

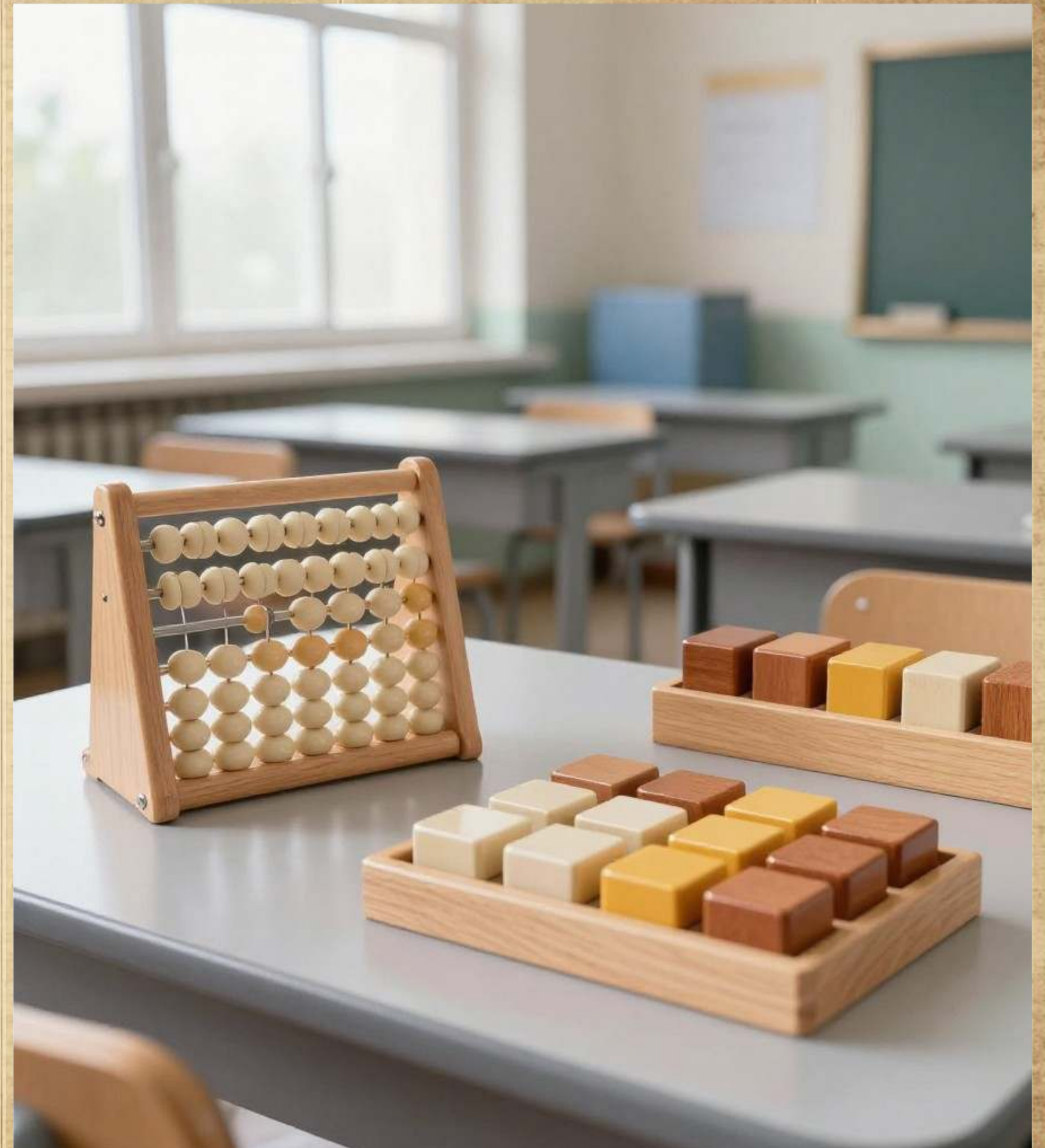
Арифметика в произвольной позиционной системе счисления

Число задаётся разрядами и основанием b ; значит, правила операций одинаковы, но переносы зависят от b .



Зачем нужны разные системы счисления в информатике

Компьютер хранит данные в двоичном виде: схемам проще различать два состояния — 0 и 1. Длинные цепочки удобно записывать короче в восьмеричной или шестнадцатеричной системах. Основание показывает, сколько разных цифр используется.



Цифры, разряды и правила переноса

В системе с основанием b допустимы цифры от 0 до $b-1$. Позиция цифры задаёт её вклад: правый разряд отвечает за «мелкие» значения, левый — за более «крупные».

При сложении в одном разряде суммарное значение может достигнуть основания b . Тогда появляется перенос в следующий разряд, а в текущем разряде остаётся только остаток после деления на b .

При вычитании ситуация обратная: если не хватает, чтобы вычесть цифру, берут заем из ближайшего старшего разряда. Он уменьшается на 1, а текущий разряд получает b , после чего операция продолжается.

Какие цифры допустимы в разных основаниях

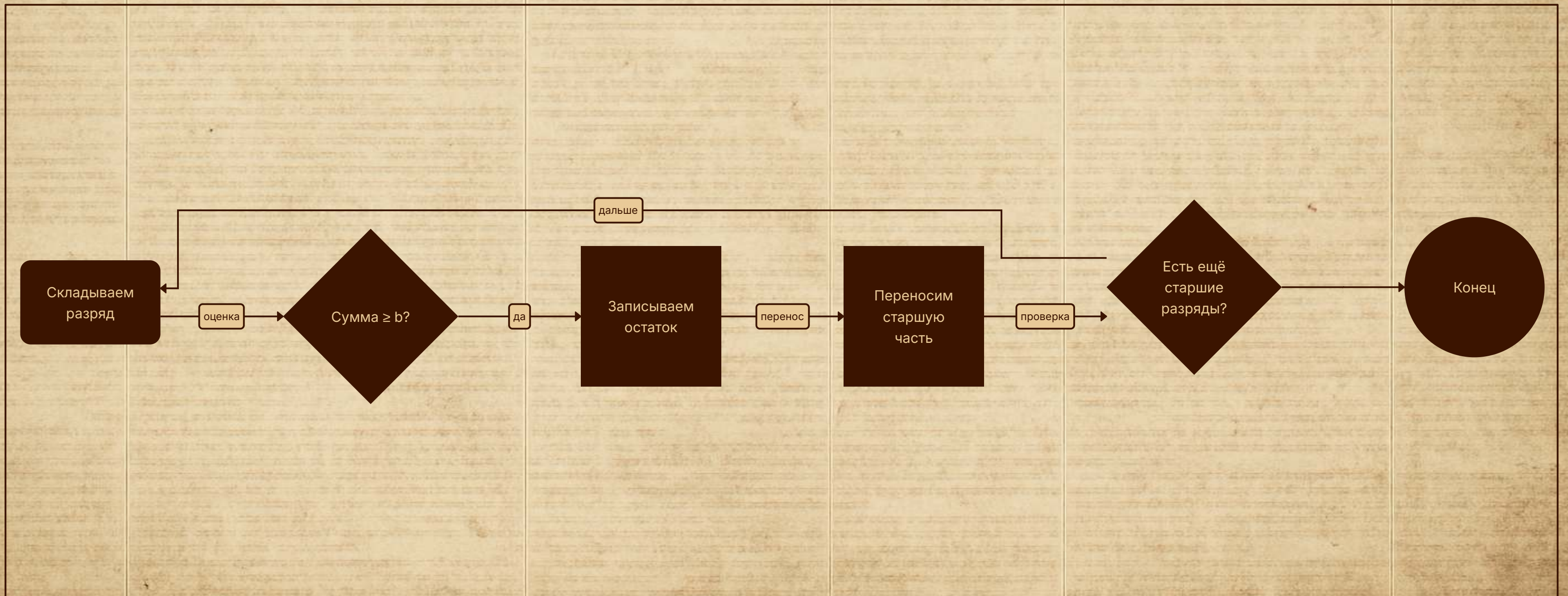
Основание	Допустимые цифры
2	0, 1
8	0–7
10	0–9
16	0–9, A–F

В любой позиционной системе самая большая цифра ровно на 1 меньше основания — потому что цифры перечисляют все значения от 0 до $b-1$.

Максимальное значение цифры всегда равно $b-1$. Это помогает быстро понимать, что «записать» можно, а что — нельзя.

Общий алгоритм сложения в любой системе

Основу b определяет момент переноса между разрядами.



Сложение без переноса через основание

Если в каком-то разряде сумма цифр меньше основания b , перенос не появляется. Тогда в текущий разряд просто записывается получившаяся цифра, и вы переходите к следующему разряду.

Пример в системе с основанием 7: $23_7 + 12_7 = 35_7$.
Здесь $3 + 2 = 5$ и $2 + 1 = 3$ — ни одна сумма не достигает 7, значит переносов нет.

По сути, правила совпадают с привычными: вы складываете «как обычно», но каждый раз контролируете границу b . Как только она не достигнута — всё остаётся на месте.

Сложение с переносом

- Если сумма цифр в разряде равна или больше основания b , нужно записать остаток, а «лишнее» отправить в следующий разряд как перенос. Это делает запись короче и сохраняет правильное значение числа.
- Пример: $1011_2 + 0110_2 = 10001_2$. Здесь в некоторых разрядах $1 + 1$ даёт 0, а перенос равен 1; на рисунке переносы показывают над теми разрядами, где они возникают.

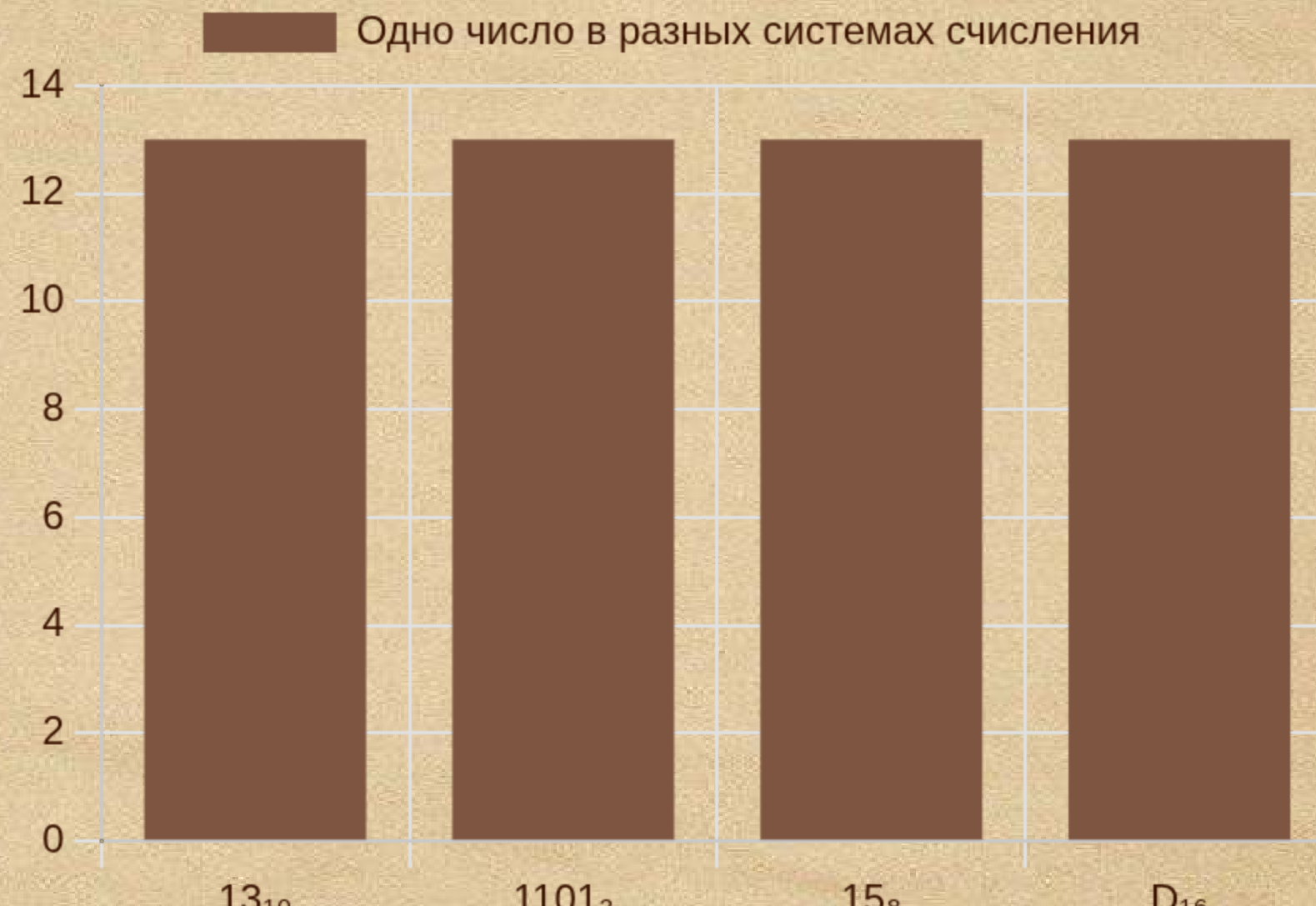


Один и тот же результат в разных системах

Чем больше основание, тем «реже» приходится менять разряды. Поэтому разные записи могут быть разной длины, но соответствовать одному и тому же количеству.

Смысл числа определяется разрядами и основанием, а не формой записи. Это объясняет, почему одни и те же данные в компьютере выглядят по-разному при разных форматах.

Сравнение по правилам позиционных систем счисления (данные для примера)



Вычитание по разрядам

Вычитание выполняется справа налево: сначала работают с младшими разрядами, затем переходят к старшим. Это помогает сразу учитывать «вклад» каждой позиции в число.

Если в текущем разряде верхняя цифра меньше нижней, берут заем из старшего разряда. Старший уменьшается на 1, а текущий разряд получает b — так становится возможным вычитание.

После займа вычитание продолжается по правилам для текущих цифр. Такой подход работает в любых основаниях, включая 2, 8 и 16, потому что во всех них граница «цифры» определяется одним и тем же b .

Пример вычитания с заимствованием

1

Записываем пример

Дано $504_8 - 276_8$.
Начинаем вычитание с младшего разряда (справа), то есть с разрядов единиц, десятков и сотен.

2

Пробуем младший разряд

В младшем разряде нужно выполнить $4 - 6$. Это невозможно, поэтому требуется заем из старшего разряда, чтобы увеличить текущий разряд на 8.

3

Оформляем заем

В системе с основанием 8 заем превращает 4 в 14_8 (то есть добавляет 8). Одновременно старший разряд уменьшается на 1, и дальнейшее вычитание становится выполнимым.

4

Доводим до конца по разрядам

После займа — выполняем вычитание по разрядам без нарушения правил системы и получаем результат 206_8 . Проверять полезно тем же способом — по разрядам и смыслу заимствований.

Короткие примеры сложения и вычитания

Основание	Пример	Результат
16	$15_{16} + 7_{16}$	$1C_{16}$
2	$101_2 + 11_2$	1000_2
5	$32_5 - 14_5$	13_5

Короткие примеры в разных основаниях: сложение и вычитание выполняются по общим правилам позиционных систем, а затем результат можно проверить переводом в десятичную систему.

Проверка через перевод в десятичную систему помогает убедиться, что переносы и заимствования выполнены верно.

Учебные преобразования в позиционных системах счисления.

Умножение как многократное сложение

Почему умножение раскладывается по разрядам

Умножение в позиционной системе можно понимать как повторное сложение: каждый разряд первого множителя создаёт частичное произведение, а затем эти частичные значения складываются с учётом разрядов. Так сохраняется порядок и вес разрядов.

Роль переносов при умножении

Например, $12_4 \times 3_4 = 102_4$. Здесь умножение 2×3 даёт 6, то есть в системе с основанием 4 это 2 с переносом 1. Перенос добавляется к следующему разряду, формируя 0 затем 1.



Умножение на однозначное число

При умножении на однозначное число каждый разряд исходного числа умножается отдельно. Затем полученные цифры приводятся к основанию b , потому что в одном разряде помещается только значение от 0 до $b-1$.

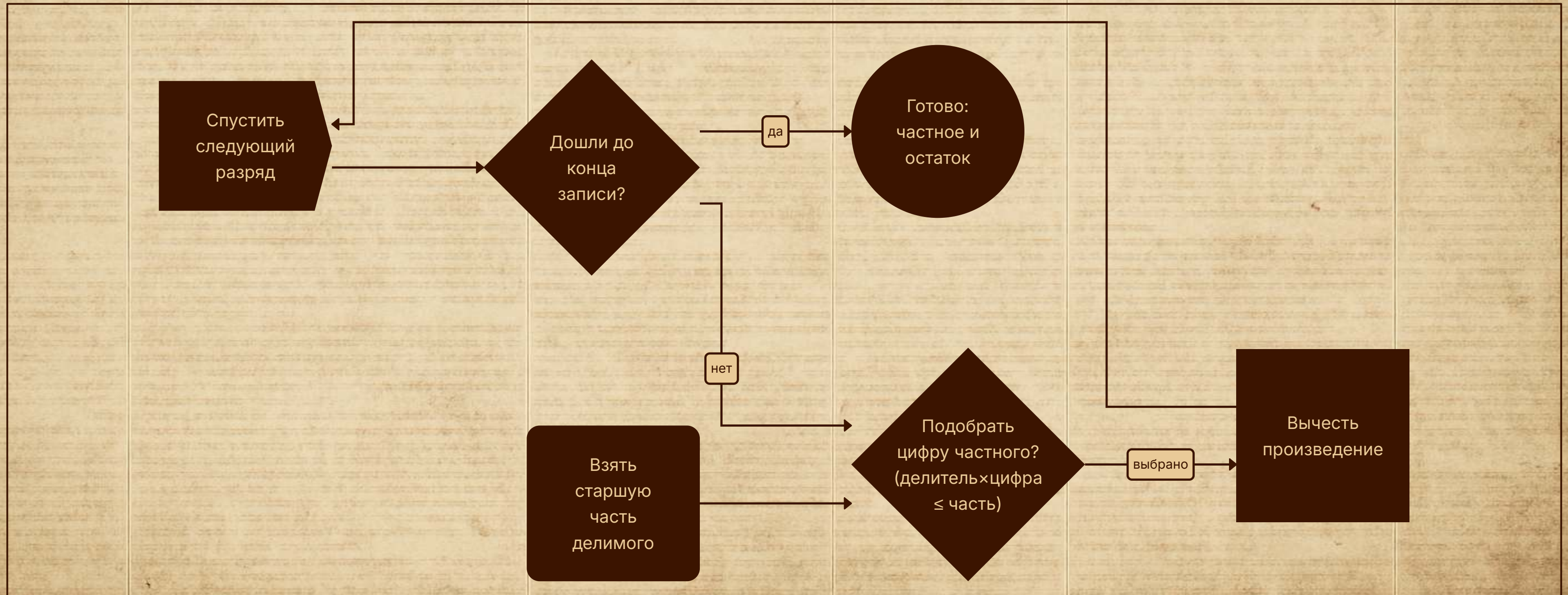
В системе с основанием 12 используются цифры 0–9, A и B, причём B означает 11. Например, в $7B_{12} \times 2_{12}$ нужно умножить 7 и 11 на 2, а затем оформить перенос по основанию 12.

После умножения обязательно учитывается перенос: если произведение превышает $b-1$, лишняя часть отправляется в следующий, более старший разряд. Так результат остаётся корректным именно для выбранной системы.

Проверка удобна: можно перевести промежуточные значения в десятичную систему или пересчитать результат обратным способом. Это особенно помогает не спутать цифру B с обычной буквой при письме.

Алгоритм деления в произвольной системе

Деление выполняется как последовательное построение частного по разрядам: берём старшую часть, выбираем цифру частного, вычитаем и «спускаем» следующий разряд.



Проверка деления обратным действием



Умножение частного

Проверяем результат: находим произведение делителя на полученное частное в той же системе счисления, где выполнялось деление, чтобы сохранить правило разрядов и переносов.



Добавление остатка

К произведению прибавляем остаток. Если сумма совпадает с делимым, значит и выбор цифр частного, и вычитания по разрядам выполнены верно.



Почему это надёжно

Этот способ особенно полезен в основаниях 2, 8 и 16: ошибки чаще возникают из-за неверного переноса или заимствования, а обратное действие сразу выявляет несоответствие.

Типичные ошибки при вычислениях

Чтобы уменьшить риск, полезно отмечать перенос и заем прямо над разрядами и каждый раз сравнивать получаемую цифру с допустимым диапазоном $0 \dots b-1$.

Чаще всего ошибки связаны не с «незнанием» правил, а с нарушением границы разряда: без учёта основания b легко получить неправильные переносы и заимствования.

Свод наблюдений из типовых учебных задач по арифметике в позиционных системах счисления (структурированная качественная оценка).

Частые причины ошибок при вычислениях



Способы проверки ответа

Первый способ: перевести найденный результат в десятичную систему и сравнить значения. Этот метод прост, но требует аккуратности при переводе, чтобы не ошибиться в разрядных весах.

Второй способ опирается на порядок числа: после сложения результат не может быть меньше каждого из слагаемых. Это быстрая «проверка здравого смысла», особенно в задачах без сложных вычислений.

Третий способ — обратное действие. Для вычитания можно проверить сложением, а для деления — умножением и добавлением остатка. Он хорошо ловит ошибки переносов и заимствований.

Тренировочные примеры для 8 класса

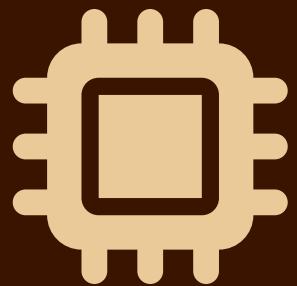
В каждой строке потренируйтесь выполнять переносы и заимствования по разрядам, а затем проверьте корректность ответа тем же способом, что и в примерах.

Практика на разных основаниях помогает привыкнуть к разным границам переноса и к допустимым цифрам, что снижает количество ошибок.

Задание	Основание	Действие
$1011_2 + 110_2$	2	сложение
$43_8 - 17_8$	8	вычитание
$2A_{16} + 5_{16}$	16	сложение
$31_5 \times 2_5$	5	умножение

Типовые учебные задания по арифметике в позиционных системах счисления.

Связь темы с работой компьютера

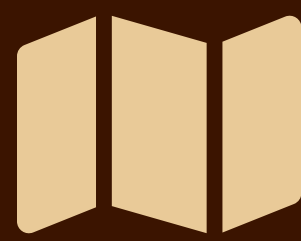


Почему в компьютере важно двоичное основание

Процессоры и память работают с двоичными сигналами, где удобно различать два состояния. Поэтому арифметика в двоичной системе лежит в основе вычислений машины.



Как помогают 8 и 16
Для записи и чтения больших двоичных чисел используют восьмеричную и шестнадцатеричную системы. Они делают запись короче, сохраняя связь с теми же разрядами.



Адреса и коды
Арифметика по разрядам нужна при обработке адресов памяти и кодов данных. В таких операциях важны корректные переносы и заимствования, чтобы данные не исказились.



Представление больших чисел
Преобразования между основаниями помогают понимать, как одно и то же количество представляется по-разному. Это полезно при отладке и при чтении результатов программ.

Арифметика по разрядам — универсальный подход

Все арифметические действия в любой позиционной системе строятся на одном принципе: считать по разрядам и учитывать основание b .
Сложение использует перенос, вычитание — заем, умножение — разложение по разрядам, деление — подбор частного.

Announcement

Office



laking
Ace

Making A

THE PLAN

Bus

Trade Rev

Issue