



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 Нисуев Нису Феликсович

Группа ИУ7 – 32Б

Преподаватель Силантьева Александра Васильевна

2023 г.

ОГЛАВЛЕНИЕ

Оглавление	2
Описание условия задачи	3
Описание технического задания	3
Описание структуры данных	4
Описание алгоритма	5
Набор тестов	6
Ответы на контрольные вопросы	6
Вывод	8

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Смоделировать операцию умножения целого числа длиной до 40 десятичных цифр на действительное число в форме $(+/-)m.n E (+/-)K$, где суммарная длина мантиссы $(m+n)$ - до 30 значащих цифр, а величина порядка K - до 5 цифр. Результат выдать в форме $(+/-)0.m1 E (+/-)K1$, где $m1$ - до 40 значащих цифр, а $K1$ - до 5 цифр.

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

ВХОДНЫЕ ДАННЫЕ:

Целое число:

строка, содержащая целое число в виде $\langle (+/-)m \rangle$. Если не указать знак перед числом – по умолчанию будет считаться за '+'. Длина модуля числа $\langle m \rangle$ - до 30 цифр.

Действительное число:

строка, содержащая вещественное число в виде $\langle (+/-)m.nE(+/-)K \rangle$. Если не указать знак перед числом и/или экспонентой – по умолчанию будет считаться за '+'. Суммарная длина $\langle m+n \rangle$ - до 31 цифры, включая точку; длина порядка — до 5 цифр.

ВЫХОДНЫЕ ДАННЫЕ:

Длинное число в виде $\langle (+/-)0.m1E(+/-)K1 \rangle$. Длина мантиссы $m1$ - до 40 цифр; длина порядка $K1$ — до 5 цифр.

ДЕЙСТВИЕ ПРОГРАММЫ:

Умножение целого числа на действительное

ОБРАЩЕНИЕ К ПРОГРАММЕ:

Запуск через терминал $(./build/app.exe)$. Сначала вводится целое число, потом вещественное

АВАРИЙНЫЕ СИТУАЦИИ:

1. Некорректный ввод: превышение длины при вводе целого числа (больше 40 значащих цифр).

2. Некорректный ввод: превышение длины при вводе действительного числа (больше 30 значащих цифр).
3. Некорректный ввод: В числах присутствуют недопустимые символы
4. Некорректный ввод/переполнение: превышение длины порядка (больше 5 символов).
5. Некорректный результат: Порядок полученного числа превышает 5
6. Некорректный результат: Размер мантиссы полученного числа превышает 40

ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ

Целое число обрабатывается при вводе и записывается в структуру **llong_t**.

Структура **llong_t**:

```
typedef struct {  
    bool sign;  
    char digits[LLONG_LEN + 1];  
} llong_t;
```

Поля структуры:

- **sign** – знак числа
- **digits** – массив цифр числа

Действительное число обрабатывается при вводе и записывается в структуру **lexp_t**.

Структура **lexp_t**:

```
typedef struct {  
    bool sign;  
    char mantiss[LEXP_LEN + 1];  
    int num_order;  
} lexp_t;
```

Поля структуры:

- **sign** – знак числа
- **mantiss** – массив цифр мантиссы экспоненциального числа
- **num_order** – порядок экспоненциального числа

Результирующее число вычисляется и записывается в структуру **llexp_t**. Структура результирующего числа ничем не отличается от структур **lexp_t**, кроме размера мантиссы

Структура **llexp_t**:

```
typedef struct {
    bool sign;
    char mantiss[LLEXP_LEN + 1];
    int num_order;
} lexp t;
```

Поля структуры:

- **sign** – знак числа
- **mantiss** – массив цифр мантииссы экспоненциального числа
- **num_order** – порядок экспоненциального числа

Размеры массивов чисел:

```
#define LLONG_LEN 40 // Максимальный размер целого числа
#define LLEXP_LEN 40 // Максимальный размер мантииссы длинного экспоненциального числа
#define LEXP_LEN 30 // Максимальный размер мантииссы короткого экспоненциального числа
#define ORDER_LEN 5 // Максимальный порядок экспоненциальных чисел
```

ОПИСАНИЕ АЛГОРИТМА

1. Программа считывает строку целого числа
2. Если в строке не найдено ошибок, то программа записывает число в структуру **llong_t**, иначе сообщает об ошибке и завершает программу
3. Программа считывает строку действительного числа
4. Если в строке не найдено ошибок, то программа записывает число в структуру **lexp_t** с определенными преобразованиями, иначе сообщает об ошибке и завершает программу
5. Если все данные корректны, происходит умножение первого числа на второе методом умножения в столбик
6. После умножения программа записывает получившиеся значение в результирующую структуру **llexp_t** с округлением, если оно требуется
7. После умножения результат выводится в нормализованном виде в соответствии со спецификацией, указанной в техническом задании (<+|-0.m1E+|-K1>). При порядке превышающем допустимый в техническом задании выводится ошибка.

НАБОР ТЕСТОВ

[illegible]

6	Некорректный ввод	23a	Не вводится	ERROR : Incorrect input
7	Некорректный ввод	1	100+E1	ERROR : Incorrect input
8	Некорректный ввод	1.0	Не вводится	ERROR : Incorrect input
9	Некорректный ввод	23	--23	ERROR : Incorrect input
10	Некорректный ввод	--23	Не вводится	ERROR : Incorrect input
11	Некорректный ввод	+123.45E+6.7	Не вводится	ERROR : Incorrect input
12	Некорректный ввод	1	E123	ERROR : Incorrect input
13	Слишком большое получившееся число	100	1e99999	ERROR: Calculated number too long

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Каков возможный диапазон чисел, представляемых в ПК?

Диапазон доступных чисел зависит от типа данных, который выбран, и разрядности процессора. Чем выше разрядность процессора, тем больший диапазон чисел он способен обрабатывать. Например, если процессор имеет 64 разряда, то максимальное значение для беззнакового целого числа составляет 18 446 744 073 709 551 615, что эквивалентно $2^{64} - 1$.

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

Точность представления вещественных чисел зависит от объема памяти, отведенного для хранения мантиссы числа. Например, для типа данных float (4 байта) выделяется 24 бита для мантиссы, а для double (8 байт) - 52 бита. Это позволяет мантиссе числа иметь значение, состоящее из 4 503 599 627 370 496 разрядов – 16 знаков.

3. Какие стандартные операции возможны над числами?

Над числами возможно выполнять следующие операции: сложения, вычитания, умножения, деление, взятия остатка и сравнение.

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

Программист может определить выбор между использованием структуры или массива символов. Структура может содержать информацию о знаках мантиссы, порядке,

мантиссе и порядке, а также другие дополнительные данные. Точно так же, можно оперировать с массивом символов.

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

Требуется разработать собственные библиотеки, которые будут включать в себя функции для обработки отдельных разрядов чисел, которые выходят за пределы машинного кода.

Вывод

Длинная арифметика расширяет возможности работы с числами, предоставляя высокую точность и способность обрабатывать очень большие числа. Однако, она потребляет больше памяти и может замедлять выполнение программы. Поэтому при выборе длинной арифметики нужно учитывать баланс между точностью и производительностью, а также обращать внимание на обработку ошибок и некорректных данных, чтобы избежать непредсказуемых сбоев программы.