

МГТУ им. Н.Э. Баумана

**Дисциплина «Типы и структуры данных»**  
**Лабораторный практикум №1**  
**по теме: «Длинная арифметика: обработка больших чисел»**

Работу выполнил:  
студент группы ИУ7-35Б  
Клименко А. К.

Работу проверила:  
Барышникова М. Ю.

Москва, 2020 г.

## Цель работы

научиться правильно выбирать структуры данных для решения конкретных задач, обосновывать свой выбор. Научиться программировать алгоритмы, на основе выбранных структур данных.

## Описание условия задачи

Необходимо составить программу, выполняющую операцию деления двух введенных пользователем с клавиатуры вещественных чисел, с последующим выводом результата на экран.

При невозможности произвести вычисление, программа должна выдавать информативное сообщение об ошибке.

## Техническое задание

### Исходные данные

Оба вводимых числа являются действительными и вводятся в любой из нижеперечисленных форм, где суммарная длина мантиссы ( **$m+n$** ) - **до 30 значащих десятичных цифр**, а величина порядка  **$K$**  - **до 5 десятичных цифр**.

Формы вводимого числа:

**$[\pm]m[. [n]][e[\pm]K]$**

**$[\pm].n[e[\pm]K]$**

Квадратные скобки [...] обозначают опциональность данной конструкции (можно опустить при вводе). Символ  $\pm$  означает, что на этом месте может быть либо плюс, либо минус. В любой форме допустимо использовать символ экспоненты в произвольном регистре.

### Результат

Результат (частное) выводится в форме  **$\pm 0.pE\pm L$** , где  **$p$**  - **до 30 значащих десятичных цифр**, а  **$L$**  - **до 5 десятичных цифр**.

При возникновении аварийных ситуаций результатом работы программы является сообщение об ошибке.

## Описание задачи, реализуемой программой

Программа при запуске запрашивает у пользователя ввести число в любом из предложенных форматов. После того, как пользователь вводит число, оно сохраняется в строку и далее обрабатывается и преобразуется в структуру, используемую для представления больших вещественных чисел. Затем описанный процесс ввода повторяется для

второго числа, после чего происходит выполнение основного алгоритма деления двух вещественных чисел с последующим выводом результата на экран консоли.

## Способы обращения к программе

Вызов программы происходит любым возможным способом запуска \*.exe программ. (Например, через консоль). Для корректной работы программы при её вызове не требуются дополнительные аргументы командной строки.

## Возможные аварийные ситуации и ошибки пользователя

При не соответствии вводимого числа формату ввода выполнение программы останавливается с предварительным оповещением пользователя о том, в каком месте была допущена ошибка при вводе.

При невозможности выполнить деление введенных чисел без потери данных (переполнении) выполнение программы останавливается и выводится соответствующее сообщение об ошибке.

## Структуры данных

Для хранения больших вещественных чисел, была выбрана структура, содержащая в качестве своих полей знак мантиссы (1 байт), модуль самой мантиссы в виде массива из 15 байт (по паре десятичных чисел в каждом байте), и степень экспоненты (целое знаковое 4 байта).

Выбор такой структуры был мотивирован балансом между компактностью хранения данных и удобством их обработки (как будет видно далее). Вся структура занимает всего **20 байт**.

```
10  enum big_float_sign_enum
11  {
12      ... POSITIVE = 1,
13      ... NEGATIVE = -1
14  };
15
16  typedef struct big_float_t
17  {
18      ... char mantissa_sign; // POSITIVE or NEGATIVE
19      ... unsigned char mantissa[15]; // decimal digit pairs
20      ... int power;
21  } big_float_t;
```

## Множество допустимых значений

Данная СД используется для хранения вещественных чисел в интервале  $[-0.9...9e-99999; +0.9...9e+99999]$  с точностью до **30 значащих цифр**.

## Набор допустимых операций

Список операций необходимых для работы с данной СД:

1. Обнуление числовой структуры.
2. Считывание числовой структуры в установленном формате из строки.
3. Конвертирование числовой структуры обратно в строку для дальнейшего вывода.
4. Проверки на определённые значения структуры (0, 1).
5. Сравнение мантисс двух числовых структур.
6. Умножение мантиссы числа на десятичное число (от 1 до 10)
7. Деление одной числовой структуры на другую.

## Описание алгоритма обработки данных

Работа алгоритма деления двух чисел, представленных с помощью описанной выше структуры, сводится к следующей последовательности действий:

1. Проверка на равенство делителя нулю. (ошибка деления на ноль)
2. Проверка на равенство делимого нулю. (возврат ответа без дальнейших расчётов)
3. Проверка на возможное переполнение при делении (ошибка переполнения)
4. Проверка на равенство делителя единице (возврат делимого без изменения)
5. Вычисление знака результирующего числа.
6. Вычисление степени результирующего числа. (учитывая возможные переносы при делении мантисс)
7. Деление мантисс чисел методом «в столбик».
8. Округление результирующей мантиссы.

## Набор тестов

№	Описание теста	Входные данные	Выходные данные
1	Базовый случай	1 2	+0.5E+0
2	"-" на "+"	-4e0 2.	-0.2E+1
3	"+" на "-"	6.0 -004	-0.15E+1
4	"-" на "-"	-800e-1 -0.64e2	+0.125E+1
5	Деление нуля	0 -.21300	+0.E+0
6	Деление на ноль	7 0	Деление на ноль не имеет смысла.
7	Деление нуля на ноль	-0.0e0 +0000.0	Деление на ноль не имеет смысла.
8	Деление равных	3.1415 0.314150e+1	+0.1E+1
9	Деление на единицу	77.56 1	0.7756E+2
10	Несоответствие формату ввода	1300,99	Неожиданный символ в числе.
11	Превышение кол-ва цифр в мантиссе	123456789012345678 9012345678901	Суммарная длина мантиссы не должна превышать 30 цифр.
12	Превышение кол-ва цифр в степени	1.0e100000	Суммарная длина степени не должна превышать 5 цифр.
13	Пустой ввод		Вы не ввели числа.

## Выводы по проделанной работе

В ходе работы я изучил способ представления больших чисел в программе с помощью структуры данных и понял, как большие числа могут быть представлены в программе. А также запрограммировал алгоритм деления двух больших чисел.

## Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон числа зависит от его представления (типа). Для целочисленных знаковых типов диапазон есть отрезок от  $-2^{(n-1)}$  до  $+2^{(n-1)}-1$ , где  $n$  - количество бит отведённых под хранение данного типа.

Для вещественных чисел диапазон преимущественно определяется количеством бит отведённых под хранение степени экспоненты. Для 32-битных чисел (float, single...) диапазоном является промежуток от  $1.2e-38$  до  $3.4e+38$ , а для 64-битных чисел (double...) диапазон — от  $2.3e-308$  до  $1.7e+308$ .

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

Точность представления вещественного числа определяется его мантиссой. Для 32-битных чисел (float, single...) - точность сопоставима с 7 значащими десятичными цифрами, а для 64-битных — с 15 значащими десятичными цифрами.

*3. Какие стандартные операции возможны над числами?*

Над числами возможны базовые арифметические операции, такие как сложение, вычитание, умножение, деление.

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

Программист вынужден создать собственный тип, подходящий под конкретную задачу, или найти и использовать открытую библиотеку, которая выполняет обработку больших чисел.

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

Можно разбить число на группы разрядов и хранить эти группы как отдельные числа в массиве. Обработка такого массива должна выполняться последовательно от элемента соответствующего группе с наименьшими разрядами до элемента соответствующего группе с наибольшими разрядами. При этом важно учитывать возможные переполнения при обработке.