

Лабораторная работа №2

Числа с плавающей точкой

Инструментарий и требования к работе

Допустимые языки	C	C++	Python	Java
Стандарты / версии	C17	C++20	3.12.5	Temurin-21.0.4
Требования для всех работ	Правила оформления и написания работ			

Задание

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

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

1. `<fmt> <округление> <число>`
2. `<fmt> <округление> <число1> <операция> <число2>`

где:

- **fmt** – формат представления входных чисел.

формат	пояснение
h	числа с плавающей точкой половинной точности (half precision floating point IEEE-754), 16 бит.
f	формат 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 ‘0х’ в любом регистре, уместающиеся в 32 бита.
- округление – задаёт тип округления:

тип округления	пояснение	общепринятое обозначение
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 получен в том же формате, что и входные числа в соответствии с IEEE-754. В случае получения nan в результате операции с не-nan значениями, должен получиться тихий nan с остальными битами мантиисы 0 и битом знака 1. Если в операции оба операнда nan, то выживает первый.

Результат операции или единственное входное число необходимо вывести в стандартный поток вывода в формате:

<value> <hex>

где:

- **value** – число в шестнадцатеричной показательной форме; степень в десятичном представлении; знак экспоненты выводится всегда; перед точкой всегда 1 (кроме нуля). Количество цифр после точки фиксировано и равно минимально необходимому для точного вывода

произвольного числа данного формата (соответственно, у half и single разное количество). Буквы в нижнем регистре. 0 выводится с экспонентой +0. $-\infty$ как **-inf**, $+\infty$ как **inf**. NaN выводится как **nan**.

- **hex** – представление числа аналогично входным данным. Буквы в верхнем регистре, ровно 4 и 8 шестнадцатеричных цифр для **h** и **f** форматов соответственно.

Примеры:

Входные аргументы	Результат
f 0 0xB9CD542	0x1.39aa84p-104 0x0B9CD542
f 0 0x8B9CD542	-0x1.39aa84p-104 0x8B9CD542
h 1 0x4145 * 0x142eB	0x1.23cp+3 0x488F
f 2 0x1 / 0x0	inf 0x7F800000
h 3 0xFF10 + 0x7F01	nan 0xFF10

Запуск автотестов

<https://skkv-itmo.gitbook.io/ct-comp-arch-course/course-format/autotests-github>

Число попыток = 30.

из Web-интерфейса

4 варианта округления. Нужно выбрать подходящий (по умолчанию выбран округление к 0).

через CLI интерфейс:

Запуск скрипта

```
gh workflow run classroom.yml --ref main -f <field>=<value>
```

Пример

```
gh workflow run classroom.yml --ref main -f rounding="0"
```

Ключ: rounding ("Округление") default: "0"

options: "0" "1" "2" "3"