Лабораторная работа №6 по курсу Дискретного Анализа: Калькулятор

Выполнил студент группы 08-308 МАИ Жерлыгин Максим Андреевич

Условие

Необходимо разработать программную библиотеку на языке C или C++, реализующую простейшие арифметические действия и проверку условий над целыми неотрицательными числами. На основании этой библиотеки нужно составить программу, выполняющую вычисления над парами десятичных чисел и выводящую результат на стандартный файл вывода.

Список арифметических операций:

Сложение (+).

Вычитание (-).

Умножение (*).

Возведение в степень (^).

Деление (/).

В случае возникновения переполнения в результате вычислений, попытки вычесть из меньшего числа большее, деления на ноль или возведении нуля в нулевую степень, программа должна вывести на экран строку Error.

Список условий:

Больше (>).

Меньше (<).

Равно (=).

В случае выполнения условия программа должна вывести на экран строку true, в противном случае — false.

Количество десятичных разрядов целых чисел не превышает 100000. Основание выбранной системы счисления для внутреннего представления «длинных» чисел должно

быть не меньше 10000.

Числа, поступающие на вход программе, могут иметь «ведущие» нули.

Формат входных данных

Входной файл состоит из последовательности заданий, каждое задание состоит из трех строк:
Первый операнд операции.
Второй операнд операции.
Символ арифметической операции или проверки условия (+, -, *, ; /, >, <, =).
Числа, поступающие на вход программе, могут иметь «ведущие» нули.

Формат результата

Для каждого задания из выходного файла нужно распечатать результат на отдельной строке в выходном файле:

Числовой результат для арифметических операций.

Строку Error в случае возникновения ошибки при выполнении арифметической операции.

Строку true или false при выполнении проверки условия.

В выходных данных вывод чисел должен быть нормализован, то есть не содержать в себе «ведущих» нулей.

Метод решения

Длинная арифметика это способ представления чисел в виде строк. Она необходима, когда дело доходит до очень больших чисел, которые не помещаются в самые большие типы. Для это я реализовал основные операции длинной арифметики с длиными числами: сложение, вычитание, умножение, деление и возведение в степень.

Исходный код

Я описал реализацию длинного числа, которая содержится в файле TBigInt.h,

```
1 class TBigInt {
```

```
public:
    TBigInt() {};
    TBigInt(const std::string &str);
    TBigInt(const size_t &size);
    TBigInt(int size);
    ~TBigInt() {};

    TBigInt operator+(const TBigInt &obj);
    TBigInt operator-(const TBigInt &obj);
    TBigInt operator*(const TBigInt &obj);
    TBigInt operator/(TBigInt &obj);
    TBigInt Power(int deg);
    bool operator<(const TBigInt &obj) const;
    bool operator>(const TBigInt &obj) const;
    bool operator==(const TBigInt &obj) const;

    friend std::ostream& operator<< (std::ostream& os, const TBigInt &
        obj);

    inline size_t size() const
    {
        return mData.size();
    }
    int Get(size_t i) const
    {
        return mData[i];
    };
    void Set(size_t i, int num)
    {
        mData[i] = num;
    };
private:
    std::vector<int> mData;
    void DeleteZeros();
};
```

а также реализовал все методы класса TBigInt в файле TBigInt.cpp.

На вход программе подается два образца (числа) и оператор. Программа считывает оба образца как строки преобразуя их в тип int и помещаются в vector. Затем, в зависимости от оператора, вызывается определнная функция и производится вычисление. Главное условие такого, что отрицательных чисел нет, то есть мы не можем вычесть и меньшего числа большее, если же мы захотим попробовать этот тест программа должна выводить $ERROR$.

Пример работы

```
mmaxim2710@DESKTOP-RDPBU3D:/mnt/c/Users/mmaxi/Desktop/coursera/DA_ex/lab6$ make clean
rm -f *.o solution
mmaxim2710@DESKTOP-RDPBU3D:/mnt/c/Users/mmaxi/Desktop/coursera/DA_ex/lab6$ make
g++ -std=c++17 -pedantic -Wall -O2 -c main.cpp -o main.o
g++ -std=c++17 -pedantic -Wall -O2 -c TBigInt.cpp -o TBigInt.o
g++ -std=c++17 -pedantic -Wall -O2 main.o TBigInt.o -o solution
mmaxim2710@DESKTOP-RDPBU3D:/mnt/c/Users/mmaxi/Desktop/coursera/DA_ex/lab6$ ./solution
38943432983521435346436
354353254328383
+
3894343333787468967 4819
90409438473849324729384738 43
2343543
-
90409438473849324729361303 00
972323
2173937
>
false
2
3
-
Error
52352352342423
52342346346456456456
*
27402449583588256240498437010328 88
27402449583588256240498437010328 88
52342346346456456456
/
52352352342423
52352367867867878
2
^
27407704213725650663291056642228 84
90409438473849324729384738 43
90409438473849324729384738 43
=
true
```

Вывод

Длинная арифметика - способ реализации работы с неограниченно большими числами. Однако она мало где реализована на уровне языка. Например, в Python длинные числа и длинная арифметика поддерживается на языковом уровне. В том же C++ такой поддержки нет. Возможно, это связано с узкой сферой применения. Большинство решаемых задач укладываются в стандартные типы данных.

Одна из областей применения длинной арифметики - криптография. Большинство систем подписывания и шифрования данных используют целочисленную арифметику по модулю m, где m — очень большое натуральное число, не обязательно простое.

Из того, что можно добавить: извлечение квадратного корня алгоритмом «Karatsuba Square Root» (сложность $5O(n)$), взятие остатка.