



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** **«ОБРАБОТКА БОЛЬШИХ ЧИСЕЛ»**

Студент Чыонг Ван Хао  
*фамилия, имя, отчество*

Группа ИУ7И-31Б

**Цель работы:** реализация арифметических операций над числами, выходящими за разрядную сетку персонального компьютера, выбор необходимых типов данных для хранения и обработки указанных чисел.

## Описание условия задачи

Смоделировать операцию деления действительного числа на действительное число в форме  $\pm m.n E \pm K$ , где суммарная длина мантииссы ( $m+n$ ) - до 30 значащих цифр, а величина порядка  $K$  - до 5 цифр. Результат выдать в форме  $\pm 0.m_1 E \pm K_1$ , где  $m_1$  - до 30 значащих цифр, а  $K_1$  - до 5 цифр.

Если при делении чисел длина мантииссы стала больше 30 знаков, то необходимо произвести округление по следующему правилу: если 31-ая цифра больше или равна 5-ти, то число округляется по законам математики, иначе все последующие знаки отбрасываются.

## Техническое задание

### Входные данные

**Действительное число:** строка, которая может состоять только из символов '+', '-', '.', 'E' и цифр, в формате ' $\pm m[n][E\pm k]$ ', где  $m$  – целая часть числа,  $n$  – дробная часть числа, ( $m+n$ ) – мантиисса числа,  $k$  – порядок числа. Ввод знака перед числом обязателен. Длина мантииссы числа ( $m+n$ ) меньше, либо равна 30 и больше 0. Длина порядка  $k$  меньше, либо равна 5 и больше 0. Если число вводится с дробной частью  $n$ , то ввод точки после целой части  $m$ , ввод символа 'E', ввод знака порядка обязательны.

### Выходные данные

**Действительное число:** строка, которая может состоять только из символов '+', '-', '.', 'E' и цифр, в нормализованном формате ' $\pm 0.mE\pm k$ ', где  $m$  – мантиисса числа,  $k$  – порядок числа. Длина мантииссы числа  $m$  меньше, либо равна 30. Длина порядка числа  $k$  меньше, либо равна 5.

## **Функция программы**

Программа делит действительное число на действительное.

## **Обращение к программе**

Программа запускается из терминала командой «./app.exe» в директории с программой.

## **Возможные аварийные ситуации и ошибки пользователя**

1. Некорректный ввод: строка с действительным числом не содержит знака числа или первый символ не является знаком.

На выходе сообщение: «Первым символом действительного числа должен быть знак числа.»

2. Некорректный ввод: строка с действительным числом содержит не цифру в мантиссе.

На выходе сообщение: «Мантисса действительного числа должна состоять только из цифр.»

3. Некорректный ввод: строка с действительным числом содержит пробелы.

На выходе сообщение: «Не должно быть пробелов между символами.»

4. Неверная длина мантиссы действительного числа.

На выходе: корректное число, длина мантиссы которого превышает 30 цифр. На выходе сообщение: «Длина мантиссы действительного числа должна быть меньше, либо равна 30 и больше 0.»

На выходе: нет числа. На выходе сообщение: «Необходимо ввести действительное число.»

5. Некорректный ввод: строка с действительным числом не содержит знака порядка или он не является знаком.

На выходе сообщение: «Перед порядком действительного числа должен быть знак порядка.»

6. Некорректный ввод: строка с действительным числом содержит не цифру в порядке.

На выходе сообщение: «Порядок действительного числа должен состоять только из цифр.»

7. Неверная длина порядка действительного числа.

На выходе: корректное число, длина порядка которого превышает 5 цифр.

На выходе сообщение: «Длина порядка действительного числа должна быть меньше, либо равна 5 и больше 0.»

8. Переполнение порядка.

На входе: в процессе деления степень полученного в результате числа превышает 99999.

На выходе сообщение: «Переполнение порядка.»

9. Действительное число равно нулю.

На выходе сообщение: «Нельзя делить на 0.»

## Внутренняя структура данных

**Ввод:** Для ввода вещественного числа используется символ `symbol` и массив `order[MAX_LEN_ORDER + 2]`.

При помощи `symbol` и `order` поля структуры `number` сразу заполняются.

```
typedef struct
{
    char number_sign;
    char mantissa[MAX_LEN_MANTISSA + 2];
    int point_index;
    char order_sign;
    int order;
} number;
```

Поля структуры:

- `number_sign` – знак числа. Принимает значения “+” или “-”;
- `mantissa[MAX_LEN_MANTISSA + 2]` – значение мантиссы числа,

$MAX\_LEN\_MANTISSA = 30$ , два дополнительных символа для знака и «\0»;

- *point\_index* — индекс точки в числе;
- *order\_sign* — знак числа. Принимает значения “+” или “-”;
- *int order* — значение порядка числа.

В результате создаются переменный структурного типа *real\_number*.

**Деление:** алгоритм деления позволяет записать окончательный результат в переменную *number result\_number*.

**Вывод:** для вывода используется созданная на этапе деления переменная структурного типа *number result\_number*.

## Алгоритм

1. Программа считывает две строки символов и записывает полученные числа в переменные структурного типа *number*, одновременно проверяя на валидность.

```
int input_number(number *real_number);
```

2. Корректные числа преобразуются в нормализованный вид: удаляются ненужные нули, изменяется порядок числа в связи со сдвигом точки в начало числа

```
void normalize_number(number *one_number);
```

```
void add_zeros(number *one_number);
```

3. Выполняется деление первого числа на второе методом “деления в столбик”, при этом контролируется округление, переполнение порядка числа и деление на ноль.

```
int compare_numbers(number *first_number, number *second_number);
```

```
int compare_zero(number *one_number);
```

```
void change_zero(number *one_number);
```

```
int subtraction(number *first_number, number *second_number);
```

```
int divide_numbers(number *first_number, number *second_number, number *result_number);
```

4. Корректный результат выводится на экран в нормализованном виде в соответствии со спецификацией, указанной в ТЗ.

```
void output_number(number real_number);
```

## **Функции программы**

```
void print_conditions(void);
```

Описание: функция выводит назначение программы и условия ввода чисел.

Входные значения: -

Выходные значения: сообщения с информацией о назначении программы и правилах ввода чисел.

```
int input_number(number *real_number);
```

Описание: функция считывает действительное число и записывает его в структуру *number real\_number*.

Входные значения: переменная структурного типа *number real\_number* для записи в нее числа.

Выходные значения: переменная структурного типа *number real\_number* с записанным в нее числом, код ошибки.

```
void normalize_number(number *one_number);
```

Описание: функция нормализует мантиссу числа, записанную в *number one\_number*, изменяя порядок числа.

Входные значения: переменная структурного типа *number one\_number* с числом для нормализации.

Выходные значения: переменная структурного типа *number one\_number* с нормализованным числом.

***void add\_zeros(number \*one\_number);***

Описание: функция добавляет нуля в структуру *number*.

Входные значения: переменная структурного типа *number*.

Выходные значения: переменная структурного типа *number* с добавленными нулями.

***int compare\_numbers(number \*first\_number, number \*second\_number);***

Описание: функция сравнивает целое действительное *number first\_number* и действительное число *number second\_number*.

Входные значения: переменные структурного типа *number first\_number*, *number second\_number*.

Выходные значения: код результата (1 — больше или равно, 0 — меньше).

***int compare\_zero(number \*one\_number);***

Описание: функция сравнивает число и 0.

Входные значения: переменная структурного типа *number*.

Выходные значения: код результата (1 — равны, 0 — различны).

***void change\_zero(number \*one\_number);***

Описание: функция меняет позицию нуля.

Входные значения: переменная структурного типа *number*.

Выходные значения: переменная структурного типа *number* с измененной позицией нуля.

***int subtraction(number \*first\_number, number \*second\_number);***

Описание: функция вычитает из действительное числа *number first\_number* действительное *number second\_number* и записывает результат в структуру действительное *number first\_number*.

Входные значения: переменные структурного типа *number first\_number*, *number second\_number*.

Выходные значения: переменная структурного типа *number first\_number* с результатом вычитания, код ошибки.

***int divide\_numbers(number \*int\_number, number \*real\_number, number \*result\_number);***

Описание: делит действительное число *number first\_number* на действительное *number second\_number* и сохраняет результат в *number result\_number*.

Входные значения: переменные структурного типа *number first\_number*, *number second\_number*.

Выходные значения: переменная структурного типа *number result\_number* с результатом деления, код ошибки.

***void output\_number(number real\_number);***

Описание: функция печатает число из структуры *number real\_number*.

Входные значения: переменная структурного типа *number real\_number*.

Выходные значения: сообщение с результатом деления в нормализованном виде.

## Тесты

№	Случай	Первое число	Второе число	Вывод
1	Отсутствие знака	12334564.1	-	Первым символом действительного числа должен быть знак числа.



2	В мантиссе числа недопустимые символы	+12.3	+1a	Мантисса действительного числа должна состоять только из цифр.
3	В мантиссе числа введены пробелы	+12 3	-	Не должно быть пробелов между символами.
4	Не введен знак порядка	+12.3	+12E5	Перед порядком действительного числа должен быть знак порядка.
5	В порядке действительного числа недопустимые символы	+12.3	+78E-a	Порядок действительного числа должен состоять только из цифр.
6	Неверная длина мантиссы числа	-123..99E+1 (31 цифра в мантиссе)	-	Длина мантиссы целого числа должна быть меньше, либо равна 30 и больше 0.
7	Неверная длина мантиссы числа	+12.3	+	Необходимо ввести действительное число.
8	Неверная длина порядка	+12.3	+123E+9...9 (6 цифра 9 в порядке)	Длина порядка действительного числа должна быть меньше, либо равна 5 и больше 0.
9	Переполнение порядка	+1E+99999	+10E+0	Переполнение порядка.
10	Делитель равно 0	-123.5	+0	Нельзя делить на 0.

11	Делимое равно 0	+0	+123.5E+2	+0.0E+0
12	Деление одного знака	-56789.8E+3	-0.2E+0	+0.283949E+9
13	Деление разных знаков	+255.5E+5	-0.5E+1	-0.511E+7
14	Округление	+99...9 (30 цифра 9)	+2	+0.5E+30

## Выводы

При написании лабораторной работы реализована функция деления вещественное числа на длинное вещественное, используя алгоритм деления «в столбик». Для ввода чисел использовался массив символов, для хранения, обработки и вывода чисел - переменные структурного типа.

## Контрольные вопросы

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

Возможный диапазон чисел зависит от выбранного типа, разрядности процессора и памяти выделенной для хранения числа. Для беззнакового целого числа выделяется 64 двоичных разряда, его максимальное значение –  $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$ .

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

Точность представления вещественных чисел зависит от размера памяти, выделенной для хранения мантииссы. Для мантииссы числа типа *double* это значение равно 52 битам, поэтому мантиисса числа может иметь значение до 4 503 599 627 370 496.

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

Числа возможно складывать, вычитать, умножать, делить, брать остаток от числа и сравнивать числа.

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

Программист может использовать массив чисел или символов, представляя числа в строку, или структуру, поля которой могут быть частями числа.

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

Можно написать собственные функции, реализующие такие операции или использовать функции дополнительных библиотек.