

Big Integer Operations

locked

Problem

Submissions

Leaderboard

Discussions

In this assignment you should implement a new data-type called Integer that can store integers with arbitrarily large number of digits. You should design such that all the standard operations that can be performed on integers will be supported as detailed below.

An integer of type Integer will have a sign and can have any number of digits in it, say 100 or even 1000. First implement the following operators on the Integer type.

Integer operator+(Integer&); //addition operation

Integer operator-(Integer&); //subtraction operation

Integer operator*(Integer&); //multiplication operation

Integer operator/(Integer&); //division operation Eg: 44/5 (=8.8) must return 8 and not 9.

Integer operator%(Integer&); //remainder operation

Integer& operator+=(Integer&); //addition and assignment

Integer& operator-=(Integer&);

Integer& operator*=(Integer&);

Integer& operator/=(Integer&);

Integer& operator%=(Integer&);

bool operator==(Integer&);

bool operator!=(Integer&);

bool operator<(Integer&);

bool operator<=(Integer&);

bool operator>(Integer&);

bool operator>=(Integer&);

friend ostream& operator<<(ostream&, Integer&);

friend istream& operator>>(istream&, Integer&);

The running time of addition and subtraction operations must be $O(d_1 + d_2)$ where d_1 and d_2 are the number of digits of the two numbers. For multiplication and division, the running time should be $O(d_1 d_2)$.

Input Format

First line contains the operation

Next two line contain the operands

End of the input contain **end** keyword to stop the execution

Constraints

1 <= #digits <= 60

No undefined operations like division by zero or % 0

Output Format

Print the result of each operation in separate lines

Note: All the digits must be printed. You should not use any scientific notation like e, E etc..

Sample Input 0

```

+
74462898442155373703893556749928174163166777
51015202530354045505560657075808590951001057
-
74462898442155373703893556749928174163166777
-51015202530354045505560657075808590951001057
*
-535835925499096664087184089
-99999999999899999
/
-535835925499096664087184089
-99999999999899999
%
-535835925499096664087184089
-99999999999899999
+=
74462898442155373703893556749928174163166777
51015202530354045505560657075808590951001057
-=
74462898442155373703893556749928174163166777
-51015202530354045505560657075808590951001057
*=
-535835925499096664087184089
-99999999999899999
/=
-535835925499096664087184089
-99999999999899999
%=
-535835925499096664087184089
-99999999999899999
==
74462898442155373703893556749928174163166777
74462898442155373703893556749928174163166777
!=
74462898442155373703893556749928174163166777
51015202530354045505560657075808590951001057
<
74462898442155373703893556749928174163166777
51015202530354045505560657075808590951001057
<=
74462898442155373703893556749928174163166777
74462898442155373703893556749928174163166777
>
74462898442155373703893556749928174163166777
51015202530354045505560657075808590951001057
>=
74462898442155373703893556749928174163166777
74462898442155373703893556749928174163166777
end

```

Sample Output 0

```

125478100972509419209454213825736765114167834
125478100972509419209454213825736765114167834
53583592549856082280332573734494617503915911
5358359254
-99632505370943343
125478100972509419209454213825736765114167834
125478100972509419209454213825736765114167834
53583592549856082280332573734494617503915911
5358359254
-99632505370943343
true
true
false
true

```

Sample Input 1

Sample Output 1

3/4

Max Score: 100

Difficulty: Medium

Rate This Challenge:

[More](#)

C++



```
1 #include <iostream>
2
3 using namespace std;
4
5
6 int main() {
7     /* Enter your code here. Read input from STDIN. Print output to STDOUT */
8     return 0;
9 }
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |