IIT Madras 13-Feb-2017

Bigg Integers!

At his workplace, Mr X needs to deal with huge numbers on a daily basis. Unfortunately, the native **int** type in C++ cannot store integers which have more than 10 decimal digits. He needs your help to perform basic operations such as those mentioned below on these big integers.

Operators to be Implemented:

- Arithmetic Operators:
  - Addition ’+’ (add)
  - Subtraction ’-’ (sub)
  - Mutliplication ’*’ (mul)
  - Division ’/’ (div)
  - Modulus ’%’ (mod)
  - Factorial ’!’ (fac)
  - Power ’ˆ’ (pow)
- Comparison Operators:
  - Equals ’==’ (eq)
  - Not Equals ’!=’ (neq)
  - Greater than Equals ’>=’ (geq)
  - Less than Equals ’<=’ (leq)

Constraints

- You must handle operations on integers with upto 1000 digits
- For pow operator, base can be negative but the exponent is positive.
- For rest of the operators except ’!’, both numbers can be negative.

Input format

<operator-keyword1> <number1> <number2>
<operator-keyword2> <number1> <number2>
...

Note:
1. Operator keywords are mentioned in parentheses above.
2. For the factorial operator, there will be only one operand.

Output format

<result1>
<result2>
...

Note:

- For comparison operators, output 1 for true, 0 for false.
- In mod operation $a\%b$, if b is negative, result lies in (b,0] otherwise in [0,b).
- In div operation a/b, print the quotient $\lfloor a/b \rfloor$.
- Result of div is in Egyptian fraction expansion's Integer e.g -1/2 => -1 not 0 but 1/2 => 0
- You do not have to print leading 0s in results.

Sample Testcase

Input:

```
add -3245643213456234 23456754324523
mul 35434653434545 234234345345
pow 2938579324832 2
eq 1234567890123 1234567890132
fac 14
```

Output:

```
-3222186459131711
8300012849767603882943025
8635248448330092971828224
0
87178291200
```

Design

We expect you to use operator overloading for all operators mentioned above.

Submission Guidelines

- Submit the design code in Moodle as <roll number>.cpp by 16:40 today. The file must include comments detailing your design decisions.
- Submit the final working code in HackerRank by 23:45 tomorrow. **(No Extension!)**