



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 <вариант 2>

«Длинная арифметика»

Студент \_\_\_\_\_ Шавиш Тарек

Группа \_\_\_\_\_ ИУ7и-31Б

Преподаватель \_\_\_\_\_ Силантьева А.В.

2023 Г

## 1. Условие задачи

Смоделировать операцию умножения действительного числа в форме  $\pm m.nE\pm K$ , где суммарная длина мантиссы ( $m + n$ ) - до 40 значащих цифр, а величина порядка  $K$  - до 5 цифр, на целое число длиной до 30 десятичных 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

## 2. Техническое задание

Программа принимает две входные строки, каждая из которых содержит два действительных десятичных числа в пределах указанного диапазона. Десятичное число может быть представлено без десятичной точки, например 123. Когда в числе присутствует десятичная точка, возможны следующие представления: .00036, +165002., -456.321. Кроме того, число может быть представлено в экспоненциальной форме: 9632587E-20, 9632587E20 или 369.5248E23.

### 2.2 Описание задачи, реализуемой программой

умножение переданных ей действительных чисел и вывести его результат в форме  $\pm 0.m_1 E \pm K_1$ , где  $m_1$  - до 40 значащих цифр, а  $K_1$  - до 5 цифр.

### 2.3 Способ обращения к программе

\_С помощью команд в терминале.

### 2.4 Описание возможных аварийных ситуаций и ошибок пользователя

- Введенные числа находятся за пределами допустимого диапазона для обработки.
- Цифры введены в неправильном формате.
- При умножении числа значение показателя степени превысило допустимый диапазон  $[-99999, 99999]$ .

### 3. Описание структур данных

В программе действительные числа представлены с использованием структуры `number_attributes`. Эта структура включает следующие компоненты числа: • Знак числа (0: положительное, 1: отрицательное) • Мантисса числа • Позиция десятичной точки • Порядок числа

Каждый байт в массиве мантиссы может содержать значение от 0 до 9, представляющее один разряд мантиссы в десятичной системе счисления. Цифры хранятся в массиве с порядком от старших к младшим (Big-Endian). Степень экспоненты ограничена программно в пределах допустимого диапазона от -99999 до 99999.

Структура, содержащая характеристики числа

```
#define LEN 100
typedef struct
{
    int mantissa[LEN];
    int sign_number;
    int power;
    int exp;
} number_attributes;
```

#### 4. Описание алгоритма

Шаги алгоритма умножения следующие:

В программе используется классический алгоритм умножения "столбиком".

Процесс выглядит следующим образом:

1. Берется вторая строка, которая посимвольно обходится с конца к началу (то есть каждый символ извлекается из массива, в котором хранятся цифры числа).
2. На каждой итерации символ (цифра) умножается на первую строку, при этом результат умножения добавляется к "накопителю" (третья строка).
3. После завершения полного прохода по числу в "накопителе" будет содержаться результат умножения.

```
void multiplication(int arr1[LEN], int arr2[LEN], int len1, int len2, int
arr_result[LEN])
{
    int resultLen = len1 + len2; // Length of the result array

    // Initialize the result array with zeros
    for (int i = 0; i < resultLen; i++) {
        arr_result[i] = 0;
    }

    // Perform multiplication and addition
    for (int i = len1 - 1; i >= 0; i--) {
        for (int j = len2 - 1; j >= 0; j--) {
            int product = arr1[i] * arr2[j];
            int sum = arr_result[i + j + 1] + product;

            arr_result[i + j + 1] = sum % 10; // Update the current digit
            arr_result[i + j] += sum / 10;    // Carry over to the next digit
        }
    }
}
```

**int correctness\_number(char number[LEN]):**

Назначение: Эта функция проверяет правильность ввода строки в виде числа с плавающей запятой.

Пояснение: Он проверяет, что входная строка содержит допустимое число с плавающей запятой, и сообщает о различных ошибках, если ввод неверен.

**int correctness\_number\_int(char number[LEN])**

Назначение: Эта функция проверяет правильность ввода строки в виде целого числа.

Пояснение: Это гарантирует, что входная строка представляет собой допустимое целое число, и сообщает об ошибках, если есть какие-либо проблемы с вводом.

**int number\_int(char number[MAX\_LEN\_INT])**

Назначение: Преобразует строковое представление целого числа в числовую структуру.

Пояснение: Он анализирует входную строку и извлекает компоненты мантиссы и экспоненты, сохраняя их в структуре для дальнейшей обработки.

**int int\_to\_number(char str\_number[LEN], number\_attributes \*number)**

Назначение: Преобразует строковое представление числа с плавающей запятой в числовую структуру.

Пояснение: Эта функция анализирует входную строку, разделяя компоненты мантиссы и экспоненты, и сохраняет их в структуре для дальнейших вычислений.

**int float\_to\_number(char str\_number[LEN], number\_attributes \*number):**

Назначение: Преобразует строковое представление числа в экспоненциальной системе счисления в числовую структуру.

Пояснение: Аналогично предыдущим функциям, она анализирует входную строку, извлекает мантиссу и показатель степени и сохраняет их в структуре для дальнейшего использования.

```
void print_result(number_attributes number, int len)
```

Назначение: Выводит результат обработки чисел.

Пояснение: Он принимает числовую структуру и печатает отформатированный результат в научной нотации (например, "0.123e4").

## Тесты

N	input	Output	Объяснение ошибки
1	999999999999999999 99999999 * 2	0.1999999999 999999999999 999998E30	-
2	1e1 * 0	0.0e0	-
3	0 * 0	0.0e0	-
4	10 * 10	0.1e3	-
5	4 * 999999999999999999	0.3999999999 99999996E19	-
6	-12 * 12	-0.144e3	-
7	0.01e-99999 * 100	0.1e-99998	-
8	-12 * -12	-0.144e3	-
9	12 * 12	0.144e3	-
10	0.0001e-99999 * 1	Error	Переполнение порядка

11	Ert	Invalid	Некорректный формат ввода
12	1e1 * 99999999999999999999 92	Big mantisa	Больше 30 цифр в целом числе
13	1e999999999999 * 1	Order big	Больше 5 цифр в порядке
14	99999999999999999999 99999999999999999999 99999999999999999999 99999999999999999999 999999 * 1	Big mantissa	Больше 40 цифр в мантиссе

## 6. Контрольные вопросы :

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

#### Целые числа:

- 8-битные (byte): от -128 до 127 (с знаком) или от 0 до 255 (без знака).
- 16-битные (short): от -32,768 до 32,767 (с знаком) или от 0 до 65,535 (без знака).
- 32-битные (int): от -2,147,483,648 до 2,147,483,647 (с знаком) или от 0 до 4,294,967,295 (без знака).
- 64-битные (long): от -9,223,372,036,854,775,808 до 9,223,372,036,854,775,807 (с знаком) или от 0 до 18,446,744,073,709,551,615 (без знака).

#### 2. Числа с плавающей запятой:

- 32-битные (float): приблизительно от  $-3.4 \times 10^{38}$  до  $3.4 \times 10^{38}$  с плавающей точкой.
- 64-битные (double): приблизительно от  $-1.7 \times 10^{308}$  до  $1.7 \times 10^{308}$  с плавающей точкой.

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

Точность представления вещественных чисел зависит от максимального количества разрядов, отведенных под хранение мантиссы. При выходе мантиссы из разрядной сетки происходит округление и точность теряется. Под хранение мантиссы числа типа double отводится 52 двоичных разряда, что соответствует не более, чем 20 десятичным разрядам.

### **Какие стандартные операции возможны на числами?**

Сложение, вычитание, умножение, деление, сравнение, деление нацело, взятие остатка.

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

Самым выгодным по памяти вариантом является массив элементов типа char. Так же можно использовать целые типы (int, short int), однако по сравнению с массивом символов массив целых чисел будет занимать больше памяти.

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

Производить действия над числами поэлементно, заранее сохранив цифры числа в массив. Для выполнения операций использовать алгоритм «столбиком или его модификации.



## 7. Вывод

Для работы с числами в компьютере существуют разные методы. Если требуемый диапазон обрабатываемых целых значений находится в пределах диапазона значений машинного слова, предпочтительно использовать встроенную целочисленную арифметику, которая находится внутри процессора. В противном случае, необходимо создавать так называемую "длинную арифметику", которая позволяет программно выполнять арифметические операции над большими числами. Длинная арифметика с целыми числами может быть реализована в виде арифметики "в столбик".

При обработке действительных чисел существуют два основных метода: числа с фиксированной точностью и числа с плавающей запятой. Числа с фиксированной точностью обрабатываются аналогично целым числам, с дополнительным соглашением о местоположении десятичной точки. Числа с плавающей запятой могут быть обработаны аппаратно в соответствии с спецификацией FPU (чаще всего это стандарт IEEE754). Если нет поддержки FPU или требуется оперировать числами, выходящими за пределы представимых значений, то также необходима реализация длинной арифметики.

Для хранения и обработки таких чисел удобно использовать структуру, имеющую поля: знак, мантисса и экспонента. Мантисса и экспонента могут быть представлены как длинные целые числа. Операции над такой структурой включают в себя обработку её частями: мантисса и экспонента обрабатываются отдельно в арифметических операциях, а затем приводятся к общему виду при нормализации числа.