

## Лабораторная работа №1

### Представление чисел с плавающей точкой

#### Инструментарий и требования к работе

Работа выполняется на C (C11 и новее). На сервере сборка под C17.

#### [ППА] Модификация

Проверка работы будет при прохождении на GitHub в новых репозиториях всех тестов из репозитория на всех конфигурациях.

#### Задание

Необходимо написать программу, которая позволяет программно выполнять арифметические действия с дробными числами в формате с плавающей точкой. Программа должна использовать только целочисленные вычисления и типы данных.

Аргументы программе передаются через командную строку в одном из двух вариантов:

1. **<формат> <округление> <число>**
2. **<формат> <округление> <число1> <операция> <число2>**

где:

- формат – задаёт формат представления чисел:

формат	пояснение
h	числа с плавающей точкой половинной точности (half precision floating point IEEE-754), 16 бит.
f	числа с плавающей точкой одинарной точности (single precision floating point IEEE-754), 32 бита.

- операция – символ арифметической операции: +, -, \*, /.
- число – одно или два числа, записанные в 16-ричной побитовой форме с префиксом '0x'. Гарантируется корректность входных чисел. Пример: 0xC4D
- округление – задаёт тип округления:

тип	пояснение
0	к нулю (toward_zero)
1	к ближайшему чётному (toward_nearest_even)
2	к $+\infty$ (toward_pos_infinity)
3	к $-\infty$ (toward_neg_infinity)

В случае присутствия операции, её результат должен быть вычислен в том же формате, что и входные числа.

Результат операции или единственное входное число необходимо вывести в стандартный поток вывода в следующем формате: шестнадцатеричная показательная форма, степень в десятичном представлении, знак экспоненты выводится всегда, перед точкой всегда 1 (кроме нуля). Буквы в нижнем регистре. Для single 6 цифр после точки, для half – 3 цифры. 0 выводится с экспонентой +0. NaN выводится как nan,  $-\infty$  как -inf,  $+\infty$  как inf.

Примеры:

Входные аргументы	Результат
f 0 0xB9CD542	0x1.39aa84p-104
f 0 0x414587dd * 0x42ebf110	0x1.6c1b72p+10
h 0 0x4145 * 0x42eb	0x1.238p+3

h 0 0x1 / 0x8000	-inf
------------------	------

Если на вход программе поданы некорректные данные, то необходимо завершаться с человекочитаемым сообщением о возникшей ошибке (по-английски) и правильным кодом из `return_codes.h`.

Полезности:

Целые числа: <https://www.youtube.com/watch?v=tq1brvF8cw4>

Числа с плавающей точкой:

<https://www.youtube.com/watch?v=P8qaJPADd68>

Частые ошибки (проверяющим грустно):

- использование `exit()` – читаем правила курса: <https://skkv-itmo.gitbook.io/ct-c-cpp-course/course-format/rules/reqs>
- проверка аргументов как `argc == 1` – а если вам подадут `f 0`, то что тогда будет?
- объявление функций в виде `int f()` – вспоминаем, что означает данная конструкция в C до 23 стандарта.



разумно, то должны это доказать на защите (и хорошо бы оставить комментарий для проверяющего в коде, что это вы сделали осознанно).

- копия паста функций – если 2 функции схожи в точности до каких-то констант, то стоит задуматься об объединении этого в одну функцию и передачи этого параметра через аргументы функции.