



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 «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

# ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1 ПО ДИСЦИПЛИНЕ: ТИПЫ И СТРУКТУРЫ ДАННЫХ Длинная арифметика

Студент \_\_\_\_\_ Палладий Е.И.

Группа \_\_\_\_\_ ИУ7-31Б

Название предприятия: НУК ИУ МГТУ им. Н. Э. Баумана

Студент	_____ Палладий Е.И.
Преподаватель	_____ Барышникова М.Ю.

2024 г.

## СОДЕРЖАНИЕ

<b>1</b>	<b>Описание условия задачи . . . . .</b>	<b>2</b>
<b>2</b>	<b>Техническое задание . . . . .</b>	<b>3</b>
	2.1. Исходные данные . . . . .	3
	2.2. Описание задачи, реализуемой программой . . . . .	3
	2.3. Способ обращения к программе . . . . .	3
	2.4. Описание аварийных случаев . . . . .	3
<b>3</b>	<b>Описание структуры данных . . . . .</b>	<b>5</b>
<b>4</b>	<b>Описание алгоритма . . . . .</b>	<b>6</b>
<b>5</b>	<b>Тестовые данные . . . . .</b>	<b>9</b>
<b>6</b>	<b>Ответы на вопросы . . . . .</b>	<b>12</b>
	<b>ЗАКЛЮЧЕНИЕ</b>	<b>14</b>

## 1 Описание условия задачи

Смоделировать операцию деления целого числа длиной до 40 десятичных цифр на действительное число в форме  $\pm m.n \text{ E } K$ , где суммарная длина мантиссы ( $m+n$ ) - до 40 значащих цифр, а величина порядка  $K$  - до 5 цифр. Результат выдать в форме  $\pm 0.m1 \text{ E } \pm K1$ , где  $m1$  - до 40 значащих цифр, а  $K1$  - до 5 цифр.

## **2 Техническое задание**

### **2.1. Исходные данные**

На вход программе подаются два числа, целое и действительное, каждое из которых записано на отдельной строке.

Формат целого числа:  $[+ -][0-9]^+$ . Количество значащих цифр не более 40.

Формат вещественного числа:  $[+ -]?(0.[0-9]^*)E[+ -]?([0-9]^*)$ . Количество значащих цифр не более 40, количество цифр в порядке не более 5

При корректно введенных данных и существовании ответа будет выведено число в формате:  $[+ -]?(0.[0-9]^*)E[+ -]?([0-9]^*)$ . Количество значащих цифр не более 40, количество цифр в порядке не более 5

При некорректно введенных данных будет выведено сообщение об ошибке. При слишком большом числе в ответе будет выведено «Достигнута машинная бесконечность» При слишком маленьком числе в ответе будет выведено: «Достигнут машинный ноль»

### **2.2. Описание задачи, реализуемой программой**

Деление целого числа на действительное число

### **2.3. Способ обращения к программе**

Запускается через терминал: `./app.exe`. Сначала вводится целое число, потом вещественное

### **2.4. Описание аварийных случаев**

**Аварийные случаи при вводе целого числа**

1. Пустой ввод
2. Наличие посторонних символов
3. Наличие более одного знака  $+ -$  или неправильное его положение
4. Количество значащих цифр больше 40

### **Аварийные случаи при вводе вещественного числа**

1. Отсутствие значащих цифр
2. Отсутствие цифр в экспоненте
3. Наличие посторонних символов
4. Наличие двух точек или двух экспонент
5. Количество значащих цифр больше 40
6. Количество цифр в экспоненте больше 5
7. Введен ноль

### 3 Описание структуры данных

Листинг 3.1 – Структура хранения длинного числа

```
1      typedef struct
2      {
3          char sign;
4          int mantissa[MANTISA_LEN];
5          int exponent;
6          size_t man_length;
7      } bdouble_t;
```

#### Объяснение полей:

1. sign - Знак числа (+ или -)
2. mantissa - Мантисса числа. Массив целых чисел, каждый элемент - одна цифра
3. exponent - Порядок числа
4. man\_length - Количество значимых элементов в мантиссе

## 4 Описание алгоритма

### 1. Ввод чисел

- Проверка корректности введенных чисел
- Запись чисел в структуру или вывод сообщения об ошибке

### 2. Деление целого числа на вещественное

- Создается переменная `ans` типа `bdouble_t`, которая будет возвращена из функции. Происходит вычисление порядка: умножить на -1 экспоненту делителя и сложить с разностью экспоненты делимого и длины мантииссы делителя
- Начинается цикл с пост условием, пока не заполнено 40 цифр в мантиссе или делимое не равно 0
- Определяется длина неполного делимого и частное неполного делимого и делителя
- Находится вычитаемое для неполного делителя, за счет умножение делимого на частное
- В мантиссу записывается частное неполного делимого и делителя
- Из неполного делимого вычитается произведение делимого и частного
- Если уже заполнено 40 цифр в мантиссе или оставшееся число равно 0, то цикл деления завершается

### 3. Вывод результата деления

- Если делитель равен нулю, выводится сообщение о невозможности деления на ноль
- Если экспонента в результате больше 99999, выводится сообщение о достижении машинной бесконечности
- Если экспонента в результате меньше 99999, выводится сообщение о достижении машинного нуля
- Иначе выводится ответ в формате `[+-]?(0.[0-9]*)E[+-]?([0-9]*)`

## Основные функции:

1. `bdouble_t (div_big_numbers(bdouble_t *divident, bdouble_t *divisor)`
  - Функция делит два длинных числа, представленных в виде структуры `bdouble_t`. Возвращает результат деления, записанный в структуре `bdouble_t`
  - `divident`: Делимое.
  - `divisor`: Делитель.
2. `size_t incomplete_dividend(size_t *len, bdouble_t *divident, const bdouble_t *divisor)`
  - Функция определяет частное неполного делимого и делителя, и первым параметром изменяет длину неполного делимого.
  - `len`: Длина неполного делимого.
  - `divident`: Делимое.
  - `divisor`: Делитель.
3. `void mul_big_small(bdouble_t *big, const int digit, const size_t base)`
  - Функция умножает длинное число на цифру
  - `big`: Длинное число
  - `digit`: Цифра
  - `base`: Система счисления
4. `int cmp_mantissa(int *man1, const size_t len1, int *man2, const size_t len2)`
  - Функция сравнивает мантиссы двух длинных чисел
  - `man1`: Мантисса первого числа
  - `len1`: Длина мантиссы первого числа
  - `man2`: Мантисса второго числа
  - `len2`: Длина мантиссы второго числа
5. `int get_sub(size_t *len, bdouble_t *subtrahend, bdouble_t *divident, const bdouble_t *divisor)`



- Функция находит вычитаемое для неполного делимого. Возвращает очередную цифру частного делимого и делителя
- len: Длина неполного делимого
- subtrahend: Вычитаемое
- dividend: Делимое
- divisor: Делитель

## 5 Тестовые данные

### Позитивные тесты

Тест	Целое число	Вещественное число	Вывод
Числа равны обычная запись	123	123	+0.1E1
Числа равны экспоненциальная запись	123	1.23e2	+0.1E1
Есть ведущие нули	005	008.0	+0.625E0
Делимое больше делителя	12345678	12.3e3	+0.100371365 853658536585 365853658536 585365853E+4
Делитель больше делимого	123	4242.24e76	+0.289941163 159085766010 409594930979 859696763E-77
Делимое кратно делителю	144	12e0	+0.12E+2
Бесконечная дробь есть округления	5	0.0003e-1	+0.166666666 666666666666 666666666666 6666667E+6
Бесконечная дробь длинное округления	9999999999 9999999999 9999999999 9999999999	2	0.5E+40
Бесконечная дробь нет округления	5	.2e-3	+0.25E+5
40 цифр в делителе	31314	123456789012 345678901234 567890123456 7890e31	+0.253643402 282790620773 394649037891 3062448E-30

Тест	Целое число	Вещественное число	Вывод
40 цифр в делимом	1234567890 1234567890 12345678901 234567890e31	4224e-3124	+0.292274595 199682004974 513655042905 9109588E+3160
40 цифр в делителе и делимом	1234567890 1234567890 1234567890 1234567890	1111111111 1111111111 1111111111 1111111111	+0.1111111 1011111111 1011111111 1011111111 01E+2
Машинная бесконечность	123	0.000001e-99999	Достигнута машинная бесконечность!
Машинный ноль	123	1e99999	Достигнут машинный ноль

### Негативные тесты

Тест	Целое число	Вещественное число	Вывод
Нет значащих цифр в вещественном числе	123	.	Ошибка при вводе действительного числа
Две точки в вещественном числе	123	1.1.1	Ошибка при вводе действительного числа
Два знака +/- в вещественном числе	123	+ -23e3	Ошибка при вводе действительного числа
Нет цифры после экспоненты	123	12.3e	Ошибка при вводе действительного числа
Нет цифры перед экспонентой	123	e3	Ошибка при вводе действительного числа

Тест	Целое число	Вещественное число	Вывод
Символ перед экспонентой	123	12ke3	Ошибка при вводе действительного числа
Символ после экспоненты	123	12.3ek	Ошибка при вводе действительного числа
Две экспоненты	123	12.3e3e5	Ошибка при вводе действительного числа
Два знака +- после экспоненты	1	12.3e+-3	Ошибка при вводе действительного числа
Переполнение порядка в большую сторону	123	1e99999	Переполнение порядка
Переполнение порядка в меньшую сторону	123	1e-99999	Переполнение порядка
Пустой ввод		12ke3	Ошибка при вводе целого числа
Символ в числе	12k3	12.3ek	Ошибка при вводе целого числа
Две экспоненты	123.12	12.3e3e5	Ошибка при вводе целого числа
Деление на ноль	123.12	0	Ошибка деление на 0

## **6 Ответы на вопросы**

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

Для 64-разрядных систем максимально возможное знаковое представление целого числа находится в отрезке  $[-9223372036854775807; -9223372036854775806]$ . Максимально возможное беззнаковое представление целого числа находится в отрезке  $[0; 18\,446\,744\,073\,709\,551\,615]$  Для хранения вещественных чисел в 64-разрядных системах максимально под представление мантиссы отводится 52 двоичных разряда, а под представление порядка – 11 двоичных разрядов. В этом случае возможные значения чисел находятся в диапазоне в отрезке  $[3.6E-4951; 1.1E+4932]$ .

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

Точность представления чисел выражается в количестве памяти, выделенной под их хранение. Если рассматривать числа с плавающей точкой, то обычно для float отводится 4 байт, для double 8 байт, а для long double 16 байт. Память для хранения таких чисел распределяется на хранение знака, мантиссы и экспоненты. Соответственно, чем больше памяти выделено под мантиссу и экспоненту - тем более точными будут вычисления. Если же точности не хватает, программист должен сам реализовать хранение чисел с плавающей точкой

### **3. Какие стандартные операции возможны над числами?**

Для любых чисел: сложение, вычитание, умножение, деление, сравнение  
Дополнительные операции только для целых чисел: взятие остатка, побитовые сдвиги

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

Никакой из стандартных. В таких случаях программист должен написать свою структуру данных, которая будет покрывать необходимый для задачи диапазон значений.

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

Необходимые операции программист должен реализовать сам, используя

структуру для хранения длинного числа

## **ЗАКЛЮЧЕНИЕ**

В случаях, когда встроенных типов данных не хватает для покрытия необходимого диапазона значений, программист должен самостоятельно создать структуру данных для хранения длинных чисел и реализовать функции для их обработки. Это необходимо, например, в астрономических расчётах