Представление чисел в формате с плавающей запятой.

Вещественные числа (конечные и бесконечные десятичные дроби) хранятся и обрабатываются в компьютере в формате *с плавающей запятой*.

В этом случае положение запятой в записи числа может изменяться.

Представление чисел в формате с плавающей запятой.

Формат чисел *с плавающей запятой* базируется на экспоненциальной форме записи, в которой может быть представлено любой число. Так число А может быть представлено в виде:

$A = m \times q^n$

где m – мантисса числа

q – основание системы счисления,

n – порядок числа.

Для однозначности представления чисел *с плавающей запятой* используется нормализованная форма, при которой мантисса отвечает условию:

$1/n \leq |m| < 1.$

Это означает, что мантисса должна быть правильной дробью и иметь после запятой цифру, отличную от нуля.

Пример 1.

Преобразуйте десятичное число 888,888, записанное в естественной форме, в экспоненциальную форму с нормализованной мантиссой.

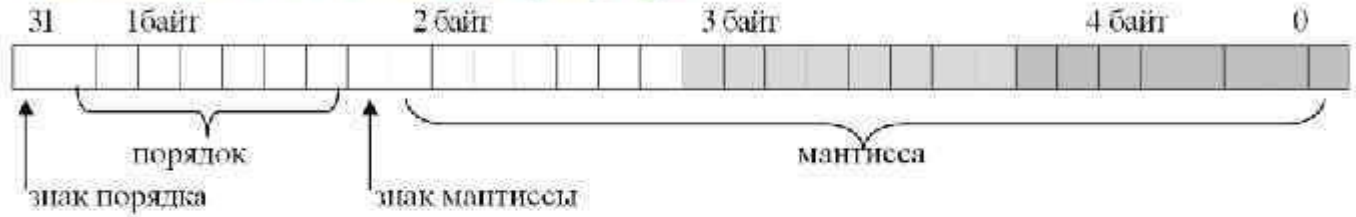
$888{,}888 = 0{,}888888 \times 10^3$

Нормализованная мантисса $m = 0{,}888888$, порядок $n = 3$.

Представление чисел в формате с плавающей запятой.

- Число в форме с плавающей запятой занимает в памяти компьютера четыре (*число обычной точности*) или восемь байт (*число двойной точности*).
- При записи числа с плавающей запятой выделяются разряды для хранения знака мантиссы, знака порядка, порядка и мантиссы.
- Диапазон изменения чисел определяется количеством разрядов, отведенных для хранения порядка числа, а точность (количество значащих цифр) определяется количеством разрядов, отведенных для хранения мантиссы.

Пример 2. Определить максимальное число и его точность для формата *чисел обычной точности*, если для хранения порядка и его знака отводится 8 разрядов, а для хранения мантиссы и ее знака 24 разряда.



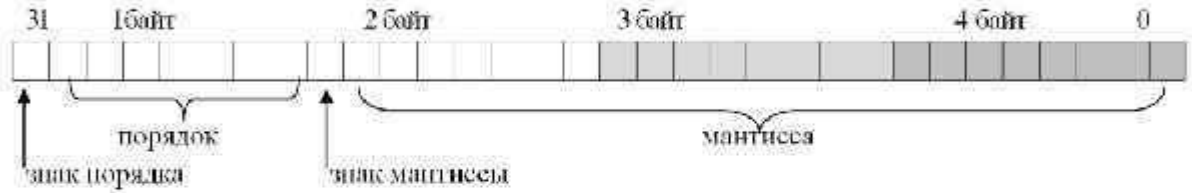
Максимальное значение порядка числа составит

$1111111_2 = 127_{10}$

Максимальное значение числа составит:

$2^{127} = 1{,}70141183460469231731687303715884 \times 10^{38}$

Пример 2. Определить максимальное число и его точность для формата *чисел обычной точности*, если для хранения порядка и его знака отводится 8 разрядов, а для хранения мантиссы и ее знака 24 разряда.



Точность вычислений определяется количеством разрядов, отведенных для хранения мантиссы чисел. Максимальное значение положительной мантиссы равно:

$2^{23} - 1 \approx 2^{23} = 2^{(10 \times 2,3)} \approx 1000^{2,3} = 10^{(3 \times 2,3)} \approx 10^{7}$

Представление чисел в формате с плавающей запятой.

- При сложении и вычитании чисел в формате *с плавающей запятой* сначала производится подготовительная операция *выравнивания порядков*.
- Порядок меньшего (по модулю) числа увеличивается до величины порядка большего (по модулю) числа.
- Для того чтобы величина числа не изменилась, мантисса уменьшается в такое же количество раз (сдвигается в ячейке памяти вправо на количество разрядов, равное разности порядков чисел).

Представление чисел в формате с плавающей запятой.

- После выполнения операции выравнивания одинаковые разряды чисел оказываются расположенными в одних и тех же разрядах ячеек памяти.
- Теперь операции сложения и вычитания чисел сводятся к сложению или вычитанию мантисс.

Представление чисел в формате с плавающей запятой.

- После выполнения арифметической операции для приведения полученного числа к стандартному формату с плавающей запятой производится нормализация, т.е. мантисса сдвигается влево или вправо так, чтобы ее первая значащая цифра попала в первый разряд после запятой.

Представление чисел в формате с плавающей запятой.

- При умножении чисел в формате *с плавающей запятой* порядки складываются, а мантиссы перемножаются.

- При делении из порядка делимого вычитается порядок делителя, а мантисса делимого делится на мантиссу делителя.

Пример 3. Произвести сложение чисел $0,1\times2^3$ и $0,1\times2^4$ в формате с плавающей запятой.

Произведем выравнивание порядков и сложение мантисс:

$$\begin{array}{rl} & 0,01\times2^4 \\ + & 0,10\times2^4 \\ \hline & 0,11\times2^4 \end{array}$$

Представление чисел в формате с плавающей запятой.

```
double sum1(std::vector<double>& v)
{
    if (v.empty())
    {
        return 0.0;
    }
    for(size_t i = 0; i < v.size() - 1; ++i)
    {
        std::sort(v.begin()+i, v.end());
        v[i+1] += v[i];
    }
    return v.back();
}
```

Представление чисел в формате с плавающей запятой.

```
const double x = 0.01;
double s = 1000000000.;
// initial sum
for (int i = 0; i < 10000; ++i )
{
        s = s + x;
}
const double e = 1000000100. - s;
std::cout << e << std::endl;
```

результат:
9.53674e-05

Представление чисел в формате с плавающей запятой.

IEEE 754

```
const double x = 0.01;
double c = 0; // для хранения ошибки
double s = 1000000000.;// начальная сумма
for (int i = 0; i < 10000; ++i )
{
        const double y = x - c;
        const double t = s + y;
        c = (t - s) - y; // Зависит от оптимизаций компиляторов!
        s = t;
}
const double e = 1000000100. - s;
std::cout << e << std::endl;
```

результат:
0