



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

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1** **«ДЛИННАЯ АРИФМЕТИКА»**

Студент Ланкин Дмитрий Леонидович

Группа ИУ7 – 34Б

# СОДЕРЖАНИЕ

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	<u>2</u>
<u>ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....</u>	<u>2</u>
ВХОДНЫЕ ДАННЫЕ.....	2
ВЫХОДНЫЕ ДАННЫЕ.....	2
ОПИСАНИЕ ЗАДАЧИ.....	3
СПОСОБ ОБРАЩЕНИЯ К ПРОГРАММЕ.....	3
ОПИСАНИЕ ВОЗМОЖНЫХ АВАРИЙНЫХ СИТУАЦИЙ И ОШИБОК ПОЛЬЗОВАТЕЛЯ.....	3
<u>ВНУТРЕННЯЯ СТРУКТУРА ДАННЫХ.....</u>	<u>4</u>
<u>ОСОБЕННОСТИ РЕАЛИЗАЦИИ АЛГОРИТМА.....</u>	<u>4</u>
<u>ОПИСАНИЕ ФУНКЦИЙ.....</u>	<u>5</u>
<u>НАБОР ТЕСТОВЫХ СЛУЧАЕВ.....</u>	<u>7</u>
Выводы.....	9
<u>КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	<u>9</u>

## ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

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

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ

### ВХОДНЫЕ ДАННЫЕ

**Целое число:** строка, первым символом которой является знак ( $\pm$ ), состоящая из не более, чем 30 цифр. Формат:  $\pm m$ , где  $m$  – мантисса числа.

**Действительное число:** строка, первым символом которой является знак ( $\pm$ ), состоящая из не более, чем 30 цифр. Формат:  $\pm m[n][E \pm K]$ , где  $m$  – целая часть действительного числа,  $n$  – дробная часть (опционально),  $K$  – порядок числа (опционально) – число, принадлежащее отрезку  $[-99999; 99999]$ .

### ВЫХОДНЫЕ ДАННЫЕ

Программа возвращает **действительное число** формата  $\pm m.n E \pm K$ , где  $m$  – целая часть действительного числа,  $n$  – дробная часть действительного числа,  $K$  – порядок действительного числа. Суммарная длина мантиссы  $m.n$  должна быть не менее 2 и не более 30 цифр. Порядок числа  $K$  принадлежит отрезку  $[-99999; 99999]$ .

### ОПИСАНИЕ ЗАДАЧИ

Программа получает от пользователя целое и действительное число и производит деление целого на действительное. По окончании работы выводит на экран результат деления.

### СПОСОБ ОБРАЩЕНИЯ К ПРОГРАММЕ

Программа вызывается из консоли.

### ОПИСАНИЕ ВОЗМОЖНЫХ АВАРИЙНЫХ СИТУАЦИЙ И ОШИБОК ПОЛЬЗОВАТЕЛЯ

1. Некорректный ввод: введен буквенный символ.  
Сообщение об ошибке: «Error: введены некорректные символы.»
2. Некорректный ввод: ввод точки при записи целого числа.  
Сообщение об ошибке: «Error: введены некорректные символы.»
3. Некорректный ввод: отсутствие знака ( $\pm$ ) в начале строки.  
Сообщение об ошибке: «Error: число должно начинаться со знака '+' или '-'»

4. Некорректный ввод: пустая строка.  
Сообщение об ошибке: «Error: Вы ввели пустую строку.»
5. Некорректный ввод: излишне длинная запись.  
Сообщение об ошибке: «Error: слишком длинная строка.»
6. Некорректный ввод: наличие пробелов в записи.  
Сообщение об ошибке: «Error: число не может содержать пробелы.»
7. Некорректный ввод: точка в записи порядка действительного числа.  
Сообщение об ошибке: «Error: точка не может находиться в записи порядка.»
8. Некорректный ввод: более 1 буквы E в записи действительного числа.  
Сообщение об ошибке: «Error: по правилам записи не должно быть более одной буквы E!»
9. Некорректный ввод: более 1 точки в записи действительного числа.  
Сообщение об ошибке: «Error: по правилам записи не должно быть более одной точки!»
10. Некорректный ввод: более 2 арифметических знаков в записи.  
Сообщение об ошибке: «Error: по правилам записи должно быть 2 знака (мантисса и порядок)!»
11. Переполнение порядка.  
Сообщение об ошибке: «Error: переполнение порядка.»
12. Деление на ноль.  
Сообщение об ошибке: «Error: деление на 0.»

## ВНУТРЕННЯЯ СТРУКТУРА ДАННЫХ

От пользователя программа получается массив символов (строку), затем происходит её валидация и разделение на элементы, которые в дальнейшем записываются в структуру следующего вида:

```
1  typedef struct
2  {
3      char sign_mantissa;
4      char mantissa[MAX_MANTISSA_LEN - 1];
5      size_t point_pos;
6      size_t E_pos;
7      char sign_power;
8      int power_int;
9  } number_t;
```

— sign\_mantissa — переменная типа char, хранящая в себе знак мантиссы (+\-).

- `mantissa` — массив элементов типа `char`, состоящий из 31 элемента (1 запасной для округления), для хранения мантиссы числа.
- `point_pos` — переменная типа `size_t`, хранящая в себе индекс позиции точки в мантиссе.
- `E_pos` — переменная, типа `size_t`, хранящая в себе индекс позиции буквы `E`.
- `sign_power` — переменная типа `char`, хранящая в себе знак порядка (+\−).
- `power_int` — переменная типа `int`, хранящая в себе числовое значение порядка числа.

Таким образом, целое число и действительное число разбиваются на элементы и записываются в структуру для дальнейшего хранения.

Такие типы данных были выбраны в связи с особенностями хранения «длинных» чисел в памяти компьютера и с целью облегчения обработки данных.

## ОСОБЕННОСТИ РЕАЛИЗАЦИИ АЛГОРИТМА

1. Программа считывает 2 строки: первая представляет целое число, вторая — действительное (функция `str_input`).
2. Программа производит валидацию каждого из чисел (для целого проверяет наличие знаков + или −, для действительного — наличие знаков + или −, наличие буквы `E`, количество букв `E`, наличие и количество точек).
3. Программа разделяет строки и помещает элементы в поля структуры.
4. Программа нормализует целое и действительное числа (приводится к виду, при котором число начинается с 0, и дополняется нулями до длины 30 символов).
5. Программа производит деление каждой цифры целого числа на действительное число и записывает результат в структуру `result_number`.
6. Программа удаляет лидирующие незначащие нули.
7. Программа удаляет нули в конце числа и преобразует порядок числа.
8. Программа выводит результат на экран.

## ОПИСАНИЕ ФУНКЦИЙ

- **`void number_normalization(number_t *number);`**

Описание: функция приводит число к нормализованному виду.

Входные данные: экземпляр структуры `number_t`.

Выходные данные: нормализованный экземпляр структуры `number_t`.

- **`int str_to_number_t(char str[], number_t *number);`**

Описание: функция преобразует массив символов в экземпляр структуры `number_t`.

Входные данные: массив символов, экземпляр структуры `number_t`.

Выходные данные: код возврата, заполненный экземпляр структуры `number_t`.

– **`int is_left_greater(number_t *l_num, number_t *r_num);`**

Описание: функция проверяет, является ли «левое» число большим, чем правое.

Входные данные: два экземпляра структуры `number_t`.

Выходные данные: `TRUE` — левое число больше, `FALSE` — правое число больше, `EQUAL` — числа одинаковые.

– **`int digits_subtraction(number_t *int_number, number_t *real_number);`**

Описание: функция производит поразрядное вычитание действительного числа из целого числа.

Входные данные: два экземпляра структуры `number_t`.

Выходные данные: 1 в случае успеха, 0 в случае неудачи.

– **`int is_zero(number_t *number);`**

Описание: функция определяет, равно ли 0 число.

Входные данные: экземпляр структуры `number_t`.

Выходные данные: 1 в случае равенства, 0 — иначе.

– **`void push_back_zero(number_t *number);`**

Описание: функция сдвигает число на 1 разряд влево и вставляет 0 в конец.

Входные данные: экземпляр структуры `number_t`.

Выходные данные: -.

– **`int division(number_t *int_number, number_t *real_number, number_t *result_number);`**

Описание: функция производит поразрядное деление целого числа на действительное и записывает результат в экземпляр структуры `number_t result_number`.

Входные данные: экземпляры структуры `number_t int_number`, `real_number` и `result_number`.

Выходные данные: код возврата и экземпляр структуры `result_number`.

– **`int str_input(char str[], const size_t max_len, size_t *len);`**

Описание: функция выполняет ввод массива символов из стандартного потока ввода.

Входные данные: массив символов, максимальная длина.

Выходные данные: код возврата и длина введенного массива символов.

- **int str\_real\_validation(char str[], const size\_t len);**  
**int str\_int\_validation(char str[], const size\_t len);**

Описание: функция производит валидацию строки по соответствующим правилам.

Входные данные: массив символов, длина массива символов.

Выходные данные: код возврата.

- **void symbol\_shift(char str[], size\_t start, size\_t end);**

Описание: функция сдвигает символ влево в массиве символов.

Входные данные: массив символов, начальный индекс символа, конечный индекс символа.

- **void num\_shift(char str[], size\_t start);**

Описание: функция сдвигает число, начиная с позиции start, на 1 разряд влево в массиве символов.

Входные данные: массив символов, индекс первого символа.

Выходные данные: измененный массив символов.

- **void throw\_out\_digit(char str[], size\_t start);**

Описание: функция удаляет элемент с позиции start.

Входные данные: массив символов, индекс элемента для удаления.

Выходные данные: измененный массив символов.

- **void delete\_leading\_zeros(char str[]);**

Описание: функция удаляет лидирующие нули из массива символов.

Входные данные: массив символов.

Выходные данные: измененный массив символов.

- **size\_t fill\_ending\_with\_zeros(char str[], size\_t start, size\_t end);**

Описание: функция дополняет массив символов с конца нулями.

Входные данные: массив символов, индекс начального элемента, индекс конечного элемента для дополнения.

Выходные данные: количество дополненных нулей, измененный массив символов.

– **void delete\_last\_zeros(char str[]);**

Описание: функция удаляет лишние нули, изменяя при этом порядок числа.

Входные данные: массив символов.

Выходные данные: измененный массив символов.

## НАБОР ТЕСТОВЫХ СЛУЧАЕВ

№	Ситуация	Целое число	Действительное число	Результат
1	Введен буквенный символ	a	—	Error: введены некорректные символы.
2	Введена точка при записи целого числа.	+34.43	—	Error: введены некорректные символы.
3	Отсутствует знак +\-. .	432	—	Error: число должно начинаться со знака '+' или '-'.
4	Пустая строка.		—	Error: Вы ввели пустую строку.
5	Излишне длинное число.	+12345678901234567890123456789012345678901234	—	Error: слишком длинная строка.
6	Наличие пробелов в записи числа.	+4324 53245	—	Error: число не может содержать пробелы.
7	Точка в записи порядка.	+324	-543576.123E-43.3	Error: точка не может находиться в записи порядка.
8	Более 1 буквы E в записи действительного числа.	+12345	+4343.654EE-4	Error: по правилам записи не должно быть более одной буквы E!
9	Более 1 точки в записи действительного числа.	+12345	+432.576.78E-3	Error: по правилам записи не должно быть более одной точки!
10	Более 2 арифметических знаков в записи	+12345	+432.432E-43E3	Error: по правилам записи должно быть 2



				знака (мантисса и порядок)!
11	Переполнение порядка.	+123456	+1223.43E-99999	Error: переполнение порядка.
12	Деление на 0.	+34324	+0	Error: деление на 0.
13	Деление нуля.	+0	-213.432E-5	+0.0E+0
14	Деление целого на целое одного знака.	+144	+12	+0.12E2
15	Деление целого на целое противоположных знаков.	+144	-12	-0.12E2
16	Деление целого на действительное одного знака.	+144	+1.2E+1	+0.12E2
17	Деление целого на действительное противоположных знаков.	+144	-1.2E+1	-0.12E2
18	Деление целого на целое противоположных знаков.	+200000	-100	-0.2E4
19	Деление максимального целого числа на единицу.	+9999999999999999 9999999999999999	+1	+0.9999999999999999 9999999999999999 99E30
20	Деление максимального целого числа на отрицательное число с отрицательным порядком.	+9999999999999999 9999999999999999	-0.1E-30	- 0.9999999999999999 9999999999999999 9E60
21	Деление целого числа максимальной длины на большое действительное число.	+1234567890123456 78901234567890	- 1234567.8901234 5E+99321	- 0.1000000000000000 005499100049491 8E-99297
22	Деление положительного числа на отрицательное малое число.	+1000000000000	-1.0E-2314	-0.1E2325

## Выводы

Проделав данную лабораторную работу, я научился делить большие целые числа на действительные числа, длина мантиссы которых не превышает 30 разрядов, а порядок находится в диапазоне от -99999 до +99999 (т.е. не более 5 разрядов). Я узнал, как можно хранить подобные числа в памяти компьютера и оперировать ими с целью деления.

## КОНТРОЛЬНЫЕ ВОПРОСЫ

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

Диапазон чисел напрямую зависит от выбранного типа данных. В случае, если выбран беззнаковый целочисленный тип размером 64 бита, то его диапазон — [0; 18 446 744 073 709 551 615]. В случае знакового целочисленного типа размером 64 бита — [-9 223 372 036 854 775 808; 9 223 372 036 854 775 807].

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

Точность действительных чисел зависит от того размера памяти, который выделяется на хранение мантииссы этого числа.

Согласно стандарту 754-1985, определяется четыре формата представления чисел с плавающей точкой (т.н. ЧПТ):

- с одинарной точностью (single-precision) — 32 бита;
- с двойной точностью (double-precision) — 64 бита;
- с одинарной расширенной точностью (single-extended precision) —  $\geq 43$  бита (редко используемый);
- с двойной расширенной точностью (double-extended precision) —  $\geq 79$  бит (обычно используют 80 бит).

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

Над числами можно производить процедуры сравнения, вычитания, сложения, умножения, деления и взятия остатка.

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

Программист может воспользоваться такими агрегированными типами данных, как массив, структура.

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

Возможно написать свои собственные функции, реализующие необходимые программисту операции, используя простейшие математические операции.