

C++ Operator Overloading

Реализирайте клас `BigNumber`, който поддържа работа с целочислени стойности, като **няма ограничение в тяхната дължина** и съответно минимална и максимална стойност.

1. Да се дефинира допълнителен конструктор, който приема като параметър целочислено число и го конвертира към `BigNumber`.
2. Да се поддържат и отрицателни стойности.
3. Да се дефинират и съответните оператори:
 - a. `>`, `==`, `<`, `>=`, `<=`, `!=` сравнение между две `BigNumber` числа.
 - b. `>`, `==`, `<`, `>=`, `<=`, `!=` сравнение между `BigNumber` и `integer` число.
 - c. `+` и `+=` с втори параметър `BigNumber`.
 - d. `+` и `+=` между `BigNumber` и `integer` число, съответно и операторите, които са семантично възможни.
 - e. Аналогични оператори `-`, `-=`, `/` (целочислено делене), `/=`, `*`, `*=`, `%` (остатък при целочислено делене), `%=`
 - f. Оператор `[]`, който връща цифрата на съответната позиция, като на нулева позиция се намират единиците на съответното число, на първа позиция се намират десетиците на съответното число и т.н.
В случай, че няма такава цифра, да се хвърли коректно изключение.
 - g. Входно-изходни операции с потоци.

Покажете в кратка програма как работи класа, който сте реализирали.