**Отчет**

**Лабораторная работа №1**

**Дисциплина:** Типы и структуры данных

Выполнила: Овчинникова Анастасия

Группа: ИУ7-35

Вариант 19

**Техническое задание**

**Применение.** В настоящее время «длинная арифметика» применяется в криптографии, при разработке математического и финансового ПО, а также при составлении кода для процессоров низкой разрядности.

**Срок выполнения**: три недели.

**Задача:** Смоделировать операцию деления действительного числа на действительное число в форме ±m.n Е ±K, где суммарная длина мантиссы (m+n) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Входные данные.** строка, содержащая действительное число в формате [-]m.n e [-]k. Длина мантиссы m+n не превышает 30 разрядов, порядок имеет не больше 5 разрядов. Допускается отсутствие точки, “e”, знака “-“. Знак “+” использовать нельзя. Возможны следующие варианты представления чисел:

* .00025
* 123001.
* [ –]123.456
* 1234567e[–]20
* [-]12.3e[-]123
* .123e123
* 0.123e123

**Выходные данные.** Результат при выдаче на печать должен быть нормализован в виде: знак 0.мантисса E знак порядок. Длина мантиссы не превышает 30 разрядов, порядок имеет не больше 5 разрядов.

**Функция программы.** Деление действительного числа на действительное число; если мантисса частного выходит за 30 значащих цифр, произвести округление.

**Способ обращения к программе.** Запуск проекта DivideBigRealNumbers в qtcreator или через командную строку файла main.exe, параметры командной строки не нужны.

**Аварийные ситуации**

1. Пользователь ввел число, порядок которого больше 99999.

Выход: “Слишком большой порядок”.

1. Пользователь ввел число, порядок которого меньше -99999.

Выход: “Слишком маленький порядок”.

1. Некорректный ввод пользователем строки, содержащей лишние буквы или цифры.

Выход: “Некорректный ввод”.

1. Ошибка при вычислениях при переполнении порядка.

Выход: “Переполнение порядка в результате вычислений”.

1. Деление на ноль.

Выход: «Деление на ноль».

**Описание внутренних структур данных.**

Для представления в памяти числа используется структура:

*struct BigReal*

*{*

*vector<short> mantissa;*

*long long int order;*

*short mantissaSign;*

*};*

Для хранения мантиссы используется вектор. Выбор обусловлен удобством использования векторов в C++, возможность легко удалить или добавить новый элемент в любое место вектора. Знак мантиссы хранится в знаковой целочисленной переменной.

Для хранения порядка используется тип длинного целого числа со знаком, так как диапазон значений типа позволяет хранить число, соответствующее ограничениям, прописанным в задаче.

Результат также хранится в структуре *BigReal.*

**Описание алгоритма.**

Если ввод корректный

Занесение мантиссы в вектор и порядка в переменную, запоминание знака;

Вычисление частного от деления (Функция divide);

Если степень переполнена:

Вывод сообщения;

Иначе:

Вывод ответа на экран;

Иначе:

Вывод соответствующей ошибки.

Алгоритм деления:

Вычисляем знак мантиссы.

Вычисляем порядок.

Делим мантиссы.

Округление.

Проверка на переполнение порядка.

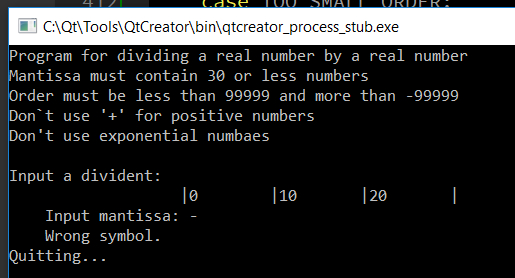
Деление мантисс происходит следующим образом. Например 123456789 поделим на 10. Число 10 будет расширено до 100 000 000.  
Далее вычитаем из 123456789 число 100 000 000, пока 123456789 > 100000000. Вычитание произошло один раз, записываем это в res. res = 1. Далее 23456789 - 10 000 000. Операция прошла 2 раза, запишем в res = 12. 3456789 - 1\_000\_000, 3 раза => res = 123… 89 - 10, 8 раз => res = 12345678. Далее, если в результате такого деления остался остаток, приписываем к остатку ноль, и повторяем всю описанную операцию до тех пор, пока результат будет содержать менее 30 символов, либо пока делимое не станет равно нулю.

Для осуществления округления в случае, если длина частного превышает 30 знаков, при делении мантисс вычисляется 31 знак. Далее, в зависимости от значения 31 знака происходит округление, если необходимо, и 31 знак отбрасывается.

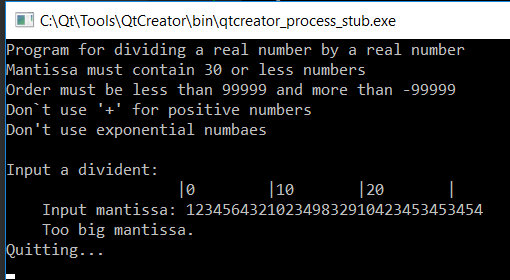
**Набор тестов.**

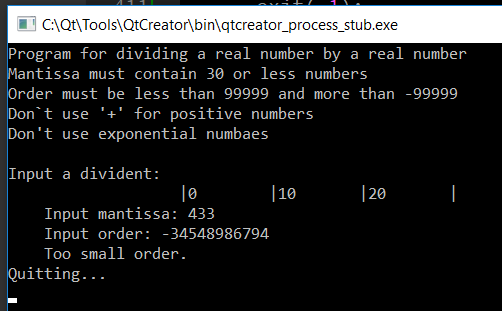
**Аварийные ситуации.**

Некорректный ввод:

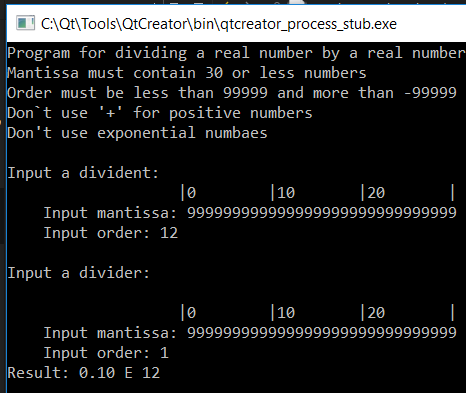


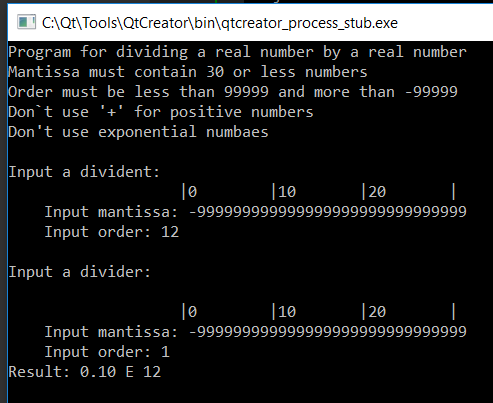
Переполнение при вводе:

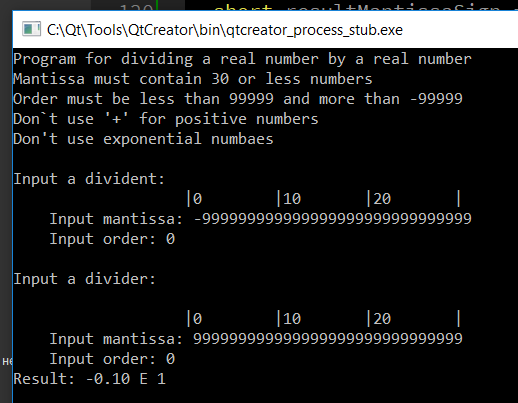




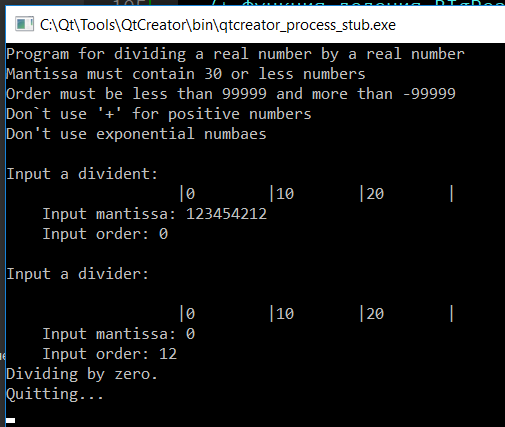
**Проверка граничных значений.**

****

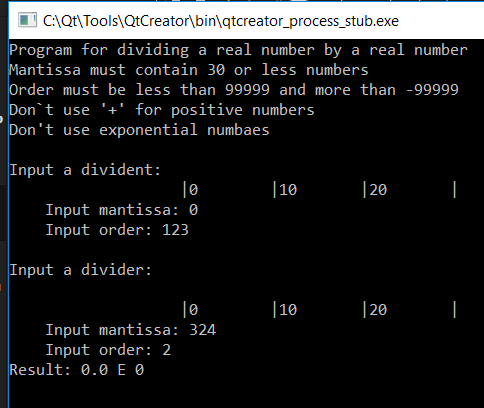
****

****

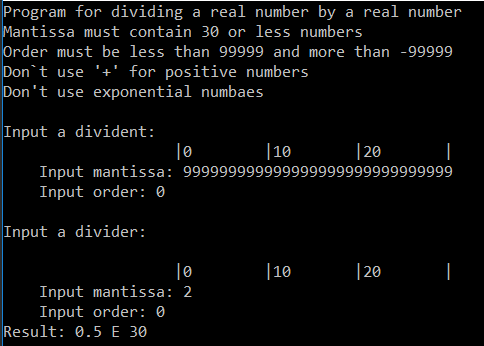
Деление на ноль:

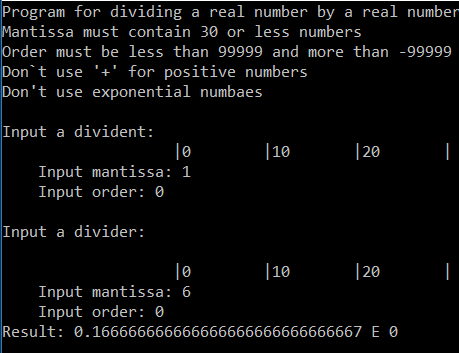


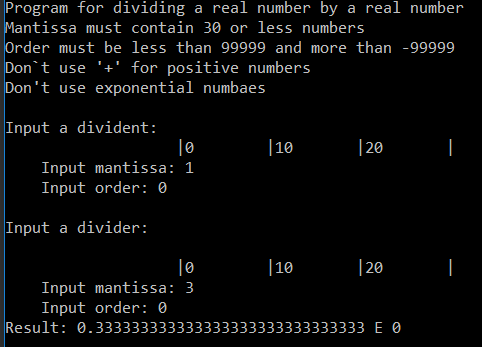
Деление нуля:

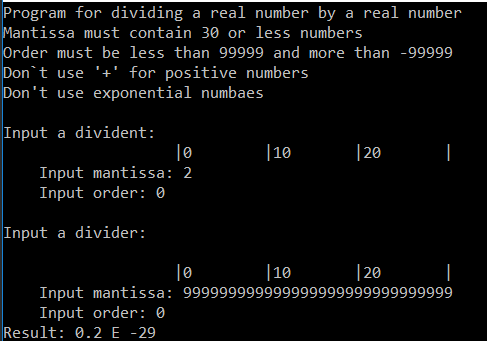


Округление:

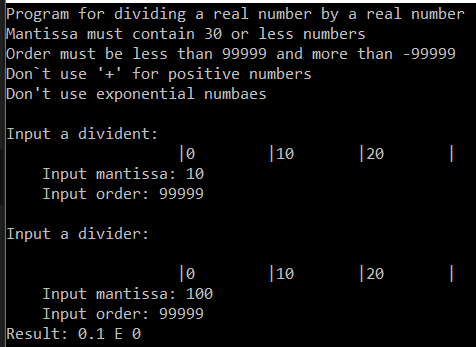


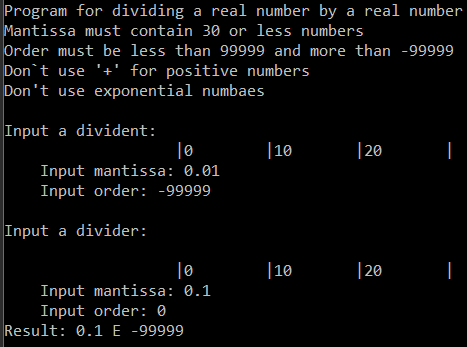






Переполнение порядка:





**Контрольные вопросы**

1. *Каков возможный диапазон чисел, представляемых в ПК?*

Возможный диапазон чисел, представляемых в ПК, зависит от размера области памяти, выделяемой под хранение переменной, наличия знака в числе и от типа представления числа (целое или вещественное). Например, для 64-х разрядного процессора под хранение целого числа выделяется 64 бита. Для 64 битов максимальное значение числа равно 264-1 = 18 446 744 073 709 551 615. То есть для 64-х разрядного процессора возможно использовать не более 20 десятичных разрядов для представления числа.

1. *Какова возможная точность представления чисел, чем она определяется?*

Максимальная возможная точность представления числа определяется максимальной длиной мантиссы. Максимальная длина мантиссы зависит от размера области памяти, выделяемой под хранение переменной и наличия знака в числе. Как уже было сказано, для 64-х разрядного процессора возможно использовать не более 20 десятичных разрядов для представления числа. При этом, если длина мантиссы выходит за границы разрядной сетки, происходит ее округление.

1. *Какие стандартные операции возможны над числами?*

Стандартные операции над числами: сложение, вычитание, умножение, деление, сравнение.

1. *Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?*

Наиболее предпочтительным типом для этого является массив, например, массив символов – для ввода числа, числовой массив – для обработки и вывода. Кроме того, можно использовать, например, структуру, содержащую массив цифр мантиссы, массив цифр порядка, а также две переменных для хранения знаков мантиссы и порядка.

1. *Как можно осуществить операции над числами, выходящими за рамки машинного представления?*

Операции над числами, выходящими за рамки машинного представления, можно осуществлять путем последовательного выполнения операций над всеми цифрами данного числа, то есть пользуясь стандартными алгоритмами арифметических операций, таких как умножение и деление в столбик.

**Вывод.**

В результате проделанной работы было выявлено, что длинные действительные числа лучше всего хранить в структурах, какие были описаны выше, так как данный вариант представления данных является наиболее простым и удобным, что позволяет выполнять меньше операций для доступа к длинным действительным числам и выполнения арифметических операций с ними.