



Министерство науки и высшего образования 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)

ФАКУЛЬТЕТ
КАФЕДРА

«Информатика и системы управления»
«Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТ № 1 **«ДЛИННАЯ АРИФМЕТИКА»**

Студент
Группа

Байрамгалин Ярослав Ринатович
ИУ7-33Б

1. Описание условия задачи

Составить программу, которая реализует операцию деления целого числа длиной до 30 десятичных цифр на действительное число в форме

$$\pm m.n E \pm K,$$

где суммарная длина мантииссы $m + n$ не превосходит 30 значащих цифр, а величина порядка K содержит не более 5 цифр.

При выводе результата на печать необходимо нормализовать и представить в виде

$$\pm 0.m_1 E \pm K_1,$$

где m_1 - мантисса длиной до 30 значащих цифр, а K_1 содержит не более 5 цифр.

Целое число должно представлять в виде последовательности цифр, без каких-либо посторонних символов, например

$$132, 0, 192, 1234657910.$$

Действительное число может представляться с точкой или без, и экспонентой или без таковой. Возможны следующие варианты записи:

$$.00025, +123001, 1234657e - 20, 123E - 20, 123.45e23.$$

Если при делении мантисса стала занимать больше 30 значащих разрядов, то необходимо выполнить округление (увеличить 30-й разряд на 1, если значение 31-го разряда больше или равно 5).

При разработке интерфейса программы следует предусмотреть:

- вывод на экран операции, производимой программой,
- указание формата и диапазона вводимых данных,
- указание формата выводимых данных,
- наличие пояснений при выводе результатов.

2. Описание технического задания

Входные данные:

Целое число: строка, состоящая из цифр. Допускается указание знака числа на первой позиции строки. Общая длина строки не должна превышать 31 символ

Действительное число: строка, удовлетворяющая следующему регулярному выражению:

$$^{+}[-+]?([0-9]+([.][0-9]*)?([eE][-+]?[0-9]+)?|$$
$$[.][0-9]+([eE][-+]?[0-9]+)?)\$$$

Выходные данные:

Строка, представляющая собой число в нормализованной форме $(\pm 0.m_1 E \pm k_1)$.

Действие программы:

Деление целого числа на вещественное

Обращение к программе:

Запускается командой `./app.exe` через терминал, находясь в директории, содержащей программу.

Аварийные ситуации:

1. Введенное целое число не соответствует формату, указанному во внешней спецификации.
2. Введенное действительное число не соответствует формату, указанному во внешней спецификации.
3. Превышена допустимая длина строки.
4. При вводе действительного числа порядок – число превосходящее по модулю 99999.
5. Ввод пустой.
6. Деление на ноль.

3. Описание структуры данных

Для осуществления операции деления было принято решения хранить как целые, так и действительные числа в одной структуре данных `big_float_t`.

Ниже приведено описание типа `big_float_t`.

```
typedef struct
{
    int sign;
    int digits[MAX_MANTISSA_LNG];
    size_t mantissa_lng;
    int exp_value;
} big_float_t;
```

Поля структуры:

- `sign` – знак мантиссы,
- `digits` – мантисса,
- `mantissa_lng` – длина мантиссы,
- `exp_value` – знаковое значение экспоненты.

Примечание: `MAX_MANTISSA_LNG = 30`

4. Описание алгоритма

1. Программа считывает строку и осуществляет проверку на соответствие введенной строки регулярному выражению числа в экспоненциальной форме. При вводе целого числа также осуществляется проверка этого числа на соответствие регулярному выражению целого числа. При возникновении ошибок возвращается соответствующий код ошибки.
2. Осуществляется перевод строки в `big_float_t`, причем в нормализованный вид.
3. Вызывается функция деления, считается значения экспоненты в результате, затем
 - создается новая строка `cur_mantissa`, хранящая значение мантиссы делителя;
 - пока значение мантиссы делителя меньше значения `cur_mantissa`, к `cur_mantissa` прибавляется мантисса делителя;
 - из мантиссы делимого вычитается `cur_mantissa`;
 - количество итераций, необходимое для того, чтобы значение `cur_mantissa` превысило мантиссу делителя записывается на соответствующую позицию мантиссы результата;операции пункта 3 повторяются пока не будут посчитаны значения всех разрядов мантиссы результата.
4. Выполняется округление полученного числа.
5. Выполняется нормализация полученного числа.
6. На экран печатается результат

5. Набор тестов

[illegible]

13	Получено минимальное допустимое значение	10 100e99999	+0.1e-99999
----	---	---------------------	-------------

6. Ответы на контрольные вопросы

1. Каков возможный диапазон чисел, представляемых в ПК?

Фактически, диапазон ограничен лишь объемом памяти в ПК, однако в «стандартных типах» невозможно представить числа превышающие 2^{64} на 64-х разрядной машине.

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

Точность представления вещественных чисел определяется длиной мантиссы. Для мантиссы типа *double* выделяется 52 бита, значит мантисса может принимать значения вплоть до 2^{52} .

3. Какие стандартные операции возможны над числами?

Сравнение, сложение, вычитание, умножение, деление, взятие остатка от деления.

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

Для представления чисел, диапазон которых превышает возможный диапазон представления в ПК необходимо выбрать «нестандартный» тип данных. Наиболее популярным решением является массив, каждый элемент которого, будет отвечать за соответствующий разряд числа.

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

Необходимо или самостоятельно разработать функции, осуществляющие операции над выбранным типом данных, либо воспользоваться одним из готовых решений.

7. Выводы

1. Для обработки чисел, диапазон которых превышает диапазон стандартных типов необходимо прибегнуть к реализации собственного типа данных, либо же воспользоваться одним из библиотечных решений.
2. Для обработки таких чисел с языке C, в отличие от многих объектно-ориентированных языков программирования, невозможно использовать стандартные операторы для работы с числами (операторы $+$, $-$, $/$, $*$, $\%$, и прочие). Поэтому для реализации арифметических операций необходимо разрабатывать функции, осуществляющие все вышеперечисленные операции.