



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

ИУ

КАФЕДРА

ИУ7

## ОТЧЕТ ПО ЛР1 ТИПОВ И СТРУКТУР ДАННЫХ

Студент

Андреев Александр Алексеевич  
фамилия, имя, отчество

Группа

ИУ7-34Б

Тип практики

учебная

Студент

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
фамилия, и.о.

Преподаватель

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
фамилия, и.о.

Оценка

21 сентября 2020

# Оглавление

<b>1. Описание условия задачи</b>	<b>1</b>
<b>2. Описание ТЗ, включающее внешнюю спецификацию</b>	<b>2</b>
a. Описание исходных данных и результатов	2
b. Описание задачи, реализуемой программой	2
c. Способ обращения программы	2
d. Описание возможных аварийных ситуаций и ошибок пользователя	3
<b>3. Описание внутренних СД</b>	<b>4</b>
<b>4. Описание алгоритма</b>	<b>4</b>
Бриф работы алгоритма:	4
Расширенное представление работы алгоритма программы:	4
Считывание целого и вещественного чисел с их верификацией, и последующим представлением в структуру.	4
Умножение чисел	5
Вывод полученного значения произведения чисел	5
<b>5. Набор тестов с указанием, что проверяется</b>	<b>5</b>
<b>6. Выводы по проделанной работе</b>	<b>7</b>
<b>7. Ответы на контрольные вопросы</b>	<b>7</b>

# 1. Описание условия задачи

Необходимо составить программу, моделирующую умножение целого числа длиной до 30 десятичных цифр на действительное число в форме  **$+m.nE+K$** , где суммарная длина мантиссы ( **$m + n$** ) - до 30 значащих цифр (без учета точки), а величина порядка  **$K$**  - до 5 цифр. Составленная программа должна выдавать их произведение в форме  **$+0.m1EK1$** , где  **$m1$**  - до 30 значащих цифр, а  **$K1$**  до 5 цифр.

## 2. Описание ТЗ, включающее внешнюю спецификацию

### а. Описание исходных данных и результатов

В качестве исходных данных программа получает на вход два числа, введенных в двух различных строках, где в первой - вводится целое число, и во второй - вводится вещественное число в формате  **$+m.nE+K$** .

Ограничения на введенное целое число:

- До 30 значащих цифр

Ограничения на введенное вещественное число:

- ( **$m + n$** ) - до 30 значащих цифр
- величина порядка  **$K$**  - до 5 цифр

После соответствующего пояснения программа должна выдавать произведение введенных целого числа и вещественного числа в определенной форме  **$+0.m1EK1$** , где  **$m1$**  - до 30 значащих цифр, а  **$K1$**  до 5 цифр.

В случае отсутствия возможности произвести умножение программа должна сообщить об этом пользователю в консоле.

### б. Описание задачи, реализуемой программой

При хранении вещественных чисел в оперативной памяти компьютера необходимо обеспечить следующий формат их представления (см. Рис. 1)

Рис. 1



числа),  **$order$**  (Порядок числа).

Данный формат представления данных должен быть обеспечен особой структурой хранения данных ***number***, содержащей четыре различных поля: ***mantisa\_sign*** (Знак мантиссы числа), ***mantisa*** (Мантисса числа), ***order\_sign*** (Знак порядка

### с. Способ обращения программы

Скомпилированная программа запускается командой “./a.out” на Unix и “./a.exe” на Windows.

При запуске программа должна вывести инструкцию по своему применению:

“Программа вычисления произведения целого и вещественного чисел, представленных в формате +-N и +-m.nE+-K, запущена.

Далее после ввода пользователем целого и вещественного чисел программа должна вывести произведенные вычисления на экран в формате “Произведение введенного целого и вещественных чисел: X”, где X - это Произведение введенного целого и вещественных чисел.

При наличии аварийных ситуаций и ошибок пользователя программа должна вывести соответствующее сообщение и не должна завершить свою работу абортно.

### d. Описание возможных аварийных ситуаций и ошибок пользователя

В представленной ниже **Таблице 1** отражены действия программы при различных возможных, допущенных пользователем, при ее использовании ошибках.

Табл.1

Действие программы при различных ошибках		
№	Действие пользователя	Реакция программы
1	В введенном целом числе пользователя, присутствуют недопустимые условием работы программы символы.	Программа выведет сообщение об ошибке “Вы некорректно ввели целое число. Исправьте и повторите заново.”
2	Разрядность введенного пользователем целого числа оказалась больше 30 .	Программа выведет сообщение об ошибке “Вы ввели целое число большее 30 разрядов. Уменьшите число и повторите попытку”
3	В введенном вещественном числе пользователя, присутствуют недопустимые условием работы программы символы.	Программа выведет сообщение об ошибке “Вы некорректно ввели вещественное число. Исправьте и повторите заново.”
4	Размер введенной пользователем мантиссы вещественного числа оказался больше 30 символов.	Программа выведет сообщение об ошибке “Вы ввели вещественное число с мантиссой большее 30 разрядов. Исправьте и

		повторите попытку”
5	Размер введенного пользователем порядка вещественного числа оказался больше 5 символов.	Программа выведет сообщение об ошибке “Вы ввели вещественное число с порядком большее 5 разрядов. Исправьте и повторите попытку”

### 3. Описание внутренних СД

Считанные из консоли числа внутри программы хранятся в виде структуры **number**, которая представляет из себя (Рис. 2)

Рис. 2

```
typedef struct number {
    char mantisa_sign;           // Знак мантиссы числа '-'/'+'
    char mantisa[MAX_NUMBERS]; // Мантисса числа

    char order_sign;           // Знак порядка числа '-'/'+'
    int order;                 // Порядок числа в виде int
} number;
```

Удобство хранения знака мантиссы и порядка числа в виде char '-'/'+' помогает избежать дополнительного кода в сравнении с хранением 1/0.

А использование хранения мантиссы числа в структуре в виде массива char-ов обусловлено тем, что в мантиссе числа может иметься точка.

### 4. Описание алгоритма

Бриф работы алгоритма:

1. Считывание целого и вещественного чисел с их верификацией, и последующим представлением в структуру.
2. Умножение чисел
3. Вывод полученного значения произведения чисел

## Расширенное представление работы алгоритма программы:

1. Считывание целого и вещественного чисел с их верификацией, и последующим представлением в структуру.
  - a. Считывание целого числа и вещественных чисел
  - b. Верификация целого и вещественных чисел
  - c. Представление чисел, как **number** структур **first\_number**, **second\_number**
2. Умножение чисел
  - a. Создание итоговой **number** структуры **answer\_number**, размером на один больше суммы размеров структур **first\_number** и **second\_number** и заполнения ее мантиссы нулями
  - b. Умножение знаков мантисс структур **first\_number** и **second\_number** в **answer\_number**
  - c. Умножение переведенных из массива char-ов в массив int-ов мантисс структур **first\_number** и **second\_number** в **answer\_number**
    - i. Обратное чтение элементов, переведенных из массива char-ов в массив int-ов, мантиссы структуры **first\_number** и ее последовательное умножение на мантиссу, также переведенную из массива char-ов в массив int-ов, структуры **second\_number**
    - ii. Каждое произведение суммируется на соответствующие позиции в мантису **answer\_number**
  - d. Умножение знаков порядка **first\_number** и **second\_number** в **answer\_number**
  - e. Умножение порядков структур **first\_number** и **second\_number** в **answer\_number**
3. Вывод полученного значения произведения чисел
  - a. Форматирование вывода
  - b. Непосредственный вывод в консоль

## 5. Набор тестов с указанием, что проверяется

В представленных ниже **Таблица 2** отражены тестирование устойчивости работы консольного меню программы и демонстрация устойчивости работы программы к различному типу выполняемых с ней операций пользователем соответственно.

Табл. 2

Тестирование устойчивости работы программы			
№	Ввод пользователя	Реакция программы	Что проверяется данной операцией?

1	-123 -12.123E-10	Программа выводит “Произведение введенного целого и вещественных чисел: +1491.129E-10”	Произведение чисел с одинаковыми знаками
2	-123 +12.123E-10	Программа выводит “Произведение введенного целого и вещественных чисел: -1491.129E-10”	Произведение чисел с разными знаками
3	-999 -12.12.3E-10	Программа выводит “Вы некорректно ввели вещественное число. Исправьте и повторите заново.”	Ошибка в вещественном числе
4	-1234567890 12345678901 23456789012 34567890 -12.123E-10	Программа выводит “Вы ввели целое число большее 30 разрядов. Уменьшите число и повторите попытку”	Разрядность целого числа больше допустимой
5	-123456789 -12.12345678 90123456789 01234567890 1234567890E -10	Программа выводит “Вы ввели вещественное число с мантиссой большее 30 разрядов. Исправьте и повторите попытку”	Размер мантиссы вещественного числа больше допустимой
6	-123456789 -12.123E-111 111	Программа выводит “Вы ввели вещественное число с порядком большее 5 разрядов. Исправьте и повторите попытку”	Размерность порядка вещественного числа больше допустимой
7	-1234567891 23456789123 45678912345 67891234567 89123456789 -12.123E-111 11	Программы выведет “Ваши действия привели к переполнению. Исправьте и повторите ошибку.”	Тестирование переполнения
8	0 -12.123E-111 11	Программа выводит “Произведение введенного целого и вещественных чисел: +0.0E+0”	Тестирование умножения нулевого целого на ненулевое вещественное
9	123 -0.0E-0	Программа выводит “Произведение введенного целого и вещественных чисел: +0.0E+0”	Тестирование умножения ненулевого целого на нулевое вещественное
10	+2	Программа выводит	Тестирование округления, когда

	999999999999 999999999999 99999.9E+0	“Произведение введенного целого и вещественных чисел: +0.2000000000000000000000 000000000E+29”	меняются все цифры числа
11	777777777777 777777777777 77777777 5.0E+0	Программа выводит “Произведение введенного целого и вещественных чисел: +0.3888888888888888888888 88888888900E+31”	Тестирование округления, когда меняются не все цифры числа
12	100 0.001E+9999 9	Программа выводит “Произведение введенного целого и вещественных чисел: +0.100E+99999”	Проверка работы верификации на выводимых значениях на границе

## 6. Выводы по проделанной работе

Не всегда стандартно используемые типы и структуры данных позволяют выполнять необходимые операции, как произошло в данной лабораторной работе. Мы бы не смогли умножить целое число размером до 30 символов на вещественное число с мантиссой из 30 символов, так как ограничения стандартного float 9 цифрами не позволило бы это сделать. Именно поэтому существуют разные типы представления данных, которые мы можем использовать под ту или иную задачу. Мы экспериментально доказали, что для данной задачи, используя массив, мы можем произвести умножение.

## 7. Ответы на контрольные вопросы

- Каков возможный диапазон чисел, представляемых в ПК?  
Возможный диапазон чисел зависит от разрядности системы компьютера.  
Так при разрядности системы 64 длина int равна числу 2 в степени 62.  
Для float с плавающей точкой - 4 Байта  
Для double с плавающей точкой - 8 Байт  
Для мантиссы float - 23 бита  
Для экспоненты float - 8 бит  
Для мантиссы float - 52 бита  
Для экспоненты float - 11 бит
- Какова возможная точность представления чисел, чем она определяется?  
Точность представления чисел определяется длиной массива и его ограничениями. Размер представленного числа зависит же от разрядности системы, например в 64-разрядной системе его максимальное значение 2 в степени 64.
- Какие стандартные операции возможны над числами?  
Над числами возможны операции сложения, умножения, деления, вычитания.



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

Программист может выбрать массив `char/int`.

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

Можно написать самостоятельные операции, выполняемые над числами и их использовать. В этих операциях нужно выбрать подходящий тип данных. Данные операции можно выполнять поразрядно.