**Лабораторная работа № 4**

**ИССЛЕДОВАНИЕ ФОРМАТОВ ПРЕДСТАВЛЕНИЯ ВЕЩЕСТВЕННЫХ ЧИСЕЛ**

**Цель работы:** формирование практических навыков обоснованного выбора подходящих форматов хранения числовых данных исходя из требований технического задания.

**Задачи:** Изучить алгоритмами представления вещественных чисел в памяти ЭВМ, выполнить арифметические операции над указанными вещественными числами. Получить навыки выполнения арифметических операций с плавающей точкой.

**Содержание отчета**

После выполнения расчетной части лабораторной работы каждый студент должен подготовить отчет, в который входит:

1. титульный лист;
2. цель работы;
3. вариант задания;
4. расчетная часть в соответствии с ходом решения задачи:

табличные модели;

выполненные расчеты;

[скриншот окна проверки.](#сайт)

1. выводы по работе**.**

###### Теоретическая часть

# ПРЕДСТАВЛЕНИЕ ВЕЩЕСТВЕННЫХ ЧИСЕЛ

При решении научно-технических задач чаще имеют дело с дробными (вещественными или действительными) числами.

Существует две формы представления действительных чисел:

* ЧФЗ – в формате с фиксированной запятой (точкой),
* ЧПЗ – в формате с плавающей запятой (точкой).

Поскольку в вычислениях используются двоичные дроби, заменяющие десятичные дробные числа, то всегда существует погрешность представления для десятичных дробей. Представление в формате с фиксированной запятой используется в компьютерах, как правило, на этапе ввода и вывода чисел. В вычислениях используются нормализованные числа в формате с плавающей запятой.

* 1. ЧИСЛА С ФИКСИРОВАННОЙ ЗАПЯТОЙ. ПОГРЕШНОСТЬ ПРЕДСТАВЛЕНИЯ

Запись дробного числа с фиксированной точкой заключается в (условном) разделении имеющихся в наличии *n* разрядов на две группы бит, описывающих целую и дробную часть числа соответственно.

**Например,** 16-разрядное число может быть разделено на 12-разрядную целую часть (1 бит – знаковый и 11 бит числовых) и 4-разрядную дробную часть, что позволяет оперировать значениями, лежащими в диапазоне [−211−(1−∆); 211 −1+(1−∆)], с точностью 4 двоичных знака после запятой. Значит, абсолютная погрешность ∆ = 2−4 = 0.0625, а диапазон представления в десятичном эквиваленте [−2048,9375; 2047,9375].

В данном случае это преобразование эквивалентно условному сдвигу разделяющей запятой из крайнего правого положения (00 ... 00000**,**) в положение между четвертым и пятым разрядом (00 ... 0**,**0000), что соответствует изменению масштаба младшего разряда числа с 1 на ∆.

Операции сложения, вычитания, умножения и деления для таких чисел эквивалентны соответствующим операциям для обычных целых чисел. При преобразовании таких чисел из десятичной системы в двоичную, целую и дробную части рассматривают по отдельности.

**Пример.** Представить десятичное число 52,16110 в двоичной системе счисления.

**Решение**

1. сначала преобразуем целую часть числа:

52/2 = 26 (остаток 0, значит цифра *a0*=0) ;

26/2 = 13 (остаток 0 – цифра *a1*);

13/2 = 6 (остаток 1 – цифра *a2*);

6/2 = 3 (остаток 0 – цифра *a3*);

3/2 = 1 (остаток 1 – цифра *a4*);

1/2 = 0 (остаток 1 – цифра *a5*) .

1. затем преобразуем дробную часть числа:

0,161∙2 = 0,322 (целая часть равна 0, значит цифра *a–1* =0) ;

0,322∙2 = 0,644 (цифра *a–2* =0);

0,644∙2 = 1,288 (цифра *a–3* =1);

0,288∙2 = 0,576 (цифра *a–4* =0);

0,576∙2 = 1,152 (цифра *a–5* =1);

0,152∙2 = 0,304 (цифра *a–6* =0);

0,304∙2 = 0,608 (цифра *a–7* =0);

0,608∙2 = 1,216 (цифра *a–8*=1) …

Таким образом, 52,16110 ≈ 110100,001010012 .

Однако если мы попробуем проверить вычисления и сделаем обратный перевод двоичного числа 110100,001010012 в десятичное, то получим приближённое число 52,1601562510.

Значит, **погрешность перевода (в десятичном выражении) составила 0.00084375.** Это происходит потому, что преобразование дробной части числа умножением – процесс, в принципе, бесконечный. И когда мы останавливаемся принудительно, то отбрасываем значащие цифры числа, которые могли бы получиться при дальнейших вычислениях.

**Правило:** умножение, используемое для преобразования дробной части числа, продолжается до тех пор, пока не наступит одно из следующих событий:

1. достигнута необходимая точность представления (набрано нужное количество разрядов);

2. при очередном умножении получается точное значение единицы;

3. процесс «зацикливается».

* 1. ЧИСЛА С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

В формате с плавающей точкой числа представляются в виде:

*А10= ± т∙**Р*,

где *m* – мантисса, записанная с основанием *; P* – порядок числа.

**Пример.** Число –273,9 может быть представлено в формате ЧПЗ множеством разных способов: –273,9 = –2739∙10–1= –2,739∙102= –0,2739∙103 .

**Нормальной формой** числа с плавающей запятой называется такая форма, в которой мантисса (без учёта знака) находится на полуинтервале [0;1[.

– 0,2739×103, – 0,02739×104, – 0,002739×105, …

Как видим, и такая форма записи имеет недостаток: некоторые числа записываются неоднозначно, поэтому распространена (особенно в информатике) также другая форма записи — **нормализованная**

**Нормализованной формой** числа с плавающей запятой называется такая форма**,** в которой мантисса десятичного числа принимает значения от 1 (включительно) до 10 (не включительно), а мантисса двоичного числа принимает значения от 1 (включительно) до 2 (не включительно). В такой форме любое число (кроме 0) записывается единственным образом

– 2,739×102.

Недостаток заключается в том, что в таком виде невозможно представить 0, это особый случай машинной арифметики.

* 1. СТАНДАРТЫ ПРЕДСТАВЛЕНИЯ ЧПЗ[[1]](#footnote-1) В ЭВМ

В памяти компьютера действительные числа хранятся в нормализованной форме в двоичной системе. Большинство производителей процессоров, в том числе Intel и AMD, придерживаются международного стандарта IEEE 754–2008 регламентирующего способы хранения и методы обработки таких чисел.

Согласно этому **стандарту числа с плавающей точкой делятся на типы с одинарной, двойной и расширенной точностью** (соответствующие типы в языке C – ﬂoat, double, long double).

Каждая ячейка памяти с числом соответствующего типа в бинарном виде содержит (последовательно): 1 бит знака (S), *p* бит, выделяемых на хранение порядка числа, и *m* бит для хранения мантиссы. Все перечисленные типы отличаются только суммарным количеством бит, отводимых для их хранения, а также величинами *p* и *m*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип данных | Размер (байт) | Точность | Разрядность (бит) | | |
| *р*  порядок | *m*  мантисса | всего |
| float | 4 | **одинарная** | 8 | 23 | 32 |
| double | 8 | **двойная** | 11 | 52 | 64 |
| long double | 10 | **расширенная** | 15 | 64 | 80 |

* 1. ДИАПАЗОНЫ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ

Числа с плавающей запятой **одинарной точности** могут хранить значение в диапазоне

[–3,37∙1038; –1,17∙10–38] U [1,17∙10–38; 3,37∙1038],

где

|  |  |  |  |
| --- | --- | --- | --- |
| минимальное вещественное отрицательное число | максимальное вещественное отрицательное число | минимальное вещественное положительное число | максимальное вещественное положительное число |

Для чисел **с двойной точностью:**

[–1,8∙10308; –2,2∙10–308] U [2,2∙10–308; 1,8∙10308].

Для чисел **с расширенной точностью:**

[–1,18⋅104932; – 3,37⋅10–4932] U [3.37⋅10–4932; 1,18⋅104932].

Отдельно рассматривается число ноль (положительный и отрицательный).

Помимо действительных чисел стандарт предусматривает возможность хранения таких результатов вычислений, как ±∞, «не число» (возникает, например, в результате деления на ноль, вычисления квадратного корня из отрицательного числа), «неопределенность» (деление бесконечности на бесконечность, сложение бесконечностей и т.д.).

Ниже приведены некоторые «особые числа», представимые при кодировке IEEE 754–2008.

|  |  |  |  |
| --- | --- | --- | --- |
|  | З | Р | М |
| Положительный нуль | 0 | 00…00 | 00…00 |
|  |  |  |  |
| Отрицательный нуль | 1 | 00…00 | 00…00 |
|  |  |  |  |
| Наименьшее нормализованное положительное число | 0 | 00…01 | 00…00 |
| ≈1,8∙10–38 (при наименее возможном значении Р=1) |  |  |  |
|  |  |  |  |
| Наименьшее денормализованное (без характеристики, | 0 | 00…00 | 00…01 |
| Р=0) положительное число ≈1,4∙10–45 |  |  |  |
|  |  |  |  |
| Наибольшее положительное число | 0 | 11…10 | 11…11 |
|  |  |  |  |
| Наибольшее нормализованное отрицательное число | 1 | 00…01 | 00…00 |
| ≈1,8∙10–38 (при наименее возможном значении Р=1) |  |  |  |
|  |  |  |  |
| Наибольшее денормализованное (без характеристики, | 0 | 00…00 | 00…01 |
| Р=0) отрицательное число ≈–1,4∙10–45 |  |  |  |
|  |  |  |  |
| Наименьшее отрицательное число | 1 | 11…10 | 11…11 |
|  |  |  |  |
| Положительная бесконечность | 0 | 11…11 | 00…00 |
|  |  |  |  |
| Отрицательная бесконечность | 1 | 11…11 | 00…00 |
|  |  |  |  |
| Не число | 0 | 11…11 | 10…00 |
|  |  |  |  |
| Неопределённость | 1 | 11…11 | 10…00 |

* 1. ТОЧНОСТЬ ПРЕДСТАВЛЕНИЯ ЧИСЕЛ В ЭВМ

**Точностью числа** в ЭВМ считается количество знаков после запятой. Поскольку для расчетов мы пользуемся десятичной системой счисления, то это количество десятичных знаков после запятой.

Для одинарной точности, количество верных знаков мантиссы после десятичной запятой составляет:

log10(мантиссы) = log10(2m) = log10(223)=6,92≈**7**

примерно семь десятичных знаков (семь значащих цифр).

Это машинная точность/погрешность (машинный эпсилон) ε≈10-7 для одинарной точности. Для вычисления полной погрешности представления дробного числа нужно учитывать значение порядка, т.к. мантисса умножается на 2Р (или известны аналогичные данные для 10Р). Так, например, для минимального порядка (10–38) итоговая десятичная погрешность ε≈10–7∙10–38≈10–45, для максимального порядка (10+38) итоговая десятичная погрешность ε≈10–7∙10+38≈10–31.

Для двойной точности, количество верных знаков мантиссы после десятичной запятой составляет:

log10(252)=15,65≈**16.**

Машинный эпсилон ε≈10-16.

Для расширенной двойной точности, количество верных знаков мантиссы после десятичной запятой составляет:

log10(264)=19,26≈**19.**

Машинный эпсилон ε≈10-19.

Кроме того, представимые числа на числовой оси расположены неравномерно: плотность их возрастает при приближении к нулю и падает при удалении от нуля. Это связано с изменением значимости младшего разряда мантиссы при изменении порядка *Р*. Чем больше значение порядка, тем больше значимость младшего разряда, равная 2*Р*.

Проиллюстрируем это утверждение. Для простоты рассмотрим десятичную систему счисления с форматом S=1, Р=3 и М=5.

|  |  |  |
| --- | --- | --- |
| 1 разряд  Знак числа | *р* разрядов  порядок | *m* разрядов  мантисса |

S Р М

Рассмотрим несколько чисел на числовой оси в таком формате с плавающей точкой:

0 А Б В +∝

|  |  |  |  |
| --- | --- | --- | --- |
| число | А= 0,0012422 | Б=12,422 | В=124 220 000 000 |
| нормализованный вид числа | 1,2422⋅10–3 | 1,2422⋅10+1 | 1,2422⋅10+11 |
| запись в формате ЧПЗ | |  |  |  | | --- | --- | --- | | 0 | 1.03 | 12422 | | |  |  |  | | --- | --- | --- | | 0 | 0.01 | 12422 | | |  |  |  | | --- | --- | --- | | 0 | 0.11 | 12422 | |
| ближайшее большее число | |  |  |  | | --- | --- | --- | | 0 | 1.3 | 12423 |   1,2423⋅10–3 | |  |  |  | | --- | --- | --- | | 0 | 0.1 | 12423 |   1,2423⋅10+1 | |  |  |  | | --- | --- | --- | | 0 | 0.11 | 12423 |   1,2423⋅10+11 |
| ближайшее меньшее число | |  |  |  | | --- | --- | --- | | 0 | 1.3 | 12421 |   1,2421⋅10–3 | |  |  |  | | --- | --- | --- | | 0 | 0.1 | 12421 |   1,2421⋅10+1 | |  |  |  | | --- | --- | --- | | 0 | 0.11 | 12421 |   1,2421⋅10+11 |

Заметим, что в интервале между двумя соседними числами во всех трёх случаях не может быть записано ни одно число. Таким образом, минимальный шаг представления чисел порядка 10–3 составляет

(1,2423–1,2422)⋅10–3=0.0001⋅10–3 или 0,000 0001.

Это также будет верхней границей абсолютной погрешности представления чисел в указанном формате порядка 10–3. Любое число из интервала между 0,0012422 и 0,0012423 или между 0,0012421 и 0,0012422 можно записать как 0,0012422 ± 0.000 000 1. Аналогично и для других случаев:

|  |  |  |  |
| --- | --- | --- | --- |
| шаг представления чисел | 10–7 | 0,0001⋅10+1=10–3 | 0,0001⋅10+11=10+7 |

Отсюда видим, что плотность представимых чисел возрастает при приближении к нулю.

В ЭВМ дробные числа формата ЧПЗ представляются в двоичной системе счисления следующим образом

А2 = (–1)З⋅ 1,M ⋅ 2Х.

|  |  |  |
| --- | --- | --- |
| 1 разряд  Знак числа | *р* разрядов  характеристика | *m* разрядов  мантисса |

(+число→ З=0) Х М

(–число→ З=1)

**M** = *a−1a−2 . . . a−m* – дробная часть мантиссы, представленная в прямом коде. Причём разряд *а*0 (целая часть =1) может

* храниться в числе (в формате long double)
* не храниться в числе (в формате float или double)

**З** – знак числа, для положительных чисел равен нулю, для отрицательных чисел равен единице.

**Х** – характеристика числа, которая вычисляется как смещенный порядок: она равна истинному порядку Р, увеличенному на значение смещения. Значение смещения для трех разных форматов равно

* 28–1=127 (8 бит для float),
* 210–1=1023 (10 бит для double) и
* 214–1=16383 (15 бит для long double).

Задание порядка в форме со смещением упрощает операцию сравнения чисел с плавающей точкой.

**Характеристика** – это целое положительное и беззнаковое число. Обработка беззнаковых чисел в ЭВМ происходит проще и быстрее.

Этот способ кодирования дробных чисел используется в стандарте IEEE 754–2008.

Рассмотрим примеры представления чисел с плавающей запятой.

**Пример 1.** Возьмем число –247,375.

1. В двоичном представлении в прямом коде число будет выглядеть так:

–11110111,011.

2. [Нормализуем число.](#одинарная) Получаем

–1,1110111011∙27,

перенося запятую на 7 знаков влево.

Соответственно, 7=+111 – истинный порядок числа.

3. Найдем **характеристику** (смещенный порядок) для разных типов чисел:

float – чисел одинарной точности 7+127=134=10000110 ,

double – чисел двойной точности 7+1023=10000000110 ,

long double – расширенной точности 7+16383=16390=100000000000110.

4. Найдем **мантиссу** для разных типов чисел. Для типа

float – чисел одинарной точности длина мантиссы должна быть 23 разряда

1,11101110110000000000000

23

double – чисел двойной точности длина мантиссы должна быть 52 разряда

1,1110111011000000000000…00000000000000000000

52

Обратите внимание, что в форматах float и double единица целой части не хранится в самой мантиссе.

long double – расширенной длина мантиссы должна быть 64 разряда

1,1110111011000000000000000000000…00000000000000000

64

Обратите внимание, что в формате long double единица целой части хранится в самой мантиссе в качестве её первого числового разряда.

**Окончательный результат: <знак><порядок><мантисса>**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| знак | порядок |  | мантисса | без целой части (без «1») | |
| float:по | 1**.**1000011 | 0**.**1110111 | 01100000 | 00000000 |  |
|  | С 3 | 7 7 | 6 0 | 0 016 | –в шестнадцатеричной системе счисления |
|  | 3ий байт | 2ой байт | 1ый байт | 0ой байт |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| знак | порядок | |  | | мантисса без целой части (без «1») | | | | | |
| double:по | 1**.** 1000000 | | 0110**.** 1110 | | 11101100 | | 0000…0000 | |  | |
|  | С 0 | | 6 Е | | Е С | | 0 … 016 | |  | |
|  | 7ий байт | | 6ой байт | | 5ый байт | | … 0ой байт | |  | |
|  | знак | | порядок | |  | | мантисса с целой частью (с «1») | | | |
| long double:по | | 1**.** 1000000 | | 00000110**.** | | 11110111 | | 01100000 … 0 | |  |
|  | | С 0 | | 0 6 | | F 7 | | 6 0 … 0 | |  |
|  | | 9ый байт | | 8ой байт | | 7ой байт | | 6 ой байт … 0ой байт | |  |

Итак, в памяти ЭВМ десятичное дробное число –247,375 может соответствовать в зависимости от точности представления различным двоичным кодам (для краткости они показаны в шестнадцатеричной системе счисления по аналогии с результатами любого отладчика при просмотре содержимого ячеек памяти и регистров процессора):

float:по 4 байта С3 77 60 0016double: 8 байт С0 6Е ЕС 00 00 00 00 0016

long double: 10 байт С0 06 F7 60 00 00 00 0016

Поскольку Intel x86-совместимые компьютеры имеют архитектуру little-endian (очень интересно происхождение термина [http://ru.wikipedia.org]), в памяти число будет выглядеть как перевернутая последовательность байт, согласно правилу: младший байт по младшему адресу. Например, при размещении числа –247,375 в формате float имеем последовательность байт:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| адрес байта в ОП |  | | | |
| содержимое байтов | 00 | 60 | 77 | С3 |

В этом можно убедиться, если скомпилировать и выполнить небольшую программу на языке Cи:

#include <stdio.h>

int main(int argc, char \*argv[])

{ float f = -0.1015625f;

unsigned char \*p = (unsigned char \*) &f;

printf("%02x %02x %02x %02x\n",

(unsigned int) p[0],

(unsigned int) p[1],

(unsigned int) p[2],

(unsigned int) p[3]);

return 0;

}

Эта программа выдаст ожидаемый результат: 00 60 77 С3.

Для других типов (double, long double) можно слегка подправить программу и также увидеть хранимые в памяти коды для любого заданного дробного числа.

**Пример 2.** Допустим, в ячейках памяти расположено действительное число В. Мы знаем, что оно 4-байтное и видим его 16-ричное представление

00 00 F2 C2

Согласно архитектуре little-endian привычное представление числа со старших разрядов слева направо наше число выглядит как C2F2000016 . Зная, что это число в формате с плавающей запятой, определим реальное значение числа в десятеричной системе. Количество байтов, занимаемых числом – 4, значит число – типа float.

* 1. Переведем число в двоичную систему:

1100 0010 1111 0010 0000 0000 0000 0000.

2. Старший знак числа «1», значит число отрицательное.

3. Характеристика в типе float занимает 8 бит, начиная со второго. Получаем 10000101=133.

4. Истинное значение порядка 133 - 127 = 6 (26).

5. Мантисса равна : (1,)111 0010 0000 0000 0000 0000. Нормализованное двоичное представление числа имеет вид:

– 1,111 ∙26.

6. Чтобы получить более привычное представление (без порядка) смещаем мантиссу на 6 знаков вправо, получаем:

– 1 111 001,02.

Что соответствует десятичному числу:

–121,0.

* 1. **АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ С ЧИСЛАМИ С ПЛАВАЮЩЕЙ ЗАПЯТОЙ**

К началу выполнения арифметического действия операнды операции помещаются в соответствующие регистры АЛУ.

**2.1. СЛОЖЕНИЕ И ВЫЧИТАНИЕ НОРМАЛИЗОВАННЫХ ЧИСЕЛ.**

Сначала производится подготовительная операция, называемая выравниванием порядков.

**В процессе выравнивания порядков мантисса числа с *меньшим* порядком сдвигается в своем регистре *вправо* на количество разрядов, равное разности порядков операндов.**

После каждого сдвига порядок увеличивается на единицу, а освободившиеся старшие разряды заполняются нулями. В результате выравнивания порядков одноименные разряды чисел оказываются расположенными в соответствующих разрядах обоих регистров, после чего мантиссы складываются или вычитаются.

В случае необходимости полученный результат нормализуется путем сдвига мантиссы результата влево.

Нормализацией называется выбор такого значения порядка, при котором старший разряд мантиссы имеет значение 1. При нормализации возможны две ситуации:

− результат меньше 1/2, то есть старшие разряды мантиссы нулевые. Если при этом результат представлен в прямом коде, мантисса сдвигается влево до тех пор, пока первая значащая 1 не окажется в старшем разряде. Если же результат представлен в обратном или дополнительном коде (отрицательный), производится сдвиг влево до появления в старшем разряде первого значащего нуля. При каждом сдвиге значение порядка уменьшается на 1;

− результат больше 1, то есть разрядная сетка переполнена. В этом случае мантисса сдвигается вправо на один разряд с одновременным увеличением порядка на 1.

Правила выполнения основных арифметических операций справедливы для чисел любой позиционной системы счисления.

**Примеры**

1. Сложить десятичные нормализованные числа **0,536·106 и 0,284·103.**

Разность порядков слагаемых здесь равна трём (6-3=3), поэтому перед сложением мантисса второго числа сдвигается на три разряда вправо:

|  |
| --- |
| 0,536 ·106  +  0,000284 ·106 |
| 0,536284 ·106 |

1. Сложить двоичные нормализованные числа 0,11011·210  и 0,10111·2–1.

Разность порядков слагаемых здесь равна трем (102+12=112=310), поэтому перед сложением мантисса второго числа сдвигается на три разряда вправо:

|  |
| --- |
| 0,11011 ·210  +  0,00010111·210 |
| 0,11101111·210 |

1. Сложить шестнадцатеричные нормализованные числа **0,1В5·163  и 0,34Е·16–1.**

Разность порядков слагаемых здесь равна четырём (316+116=416=410), поэтому перед сложением мантисса второго числа сдвигается на четыре разряда вправо:

|  |
| --- |
| 0,1В5 ·163  +  0,000034Е·163 |
| 0,1В5034Е·163 |

1. Выполнить вычитание двоичных нормализованных чисел **0,10101·210 и 0,11101·21.**

Разность порядков уменьшаемого и вычитаемого здесь равна единице (102-12=12=110), поэтому перед вычитанием мантисса второго числа сдвигается на один разряд вправо:

|  |
| --- |
| 0,10101 ·210  -  0,011101·210 |
| 0,001101·210 |

Результат получился не нормализованным, поэтому его мантисса сдвигается влево на два разряда с соответствующим уменьшением порядка на две единицы: 0,1101·20.

4) Выполнить вычитание восьмеричных нормализованных чисел **0,125·83 и 0,32·8-2** Разность порядков уменьшаемого и вычитаемого здесь равна пяти (38+28=58=510), поэтому перед вычитанием мантисса второго числа сдвигается на пять разрядов вправо:

|  |
| --- |
| 0,125 ·83  -  0,0000032 ·83 |
| 0,0147746 ·83 |

Результат получился не нормализованным, поэтому его мантисса сдвигается влево на один разряд с соответствующим уменьшением порядка на одну единицу: 0,147746 ·82.

**2. УМНОЖЕНИЕ И ДЕЛЕНИЕ НОРМАЛИЗОВАННЫХ ЧИСЕЛ.**

Для получения результата в данном случае производятся следующие действия, причем не важно, какое из двух данных чисел больше.

1. Мантиссы перемножаются или одна делится на другую;
2. Порядки при умножении складываются, а при делении вычитаются;
3. При необходимости мантисса результата нормализуется.

**Примеры:**

1. Выполнить умножение двоичных нормализованных чисел:

(0,11101 **.** 2101) **.** (0,1001 **.** 211) = (0,11101 **.** 0,1001) **.** 2(101+11) = 0,100000101 **.** 21000.

1. Выполнить умножение восьмеричных нормализованных чисел:

(0,765 **.** 84) **.** (0,537 **.** 85) = (0,765 **.** 0,537) **.** 8(4+5) = 0,527353 **.** 811.

1. Выполнить умножение шестнадцатеричных нормализованных чисел:

(0,А25 **.** 167) **.** (0,6С **.** 169) = (0,А25 **.** 0,6С) **.** 16(7+9) = 0,4479С **.** 1610.

1. Выполнить деление двоичных нормализованных чисел:

(0,100000101 **.** 21000) : (0,1001 **.** 211) = (0,100000101 : 0,1001) **.** 2(1000-11) = 0,11101 **.** 2101

1. Выполнить деление восьмеричных нормализованных чисел:

(0,527353 **.** 811) : (0,537 **.** 85) = (0,527353 :0,537) **.** 8(11-5) = 0,765 **.** 84

1. Выполнить деление шестнадцатеричных нормализованных чисел:

(0,4479С **.** 1610) :(0,6С **.** 169) = (0, 4479С :0,6С) **.** 16(10-9) = 0,А25 **.** 167

**ВАРИАНТЫ ЗАДАНИЙ**

**ЗАДАНИЕ 1**

1. Запишите целое число А=± *a1 a2 a3* в десятичной системе счисления, представляющее (месяц вашего рождения+текущий месяц)×10.
2. Представьте числа+А и –А в двоичной системе счисления.
3. В каком из целочисленных форматов могут быть записаны указанные числа? Запишите содержимое последовательности байт памяти, хранящих числа в этих форматах.

**ЗАДАНИЕ 2**

1. Запишите число F=± *a1 a2 a3 a4* **,** *a5 a6*в десятичной системе счисления, представляющее **дату вашего рождения**, где

*a1 a2*– год

*a3 a4* – месяц

*a5 a6*– число,

знак числа выберите по правилу: минус, если *a4* – чётное, иначе – плюс.

1. Представьте это числоFв формате IEEE754-2008 с одинарной, двойной и расширенной точностью, используя следующую последовательность действий:

а) перевести число F в двоичную систему счисления,

б) нормализовать число F (выделить мантиссу и истинный порядок),

в) вычислить характеристику (смещённый порядок) для разных форматов и перевести характеристику в двоичную систему счисления:

* float: 8-разрядная характеристика = порядок +127,
* double: 11-разрядная характеристика = порядок +1023,
* long double: 15-разрядная характеристика = порядок +16383;

г) записать результат в виде:

* float (4 байта):

|  |  |  |
| --- | --- | --- |
| 31 | 30 23 | 22 0 |
| 1 разряд (знак числа) | 8 разрядов (характеристика) | 23 разряда (мантисса) |

* double (8 байт) :

|  |  |  |
| --- | --- | --- |
| 63 | 62 52 | 51 0 |
| 1 разряд (знак числа) | 11 разрядов (характеристика) | 52 разряда (мантисса) |

* long double (10 байт) :

|  |  |  |
| --- | --- | --- |
| 79 | 78 64 | 63 0 |
| 1 разряд (знак числа) | 15 разрядов (характеристика) | 52 разряда (мантисса) |

1. переведите получившееся многобайтовое число из двоичной в шестнадцатеричную систему счисления;
2. проверьте ваши вычисления.

Для этого можно создать программу на любом языке программирования, которая выведет на печать последовательно все байты числа для каждого формата (float, double, long double). Пример такой программы на С приводится выше (на стр. 9).

Либо можно **воспользоваться сервисом** **перевода на сайте «Википедия» в разделе «IEEE 754-2008/ IEEE754 онлайн двоично-десятичный преобразователь» или аналогичными интернет-сервисами.**

**ЗАДАНИЕ 3**

Выполнить [арифметические действия](#АрифметическиеД) с числами с плавающей запятой по вариантам:

|  |  |
| --- | --- |
| **№ варианта** | **Числа с плавающей запятой** |
|  | P = 2 x=1110011-1 y=101110 |
|  | P = 8 x=71627 y=533 |
|  | P = 16 x=72ABF11 y=B58 |
|  | P = 2 x=1000001101 y=1110-11 |
|  | P = 8 x=1314712 y=37-7 |
|  | P = 16 x=3CC512 y=2F5 |
|  | P = 2 x=11001110 y=100111 |
|  | P = 8 x=10274-6 y=52-2 |
|  | P = 16 x=529B35 y=67-3 |
|  | P = 2 x=1001110-11 y=100-111 |
|  | P = 8 x=41416 y=15-4 |
|  | P = 16 x=50385-3 y=63-2 |
|  | P = 2 x=1100100110 y=1001-100 |
|  | P = 8 x=32414-3 y=346 |
|  | P = 16 x=72D9B10 y=E9-7 |
|  | P = 2 x=11000101-11 y=1001-10 |
|  | P = 8 x=3156710 y=37-7 |
|  | P = 16 x=37A509 y=38-8 |
|  | P = 2 x=1011011-101 y=11110 |
|  | P = 8 x=7603-5 y=23-2 |
|  | P = 16 x=9236512 y=F1-9 |
|  | P = 2 x=1110011111 y=101-101 |
|  | P = 8 x=521112 y=37-7 |
|  | P = 16 x=5D8CC12 y=8C-7 |
|  | P = 2 x=1110011-11 y=110-10 |
|  | P = 8 x=143357 y=235 |
|  | P = 16 x=4EE2E-13 y=7620 |
|  | P = 2 x=11001100111 y=11111 |
|  | P = 8 x=52117 y=37-2 |
|  | P = 16 x=5A858-5 y=A8-10 |

1. Число в формате с плавающей запятой [↑](#footnote-ref-1)