All Contests > APL-Lab-03 > Big Integer Operations

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

```
+
744628984421553737038935567499281741631667 77
```

```
true
true
```

Sample Input 1

```
+
123456789123456789123456789
987654321987654321987654321
-
123456789123456789123456789
987654321987654321987654321
*
123456789123456789123456789123456789123456789
987654321987654321987654321987654321987654321
/
987654321987654321987654321987654321987654321
123456789123456789123456789123456789123456789
%
987654321987654321987654321
123456789123456789123456789
+=
123456789123456789123456789
987654321987654321987654321
-=
123456789123456789123456789
987654321987654321987654321
*=
123456789123456789123456789
-987654321987654321987654321
/=
-987654321987654321987654321
123456789123456789123456789
%=
987654321987654321987654321
-123456789123456789123456789
==
999999999999999999999999999
999999999999999999999999999
!=
999999999999999999999999999
999999999999999999999999991
<
-999999999999999999999999999
999999999999999999999999991
<=
999999999999999999999999999
999999999999999999999999991
>
999999999999999999999999999
999999999999999999999999991
>=
999999999999999999999999999
999999999999999999999999991
end
```

Sample Output 1

```
1111111111111111111111111110
-864197532864197532864197532
121932631356500531591068431825636332060204232050906869816338969581771069347203169112635269
8
9000000009000000009
1111111111111111111111111110
-864197532864197532864197532
-121932631356500531591068431581771069347203169112635269
-8
9000000009000000009
true
true
true
false
true
true
```

Submissions: 100

Max Score: 100
Difficulty: Medium

Rate This Challenge:
☆ ☆ ☆ ☆ ☆

More

C++

```
#include <iostream>

using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File □ Test against custom input

Run Code Submit Code