

Краткое информационное руководство

Двоичная система счисления

Предварительный просмотр двоичной системы счисления

1 0 1 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 1 0 1 1
0 1 0 0 1 0 1 1 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0
1 0 0 0 0 1 0 0 0 1 1 0 0 0 1 0 0 0 1 1 0 0 0
0 1 0 1 1 0 1 0 0 0 0 1 0 1 1 0 1 0 0 0 1 0 1
1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1
0 1 0 0 0 0 1 1 0 1 0 1 0 0 0 0 1 1 0 1 0 0
1 0 0 0 1 1 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 0
0 1 1 1 0 1 1 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1
0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0
0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0
1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1
1 1 0 0 1 0 1 1 1 1 1 1 0 0 1 1 0 1 1 1 0 0
0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0
1 0 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1
0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0
0 1 0 1 1 1 1 0 0 1 0 1 0 1 1 1 1 0 0 1 0 1
1 0 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 0 1 0 0 0
0 1 1 0 0 0 0 1 0 1 0 1 0 0 0 0 1 0 1 0 1 1
1 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 0 0
0 0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0
1 0 0 1 1 1 1 0 1 0 1 0 0 1 1 1 1 0 1 0 0 1
0 0 0 1 0 0 0 1 1 0 0 0 0 1 0 0 1 1 0 0 0 0
0 1 1 0 1 1 0 1 0 0 0 1 1 0 1 1 0 1 0 0 1 0
0 1 1 1 0 0 0 1 0 1 0 1 1 1 0 0 1 0 1 1 1

Agenda

Ключевые темы,
обсуждаемые
в этой презентации

Общая теория систем счисления

Двоичные числа

Преобразование чисел из и в двоичные
числа

Восьмеричные и шестнадцатеричные числа

Почему двоичные числа?

- Компьютеры оперируют данными, представленными в двоичной системе счисления
- Двоичные числа состоят всего из двух цифр, 1 и 0
- Сначала нам нужно понять, как работают системы счисления в целом, затем мы изучим двоичные числа



Система счисления



- Позиционная система счисления - система записи для выражения чисел
- Каждая система счисления имеет **основание**
- Основание определяет диапазон цифр и способ интерпретации этих цифр
- Основание 10 (или десятичная) использует цифры 0,1,2 ... 9
- Основание 2 (или двоичная) использует цифры 0 и 1

Система счисления	Определение
Двоичный	Компьютеры обрабатывают и хранят информацию в этой системе счисления
Восьмеричный	Цифры 0-7. Можно легко преобразовать в/из двоичной системы счисления. Восьмеричные значения легче считывать, чем двоичные
Десятичный	Система счисления людей.
Шестнадцатеричный	Шестнадцатеричный формат легче читать, чем двоичный Создает более компактное число, чем восьмеричное значение Использует цифры 0,1,2, ... 8,9, A, B, C, D, E, F

Формула системы счисления

- Любое число в этом основании будет содержать цифры от **0** до **k – 1**
- Интерпретация цифры = значение * степень(power).
- Степень зависит от столбца или его положения, используя уравнение

$$\text{степень} = \text{основание}^{\text{столбец}}$$
, где столбец отсчитывается справа налево
(самый правый столбец - это столбец 0).

Пример с числами с основанием 10

Номера столбцов: 2 1 0

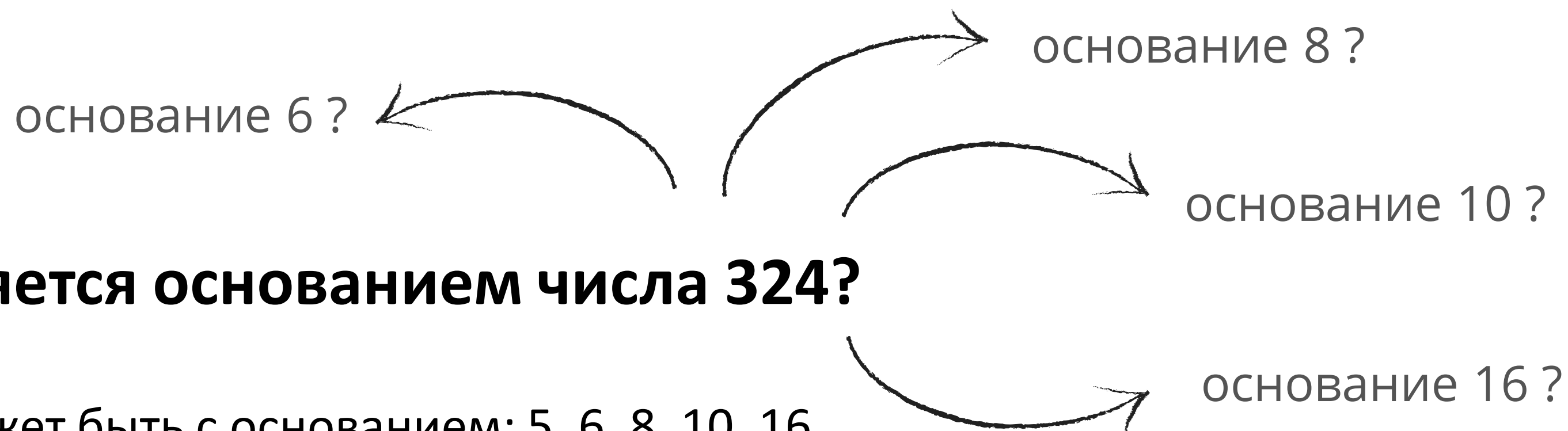
259

номера столбцов
в качестве степени

$2 * 10^2 + 5 * 10^1 + 9 * 10^0$

основание

Равно 200 + 50 + 9 = 259



Что является основанием числа 324?

- Это число может быть с основанием: 5, 6, 8, 10, 16
- Конечно, основанием **не** является 1, 2, 3 или 4
- Потому что у нас нет цифры “4” в этих системах счисления

Как показать основание?

$$1011001_2 = 89_{10}$$

- Запишите основание в виде нижнего индекса в конце числа
- 1234_5 число в системе счисления с основанием 5
- 1234_8 число в системе счисления с основанием 8
- 1234_{10} число в системе счисления с основанием 10

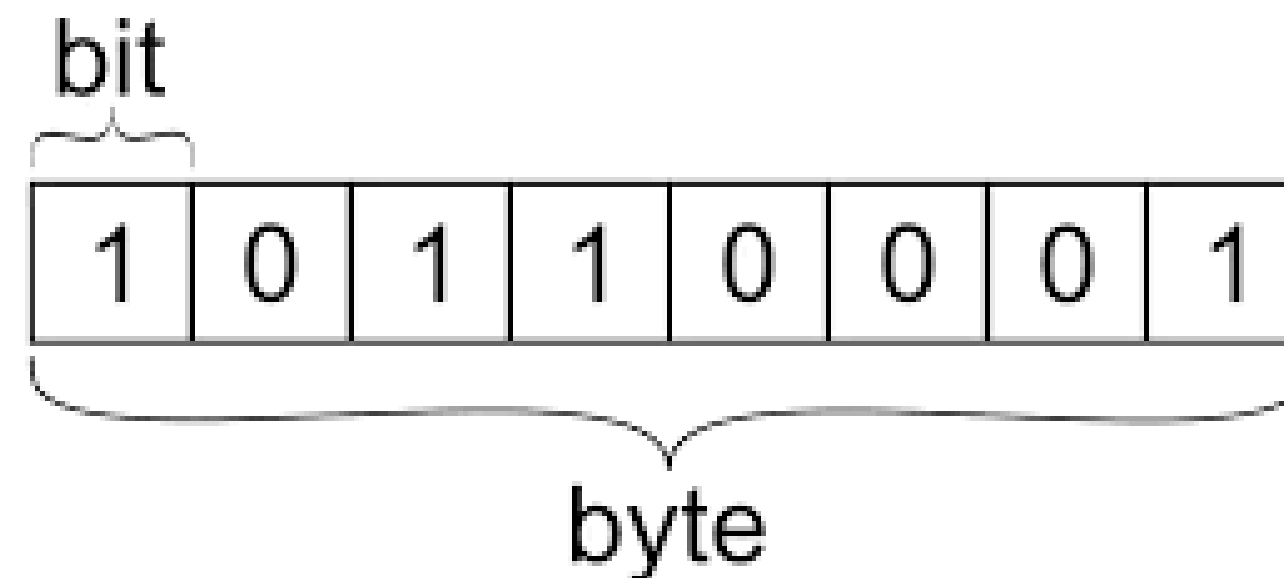
```

1 1 0 1 1 1 1 1 1 0 0 0 0 1 0
1 1 1 1 0 0 0 1 0 0 1 1 0 1 0
0 0 0 0 0 1 1 0 1 1 1 0 1 0 1
0 0 1 0 0 0 0 0 1 0 0 1 0 1 0
1 1 0 0 1 0 0 0 0 1 1 0 1 0 1
0 0 1 1 0 1 1 1 0 0 0 0 0 0 0
1 1 0 0 1 0 0 1 0 1 1 1 1 1 1
0 0 1 1 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 1 0 1 1 0 1 0 0 0 1 1
1 1 0 1 0 1 1 0 0 0 1 1 1 1 0
1 0 1 0 1 0 0 0 0 1 0 1 0 0 0
0 1 0 1 0 0 0 1 1 0 1 0 1 0 1
1 0 1 0 1 1 1 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 1 1 0 0 0 1 0 1
1 1 1 1 1 1 0 0 1 1 0 1 0 1 0
0 0 0 0 1 0 0 1 0 0 1 0 1 0 1
0 0 0 1 1 1 1 0 0 0 0 1 0 1 0
1 1 1 1 1 0 0 1 1 0 0 0 1 0 1
0 1 0 0 0 0 0 0 0 0 1 1 0 0 1
  0 1 1 0 1 0 1 0 1 0 0 1 1 0
    1 0 0 0 0 0 1 1 1 1 0 0 1
      0 1 0 1 0 1 1 1 0 0 1 0 0
        1 0 0 0 0 1 1 0 1 0 0 0
          0 1 1 1 0 0 0 1 0 0 1
            1 0 0 0 0 1 1 0 0 0
              1 0 1 1 0 0 0
                0 1 0 1
                  1 0
                    0
  
```

Двоичные числа

- Одна из самых важных систем счисления
- Компьютеры представляют любую информацию в двух состояниях: в высоком и низком напряжениях
- В двоичном виде это 1 и 0 соответственно
- Мы называем цифры 1 и 0 - **битами**

Бит и байт



- Мы группируем 8 битов вместе, чтобы получился байт
- Компьютеры оперируют данными большего размера, называемыми словами
- Более ранние компьютеры имели 8- или 16-разрядный размер слова
- Современные компьютеры используют 32- или 64-разрядный размер слова
- 8, 16, 32 и 64 бита равны 1, 2, 4 и 8 байтам соответственно

00000000 = 0 в десятичном

00000001 = 1 в десятичном

00000010 = 2 в десятичном

.....

11111110 = 254 в десятичном

11111111 = 255 в десятичном

Комбинации

- В байте может храниться любая комбинация из 8 нулей и единиц
- Примеры байтов: 10101010, 1000, 01111110
- На 1 байт приходится 256 (2^8) комбинаций единиц и нулей
- Если мы храним положительное целое число - диапазон от 0 до 255
- На 2 байта приходится 65 536 (2^{16}) комбинаций единиц и нулей

Преобразование из двоичного в десятичный

Номера столбцов: 7 6 5 4 3 2 1 0

$$\begin{array}{cccccccc}
 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
 & \swarrow & \swarrow & \swarrow & \swarrow & \downarrow & \swarrow & \swarrow & \swarrow \\
 0 * 2^7 & + & 1 * 2^6 & + & 0 * 2^5 & + & 1 * 2^4 & + & 0 * 2^3 & + & 1 * 2^2 & + & 0 * 2^1 & + & 1 * 2^0 \\
 \parallel & & \parallel & & \parallel & & \parallel & & \parallel & & \parallel & & \parallel & & \parallel \\
 0 & + & 64 & + & 0 & + & 16 & + & 0 & + & 4 & + & 0 & + & 1
 \end{array}$$

Итак, как вы видите, $01010101_2 = 85_{10}$

Простой способ

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
0	1	0	1	0	1	0	1	
	64		16		4		1	= 85

- Каждая цифра в двоичном числе равна либо 0, либо 1
- Любое число * 1 равна этому же числу
- Любое число * 0 равно нулю
- Поэтому игнорируйте числа, умноженные на 0
- И берите только числа, умноженные на 1

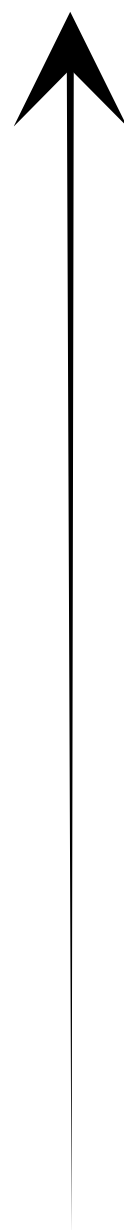
Преобразование из десятичной системы счисления в двоичную

Одним из распространенных подходов является **метод деления**

- Повторяйте, пока значение не станет равным 0
 - Разделите значение на 2, записав частное и остаток.
 - Замените значение на частное.
- Запишите остатки в порядке, записанном в виде двоичного числа справа налево.

От десятичного к двоичному

- $89/2 = 44$ с остатком 1
- $44/2 = 22$ с остатком 0
- $22/2 = 11$ с остатком 0
- $11/2 = 5$ с остатком 1
- $5/2 = 2$ с остатком 1
- $2/2 = 1$ с остатком 0
- $1/2 = 0$ с остатком 1



- Наконец, частное равно 0.
- Соберите все остатки снизу вверх, и результат будет двоичным числом.
- Таким образом, $89 = 1011001$.
- Но обычно мы группируем по 8 бит, поэтому $89 = 01011001$.

Группировка битов

- Мы всегда будем группировать биты в байты (= 8 бит)
- Предположим, что мы храним число 251 в 32 битах
- Оно равно 0000000000000000000000000000000011111011
- Лучше разделить 32 бита на 8-битные блоки, чтобы было легче читать
- Итак, 251 - это 00000000 00000000 00000000 11111011

Преобразования, требующие более 8 бит

- В 8 битах мы можем хранить только 256 различных значений
- Что делать, если мы хотим хранить числа больше 256
- Для хранения значений, которым требуется более 8 бит, мы обычно переходим к следующему по величине размеру байта - 16 битам
- Если 16 бит недостаточно, то мы переходим к 32 битам и так далее
- В 16 битах мы можем хранить 65 536 различных значений ($2^{16} = 65\,536$)
- Таким образом, диапазон значений составляет от 0 до 65 535

Преобразование большого числа

- $1234/2 = 617$, остаток = 0
- $617/2 = 308$, остаток = 1
- $308/2 = 154$, остаток = 0
- $154/2 = 76$, остаток = 0
- $76/2 = 38$, остаток = 0
- $38/2 = 19$, остаток = 0
- $19/2 = 9$, остаток = 1
- $9/2 = 4$, остаток = 1
- $4/2 = 2$, остаток = 0
- $2/2 = 1$, остаток = 0
- $1/2 = 0$, остаток = 1



- И так $1234_{10} = 10011000010_2$
- Длина двоичного числа должна быть кратна 8.
- $1234_{10} = 00000100\ 11000010_2$

32 и 64 бита информации

- Обычно компьютеры хранят данные в 32 или 64 битах
 - В 32 битах мы можем сохранить 4 294 967 296 различных значений:
 - 00000000 00000000 00000000 00000000 = 0_{10}
 - 00000000 00000000 00000000 00000001 = 1_{10}
 - 00000000 00000000 00000000 00000010 = 2_{10}
 -
 - 11111111 11111111 11111111 11111110 = $4,294,967,295_{10}$
 - 11111111 11111111 11111111 11111111 = $4,294,967,296_{10}$

Восьмеричные числа

- В восьмеричных числах мы используем цифры 0-7
- Десятичное значение 7 равно 111 в двоичном формате
- Таким образом, мы можем хранить одну восьмеричную цифру в 3 битах
- Это означает, что мы можем сгруппировать 3 бита двоичного числа для хранения одной восьмеричной цифры

3 бита		восьмеричное число
000	→	0
001	→	1
010	→	2
011	→	3
100	→	4
101	→	5
110	→	6
111	→	7

Преобразование двоичного в восьмеричный

- 10110111 10101010 00110011 00011100

||

- 010 110 111 101 010 100 011 001 100 011 100

- 2 6 7 5 2 4 3 1 4 3 4 в восьм.

- Таким образом, результат в восьмеричном представлении равен: 26752431434₈

К крайним левым 2 битам мы добавляем начальный 0, чтобы длина была кратна 3

Шестнадцатеричные числа

- Для формирования шестнадцатеричной цифры мы используем 4 бита
- Шестнадцатеричное число имеет основание 16, поэтому наибольшая “цифра” равна 15, что равно 1111 в двоичном формате
- 15 - это не однозначная цифра, поэтому вместо нее мы используем буквы.
- Мы используем буквы A, B, C, D, E и F для обозначения цифр” 10, 11, 12, 13, 14, и 15 соответственно.

4 бита	16-знач. число	10-знач. число
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Из двоичного в шестнадцатеричный

• 10110111 10101010 00110011 00011100

||

• 1011 0111 1010 1010 0011 0011 0001 1100

• B 7 A A 3 3 1 C = B7AA331C₁₆

Из восьмеричного в двоичный

$$\bullet 76_8 = 00 \underbrace{111}_7 \underbrace{110}_6$$

*Добавление начальных
нулей, чтобы длина
двоичного
числа была кратна 8*

Из шестнадцатеричного в двоичный

- $A1_{16} = \underbrace{1010}_A \underbrace{0001}_1$



*Нет необходимости
добавлять нули, так
как количество битов
кратно 8*

Шестнадцатеричные числа

- Обычно мы используем шестнадцатеричные числа для определения местоположений в памяти (адресов памяти)
- Также при определении цветов на веб-страницах
- Каждый основной цвет – красный, зеленый и синий - характеризуется двумя шестнадцатеричными цифрами
- #RRGGBB, где RR означает красный, GG - зеленый, а BB - синий.
 - #FF0000 **красный** цвет
 - #00FF00 **зеленый** цвет
 - #0000FF **синий** цвет

Итог

- Люди используют десятичные числа, в то время как компьютеры оперируют двоичными числами.
- В двоичных числах используются 1 и 0.
- Мы группируем 8 битов, чтобы сформировать байт.
- Восьмеричные числа состоят из цифр: 0-7
- Шестнадцатеричные числа состоят из цифр: 0,1,2 ... 8,9,A, B, C, D, E, F
- Мы используем шестнадцатеричный формат для определения ячеек памяти или цветов на веб-страницах.
- Любая система счисления имеет основание K , и любое число с основанием K состоит из цифр от 0 до $K-1$.
- Мы можем легко преобразовать любое число из одного основания в другое.



Спасибо за внимание!

Из двоичной в десятичную

$$101011_2 = ?_{10}$$

Из двоичной в восьмеричную

$$101011_2 = ?_8$$

3 бита		восьмеричное число
000	→	0
001	→	1
010	→	2
011	→	3
100	→	4
101	→	5
110	→	6
111	→	7

Из двоичной в шестнадцатеричную

$$101011_2 = ?_{16}$$

4 бита

16-знач.
число

10-знач.
число

0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

Из десятичной в двоичную

$$107_{10} = ?_2$$

Из восьмеричной в двоичную

$$317_8 = ?_2$$

3 бита		восьмеричное число
000	→	0
001	→	1
010	→	2
011	→	3
100	→	4
101	→	5
110	→	6
111	→	7

Из шестнадцатеричной в двоичную

$$1AC_{16} = ?_2$$

4 бита	16-знач. число	10-знач. число
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15