



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)

ФАКУЛЬТЕТ \_\_\_\_\_ Информатика и системы управления

КАФЕДРА \_\_\_\_\_ Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**  
**“ТИПЫ И СТРУКТУРЫ ДАННЫХ”**  
**ЛАБОРАТОРНАЯ РАБОТА №1**

Студент \_\_\_\_\_ Ляпина Наталья Викторовна  
*фамилия, имя, отчество*

Группа \_\_\_\_\_ ИУ7-32Б

Вариант \_\_\_\_\_ 1

Студент \_\_\_\_\_ Ляпина Н.В.  
*подпись, дата* *фамилия, и.о.*

Преподаватель \_\_\_\_\_ Силантьева А.В.  
*подпись, дата* *фамилия, и.о.*

2021 г.

## **Оглавление**

<b>1)</b>	Условие задачи.....	<b>3</b>
<b>2)</b>	Структура данных.....	<b>4</b>
<b>3)</b>	Интерфейс модулей.....	<b>5</b>
<b>4)</b>	Описание алгоритма.....	<b>6</b>
<b>5)</b>	Тесты.....	<b>7</b>
<b>6)</b>	Вывод.....	<b>8</b>
<b>7)</b>	Ответы на контрольные вопросы.....	<b>8</b>

## Задание

### 1. Условие

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

### 2. Входные данные

- 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.
- Целое число длиной до 30 десятичных цифр.

### • Требования к входным данным

- 1) Все именованные части чисел ( $m$ ,  $n$ ,  $K$ ) должны представлять собой корректные натуральные числа или 0;
- 2) Знаки при вводе не обязательны;
- 3) Примеры корректного ввода: 0.125 125. 125.521E5 12E5 +25.000E-80

### 3. Выходные данные

Действительное число в форме  $+(-)0.m1 \text{ E } +(-)K1$ , где  $m1$  - до 30 значащих цифр, а  $K1$  - до 5 цифр.

В случае ошибки будет выведено соответствующее сообщение.

### 4. Действие программы

Программа выполняет деление действительного числа на целое при корректных входных данных.

### 5. Обращение к программе

Программа может быть вызвана через консоль компилятора с помощью команды `“./app.exe”`

## 6. Аварийные ситуации

В случае аварийной ситуации выводится сообщение об определенной ошибке.

### **Неверный ввод:**

- Мантисса действительного числа содержит символы отличные от цифр и символа “.”;
- Порядок действительного числа содержит символы отличные от цифр и символа “E”;
- В целой части действительного числа нет цифр, отличных от 0 и количество 0  $\geq 2$ ;
- Используются пробелы при вводе чисел;
- Строка с действительным числом содержит больше одного символа “.”;
- Порядок превышает “99999”;
- Превышение длины мантиссы действительного числа (длина больше 31 символа, включая “.”);
- Целое число содержит символ, отличный от цифры;
- Превышение длины целого числа (более 30 цифр).

### **Другие ошибки:**

- Деление на 0;
- Переполнение порядка при делении.

## Структура данных

Для реализации данной задачи была создана структура *num\_t*, состоящая из знака числа типа *char*, мантиссы типа *man\_t* (массива типа *int*), знака порядка типа *char* и порядка типа *int*.

Выбор данной структуры обусловлен необходимостью хранить части числа по отдельности, т.к. другие типы данных не могут вместить такое большое число. При этом большое количество переменных нецелесообразно для данной реализации. Использование такого типа данных как *struct* упрощает работу с вводимыми числами.

Хранение целого числа происходит посредством такой же структуры данных, как и действительное число.

### **Хранение чисел**

**typedef struct**

```
{  
    char num_sing;  
    man_t mantissa;  
    char exp_sing;  
    int order;  
} num_t;
```

**typedef char real\_t[SRMAX]** - хранение введенного действительного числа до разбиения;  
**typedef char int\_t[SIMAX]** - хранение введенного целого числа до разбиения;  
**typedef int man\_t[NMAX]** - хранение мантиссы.  
**NMAX = 32** (используется для хранения мантиссы вместе с “.” и “\0”);  
**SRMAX = 41** (используется для хранения 30 символов мантиссы, 2 символов-знаков, 1 символа точки, 1 символа “E”, 5 символов порядка, 1 символ для проверки переполнения и 1 символ для “\0”);  
**SIMAX = 32** (используется для хранения целого числа, введенного с клавиатуры, 30 символов для числа, 1 символ для проверки переполнения и 1 символ “\0”).

## Интерфейс модулей

*Для обработки типа **real\_t** используются функции:*

```
int read_real(real_t s)
// Функция проверки ввода действительного числа
// s - строка-буфер для введенного действительного числа

int is_real(real_t string, num_t *real, size_t *man_r)
// Функция проверяет валидность введенного действительного числа и записывает
его в структуру num_t
// string - буфер с введенным действительным числом; real - структура, в которую
будет записано число; man_r - размер мантиссы .
```

*Для обработки типа **int\_t** используются функции:*

```
int read_int(int_t s)
//Функция проверки ввода целого числа
// s - строка-буфер для введенного целого числа

int is_integer(int_t string, num_t *integer, size_t *man_i)
// Функция считывает в структуру целого числа данные из строки
// string - буфер с введенным целым числом; integer - структура, в которую будет
записано число; man_i - размер мантиссы.
```

*Для обработки структуры **num\_t** используются функции:*

```
void zeros_in_massive(num_t *num)
// Функция присваивает нули всем элементам массива мантиссы
// num - структура с действительным или целым числом.
```

```
int normalization(num_t *data, size_t *n)
```

```

// Удаление лидирующих нулей массива
// data - структура с числом; n - размер мантиисы.

int move_num_end(num_t *data, size_t *n)
// Функция сдвигает число из начала в конец мантиисы
// data - структура с числом; n - размер мантиисы.
void move_digits(num_t *num, int size, int offset)
// функция сдвигает данные массива относительно его конца для выравнивания
// порядков и для того, чтобы было возможно произвести операцию вычитания
// num - структура с числом; size - размер мантиисы; offset - размер сдвига и его
// направление.

int comparison(num_t *real, num_t *integer)
// функция поразрядного сравнения чисел
// real - структура с действительным числом; integer - структура с целым числом.

void rounding(num_t *result, size_t size, int last_digit)
// функция округления при нахождении 31 цифры мантиисы
// result - результат деления ; size - размер массива мантиисы; last_digit - последняя
// цифра мантиисы (31 цифра).

void subtract(num_t *real, num_t *integer, int size)
// функция поразрядного вычитания цифр массивов мантиис
// real - структура с действительным числом; integer - структура с целым числом;
// size - размер массива мантиисы.

int division(num_t *real, num_t *integer, num_t *result, int size)
// функция, объединяющая в себе логику деления
// real - структура содержащая действительное число; integer - структура
// содержащая целое число; result - структура для записи результата деления; size -
// размер массива мантиисы.

```

## Описание алгоритма

1. Ввод действительного числа. Использование функции `read_real` позволяет прочесть все действительное число и записать его как массив `char`, для последующего разбиения на составные части с помощью функции `is_real`. При этом в первой функции проверяется выход за пределы допустимых значений для

действительного числа, а во второй функции проверяется непосредственно валидность ввода.

2. Ввод целого числа осуществляется с помощью функции `read_int`, где также проверяется переполнение. Затем число разбивается на составные части с помощью функции `is_int` и проверяется валидность ввода. Переменная, хранящая целое число имеет ту же структуру данных, что и действительное число.
3. Перед операцией деления числа нормализуются(удаляются лидирующие нули массива) и проверяется ошибка “деления на 0”. Затем числа сдвигаются в конец мантиссы и вычисляется порядок результата деления.
4. Происходит сдвиг относительно конца массива мантиссы для выравнивания порядков и для того, чтобы было возможно провести операцию вычитания.
5. Проверяется переполнение порядка результата деления и происходит получение результирующего знака.
6. Непосредственно операция деления чисел:
  - Все вычисление происходит в цикле, который зависит от текущей позиции в результирующем массиве мантиссы.
  - Вычисляется поразрядная разность между разрядами мантиссы действительного и целого чисел.
    - Если эта разность равна 0, то цикл аварийно завершается и в результат записывается единица, т.к. целое и действительное число равны.
    - Если разность больше 0, то происходит циклическое вычитание разрядов целого из разрядов мантиссы, пока первый разряд мантиссы  $\geq 0$ . Количество таких вычитаний записывается в результирующую структуру на позицию, относительно которой был начат цикл. Перед записью в структуру проверяется ее переполнение, и если оно есть, то происходит округление последней цифры с учетом результата поразрядного вычитания.
    - Если разность меньше 0 (разряд мантиссы меньше разряда целого), то разряды мантиссы сдвигаются на 1 влево
7. Затем печатается знак результата деления, считается размер мантиссы результата и она печатается на экран. После этого печатается порядок и работа программы завершается.

## Тесты

## Негативные тесты

№	Описание	Действительное число	Целое число	Результат
1	Недопустимый символ в мантиссе	1.r34E1	-	Ошибка
2	Недопустимый символ в порядке	1.34E-2g4	-	Ошибка
3	Лишний ведущий 0	00.13	-	Ошибка
4	Пробелы между цифрами	1. 3E 7	-	Ошибка
5	Две точки в действительном числе	123.4566.334	-	Ошибка
6	Превышение размера порядка	12.3E100000	-	Ошибка
7	Превышение размера мантиссы	12356.67890875465432189075435271	-	Ошибка
8	Недопустимый символ в целом	1.2	1g4	Ошибка
9	Превышение размера целого	1.3	123456789012345678901	Ошибка
10	Деление на 0	1.0	0	Ошибка
11	Переполнение порядка при делении	0.34444444444444444444444444444444E-99999	-13425	Ошибка

### *Позитивные тесты*

№	Описание	Действительное число	Целое число	Результат
1	Введена только мантисса	0.42	2	0.21E0
2	Число делится на 1 Пограничное значение	0.9999999999999999 9999999999999999E-99 999	1	0.9999999999999999 9999999999999999E- 99999



3	Пограничное значение в результате	0.1	3	0.3333333333333333 333333333333333E -1
4	Периодическая дробь в результате	125.5E10	55	0.228181818181818 181818181818182E 11
5	Действительное больше целого	6444.	20	0.3222E3
6	Делимое - 0	0.0	12	0.0E0
7	Деление одинаковых чисел разных знаков	-1.	1	-0.1E1
8	Деление на 1	0.2	1	0.2E0
9	Целое больше действительного	400.E4	2	0.2E7
10	Деление на число кратное 10	0.3E-100	1000000000000 000	0.3E-115
11	Деление на кратное число	12E4	4	0.3E6
12	Тест с периодической дробью	2.	9999	0.200020002000200 02000200020002E- 3

## ***Вывод***

Для хранения длинных целых и действительных чисел наиболее рационально использовать структуру, в которой располагаются поля с частями числа: знаком, мантиссой и порядком.

При реализации арифметических операций, в частности деления наиболее удобным алгоритмом оказался столбик. Он наиболее прост в реализации и понятен при чтении кода.

## ***Ответы на контрольные вопросы***

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

Каждый тип данных характеризуется диапазоном значений чисел, который, в свою очередь, зависит от размера области памяти, выделяемой под хранение переменной этого типа, от наличия знака в числе и от типа представления числа (целое или вещественное). Так, для целого положительного числа  $X$ , для которого выделено  $k$  разрядов максимальное значение не может превышать  $2^k - 1$ . Для 64 разрядов максимальное значение составит  $2^{64} - 1 = 18\,446\,744\,073\,709\,551\,615$ .

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

В случае чисел с плавающей точкой, точность определяется количеством разрядов, выделенных под хранение мантиссы числа. Возможная точность представления чисел зависит от архитектуры процессора, то есть количества разрядов выделенных для хранения числа с плавающей точкой. Так, для 64-х разрядных процессоров количество разрядов, выделенных под мантиссу, равно 52, а под порядок - 11 разрядов. Но необходимо помнить, что разряды представлены в двоичной системе счисления. А для хранения и обработки больших вещественных чисел в десятичной системе счисления отведено 15-17 разрядов.

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

Операции сложения, вычитания, умножения, деления, сравнения, инкрементирования, декрементирования, изменения знака.

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

Можно использовать массив, например, массив символов или чисел - для ввода и вывода числа, числовой массив - для обработки. Также можно использовать структуру для хранения всего числа, например, знак мантиссы, мантисса, знак порядка, порядок.

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

Так же как нет стандартных типов для хранения таких больших чисел, так нет и стандартных арифметических операций для их обработки, поэтому необходимо разработать эти операции самостоятельно. При этом удобно использовать алгоритм умножения и деления в "столбик".