Project 1 (18%)

Class BigInt

You have to create class BigInt representing signed integer numbers of arbitrary length. The length of these numbers must be limited only by the size of operative memory. The inner representation of such a number is a vector of digits. Use std:;vector from C++ standard library. You have to use regular (school) algorithms for arithmetic operations (addition, subtraction, multiplication, division, remainder). The class has to be tested with units-test library CATCH2.

Your class BigInt should be similar to built-in primitive type int. It means that you have to implement following operations:

- Input-output operators (<<, >>)
- Constructors: default constructor, constructor with int parameter, constructor with string representation of integer number. Last constructor should throw runtime_error exception if the string argument is invalid.
- Operators: +, -, *, /, %, += , -=, *=, /=, %=, prefix and postfix ++ and --, unary -, unary +. In the case of division by zero you have to throw runtime_error exception. Check how our g++ compiler (in C++11 mode) works with expressions like 3 / -2 or -5 % 10 and emulate this behavior in your class. Multiplication and division by numbers of type int can be implemented more efficiently and should be overloaded in your class.
- Operators: ==, !=, <, >, <=, >=
- function abs

You have to test your class on a set of simple problems from https://informatics.msk.ru/

130, 131, 132, 133, 134, 135, 2798, 136, 137, 138, 139, 140, 141, 142, 143, 145, 615, 621, 627, 633, 639.

Grades:
5%: constructors, input-output operators, addition for non-negative numbers; accepted problems 130, 132, 621

10%: previous requirements + addition and subtraction of signed numbers, increment, decrement operators; accepted problems 133, 134 .

15%: previous requirements + multiplication and comparisons operators; accepted problems 2798, 136, 143, 615, 621

18%: previous requirements + division and remainder operators; all problems must be accepted.