



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 **«ДЛИННАЯ АРИФМЕТИКА»**

Студент Шанин Матвей Владимирович

Группа ИУ7 – 31Б

Преподаватель Барышникова М. Ю.

Отчет по лабораторной работе №1 Типы и структуры данных

Шанин Матвей ИУ7-31Б

1. Описание условия задачи

Задача представляет из себя написание программы, которая считывает от пользователя целое и вещественное число, которые могут выходить за рамки машинного представления и выполнения над ними операции умножения

2. ТЗ

а) Первым числом пользователь вводит целое число (отрицательное или положительное), ограниченное длиной в 40 десятичных цифр, знак числа и ведущие нули не входят в его длину. Далее пользователь вводит вещественное число в любой форме. Оно ограничено длиной мантииссы в 30 десятичных цифр и величиной экспоненты - от -99999 до 99999, число может быть отрицательным или положительным. Знак и ведущие нули не входят в длину.

б) Программа выполняет умножение введенного целого числа на введенное вещественное и выводит результат в нормализованном виде $\pm 0.m_1 E \pm K_1$ - m_1 - мантиисса до 40 символов, K_1 - значение экспоненты от -99999 до 99999.

в) Пользователь общается с программой в консоли

`./app.exe`

Ввод дополнительных параметров командной строки не требуется.

г) Возможные ошибки:

Ввод:

1. Посторонние символы в числе (буквы и т. д.)
2. Неверный формат вещественного числа
3. Целое длиннее 40 символов
4. Мантисса вещественного числа более 30 символов
5. Экспонента не в диапазоне

Вычисления:

1. Экспонента результата вышла из диапазона

3. Описание внутренних СД

Большие числа в программе представлены через структуры данных:

typedef struct

{

int nums[MAX_LEN_INT]; - массив цифр большого
целого

size_t len; - длина целого

bool is_negative; - поле, которое хранит информацию о
знаке числа

} big_int;

typedef struct

{

int mantissa[MAX_LEN_MANTISSA_IN]; - массив цифр
мантииссы большого вещественного

int point_index; - индекс точки, отсчитывая с конца

числа, если ее нет, то 0

size_t len_mantissa; - длина мантиссы вещественного
int exp; - экспонента вещественного
bool is_negative; - поле, хранящее знак числа
} big_float;

typedef struct

{
 int mantissa[MAX_RESULT]; - массив мантиссы
результата умножения
 size_t start; - индекс, с которого начинаются цифры в
нужном диапазоне
 size_t len_mantissa; - длина мантиссы результата
 int exp; - экспонента результата
 bool is_negative; - знак результата
} big_float_result;

4. Описание алгоритма

Алгоритм умножения больших чисел предполагает из себя простое умножение в столбик, воспользовавшись таким свойством, что

$$\text{num} = \text{num}[0] * p^{**0} + \text{num}[1] * p^{**1} \dots + \text{num}[n] * p^{**n}$$

$$a * b = a[i] * p^i * a[j] * p^j = ab[i + j] * p^{(i + j)}$$

$$ab[i + j] += a[i] * b[j]$$

Код:

```

for (size_t i = 0; i < len_int; i++)
{
    long remainder = 0;
    for (size_t j = 0; j < len_float; j++)
    {
        int cur = result->mantissa[i + j] + integer->nums[i] *
floating->mantissa[j] + remainder;
        remainder = cur / 10;
        result->mantissa[i + j] = cur % 10;
    }

    int cur = len_float;
    while (remainder)
    {
        result->mantissa[i + cur] = remainder % 10;
        cur++;
        remainder /= 10;
    }
}

```

В переменной **remainder** хранится остаток, если он остался после умножения, то дописываем оставшиеся цифры пока остаток не закончится

5. Описание тестов

Позитивные тесты

Ввод:

№	Входные данные	Выходные данные	Описание

0	0000123 1	Результат: 0.123 E3	Ведущие нули
1	123 1	Результат: 0.123 E3	Беззнаковое число
2	+123 1	Результат: 0.123 E3	+
3	-123 1	Результат: - 0.123 E3	-
4	9999999999 9999999999 9999999999 9999999 1	Результат: 0.9999999999 9999999999 9999999999 9999999999 E40	Беззнаковое 40 цифр
5	- 9999999999 9999999999 9999999999 9999999 1	Результат: - 0.9999999999 9999999999 9999999999 9999999999 E40	с - 40 цифр
6	+9999999999 9999999999 9999999999	Результат: 0.9999999999 9999999999	с + 40 цифр

	99999999 1	999999999999 99999999 E40	
8	1 123	Результат: 0.123 E3	Целое число без знака
9	1 -123	Результат: - 0.123 E3	Целое число с -
10	1 1.0	Результат: 0.10 E1	Число с точкой без эксп. без знака
11	1 -10.0	Результат: - 0.100 E2	Число с точкой без эксп. с -
12	1 +10.0	Результат: 0.100 E2	Число с точкой без эксп. с +
13	1 .123	Результат: 0.123 E0	Точка в начале
14	1 123.	Результат: 0.123 E3	Точка в конце

15	1 123.321	Результат: 0.123321 E3	Точка в середине
16	1 123 e2	Результат: 0.123 E5	ЧПТ с эксп. - без знака
17	1 123 e-2	Результат: 0.123 E1	ЧПТ с эксп. - с -
18	1 123 e+2	Результат: 0.123 E5	ЧПТ с эксп. - с +
19	1 0.0001 e99999	Результат: 0.1 E99996	Экспонента 99999
20	1 1000 e-99999	Результат: 0.1000 E- 99995	Экспонента - 99999
21	1 0.0001 e+99999	Результат: 0.1 E99996	Экспонента +99999

22	1 123 e 123	Результат: 0.123 E126	После E - пробел
23	1 123 e123	Результат: 0.123 E126	После E - нет пробела
24	1 0000012.2 e1	Результат: 0.122 E3	Ведущие нули у мантиисы
25	1 123.2 e00002	Результат: 0.1232 E5	Ведущие нули у экспоненты

Вычисления:

№	Входные данные	Выходные данные	Описание
26	2 5	Результат: 0.10 E2	Целое на целое

27	3 0.1	Результат: 0.3 E0	Целое на вещественн ое с точкой
28	1 -2	Результат: - 0.2 E1	Положитель ное на отрицатель ное
29	3 4.2	Результат: 0.126 E2	Положитель ное на положитель ное
30	-2 -5	Результат: 0.10 E2	Отрицатель ное на отрицатель ное
31	2 14.4 e2	Результат: 0.288 E4	Целое на вещественн ое с экспонентой

32	111111111111 111111111111 111111111111 11111111 1.0	Результат: 0.1111111111 111111111111 111111111111 1111111111 E40	Результат - 40 знаков в мантиссе
33	1 1 e999998	Результат: 0.1 E99999	Результат - 99999 в экспоненте
34	1 0.01 e-99998	Результат: 0.1 E-99999	Результат - - 99999 в экспоненте
35	999999999999 999999999999 999999999999 99999999 999999999999 999999999999 99999999	Результат: 0.9999999999 999999999999 999999999989 9999999999 E70	Результат с округление м

36	1 0	Результат: 0.0 E0	Результат - ноль
-----------	----------------	------------------------------	-----------------------------

Негативные тесты

Ввод:

№	Входные данные	Описание
1	asda 2	Буквы в целом
2	2	Пустая строка в целом
3	+	+
4	-	-
5	112093812903812 098310293091283 091283901823091 28390182	Длина целого > 40
6	1231232.22323	Точка в целом числе
7	1 qwewqe	Буквы в вещественном

8	1	Пустая строка в вещественном
9	1 123123123231231 231231231231231 231231231231231 23123	Длина мантиссы > 30
10	1 123 e999999	Экспонента > 99999
11	1 123 e-999999	Экспонента < - 99999
12	1 123e3	Неверная форма вещественного числа

Вычисления

№	Входные данные	Описание
13	9999999999999999 9999999999999999 9999999 999 e99999	Экспонента результата > 99999

6. Вывод

В результате работы была написана программа, реализующая умножение целого на вещественное чисел, программа может работать с числами выходящими за рамки машинного представления. Программа покрыта тестами разной

Контрольные вопросы:

1. Каков возможный диапазон чисел, представляемых в ПК?

В зависимости от разрядности процессора и выбранного типа переменной. Макс. значение для хранения целого числа у 64 битного процессора **18 446 744 073 709 551 615** (тип long). Максимально точный тип для хранения вещественных чисел - double. Макс. значение вещественного числа = **1.7E+308**

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

Точность зависит от кол-ва выделяемой памяти для хранения мантииссы. для double это 52 бита. Чем больше битов выделяется для мантииссы, тем точнее вещественное число

3. Какие стандартные операции возможны над числами?

Операции сложения, вычитания, умножения, деления, взятие остатка, сравнение, возведение в степень

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

Программист может выбрать хранить информацию о числах в структурах, где каждое поле будет предоставлять данные о большом числе. Например для хранения цифр числа подойдет статический массив типа `int`, для знака `bool` переменная, для экспоненты поле типа `int`

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

Можно сохранять информацию о числах в структуры, в которых хранить цифры чисел в массивах. Для операций над такими представлениями можно воспользоваться алгоритмами умножения\деления\сложения\вычитания в столбик