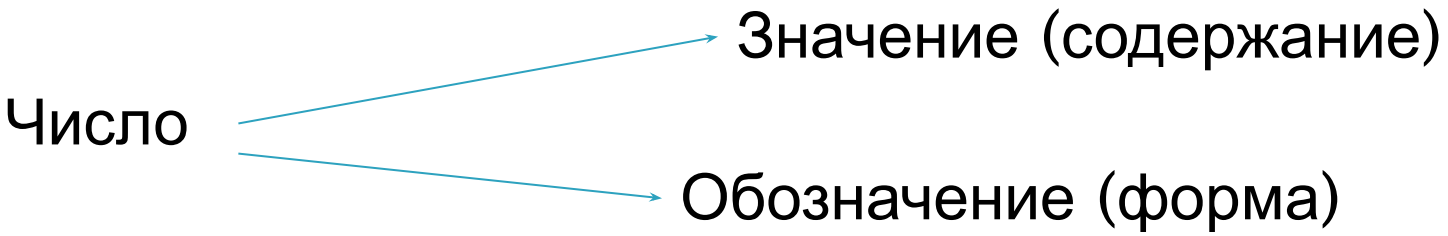
Системы счисления

Лекция 3

Значение и обозначение числа

9, IX, девять, nine, $1001_{(2)}$



Значение конкретного числа – это числовая величина, «чистая», отвлеченная от каких-либо измеряемых объектов и единиц измерения, количественная мера, выраженная в стандартных единицах.

Обозначение (форма, внешнее представление) числа – это его название или знак в некотором языке или системе обозначений, позволяющих отличать данное число от других.

Значение числа инвариантно (не зависит от обозначения).

Система счисления (с.с.) -

это система правил, позволяющих конструировать названия чисел (знаковые обозначения) некоторым регулярным способом.

Непозиционные системы счисления возникли первыми, они основаны на простом суммировании «весов» – цифр - «разновесов», занятых в записи числа.
Например, римская с.с., где все цифры могут браться плюсом или минусом, в зависимости от позиции этой цифры относительно более «тяжелых».

IX, XI

Позиционные системы счисления : число цифр конечно, вклад каждой цифры зависит от «веса» ее позиции (разряда) в записи числа.

Представление целых чисел в позиционных системах счисления с произвольным основанием

Общие свойства b-ичных позиционных систем счисления (b-с.с.) определяются параметром b - основанием с.с., которое определяет количество цифр, используемых для записи числа:
от 0 до b – 1, если $b \leq 10$.
Если $b > 10$, то используются буквы:

10 – A
11 – B
12 – C
13 – D
14 – E
15 – F

Запись целого числа

$$S = a_{k-1}a_{k-2}a_{k-3}\ldots a_2 a_1 a_{0(b)} \quad (1)$$

$0 \le a_i < b,$

i – индекс позиции (разряда), в которой расположена цифра a_i .

Запись числа называется k-значной, если индекс разряда первой значащей цифры числа равен $k-1$.

Примеры

$10011001_{(2)}$, 248933, $7DAB_{(16)}$,
$12345_{(5)}$ – неправильная запись

Соотношение записи целого числа со значением

$$S = a_{k-1}a_{k-2}a_{k-3}\ldots a_2 a_1 a_{0(b)}$$

S – запись числа.

$N(S)$ – его значение.

$$N(S) = a_{k-1}b^{k-1} + a_{k-2}b^{k-2} + \ldots + a_1 b^1 + a_0 b^0 = \sum_{i=0}^{k-1} a_i b^i \qquad (2)$$

b^i – явно указывают веса разрядов, определяющих вклад каждой цифры в значение числа,

b^i называется единицей i-го разряда b-ичного числа.

a_i – количество полных единиц i-го разряда, которое останется после вычета всевозможного числа единиц старших разрядов.

Соотношение записи целого числа со значением

$$N(S) = a_{k-1}b^{k-1} + a_{k-2}b^{k-2} + \ldots + a_1 b^1 + a_0 b^0 = \sum_{i=0}^{k-1} a_i b^i \quad (2)$$

$$N(s) = (((\ldots((a_{k-1}b + a_{k-2})b + a_{k-3})b \ldots + a_1)b + a_0 \quad (3)$$

$$\begin{cases} N_k = 0; \\ N_{i-1} = N_i b + a_{i-1}, \qquad i = k \ldots 1; \\ N(s) = N_0 \end{cases} \quad (4)$$

Значение N_i равно количеству полных единиц i-го разряда в числе.

Примеры

$N(10011_{(2)}) = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 19$

$N(10011_{(2)}) = (((1 \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1 = 19$

$N(30A_{(16)}) = 3 \cdot 16^2 + 0 \cdot 16^1 + 10 \cdot 16^0 = 778$

$N(30A_{(16)}) = (3 \cdot 16 + 0) \cdot 16 + 10 = 778$

$N(30A_{(11)}) = ?$

$N(30A_{(11)}) = 373$

Теорема 1

Любое число однозначно представимо в виде цифр заданной b-с.с.

Доказательство (от противного).

Алгоритм А1:

(перевод *b*-ичного числа в 10-с.с.)

Вход: $b > 1$, $k > 0$ (число цифр), набор $a_{k-1}, a_{k-2}, \ldots, a_1, a_0$.

$N := 0$;

$S := 1$; (S – накапливает степень, N – значение)

цикл по $i := 0$ до $k - 1$

$$N := N + a_i \cdot S;$$

$$S := S \cdot b;$$

конец цикла;

Выход: N

2k операций *

k операций +

Схема Горнера

Алгоритм А2: (перевод *b*-ичного числа в 10-с.с)

Вход: $b > 1$, $k > 0$ (число цифр), набор $a_{k-1}, a_{k-2}, \ldots, a_1, a_0$

$N := 0$;

цикл по i от k-1 вниз до 0

$N := N \cdot b + a_i$;

конец цикла;

Выход: N

k операций *

k операций +

Алгоритм А3: (перевод числа из 10-с.с. в *b*-с.с)

Вход: $N \geq 0, b > 1$;

i := 0;

цикл

a_i := N mod b; (*mod* – остаток от деления нацело)

N := N div b; (div – целое деление)

i := i + 1;

пока $N \neq 0$;

k := i;

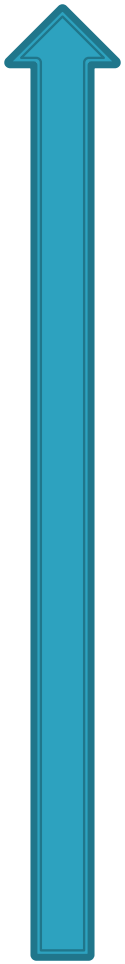
Выход: набор a_i , k (число значащих цифр)

минимальное число операций деления = k

Пример: перевод из 10-с.с. в 2-с.с.

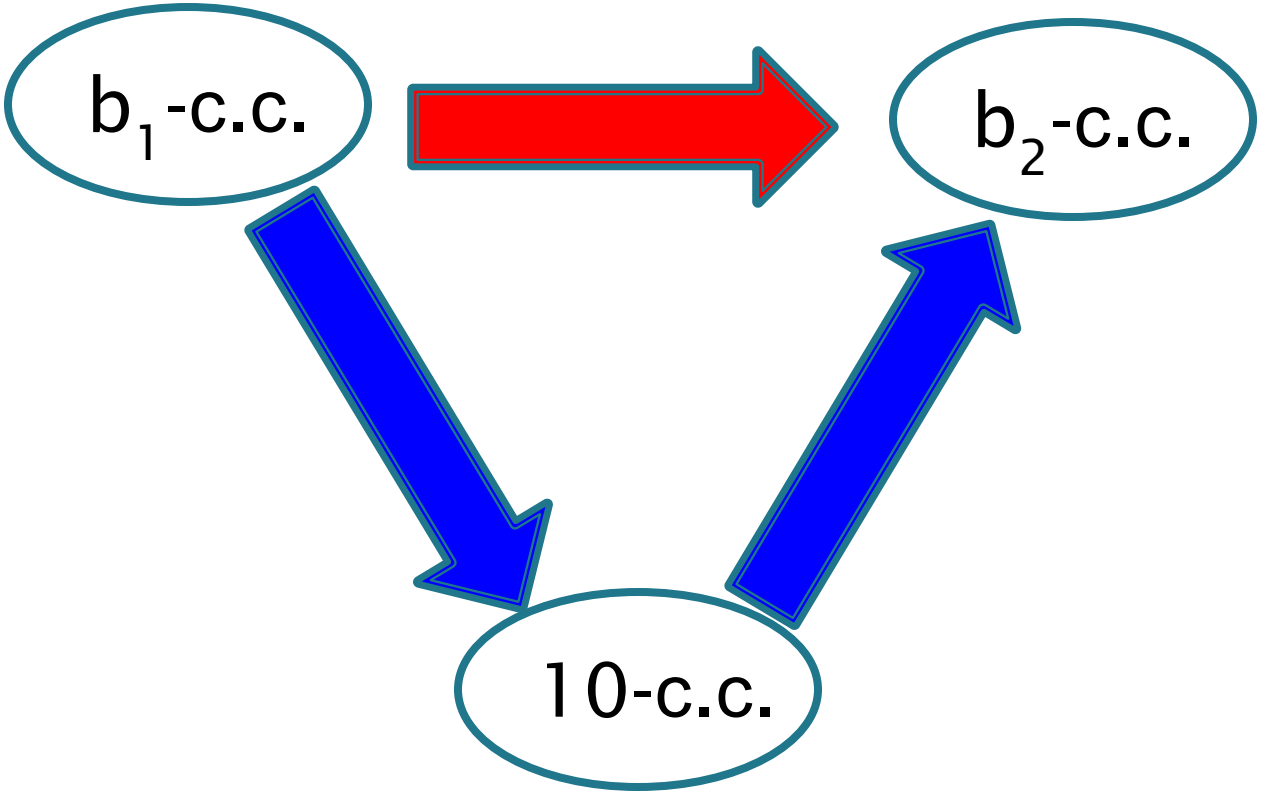
325

Целая часть	Остаток от деления на 2
162	1
81	0
40	1
20	0
10	0
5	0
2	1
1	0
0	1



$$325_{(10)} = 101000101_{(2)}$$

Перевод числа из b_1-с.с. в b_2-с.с.



Представление действительных чисел

$$S = a_{l-1}a_{l-2}a_{l-3}\ldots a_2 a_1 a_0 . a_{-1}a_{-2}a_{-3}\ldots a_{-1}\ldots_{(b)}$$

$$N(s) = a_{l-1}b^{l-1} + \ldots + a_0 b^0 + a_{-1}b^{1-} + \ldots + a_{-i}b^{-i} + \ldots = \sum_{i=\infty}^{l-1} a_i b^i. \qquad (5)$$

Если в дробной части числа конечное число знаков *k*, то нижний индекс суммы равен —*к* .

$$N_f(s) = (a_{-1} + (a_{-2} + (a_{-3} + (\ldots)/b)/b)/b)/b, \qquad (6)$$

0.375=(3+(7+5/10)/10)/10=(3+(7+(5+0)/10)/10)/10

Связь дробной части числа со значением

$$N_f(s) = (a_{-1} + (a_{-2} + (a_{-3} + (...)/b)/b)/b)/b, \qquad (6)$$

$$\begin{cases} N_{-k-1} = 0; \\ N_{-i} = (a_{-i} + N_{-i+1})/b; \quad \text{где } i = k, \ldots, 1; \\ N_f(s) = N_{-1}. \end{cases} \qquad (7)$$

Примеры

$$N(\text{«}1.101_{(2)}\text{»}) = 1\cdot 2^{0} + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3}$$
$$= 1 + 0.5 + 0.125$$
$$= 1.625$$

$$N_f(\text{«}1.101_{(2)}\text{»}) = (1 + (0 + (1 + 0)/2)/2)/2$$
$$= (1 + (0 + 0.5)/2)/2$$
$$= (1 + 0.25) / 2 = 0.625$$

$$N_f(\text{«}0.01_{(3)}\text{»}) = 1\cdot 3^{-2} = \frac{1}{9} = 0.(1)$$

Целая часть числа N_f*b ($0 < N_f < 1$) равна первой цифре дробной части числа N_f

Алгоритм А4: перевод дробной части из 10-с. с. в b-с.с

Вход: N_f ($0 \le N_f < 1$), $b > 1$;

$i := -1$;

цикл

$a_i := [N_f \cdot b]$; (взятие целой части числа)

$N_f := N_f \cdot b - a_i$; остается в том же диапазоне)

$i := i - 1$;

пока $N_f \neq 0$;

$k := i$;

Выход: набор a_i, k (число значащих цифр).

Алгоритм **А4** может не завершиться, если данное число не представимо конечной дробью в *b-c.c*

Требуется ***k*** умножений (выражение N_f*b можно вычислять в цикле один раз и хранить в промежуточной переменной).

Пример:

$$0.375 = (3 + (7 + 5/10)/10)/10 = (3 + (7 + (5 + 0)/10)/10)/10$$

$$N_f = 0.375;\ N_{-1} = 0.375;$$

$$0.375 * 2 = 0.75;\ N_{-2} = 0.75; a_{-1} = 0;$$

$$0.75 * 2 = 1.5;\ N_{-3} = 0.5; a_{-2} = 1;$$

$$0.5 * 2 = 1.0;\ N_{-4} = 0; a_{-3} = 1;$$

$$0.375_{(10)} = 0.011_{(2)}$$

$$1/4 + 1/8 = 3/8 = 0.375$$

Теорема Т2

Несократимая дробь p/q конечно представима в системе счисления с основанием b в том и только в том случае, когда все числа из разложения q на простые множители входят в такое же разложение b (количество повторений не учитывается).

Пример

121/675 конечна в 15-с.с.:

$675 = 3^3*5^2;\ 15 = 3*5;$

$1/675 = 5*15^{-3} = 0.005_{(15)};$

$121*5/15^3 = (2*15^2 + 10*15^1 + 5)/15^3 = 2/15^1 + 10/15^2 + 5/15^3$

$121/675 = 0.2A5_{(15)};$

1/10 бесконечна в 2-с.с. !!!!

Алгоритм А5:
(перевод дробной части из *b*-с.с. в 10-с.с)

Вход: $b > 1$, $\kappa > 0$ (число дробных цифр), набор a_i

$N_f := 0; S := 1;$ (*S* — накапливает степень, N_f — значение)

цикл по *i* от -1 вниз до -*k*

$$N_f := N_f + a_i \cdot S;$$

$$S := S / b$$

конец цикла

Выход: N_f

- *2k* операций *, /
- *k* операций +

Алгоритм А6:
перевод дробной части из b-с.с. в 10-с.с.

(из формулы (7) по схеме Горнера)

Вход: $b > 1$, $k > 0$ (число цифр), набор a_i

$N_f := 0;$

цикл по i от $-k$ до -1

$N_f := (a_i + N_f) / b$

конец цикла;

Выход: N_f

k операций + и /

Число N в *b*-с.с. имеющее *k* дробных цифр, при умножении на *b* становится целым (это умножение соответствует сдвигу точки на *k* позиций вправо)

Алгоритм А7

- найти целое $N_1 = N * b_1^k$ (умножением или сдвигом точки);
- выполнить для N_1 один из алгоритмов А1 или А2, затем А3;
- разделить полученный результат на b_1^k в системе b_2

Пример

Перевести $101.101_{(2)}$ в 10-с.с.

1) умножим на 2^3 → $101101_{(2)}$
2) переведем в 10-с.с. → 45
3) разделим: $45/8 = 5.625_{(10)}$

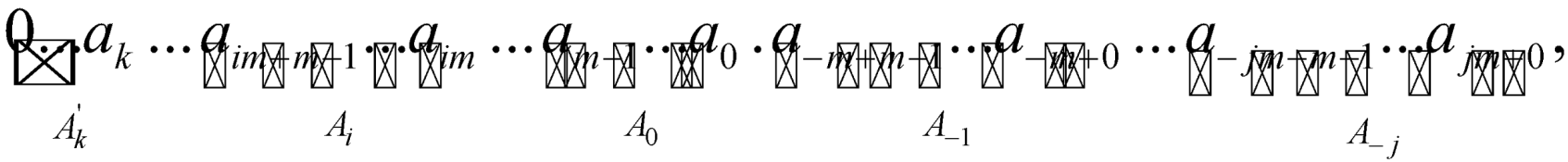
$$101.101=1*2^2+1*2^0+1*2^{-1}+1*2^{-3}=5+1/2+1/8=5.625$$

Кратные системы счисления

Если основания двух систем счисления b_1 и b_2 связаны соотношением $b_2 = b_1^m$ для некоторого натурального m, то такие системы счисления называются *кратными.*

Перевод числа из одной с. с. в другую для таких систем можно выполнить проще.

Сгруппируем цифры в b_1-записи числа по m от точки влево и вправо (добавив при нехватке цифр нужное количество незначащих нулей):



затем также сгруппируем слагаемые в формуле (5) (они содержат множитель b_1 в степени, равной индексу цифры), вынесем за скобки из каждой группы i общий множитель

$$(b_1^{im} = (b_1^{m})^i = b_2^{i})$$

и обозначим для каждой группы

$$A_i = a_{im+m-1} b_1^{m-1} + a_{im+m-2} b_1^{m-2} + \ldots + a_{im+1} b_1^{1} + a_{im} b_1^{0} \qquad (8)$$

Тогда значение исходного числа может быть представлено в виде:

$$N(S') = A_{k'} * b_2^{k'} + \ldots + A_i * b_2^{i} + \ldots + A_0 * b_2^{0} + A_{-1} * b_2^{-1} + \ldots A_{-j}\ b_2^{-j},$$

что по определению совпадает со значением записи того же числа в b_2-с.с. с цифрами A_i, если заметить, что A_i, действительно могут принимать все значения от 0 до $b_1^{m} - 1 = b_2 - 1$.

Таблицы соответствия последовательностей цифр кратных с.с.

9-с.с.	3-с.с.
0	00
1	01
2	02
3	10
4	11
5	12
6	20
7	21
8	22

8-с.с.	2-с.с.
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

16-с.с.	2-с.с.
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Алгоритм А8: перевод из меньшей кратной с.с. в большую

Вход: $b_1 > 1$, $b_2 = b_1^m$, b_1 - представление числа;

- разбить число на группы по m цифр, начиная от точки, в обе стороны (если в крайних группах цифр меньше m, добавить незначащие нули: в целой части спереди, в дробной сзади);
- заменить каждую группу b_2-цифрой по формуле (8) или таблице.

Выход: b_2 -представление исходного числа.

Алгоритм А9: перевод из большей кратной с.с. в меньшую

Вход: $b_1 > 1$, $b_2 = b_1^m$; b_2-представление числа;

- заменить каждую b_2-цифру цепочкой из m b_1-цифр по формуле (8) или таблице;
- отбросить незначащие нули слева и справа.

Выход: b_1-представление исходного числа.

Универсальные алгоритмы для арифметических операций

- Все так называемые *численные* алгоритмы для арифметических операций сложения, вычитания, умножения и деления (в том числе, вычисления «столбиком») являются *символьными,* потому что оперируют входными, выходными и промежуточными данными как строками символов.
- Символьные вычисления являются *формальными в* том смысле, что манипулируют только знаками, не обращаясь к их значениям.
- *Абстрагирование* от смысла данных различной природы и описание алгоритма в терминах чисто символьных преобразований является одним из основных методов программирования обработки данных произвольной природы

Алгоритм А10: сложение двух чисел

Вход: две строки цифр, представляющие слагаемые;

- **выравнивание:** расположить слагаемые одно под другим в произвольном порядке так, чтобы разряды с одинаковым весом находились друг под другом; если какое-то число короче других слева или справа, дополнить его нулями;
- **начальные установки**:

  обнулить цифру переноса в следующий разряд;

  установить результат равным пустой строке;
- ***цикл*** по текущему разряду от младшего до старшего:

  определить сумму переноса и цифр в столбце текущего разряда чисел; младшую цифру суммы записать в текущий разряд результата, старшую — в перенос;

  конец цикла;
- **окончание:** если перенос не равен 0, то дописать перенос в начало результата

Выход: строка, представляющая результат.

Единственное место в этом алгоритме, где присутствует обращение к значениям цифровых символов, — это поразрядное сложение в цикле.

Действительно, из одного лишь вида знаков «2» и «3» нельзя извлечь информацию, что результатом их сложения будет знак «5».

Эти сведения можно задать, например, двумя таблицами сложения: в одной для каждой пары цифр записать младшую цифру результата, в другой — цифру переноса («0» или «1»);

исчерпав таким образом все немногочисленные случаи, можно заменить операцию сложения значений операцией выборки знака из таблицы.

Чтобы учесть сложение с переносом, можно завести две пары таблиц или записать в каждую клетку по две цифры.

Алгоритм А10 замечателен тем, что применим к произвольной позиционной с. с. при соответствующей замене таблиц сложения.

+	0	1
0	0	1
1	1	0

++	0	1
0	0	0
1	0	1

*	0	1
0	0	0
1	0	1

Затраты памяти на хранение чисел и времени на выполнение операций с ними зависят от длины записи числа в цифрах рабочей системы счисления.

Для заданной *b*-с. с. следующие величины:

k_n — длина записи (натурального) числа ***N***,

N_k — максимальное натуральное число, записываемое ***k*** цифрами, связаны соотношениями:

$k_n = [log_b N] + 1$, где $[x]$ — наибольшее целое, не превышающее ***x***;

$N_k = b^k - 1$.

Верхние оценки для размера результата арифметической операции над парой целых чисел N_1 и N_2 (пусть $N_1 > N_2$):

для сложения и вычитания — $k_{N1} + 1$,

для умножения — $k_{N1} + k_{N2}$,

для деления — $k_{N1} + 1$, (так как $N_2 > 1$).

Особенности умножения и деления на основание системы счисления

В любой b-с.с. b записывается как «$10_{(b)}$».

Умножение на b сводится к дописыванию 0 справа к целому числу или сдвигу b-ичной точки на один разряд вправо.

Обратно: деление на b равносильно сдвигу b-ичной точки на один разряд влево, или отбрасыванию младшей цифры целого числа при делении нацело.

b^k всегда представляется единицей с k нулями.

Умножение (деление) на b^k сводится к сдвигу b-ичной точки на k разрядов вправо (влево).

Остатком от деления на b^k является число, составленное из k младших цифр.

<u>Примеры:</u>

$7 = 10_{(7)}$

$5^4 = 10000_{(5)}$

$1001.1101_{(2)} \cdot 2^3 = 1001.1101_{(2)} \cdot 1000_{(2)} = 1001110.1_{(2)}$

Арифметические сдвиги

Добавление *k* нулей справа и отбрасывание *k* младших цифр можно рассматривать как операции арифметического сдвига на *k* позиций.

В Си определены операции арифметического сдвига на *k* позиций, которые равносильны умножению или целочисленному делению на 2^k.

<< — сдвиг влево

>> — сдвиг вправо

Примеры:

```
a = 5 << 3;  /*  после выполнения присваивания a будет иметь
     значение 40 */
b = 112 >> 4; /* b будет равно 7 */
```

Особенности двоичной арифметики

+	0	1
0	0	1
1	1	0

++	0	1
0	0	0
1	0	1

*	0	1
0	0	0
1	0	1

Логическая аналогия

Если сопоставить 0 – логическую «ложь», а 1 – «истину», то таблица сложения соответствует логической операции «исключающее или», а таблицы переноса и умножения – логической операции «и».

Минимаксная аналогия

$a \cdot b = \min(a, b)$

$a + b = \max(a, b) - \min(a, b)$

Умножение столбиком многозначных чисел в 2-с.с. реализуется с помощью сложения и сдвига:

```
    1001001
  *   1011
  ---------
    1001001
+  1001001
  1001001
 ---------
 1100100011
```

Задачи

Задача 1. Выразить целую часть 17.5 * X через сложение и операции поразрядных сдвигов числа X вправо и влево.

$17.5_{(10)} = 16 + 1 + 0.5 = 2^4 + 2^0 + 2^{-1} = 10001.1_{(2)}$

$17.5 * X = X * (2^4 + 2^0 + 2^{-1}) =$

$= X*2^4 + X*2^0 + X*2^{-1} =$

$= (X << 4) + X + (X >> 1)$

Задача 2. Если $120_{(x)}$ делится на $11_{(10)}$, то как выглядит (чему равно?) 3^{10} в системе счисления с основанием *x (x* > 3*)*?

Подбором можно определить, что x = 9, т.к. $120_{(9)} = 99_{(10)}$ – делится на 11 без остатка.

$3^{10} = 3^{2*5} = (3^2)^5 = 9^5 = 100000_{(9)}$