|  |  |
| --- | --- |
| Изображение выглядит как герб, эмблема, символ, нашивка  Автоматически созданное описание | **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**

**ПО ДИСЦИПЛИНЕ «ТИПЫ И СТРУКТУРЫ ДАННЫХ»**

***«Длинная арифметика»: Обработка больших чисел***

**Вариант №2**

Студент: **Коротков Е.Д.**

Группа: **ИУ7-34Б**

Преподаватель: **Барышникова М. Ю.**

**2025 г.**

**Описание условия задачи**

Смоделировать операцию умножения действительного числа в форме ±m.n Е ±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 ±0.m1 Е ±K1, где m1 - до 30 значащих цифр, а K1 - до 5 цифр.

**Техническое задание**

**Исходные данные:**

**Вещественное число:** строка, содержащая число в формате: [+-]?m.?n(E[+-]?K)? где (m+n) до 40 значащих цифр, K по модулю не превышает 99999. Если не указан знак (в обоих случаях), то он принимается за положительный (“+”)

**Вещественное число:** строка, содержащая число в формате: [+-]?m1, где m1 до 30 значащих цифр. Если не указан знак, то он принимается за положительный (“+”).

**Выходные данные:** строка, содержащая вещественное число в формате: [+-]0.m1 E[+-]K1, где m1 – мантисса, длиной до 30 символов, а K1- порядок (до 5 цифр)

**Реализуемая задача:**

Умножение вещественного числа на целое.

**Способ обращения к программе:**

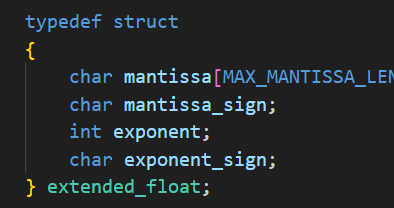
Запуск программы в CLI “./a.exe”. После запуска происходит ввод чисел. Признак окончания ввода – переход на следующую строку (ENTER)

**Аварийные ситуации:**

1. Пустой ввод
2. Слишком большой ввод числа
3. Некорректный ввод числа
4. Получение машинного нуля
5. Получение машинной бесконечности

**Внутренние структуры данных**

Введенные числа (и дробное, и целое) приводятся к единому – нормализованному виду и записываются в структуру extended\_float



Подробнее о каждом поле:

Mantissa- массив байтов для хранения мантиссы числа

Mantissa sign – байт знака мантиссы (+ или -)

Exponent – число, содержащее порядок числа

Exponent\_sign – байт знака порядка (+ или -)

И вещественное, и дробное число через алгоритмы, описанные далее, приводятся к единому виду и записываются в структуру extended float  
  
**Описание алгоритма:**

1. Программа запрашивает вещественное число, считывает его, проверяет на валидность (математическую корректность записи числа), нормализует и записывает в структуру extended\_float
2. Программа запрашивает целое число, считывает его, проверяет на валидность (математическую корректность записи числа), нормализует и записывает в структуру extended\_float
3. Программа производит умножение вещественного и целого чисел, округляя результат до 30 символов (при необходимости), выполняет проверку на переполнение и получение крайних значений (машинный ноль и машинная бесконечность)
4. Вывод результата на экран в нормализованном виде

Основные функции:

get\_sign(char letter)

Эта функция определяет знак числа. Если первый символ строки — это + или -, она возвращает соответствующий знак. В противном случае, по умолчанию возвращается +.

print\_lineal(int max\_digits)

Вспомогательная функция, которая печатает линейку с разметкой для удобства пользователя. Она помогает визуально оценить длину введённых чисел.

print\_num\_struct(extended\_float\* num)

Форматирует и выводит число, хранящееся в структуре extended\_float, в заданном виде: ±0.mantissaE±exponent.

string\_to\_arr\_num(char \*s, int \*d)

Преобразует строку, представляющую мантиссу, в массив целых чисел. Также производит переворачивание чисел, чтобы каждая цифра числа соответствовала своему разряду. Например, число 123 будет храниться как массив [3, 2, 1].

validate\_number(char \*s)

Выполняет проверку на корректность введённого числа. Она проверяет наличие допустимых символов (+, -, ., E), правильное расположение точки и экспоненты, а также следит за тем, чтобы количество значащих цифр в мантиссе и порядке не превышало заданный лимит.

norm\_res(extended\_float\* num)

Нормализует результат вычисления. Если после нормализации мантисса оказывается пустой (например, для результата 0), функция устанавливает её значение на "0".

norm\_double(char \*str, char \*out)

Эта функция нормализует действительное число в стандартный формат ±0.mantissaE±exponent

norm\_int(char\* str, char\* out)

Эта функция нормализует целое число в стандартный формат ±0.mantissaE±exponent. Она убирает ведущие нули и вычисляет порядок, который равен количеству цифр в числе.

norm\_exp(const char \*str, char \*out)

Эта функция обрабатывает числа, содержащие экспоненциальную часть (например, 12E+5). Она вызывает одну из нормализующих функций (norm\_double или norm\_int) для основной части числа, а затем корректирует порядок с учетом исходной экспоненты.

move\_num\_to\_struct(extended\_float \*num\_struct, char \*number)

Функция разделяет строку на знак мантиссы, саму мантиссу и порядок и записывает нормализованное число из временного буфера в структуру extended\_float.

parse\_num(extended\_float \*num)

Основная функция для ввода и анализа числа. Она считывает строку, вызывает validate\_number для проверки, а затем, в зависимости от формата числа, использует одну из функций нормализации (norm\_double, norm\_int или norm\_exp)

mult\_arr(int \*a, int len\_a, int \*b, int len\_b, int \*res)

Выполняет умножение двух чисел, представленных в виде массивов цифр. Она имитирует умножение «в столбик» (каждый элемент первого массива умножается на каждый элемент второго, и результаты складываются)

mul\_mant\_strings(char \*a, char \*b, char \*out)

Функция преобразует строковые представления мантисс в массивы цифр, вызывает mult\_arr для вычисления произведения и затем преобразует результат обратно в строку.

round\_mantissa(char \*mantissa, int \*exponent)

Выполняет округление мантиссы до 30 значащих цифр. Если 31-я цифра равна или больше

5, она округляет мантиссу в большую сторону и, при необходимости, корректирует порядок.

multiply\_long(extended\_float \*a, extended\_float \*b, extended\_float \*res)

Функция для выполнения умножения: вызывает mul\_mant\_strings для умножения мантисс, вычисляет итоговый порядок, затем вызывает round\_mantissa для округления и проверяет значение на машинный нуль.

main()

Основная функция программы

Выводит сообщения для пользователя.

Вызывает parse\_num для считывания двух чисел (действительного и целого) с проверкой на корректность.

Выполняет проверку на то, что второе число является целым, и завершает программу с ошибкой, если это не так.

Вызывает multiply\_long для вычисления произведения.

Если вычисление прошло успешно, вызывает norm\_res для финальной нормализации результата и печатает его на экран.

**Набор тестов**

|  |  |  |  |
| --- | --- | --- | --- |
| Описание теста | Вещественное число | Целое число | Результат |
| Умножение двух положительных чисел | 15E5 | 2 | +0.3E+7 |
| Умножение положительного целого и отрицательного вещественного числа | -20e5 | 2 | -0.4E+7 |
| Умножение положительного вещественного и отрицательного целого числа | 20e5 | 2 | -0.4E+7 |
| Умножение вещественного числа с отрицательной мантиссой | 15e-5 | 13 | +0.195E-2 |
| Умножение целого на целое | 15 | 15 | +0.225E+3 |
| Умножение максимально возможного вещественного на 1 | 999…999 (40 девяток) | 1 | +0.1E+41 |
| Умножение максимально возможного вещественного на 3 | 999…999 (40 девяток) | 3 | +0.3E+41 |
| Длинное вещественное число | 1.1234567891234567891234567891234567891234 | 1 | +0.112345678912345678912345678912345678912E+1 |
| Округление | 15.999999999999999999999999999999 | 3 | +0.48E+2 |
| Максимальное целое и макс. Вещественное | 999…999 (40 девяток) | 999…999 (30 девяток) | +0.999999999999999999999999999999E+70 |
| Вещественное с слишком большой экспонентой | 9E999999 | 1 | Input error (float num) |
| Вещественное с длинной мантиссой | 0.9…999 (41 девятка) | 1 | Input error (float num) |
| Большое целое число | 1 | 9..9(35 девяток) | Input error (int num) |
| Вещественное вместо целого | 15 | 15E3 | Input error (float num) |
| Переполнение порядка | 15E99999 | 10 | Error: exponent value out of range [-99999, 99999] |
| Пустой ввод |  |  | Input error (float num) |
| Вещественное меньше 1 | 0.5 | 25 | +0.125E+2 |
| Округление результата с увеличением порядка |  |  |  |
| Некорректное вещественное | ASDASD |  | Input error (float num) |
| Некорректное целое | 1.2E. | AA | Input error (float num) |
| Некорректное вещественное | 12E5.5 | 1 | Input error (float num) |
| Некорректное вещественное | 15.5.5E12 | 1 | Input error (float num) |
| Некорректное вещественное | 12E12E12 | 5 | Input error (float num) |

**Вывод**

Для реализации операций и действий над числами, допустимые значения которых не предусмотрены стандартными типами данных, структуры данных и операции над ними реализуются непосредственно программистов. В рамках данной лабораторной работы реализован тип (структура) для вещественных чисел, имеющих мантиссу длиной не более 40 и порядок, длиной не более 5, а также операция их умножения, позволяющая получать в результате вещественное число, с мантиссой, длиной не более 30 символов и порядком не превосходящим по длине 5. Тип (структура) extended\_float может использоваться в финансовой, а также инженерной сферах – там, где требуется высокая точность вычислений и числа, которые выходят за диапазон допустимых значений стандартных типов данных.