

## Представление целых чисел

Самым первым видом данных, с которыми начали работать компьютеры, были числа. ЭВМ первого поколения могли производить только математические расчёты (вычисления).

Из курса информатики основной школы вы помните, что компьютеры работают с целыми и вещественными числами. Их представление в памяти осуществляется разными способами.

Во многих задачах, решаемых на компьютере, обрабатываются целочисленные данные. Прежде всего, это задачи экономического характера, при решении которых данными служат количества акций, сотрудников, деталей, транспортных средств и др. Целые числа используются для обозначения даты и времени, для нумерации различных объектов: элементов массивов, записей в базах данных, машинных адресов и т. д. По своей природе множество целых чисел дискретно, т. к. состоит из отдельных элементов.

И хотя любое целое число можно рассматривать как вещественное, но с нулевой дробной частью, предусмотрены специальные способы представления целых чисел. Это обеспечивает: эффективное расходование памяти, повышение быстродействия, повышение точности вычислений за счёт введения операции деления нацело с остатком.

Для компьютерного представления целых чисел используется несколько различных способов, отличающихся друг от друга количеством разрядов (под целые числа обычно отводится 8, 16, 32 или 64 разряда) и наличием или отсутствием знакового разряда.

**Беззнаковое представление** можно использовать только для неотрицательных целых чисел.

Для получения компьютерного представления беззнакового целого числа в  $n$ -разрядной ячейке памяти достаточно перевести его в двоичную систему счисления и, при необходимости, дополнить полученный результат слева нулями до  $n$ -разрядов.

Например, десятичные числа 130 и 39 в восьмиразрядном представлении будут иметь вид:



Понятно, что существуют ограничения на числа, которые могут быть записаны в  $n$ -разрядную ячейку памяти. Максимальное значение целого неотрицательного числа достигается в случае, когда во всех разрядах ячейки хранятся единицы. Для  $n$ -разрядного представления оно будет равно  $2^n - 1$ . Минимальное число соответствует  $n$  нулям, хранящимся в  $n$  разрядах памяти, и равно нулю. Далее приведены диапазоны значений для беззнаковых целых  $n$ -разрядных чисел:

Количество разрядов	Диапазон чисел
8	[0; 255]
16	[0; 65 535]
32	[0; 4 294 967 295]
64	[0; 18 446 744 073 709 551 615]

**При знаковом представлении** целых чисел старший разряд ячейки отводится под знак (0 — для положительных, 1 — для отрицательных чисел), а остальные разряды — под цифры числа.



Представление числа в привычной для человека форме «знак-величина», при которой старший разряд ячейки отводится под знак, а остальные разряды — под цифры числа, называется **прямым кодом**.

Например, прямые коды чисел 48 и -52 для восьмиразрядной ячейки равны:

0 0 1 1 0 0 0 0

1 0 1 1 0 1 0 0

Минимальное отрицательное число, которое можно записать в знаковом представлении в  $n$  разрядах, равно  $2^{n-1}$ . Максимальное положительное число, которое можно записать в знаковом представлении в  $n$  разрядах, равно  $2^{n-1} - 1$ . Ниже приведены диапазоны значений для знаковых представлений целых чисел в ячейках с различной разрядностью:

Количество разрядов	Диапазон чисел
8	[-128; 127]
16	[-32 768; 32 767]
32	[-2 147 483 648; 2 147 483 647]
64	[-9 223 372 036 854 775 808; 9 223 372 036 854 775 807]




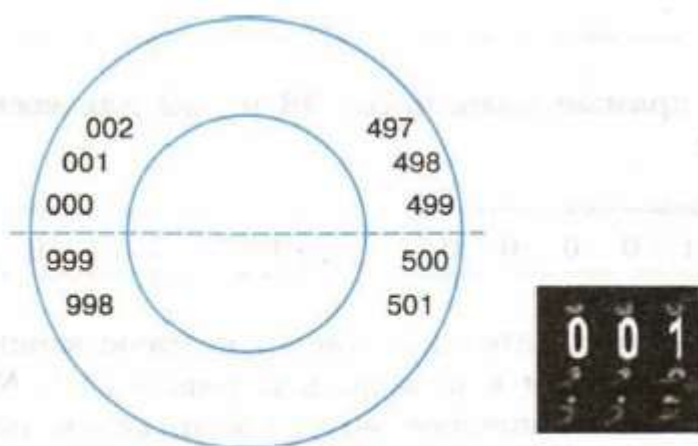
В математике множество целых чисел бесконечно.

Компьютер работает с ограниченным множеством целых чисел.

Прямой код положительного числа отличается от прямого кода равного по абсолютной величине отрицательного числа только содержимым знакового разряда.

В прямом коде числа можно хранить, но выполнение арифметических операций над числами в прямом коде затруднено — оно требует более сложной архитектуры центрального процессора, «умеющего» выполнять не только сложение, но и вычитание, а также «знающего» особый алгоритм обработки не имеющего «веса» знакового разряда. Этим трудностям позволяет избежать использование дополнительного кода.

 Чтобы понять сущность дополнительного кода, рассмотрим работу реверсивного счётчика, последовательность показаний которого можно представить в виде замкнутого кольца из чисел (рис. 3.5).



**Рис. 3.5.** Реверсивный счётчик

При возрастании показаний счётчика до максимального, например до 999, следующими его состояниями должны быть 1000, 1001, 1002 и т. д. Но для изображения старшей единицы в счётчике не хватает разряда, происходит переполнение разрядной сетки. Поэтому мы увидим 000, 001, 002 и т. д.

При убывании показаний счётчика после состояния 000 будут идти 999, 998, 997 и т. д. Но после достижения нуля последовательное вычитание единицы должно давать -1, -2, -3 и т. д.

Будем рассматривать числа 999, 998, 997 как коды чисел -1, -2, -3 и проверим на их примере соотношение:  $y + (-y) = 0$ :

$$001 + 999 = 1000;$$

$$002 + 998 = 1000;$$

$$003 + 997 = 1000.$$

С учётом того что единица переполнения теряется, мы, сложив число и код противоположного ему числа, получаем ноль!

Вот ещё несколько примеров:

$$5 - 2 = 5 + [-2] = 5 + 998 = 1003;$$

$$7 - 5 = 7 + [-5] = 7 + 995 = 1002.$$

Для устранения неоднозначности в кольце будем считать половину состояний (0-499) кодами нуля и положительных чисел, а оставшуюся половину (500-999) — кодами отрицательных чисел.

Таким образом, дополнительный код положительного числа совпадает с этим числом, а для отрицательного числа он равен дополнению его величины до числа  $q^n$ , возникающего при переполнении разрядной сетки. Здесь  $q$  — основание системы счисления,  $n$  — число разрядов в разрядной сетке.

Рассмотрим алгоритм получения дополнительного  $n$ -разрядного кода отрицательного числа:

- 1) модуль числа представить прямым кодом в  $n$  двоичных разрядах;
- 2) значения всех разрядов инвертировать (все нули заменить единицами, а единицы — нулями);
- 3) к полученному представлению, рассматриваемому как  $n$ -разрядное неотрицательное двоичное число, прибавить единицу.



**Пример 1.** Найдём 16-разрядный дополнительный код отрицательного числа  $-2017_{10}$ .

Прямой код $ -2017_{10}  = 2017_{10}$	$00000111\ 11100001_2$
Инвертирование	$11111000\ 00011110_2$
Прибавление единицы	+ $1$
Дополнительный код	$11111000\ 00011111_2$

Использование дополнительного кода позволяет свести операцию вычитания чисел к операции поразрядного сложения кодов этих чисел.



**Пример 2.** Как известно,  $48 - 2017 = -1969$ .

Выполним эту операцию в 16-разрядных машинных кодах.

Нам потребуются прямой код числа 48 и дополнительный код числа -2017.

Прямой код числа 48	$00000000\ 00110000_2$
Дополнительный код числа -2017	$11111000\ 00011111_2$
Поразрядное сложение кодов	$11111000\ 01001111_2$

Рассмотрим полученный результат. Это отрицательное число (об этом говорит 1 в знаковом разряде), представленное в дополнительном коде. Перейдём к прямому коду модуля соответствующего числа, по которому сможем восстановить десятичное представление результата.

Прямой код можно получить из дополнительного кода, если применить к нему операцию инвертирования и прибавить единицу.

Инвертирование	$00000111\ 10110000_2$
Прибавление единицы	+ $1$
Прямой код модуля результата	$00000111\ 10110001_2$

Получаем:  $-11110110001_2 = -1969$ .

## Представление вещественных чисел

В математике множество вещественных чисел непрерывно, бесконечно и не ограничено.



Попробуйте обосновать это утверждение.

Вещественные числа записываются в естественной или в экспоненциальной форме.

В жизни мы чаще пользуемся естественной формой записи чисел, при которой: число представляется последовательностью десятичных цифр со знаком плюс или минус, знак плюс может опускаться, для разделения целой и дробной частей числа используется запятая. Например: 12,34; 0,0056; -708,9.



В экспоненциальной форме вещественное число  $a$  представляется как  $a = \pm m \cdot q^p$ , где  $m$  — мантисса числа,  $q$  — основание системы счисления,  $p$  — порядок числа.

Например, длину некоторого отрезка, равного 47,8 см, можно записать так:

- 1)  $478 \cdot 1^{0-1}$  см;
- 2)  $47,8 \cdot 10^0$  см;
- 3)  $4,78 \cdot 10^1$  см;
- 4)  $0,478 \cdot 10^2$  см;
- 5)  $0,000478 \cdot 10^5$  см.

Такое многообразие вариантов записи в экспоненциальной форме одного и того же числа не всегда удобно. Для однозначного представления вещественных чисел в компьютере используется нормализованная форма.



Нормализованная запись отличного от нуля вещественного числа<sup>1</sup> — это запись вида  $a = \pm m \cdot q^p$ , где  $p$  — целое число (положительное, отрицательное или ноль),  $m$  — дробь, целая часть которой содержит одну значащую (ненулевую) цифру, т. е.  $1 \leq m < q$ .

<sup>1</sup> Стандарт IEEE 754.

Примеры нормализации чисел:

- 1)  $31,415926 = 3,1415926 \cdot 10^1$ ;
- 2)  $1000 = 1,0 \cdot 10^3$ ;
- 3)  $0,123456789 = 1,23456789 \cdot 10^{-1}$ ;
- 4)  $0,0000107_8 = 1,07_8 \cdot 10_8^{-5}$ ;
- 5)  $1000,0001_2 = 1,0000001_2 \cdot 10_2^{11}$ ;
- 6)  $AB,CDEF_{16} = A,BCDEF_{16} \cdot 10_{16}^1$ .

Диапазон вещественных чисел в памяти компьютера очень широк, но, тем не менее, ограничен. Множество вещественных чисел, которые могут быть представлены в компьютере, конечно.

Поясним это на примере калькулятора, который производит вычисления в десятичной системе счисления. Пусть это будет калькулятор с десятью знаковыми местами на дисплее:





- 6 знакомест отводится под мантиссу (одно знакоместо отводится под знак мантиссы, четыре — под цифры мантиссы, одно — под точку, разделяющую целую и дробную части мантиссы);
- одно знакоместо отводится под символ «Е»;
- три знакоместа отводятся под порядок (одно — под знак порядка, два — под цифры порядка).

У калькуляторов первая значащая цифра, с которой и начинается мантисса, изображается перед точкой.

Число 12,34 в таком калькуляторе будет представлено как +1.234E+01.

Число 12,35 будет представлено как +1.235E+01.

Как известно, между числами 12,34 и 12,35 находится бесконечное множество вещественных чисел, например: 12,341; 12,3412; 12,34123 и т. д.

Каждое из этих чисел в нашем калькуляторе будет представлено как + 1.234E+01. Для последних разрядов у нас просто не хватает знакомест! Аналогичная ситуация имеет место и в компьютерном представлении вещественных чисел, независимо от того, ячейки какой разрядности там использованы.

Получается, что точно мы можем представить в компьютере лишь некоторую конечную часть множества вещественных чисел, а остальные числа — лишь приближённо.

Таким образом, множество вещественных чисел, представляемых в компьютере, дискретно, конечно и ограничено.

## Самое главное

В математике множество целых чисел дискретно, бесконечно и не ограничено.

Для компьютерного представления целых чисел используется несколько различных способов, отличающихся друг от друга количеством разрядов (8, 16, 32 или 64 разряда) и наличием или отсутствием знакового разряда. В любом случае компьютерное представление целых чисел дискретно, конечно и ограничено.

В математике множество вещественных чисел непрерывно, бесконечно и не ограничено.

Для компьютерного представления вещественных чисел используется нормализованная запись вещественного числа  $a = \pm m \cdot q^p$ , где  $q$  — основание системы счисления,  $p$  — целое число (положительное, отрицательное или ноль),  $m$  — дробь, целая часть которой содержит одну значащую (ненулевую) цифру, т. е.  $1 \leq m < q$ .

Компьютерное представление вещественных чисел дискретно, конечно и ограничено.

## Вопросы и задания

1. Представьте в восьмиразрядном формате прямые коды десятичных чисел:

1) 64; 2) 58; 3) 72; 4) -96.

2. Можно ли числа  $43_{16}$ ,  $101010_2$ ,  $129_{10}$  и  $-52_{10}$  сохранить в однобайтовом формате?

3. Как определяется диапазон представления в компьютере целых чисел без знака? Со знаком?

4. Почему множество целых чисел, представимых в памяти компьютера, дискретно, конечно и ограничено?

5. Представьте в восьмиразрядном формате дополнительные коды двоичных чисел:

1) +1010; 2) -1001; 3) -11; 4) -11011.

6. Найдите десятичные эквиваленты чисел, представленных в прямом коде:

1) 00000100; 2) 00001001; 3) 10000011; 4) 10000110.

\*7. Найдите десятичные эквиваленты чисел, представленных в дополнительном коде:

1) 00000100; 2) 11111001.

8. Для хранения целого числа со знаком в компьютере используется два байта. Сколько единиц содержит внутреннее представление числа -101, записанного:

- 1) в прямом коде;
- 2) в дополнительном коде?



9. Вычислите с помощью калькулятора (приложение Windows) в режиме «Программист» следующие примеры:

- 1)  $111011101_2 - 1101110110_2$ ;
- 2)  $1101101001_2 - 11000100100_2$ .

Как вы можете объяснить полученные результаты?

10. Запишите десятичные числа в нормализованной форме:

1) 217,934; 2) 75321; 3) 10,0101; 4) 200450.

11. Сравните следующие числа:

1)  $318,4785 \cdot 10^9$  и  $3,184785 \cdot 10^{11}$ ;

2)  $218,4785 \cdot 10^{-3}$  и  $1847,85 \cdot 10^{-4}$ .

12. Выполните операцию сложения:

1)  $0,397621 \cdot 10^3 + 0,2379 \cdot 10^1$ ;

2)  $0,251452 \cdot 10^{-3} + 0,125111 \cdot 10^{-2}$ .

13. Чем ограничивается диапазон представимых в памяти компьютера вещественных чисел?

14. Почему множество вещественных чисел, представимых в памяти компьютера, дискретно, конечно и ограничено?

\*15. Попробуйте самостоятельно сформулировать основные принципы представления данных в компьютере.