

	<p>Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)</p>
---	--

ФАКУЛЬТЕТ _____ Информатика и системы управления (ИУ) _____

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии (ИУ7) _____

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 **«ОБРАБОТКА БОЛЬШИХ ЧИСЕЛ»**

Студент, группа

Кононенко С., ИУ7-33Б

2019 г.

Описание условия задачи

Смоделировать операцию деления целого числа длиной до 30 десятичных цифр на действительное число в форме $m.n \text{ E } K$, где суммарная длина мантиссы ($m+n$) - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $0.m1 \text{ E } K1$, где $m1$ - до 30 значащих цифр, а $K1$ - до 5 цифр.

Техническое задание

Входные данные:

1. **Целое число:** строка, содержащая вещественное число в виде $[+/-]m$. Суммарная длина модуля числа (m) – до 30 цифр. Наличие знака перед числом обязательно.
2. **Действительное число:** строка, содержащая вещественное число в виде $[+/-]m[.n][e[+/-]K]$
Суммарная длина мантиссы ($m+n$) – до 31 цифры (считая точку), порядка K – до 5 цифр. Наличие точки, знака порядка и знака числа обязательно. Наличие знака экспоненты e/E необязательно.

Выходные данные:

- длинное число, нормализованное в виде $[-]0.m1 \text{ e}[-]K1$, где длина $m1$ – до 30 цифр, $K1$ – до 5.

Функция программы: деление целого числа на действительное.

Обращение к программе: запускается из терминала.

Аварийные ситуации:

1. Некорректный ввод строки с целым числом.
На входе: строка, хотя бы один символ в которой не цифра и не символ $+/-$ (если это первый элемент в строке).
На выходе: сообщение «Ошибка ввода! Соблюдайте правила ввода!» или сообщение «Ошибка типа! Введено действительное число!» (если обнаружена точка или знак экспоненты)
2. Некорректный ввод строки с дробным числом.
На входе: строка, хотя бы один символ в которой не цифра и не символ из набора $+ - . e E$.
На выходе: сообщение «Ошибка ввода! Соблюдайте правила ввода!» или сообщение «Ошибка типа! Введено целое число!» (если не обнаружено ни точки, ни экспоненты).
3. Превышение длины строки при вводе целого числа.
На входе: корректное число, длина которого превышает 30 цифр.
На выходе: сообщение «Ошибка ввода! Соблюдайте правила ввода!»
4. Превышение длины строки при вводе дробного числа.
На входе: корректное число, длина мантиссы которого превышает 31

- цифру (считая точку) или длина порядка превышает 5 цифр.
На выходе: сообщение «Ошибка ввода! Соблюдайте правила ввода!»
5. Переполнение порядка.
На входе: в процессе деления степень полученного в результате числа превышает 99999.
На выходе: сообщение «Ошибка! Переполнение порядка!»
6. Деление на ноль.
На входе: дробное число, равное 0.
На выходе: сообщение «Ошибка! Деление на ноль!»

Структуры данных

Для изначального хранения введенного числа используется именованный тип данных **long_t**, который представляет собой символьный массив. Он описывается как:

```
typedef char long_t[MAX_LONG_T_NUM_LEN],
```

где **MAX_LONG_T_NUM_LEN** = 40 (с учетом всех служебных знаков).

После ввода числа, оно обрабатывается и разбивается на части, которые описываются в структуре **num_t**. Он описывается как:

```
typedef struct
{
    char num_sign;
    char mantissa_part[MAX_MANTISSA_PART_LEN * 2];
    char order_part[MAX_ORDER_PART_LEN];
    short int dot_position;
    short int exp_position;
    int order_int;
} num_t;
```

Поля структуры:

- **num_sign** – знак числа. Принимает значения “+” или “-”;
- **mantissa_part**[MAX_MANTISSA_PART_LEN * 2] – значение мантиссы числа (часть между знаком числа и знаком экспоненты), **MAX_MANTISSA_PART_LEN** = 32 (учитывается символ точки и терминального нуля);
- **order_part**[MAX_ORDER_PART_LEN] – значение порядка числа (часть после знака экспоненты), **MAX_ORDER_PART_LEN** = 7 (учитывается знак экспоненты и знак терминального нуля);
- **dot_position** – позиция десятичной точки, нумерация начинается с начала мантиссы (принимает значение -2, если точки в числе нет);

- **exp_position** – позиция знака экспоненты, нумерация начинается с начала числа (принимает значение -1, если экспоненты в числе нет);
- **order_int** – целочисленное представление числового порядка.

Алгоритм

1. На вход программе подаётся вдве строки символов, сначала содержащая целое число, затем содержащая вещественное число.
2. Выполняется запись полученных чисел в структуры типа **num_t**, затем выполняется проверка данных на валидность.
3. Если второе число равно нулю, то пользователю выводится соответствующее сообщение об ошибке.
4. Если первое число равно нулю, при этом второе не равно нулю, то результат принимается равным за 0.
5. Выполняется деление первого числа на второе методом “деления в столбик”, при этом контролируется округление.
6. При переполнении порядка пользователю выводится соответствующее сообщение об ошибке.
7. Если результат деления корректен, то он выводится пользователю в нормализованном виде в соответствии со спецификацией, указанной в ТЗ.

При успешном делении число выводится согласно спецификации, описанной в ТЗ.

Тесты

	Тест	Число 1	Число 2	Результат
1	Некорректный ввод	q	-	Ошибка ввода! Соблюдайте правила ввода!
2	Некорректный ввод	+123.45	-	Ошибка ввода! Соблюдайте правила ввода!
3	Некорректный ввод	+123	+1.23.45	Ошибка ввода! Соблюдайте правила ввода!
4	Некорректный ввод	+123	+123E+45.67	Ошибка ввода! Соблюдайте правила ввода!
5	Некорректный ввод	+123	+E123	Ошибка ввода! Соблюдайте правила ввода!

6	Превышение длины мантиссы	+999...999 (31 девятка)	-	Ошибка ввода! Соблюдайте правила ввода!
7	Превышение длины мантиссы	+123	+0.999...999 (31 девятка)	Ошибка ввода! Соблюдайте правила ввода!
8	Превышение длины порядка	+123	+0.1E+123456	Ошибка ввода! Соблюдайте правила ввода!
9	Деление нуля	+0	+123.	+0.0E+0
10	Деление на ноль	+123	+0.	Ошибка! Деление на ноль!
11	Деление на ноль	+0	+0.	Ошибка! Деление на ноль!
12	Округление	+2	+0.9999	+0.(2000)E+0
13	Округление	+2	+3.	+0.666...667E+0
14	Округление	+99999	+2	+0.5E+5
15	Переполнение порядка	+10000	+1E-99999	Ошибка! Переполнение порядка!
16	Деление целых чисел	+100	+5.	+0.2E+2
17	Деление чисел разных знаков	+5	-1E+5	-0.5E-4
18	Граничные значение	+999...999 (30 девяток)	+999...999.E+99999 (30 девяток)	+0.1E-99998
19	Граничные значения	+99..99	+2.	+0.5E+30
20	Граничные значения	+1000	+10E+99999	+0.1E-99997

Контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Возможный диапазон чисел зависит от их типа, размера выделенной для их хранения памяти, разрядности процессора. Для беззнакового целого числа выделяется 64 двоичных разряда, то есть его максимальное значение – 18 446 744 073 709 551 615 (long long unsigned int).

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

Точность представления вещественных чисел зависит от количества памяти, выделенного для хранения мантиссы. Для мантиссы типа double выделяется 52 бита, то есть мантисса может принимать значения до 4 503 599 627 370 496.

3. Какие стандартные операции возможны над числами?

Предусмотрено выполнение сравнения, сложения, вычитания, умножения, деления, взятия остатка от деления.

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

Программист может выбрать массив символов для представления длинных чисел, а также структуру, в которую можно включить не только мантиссу числа, но и порядок со знаком.

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

Для этого можно использовать специальные библиотеки, а также самостоятельно разработанные решения.

Вывод

При выполнении лабораторной работы я познакомился с представлением чисел в памяти компьютера, а также сам реализовал обработку и операцию деления длинных чисел. При работе с длинной арифметикой эффективно использовать массив цифр числа, представленных в виде символьного типа.