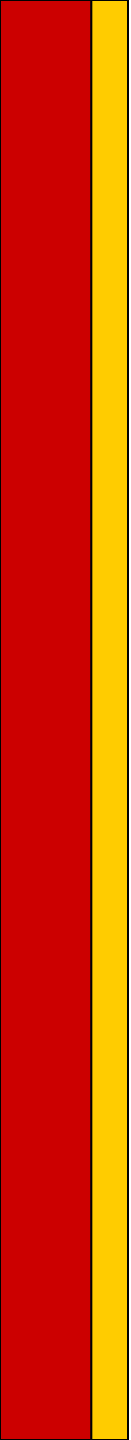


Десятичные числа	Прямой код	Обратный код	Дополнительный код
20	0000000000010100		
-60	0000000000111100	1111111111000011 <u>0000000000000001</u> 1111111111000100	1111111111000100 <u>0000000000010100</u> 1111111111011000
-40	0000000000101000	0000000000100111 <u>0000000000000001</u> 0000000000101000	



Представление чисел в формате  
с плавающей запятой

# Повторение

1. Какие числа хранятся в формате с фиксированной запятой?
2. Сколько необходимо памяти для хранения целых неотрицательных чисел? Диапазон этих чисел.
3. Сколько необходимо памяти для хранения целых чисел со знаком? Диапазон этих чисел.
4. Сколько необходимо памяти для хранения больших целых чисел со знаком? Диапазон этих чисел.
5. Что такое дополнительный код, для чего он необходим?
6. Алгоритм нахождения дополнительного кода.
7. Алгоритм арифметических чисел.



- ***Целые числа***
- ***Целые числа со знаком***
- ***Большие целые числа со знаком***



- ***Целые числа***



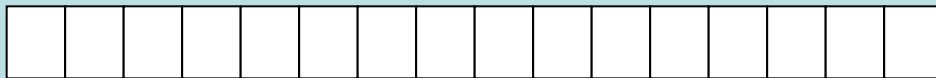
**8 бит**

**Диапазон**

**[0;255]**



- ***Целые числа со знаком***



**16 бит**

**Диапазон**

**[-32768;32767]**



32 бита

- Большие целые числа со знаком



32 бита

Диапазон

$$A = -2^{31}$$

$$A = 2^{31} - 1$$



# Дополнительный код

- Дополнительный код представляет собой дополнение модуля отрицательного числа до нуля.
- Дополнительный код позволяет заменить операцию вычитания операцией сложения.





## Алгоритм нахождения дополнительного кода

1. Модуль числа записать прямым кодом в двоичном разряде
2. Получить обратный код числа (инвертирование – единицы заменить нулями, а нули единицами.)
3. К обратному коду прибавить единицу.



# Алгоритм арифметических действий

1. Сложить прямой код положительного числа с дополнительным кодом отрицательного числа.
2. Перевести полученный дополнительный код в десятичное число
  - Инвертировать дополнительный код
  - Прибавить к дополнительному коду единицу
  - Перевести в десятичное число и приписать знак отрицательного числа.



# Задачи:

- **Выяснить какие числа хранятся в формате с плавающей запятой**
- **Научиться определять точность записи чисел**
- **Научиться выполнять арифметические действия с числами в формате с плавающей запятой.**



# Формат с плавающей запятой

- **Вещественные числа (конечные и бесконечные десятичные дроби)**

$$A = m * q^n$$

**m** – мантисса числа, где  $1/n \leq |m| < 1$

**q** – основание системы счисления

**n** – порядок числа

$$234,4567 = 0,2344567 * 10^3$$

# Числа в формате с плавающей запятой

## 1. Число обычной точности

0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1
Знак и порядок										Знак и мантисса																					

**Максимальное значение порядка числа**

$$1111111_2 = 127_{10}$$

**Максимальное число**

$$2^{127} = 0,17014118346923173168730371588 * 10^{39}$$

# Точность вычислений

- **Количество разрядов, отведенных для мантиссы**

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

$$A_{\max} = 0,1701412 * 10^{39}$$

# Числа в формате с плавающей запятой

**2. Числа двойной точности занимают в памяти восемь байтов.**

# Алгоритм сложения и вычитания

- Выравнивание порядков
- Меньший порядок увеличивается до величины большего порядка
- Мантисса уменьшается в такое же количество раз, т.е. запятая сдвигается влево на число равное разности порядков

$$A_1 = 0,1 * 2^5$$

$$A_1 = 0,100 * 2^5$$

$$A_2 = 0,1 * 2^3$$

$$A_2 = 0,001 * 2^5$$

**сумма**

$$A = 0,101 * 2^5$$

**разность**

$$A = 0,010 * 2^5 = 0,10 * 2^4$$



# Умножение и деление

## Умножение

$$A_1 = 0,1 * 2^5$$

$$A_2 = 0,1 * 2^3$$

$$A = 0,01 * 2^8 = 0,1 * 2^7$$

## Деление

$$A_1 = 0,1 * 2^5$$

$$A_2 = 0,1 * 2^3$$

$$A = 1 * 2^2 = 0,1 * 2^3$$

# Задание

- Выполнить сложение, вычитание, умножение
- и деление чисел:

$$A_1 = 0,11 * 2^2$$

$$A_2 = 0,101 * 2^4$$



# Домашнее задание

- § 2.10.2
- С.150 №2.16