ФПМИ МФТИ, весна 2020
ООП 1 курс, основной поток. Задания по C++

Задача 1. Длинная арифметика

Дедлайн: 23.03.2020 09:00 мск

В этой задаче разрешается подключать <iostream>, <vector> и <string> и только их.

Часть 1

Напишите класс BigInteger для работы с длинными целыми числами. Должны поддерживаться операции:

- сложение, вычитание, умножение, деление, остаток по модулю, работающие так же, как и для int; составное присваивание с этими операциями. Умножение должно работать **за о-малое от n^2.**
- унарный минус, префиксный и постфиксный инкремент и декремент. Префиксный инкремент и декремент должны работать за O(1) в среднем.
- операторы сравнения == != < > <= >=.
- вывод в поток и ввод из потока;
- метод toString(), возвращающий строковое представление числа;
- конструирование из int (в том числе неявное преобразование, когда это надо);
- преобразование в bool, когда это надо (должно работать в условных выражениях).
- опционально - литеральный суффикс bi для написания литералов типа BigInteger, см. справку здесь https://en.cppreference.com/w/cpp/language/user_literal

В вашем файле должна отсутствовать функция main(), а сам файл должен называться biginteger.cpp. Ваш код будет вставлен посредством #include в программу, содержащую тесты.

Часть 2

Используя класс BigInteger, напишите класс Rational для работы с рациональными числами сколь угодно высокой точности. Числа Rational должны представляться в виде несократимых обыкновенных дробей, где числитель и знаменатель – сколь угодно длинные целые числа. Должны поддерживаться операции:

- конструктор из BigInteger, из int;
- сложение, вычитание, умножение, деление, составное присваивание с этими операциями; унарный минус;
- операторы сравнения == != < > <= >=
- метод toString(), возвращающий строковое представление числа (вида [минус]числитель/знаменатель), где числитель и знаменатель - взаимно простые числа; если число на самом деле целое, то знаменатель выводить не надо;
- метод asDecimal(size_t precision=0), возвращающий строковое представление числа в виде десятичной дроби с precision знаками после запятой;
- оператор приведения к double.

В вашем файле должна отсутствовать функция main(), а сам файл должен называться rational.cpp. Ваш код будет вставлен посредством #include в программу, содержащую тесты.

Задачи 2, 3, 4 будут еще добавлены