Lab7

It is well-known that there is a strict limit for type int, long or long long. We have Bignum in JAVA, however, you need a extra bignum library in C++ to achieve the same goal.

In this lab, you are required to write your own Bignum class. After completing your code, you will get a better understanding of operator overloading in C++.

Release

You are required to write all code by yourself. We only provide two files which are sample input and output:

- sample.in
- sample.out

Please notice that we will test your code with other data.

Requirement

You are required to implement following operator:

- +, the sum of two large integer
- -, the difference of two large integer
- >>, input a large integer
- <<, output a large integer

In the meanwhile, your operator should support:

- out-of-range warning, if input integer is too large

And since life is short, you don't have to support:

- expression with parentheses
- operator precedence: in one expression, we have at most *ONE* operator.

Input && Output

Input:

```
"sample.in"
first line: n <= 100, the number of expressions
following: n lines, with one expression in one line (length of large
integer l <= 100)
```

Output:

```
"sample.out"
n lines, with one answer in one line
```

Expressions please refer to released files.

Score

Item	Score	Num	Total
simple testcase	5	6	30
hard testcase	10	4	40
programming style	30	1	30
Sum	-	-	100

Notice:

- we test your code *AUTOMATICALLY*, please use *FILE* intput and output !!! Any problems with input and output filename lead

you to 50% of your score. And any other problems lead you to 0 score.

- 0 score for plagiarist!!!
- 10%/day penalty for any delay, 0 score for more than 3-day delay
- The large integers in sample.in are all positive. But the calculation result in sample.out may be negative

Handin

Name your directory as "studentid_name" like "11300180158_张三", containing:

- Lab7.h
- Lab7.cpp

Be sure all your files are encoding in UTF-8 !