



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 (ИУ)

КАФЕДРА \_\_\_\_\_ Программное обеспечение ЭВМ и информационные технологии (ИУ7)

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1** **«ОБРАБОТКА БОЛЬШИХ ЧИСЕЛ»**

Студент, группа

**Миронов Г., ИУ7-33Б**

2020 г.

## Описание условия задачи

Смоделировать операцию умножения целого числа длиной до 30 десятичных цифр на 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. Результат выдать в форме  $0.m1 \text{ E } K1$ , где  $m1$  - до 30 значащих цифр, а  $K1$  - до 5 цифр.

## Техническое задание

### Входные данные:

1. **Целое число:** строка, содержащая целое число в виде  $[+/-]m$ . Суммарная длина модуля числа ( $m$ ) – до 30 цифр. Наличие знака “+” перед числом необязательно.
2. **Действительное число:** строка, содержащая вещественное число в виде  $[+/-]m[n]e[+/-]K$   
Суммарная длина мантиссы ( $m+n$ ) – до 31 цифры (считая точку), порядка  $K$  – до 5 цифр. Наличие знака экспоненты  $e/E$  обязательно. Наличие знака “+” перед числом и порядком не обязательны.

### Выходные данные:

Длинное число, нормализованное в виде  $[+/-]0.m1 \text{ E } [+/-]K1$ , где длина  $m1$  – до 30 цифр,  $K1$  – до 5.

### Функция программы:

Умножение целого числа на действительное.

### Обращение к программе:

Запускается из терминала последовательным вызовом команд из корневой директории проекта:

1. `make` (Только при первом запуске)
2. `make run`

## **Аварийные ситуации:**

1. Некорректный ввод строки с целым числом.

На входе: строка, хотя бы один символ в которой не цифра и не символ +/- (если это первый элемент в строке).

На выходе: сообщение «Неверный формат целого числа!»

2. Некорректный ввод строки с действительным числом.

На входе: строка, хотя бы один символ в которой не цифра и не символ из набора "+ - . e E".

На выходе: сообщение «Неверный формат вещ-ого числа!»

3. Превышение длины строки при вводе целого числа.

На входе: корректное число, длина которого превышает 30 цифр.

На выходе: сообщение «Неверный формат целого числа!»

4. Превышение длины строки при вводе действительного числа.

На входе: корректное число, длина мантиссы которого превышает 31 цифру (считая точку) или длина порядка превышает 5 цифр.

На выходе: сообщение «Неверный формат вещ-ого числа!»

5. Переполнение порядка.

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

На выходе: сообщение «Ошибка умножения. Переполнение порядка.»

## Структуры данных

Для изначального хранения введенного числа используется динамически выделяемый символьный массив.

После ввода числа, оно обрабатывается и разбивается на части, которые описываются в структуре *LongDouble*. Она описывается как:

```
typedef struct LongDouble
{
    void *value;
    char *(*toString)(struct LongDouble *);
    int (*multiply)(
        struct LongDouble *,
        struct LongDouble *,
        struct LongDouble *
    );
} LongDouble;
```

Поля структуры:

**value** – указатель на структуру, хранящую само число; Она описывается так:

```
typedef struct
{
    char sign,
    char *digits,
    int digits_size,
    int exponent
} LongDoubleValue;
```

Поля структуры *LongDoubleValue*:

sign - знак числа. Принимает значения '+' или '-'  
digits - массив цифр данного числа в десятичной СС  
digits\_size - размер массива digits  
exponent - значение экспоненты числа

**toString** – указатель на функцию перевода числа в строковое представление;

**multiply** – указатель на функцию умножения двух чисел;

## Алгоритм

1. На вход программе подаются две строки символов, сначала содержащая целое число, затем содержащая вещественное число.
2. Выполняется запись полученных чисел в структуры типа LongDouble с проверкой данных на корректность.
3. Если одно из чисел равно нулю, то результат принимается равным за 0.
4. Выполняется умножение первого числа на второе методом “умножения в столбик”, при этом контролируется округление.
5. При переполнении порядка пользователю выводится соответствующее сообщение об ошибке.
6. Если результат умножения корректен, то он выводится пользователю в нормализованном виде в соответствии со спецификацией, указанной в ТЗ.

При успешном умножении число выводится согласно спецификации, указанной в ТЗ.

## Тесты

	Тест	Число 1	Число 2	Результат
1	Некорректный ввод	q	-	Неверный формат целого числа!
2	Некорректный ввод	123.45	-	Неверный формат целого числа!
3	Некорректный ввод	123	1.23.45	Неверный формат вещ-ого числа!
4	Некорректный ввод	123	123E+45.67	Неверный формат вещ-ого числа!
5	Некорректный ввод	123	E123	Неверный формат вещ-ого числа!
6	Превышение длины мантиссы	999...999 (31 девятка)	-	Неверный формат целого числа!
7	Превышение длины мантиссы	1	0.999...999e0 (31 девятка)	Неверный формат вещ-ого числа!
8	Превышение длины порядка	1	0.1E+123456	Неверный формат вещ-ого числа!
9	Переполнение порядка	10000	1E99999	Ошибка умножения. Переполнение порядка.
10	Умножение целых чисел	100	5.0e0	+0.5E+3
11	Умножение чисел одного знака	2	0.2e0	+0.4E+0

12	Умножение чисел разных знаков	5	-1e+5	-0.5E+6
13	Округление	99..99 (30 девяток)	2e0	+0.2E+31
14	Округление	5	99..99e0 (30 девяток)	+0.5E+31
15	Граничные значения	99...99 (30 девяток)	1E0	+0.1E+31
16	Граничные значения	1	99..99E0 (30 девяток)	+0.1E+31

## Функции

**int input\_long\_int(char \*\*);**

Функция ввода строки, содержащей целое число.  
Указывает верный формат ввода и считывает строку.

Аргументы

1. Указатель на строку, в которую будет прочитано число.

Возвращаемые значения

1. Строка, содержащая число (корректность не проверяется).
2. Код ошибки.

**int input\_long\_double(char \*\*str);**

Функция ввода строки, содержащей вещественное число.  
Указывает верный формат ввода и считывает строку.

Аргументы

1. Указатель на строку, в которую будет прочитано число.

Возвращаемые значения

1. Строка, содержащая число (корректность не проверяется).
2. Код ошибки.

**int newLongDouble(LongDouble \*\*, char \*, int);**

Функция перевода строки, содержащей целое число, во внутреннее представление числа в программе.

Аргументы

1. Указатель на указатель на структуру, в которую будет прочитано значение.
2. Указатель на строку, из которой будет прочитано число.
3. Флаг строгости проверки (Не строгая - целые числа, строгая - только экспоненциальная форма вещественных).

Возвращаемые значения

1. Структура содержащая прочитанное число.
2. Код ошибки.



**int newEmptyLongDouble(LongDouble \*\*);**

Функция создания длинного числа, содержащего стандартное значение (+0E+0).

Аргументы:

1. Указатель на указатель на структуру, которая будет проинициализирована.

Возвращаемые значение:

1. Структура, содержащая значение по умолчанию (+0E+0).
2. Код ошибки.

**void deleteLongDouble(LongDouble \*\*);**

Функция удаления длинного числа из памяти.

Аргументы:

1. Указатель на указатель на структуру, содержащую число, которое нужно удалить.

Возвращаемые значения - Отсутствуют.

**int removeZeros(longDoubleValue \*);**

Функция удаления лишних нулей в числе, во внутреннем представлении числа.

Аргументы:

1. Указатель на структуру, хранящую число во внутреннем формате.

Возвращаемые значения:

1. Структура с удаленными лишними нулями.
2. Код ошибки.

```
static int rounding(longDoubleValue *);
```

Функция округления числа во внутреннем представлении.

Аргументы:

1. Указатель на структуру, представляющую округляемое число.

Возвращаемые значения:

1. Структура, содержащая округленное (Математическое округление) число.
2. Код ошибки.

```
static int multiplication(  
    struct LongDouble *,  
    struct LongDouble *,  
    struct LongDouble *  
);
```

Функция умножения длинных чисел “столбиком”.

Аргументы:

1. Указатель на структуру представляющую первый сомножитель
2. Указатель на структуру представляющую второй сомножитель
3. Указатель на структуру, в которую будет записан результат умножения

Возвращаемые значения:

1. Результат умножения первого и второго сомножителей
2. Код ошибки

```
static int toString(LongDouble *, char **);
```

Функция перевода числа из внутреннего представления программы в строку.

Аргументы:

1. Структура, которую необходимо перевести в строковое представление, описанное в ТЗ.
2. Указатель на строку, в которую необходимо записать результат.

Возвращаемые значения:

1. Строковое представление числа из переданной структуры.
2. Код ошибки.

## **Вывод**

При выполнении лабораторной работы я ознакомился с представлением чисел в памяти компьютера, а также сам реализовал обработку и операцию умножения длинных чисел. При работе с длинной арифметикой эффективно использовать структуры, хранящие длинные числа.