Представление чисел в памяти компьютера

презентация подготовлена

учителем информатики МОУ СОШ №8

Константиновой Еленой Ивановной

Как представляются в компьютере целые числа?

Целые числа могут представляться в компьютере со знаком или без знака.

Целые числа без знака обычно **занимают в памяти один или два байта** и принимают в однобайтовом формате значения от 00000000_2 до 11111111_2 , а в двубайтовом формате - от $00000000\ 00000000_2$ до 1111111111111111_2.

Диапазоны значений целых чисел без знака

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	$0 \ldots 2^{8}-1$	0 ... 255
2	$0 \ldots 2^{16}-1$	0 ... 65535

Число 39_{10} = 100111_2 в *однобайтовом* формате:

Номера разрядов	7	6	5	4	3	2	1	0
Биты числа	0	0	1	0	0	1	1	1

Число 39_{10} = 100111_2 в *двубайтовом* формате:

Номера разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1

Число $65\ 535_{10}$ = $11111111\ 11111111_2$ в *двубайтовом* формате:

Номера разрядов	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Биты числа	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Целые числа со знаком обычно занимают в памяти компьютера один, два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа. Знак "плюс" кодируется нулем, а "минус" - единицей.

Диапазоны значений целых чисел со знаком

Формат числа в байтах	Диапазон	
	Запись с порядком	Обычная запись
1	-2^{7} ... 2^{7}-1	-128 ... 127
2	-2^{15} ... 2^{15}-1	-32768 ... 32767
4	-2^{31} ... 2^{31}-1	-2147483648 ... 2147483647

Рассмотрим особенности записи целых чисел со знаком на примере **однобайтового формата**, при котором для знака отводится один разряд, а для цифр абсолютной величины - семь разрядов.

1	1	1	1	0	0	0	0

В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком: *прямой* код, *обратный* код, *дополнительный* код.

Положительные числа **в прямом, обратном и дополнительном кодах изображаются** **одинаково** **- двоичными кодами с цифрой 0 в знаковом разряде.**

Отрицательные числа **в прямом, обратном и дополнительном кодах имеют** **разное** **изображение.**

1. **Прямой код. В знаковый разряд помещается цифра 1, а в разряды цифровой части числа - двоичный код его абсолютной величины**

2. **Обратный код.** **Получается инвертированием всех цифр двоичного кода абсолютной величины числа, включая разряд знака: нули заменяются единицами, а единицы – нулями.**

3. **Дополнительный код.** **Получается образованием обратного кода с последующим прибавлением единицы к его младшему разряду.**

Формы записи целых положительных чисел

Десятичное представление	Двоичное представление	Представление в прямом коде	Представление в обратном коде	Представление в дополнительном коде
23	**10111**	**00010111**	**00010111**	**00010111**
127	**1111111**	**01111111**	**01111111**	**01111111**
1	**1**	**00000001**	**00000001**	**00000001**

имеют одинаковое представление

Число $23_{10}=10111_2$

прямой, обратный и дополнительный код

0	0	0	1	0	1	1	1
«+»							

Число $127_{10}=1111111_2$

прямой, обратный и дополнительный код

0	1	1	1	1	1	1	1
«+»							

Число $1_{10}=1_2$

прямой, обратный и дополнительный код

0	0	0	0	0	0	0	1
«+»							

Формы записи целых отрицательных чисел

Десятичное представление	Двоичное представление	Представление в прямом коде	Представление в обратном коде	Представление дополнительном коде
-1	**-1**	**10000001**	**11111110**	**11111111**
-17	**-10001**	**10010001**	**11101110**	**11101111**
-127	**-1111111**	**11111111**	**10000000**	**10000001**

Прямой код числа -17:

1	0	0	1	0	0	0	1
«-»							

Прямой код числа -127:

1	1	1	1	1	1	1	1
«-»							

Обратный код числа -17:

1	1	1	0	1	1	1	0
«-»							

Обратный код числа -127:

1	0	0	0	0	0	0	0
«-»							

Дополнительный код числа -17:

1	1	1	0	1	1	1	1
«-»							

Дополнительный код числа -127:

1	0	0	0	0	0	0	1
«-»							

Операции над числами с фиксированной точкой.

1. А и В положительные. При суммировании складываются все разряды, включая разряд знака. Так как знаковые разряды положительных слагаемых равны нулю, разряд знака суммы тоже равен нулю. Например:

Десятичная запись	Двоичные коды
+ 3	+ 0 0000011
7	0 0000111
10	0 0001010

Получен правильный результат.

2. А положительное, В отрицательное и по абсолютной величине больше, чем А.

Например:

Десятичная запись	Двоичные коды	
+ 3	+ 0 0000011	
-10	1 1110101	Обратный код числа -10
-7	1 1111000	Обратный код числа -7

Получен правильный результат в обратном коде. При переводе в прямой код биты ***цифровой части*** результата инвертируются: 1 0000111 = -7_{10}.

3. А положительное, В отрицательное и по абсолютной величине меньше, чем А. Например:

Десятичная запись	Двоичные коды	
+ 10	+ 0 0001010	
-3	1 1111100	Обратный код числа -3
7	0 0000110	
	+1	
	0 0000111	

Компьютер исправляет полученный первоначально неправильный результат (6 вместо 7) **переносом единицы** из знакового разряда в младший разряд суммы!!!

4. **А и В отрицательные.** Например:

Полученный первоначально неправильный результат (обратный код числа -11_{10} вместо обратного кода числа -10_{10}) компьютер исправляет переносом единицы из знакового разряда в младший разряд суммы.

Десятичная запись	Двоичные коды	
-3	1 1111100	Обратный код числа -3
+ -7	+ 1 1111000	Обратный код числа -7
-10	1 1110100	
	+1	
	1 1110101	Обратный код числа -10

При переводе результата в прямой код биты ***цифровой части числа*** инвертируются: 1 0001010 = -10_{10}.

5. A и B положительные, сумма A+B больше, либо равна 2^{n-1}, где n – количество разрядов формата чисел (для однобайтового формата n=8, $2^{n-1} = 2^7 = 128$). Например:

Десятичная запись	Двоичные коды	
65	0 1000001	
+ 97	+ 0 1100001	
162	1 0100010	Переполнение

Семи разрядов цифровой части числового формата **недостаточно** для размещения восьмиразрядной суммы ($162_{10} = 10100010_2$), поэтому **старший разряд суммы оказывается в знаковом разряде.** Это вызывает **несовпадение знака суммы и знаков слагаемых (знак суммы – отрицателен, знак слагаемых – положительный)**, что **является свидетельством переполнения разрядной сетки**.

6. А и В отрицательные, сумма абсолютных величин А и В больше, либо равна 2^{n-1}.

Например:

$63_2 = 0111111_2$

Десятичная запись	Двоичные коды	
+ - 63	+ 1 1000000	Обратный код числа -63
-95	1 0100000	Обратный код числа -95
158	0 1100000	Переполнение
	+1	

Здесь **знак суммы** тоже **не совпадает со знаками слагаемых**, что свидетельствует о **переполнении разрядной сетки**.

1. **А и В положительные**. Здесь нет отличий от случая 1, рассмотренного для обратного кода, т.к. дополнительный код используется только для отрицательных чисел.

Десятичная запись	Двоичные коды
+ 3	+ 0 0000011
7	0 0000111
10	0 0001010

2. А положительное, В отрицательное и по абсолютной величине больше, чем А. Например:

Десятичная запись	Двоичные коды	
3	0 0000011	
+ -10	+ 1 1110110	Дополнительный код числа -10
-7	1 1111001	Дополнительный код числа -7

Получен правильный результат в дополнительном коде.

При переводе в прямой код биты цифровой части результата инвертируются и к младшему разряду прибавляется единица: 1 0000110 + 1 = 1 0000111 = -7_{10}.

3. A положительное, B отрицательное и по абсолютной величине меньше, чем A.

Например:

Десятичная запись	Двоичные коды	
10	0 0001010	
+ -3	+ 1 1111101	Дополнительный код числа -3
7	0 0000111	
	→ перенос отбрасывется	

Получен правильный результат. Единицу переноса из знакового разряда компьютер отбрасывает.

4. А и В отрицательные.

Например:

Десятичная запись	Двоичные коды	
+ -3	+ 1 1111101	Дополнительный код числа -3
-7	1 1111001	Дополнительный код числа -7
-10	1 1110110	Дополнительный код числа -10
	→ перенос отбрасывется	

Получен правильный результат в дополнительном коде. **Единицу переноса** из знакового разряда компьютер **отбрасывает**. **Случаи переполнения** для дополнительных кодов рассматриваются по аналогии со случаями 5 и 6 для обратных кодов.

Задача.

Выполнить действия над машинными кодами чисел:
с фиксированной точкой. Формат 16 двоичных разрядов.

Дано: A=190;   B=250

Найти: C1=A + B;   C2=A – B.

Решение:

A(10) = 190;   A(16)=BE=10111110(2)

B(10) = 250;   B(16)=FA=11111010(2)

C1 = A+B

A= 0 000000010111110
+B= 0 000000011111010

C1= 0 000000110111000

Проверка:

C1=110111000(2)

C1(16) = 1B8 = 1*16*16+11*16+8*1 = 440(10)

C2 = A – B

A = 0 000000010111110 (прямой код)
\- B = 1 111111100000110 (дополнительный код)

C2 = 1 111111111000100

Проверка:

C2 = - 111100 = - BC= - 3*16 +12*1 = = - 60 (10)

Ответ:

C1 = 0 000000110111000

C2 = 1 000000000111100

Задача.

Выполнить действия над машинными кодами чисел: с фиксированной точкой.

Формат 16 двоичных разрядов.

Дано: **A= - 387; B= - 128**

Найти: **C1=A + B;**

Решение:

X = A+B X = (-A) + (- B)

A(10) = - 387; A(16)=- 183(16)= - 110000011(2)

B(10) = - 128; B(16)=- 80(16)= - 10000000(2)

A(2) = 1 000000110000011 –прямой код

A(2) = 1 111111001111100 –обратный код

A(2) = 1 111111001111101 – дополн. код

B(2) = 1 000000010000000 – прямой код

B(2) = 1 111111101111111 – обратный код

B(2) = 1 111111110000000 – дополн.код

(-A) = 1 111111001111101

\+ (-B) = 1 111111110000000

X = 1 111110111111101 –доп. код

X = 1 000001000000010 – обр.код

X = 1 000001000000011 – пр.код

X = - 203(16) = - (2*16*16+0*16+3*1) =

= - (256*2+3) = - (512+3)+ - 515

Представление чисел с плавающей точкой.

Этот способ представления опирается на нормализованную (экспоненциальную) запись действительных чисел.

Нормализованная запись отличного от нуля действительного числа А - это запись вида:

A= m* qn,

где m – мантисса числа (правильная дробь, у которой первая цифра после запятой не равна нулю),

q – основание системы,

n – порядок числа.

Примеры:

1. Мантисса числа 64.5 – это число 0.645, а порядок – число 2, так как 64.5 = 0.645*10 степень (2).

2. Мантисса числа 0.0000012 – это число 0.12, а порядок – число -5, потому что 0.0000012= =0.12*10 степень(-5).

При представлении чисел с плавающей запятой часть разрядов ячейки отводится для записи порядка числа, остальные разряды - для записи мантиссы. По одному разряду в каждой группе отводится для изображения знака порядка и знака мантиссы.

Операции над числами с плавающей точкой.

Дано: A = 12,75; B = 250

Найти: C3 = A + B, C4 = A – B

Формат – 32 двоичных разряда со смещенным порядком.

A(10) = 12,75 = A(16) = C.C;

B(10) = 250 = B(16) = FA

Нормализация мантисс

mA = 0.CC;   pxA = 40 + 1 = 41

mB = 0.FA;   pxB = 40 + 2 = 42

Выравнивание характеристик:

Δp = pxA – pxB = -1

m*A = mA * 16^{-1} = 0.0CC;

pxA = 41+ 1 = 42

C3 = A + B;

mA = 00 0CC000   pxA = 42

mB = 00 FA0000   pxB = 42

mC3 = 01 06C000   pxC = 42

Нормализация мантиссы результата

mxC3 = 00 106C00;

pxC3 = 42 + 1 = 43

Проверка

C3(16) = 106,C = (C3) = 262,75

C3 = 0 1000011000100000110110000000000

C4 = A – B

mA = 00 0CC000 pxA = 42

mB = 10 06000 pxB = 42

mC3 = 10 12C000 pxC = 42

Нормализация мантиссы результата:

mC4 = 10 ED4000 pxC4 = 42

Проверка:

C4 = - ED.4 = (C4) = - (14 * 16 + 13 * 1 + 4/16) = - 237, 25

C4=11000010111011010100000000000000

Получение дополнительного кода числа

Пусть $k = 8$

Надо представить дополнительным кодом $n = -58$

$|-58| = 58 = 111010_2$

0	0	1	1	1	0	1	0

Заменяем единицы на нули и нули на единицы.

1	1	0	0	0	1	0	1

$$\begin{array}{r} + \begin{array}{r} 11000101 \\ 1 \end{array} \\ \hline 11000110 \end{array}$$

1	1	0	0	0	1	1	0

Целые отрицательные числа в компьютере представляются в так называемом *дополнительном коде*. Это позволяет заменить операцию вычитания в компьютере операцией сложения, что в свою очередь позволяет существенно ускорить производимые вычисления.

Для *k*-разрядной ячейки дополнительный код отрицательного числа получается следующим образом:

1. Модуль числа представляется *прямым кодом* в *k* разрядах.
2. В прямом коде все нули заменяются на единицы, а единицы – на нули. Таким образом получаем *обратный код* исходного числа.
3. К полученному обратному коду прибавляется единица.

Таким образом, дополнительный код отрицательного числа *n* в *k*-разрядной ячейке – это дополнение модуля этого числа до 2^k.

Обратите внимание, что 2^k в *k*-разрядной арифметике равно 0. Дело в том, что в *k* разрядах умещается только *k* цифр, а в числе 2^k цифр *k*+1. Например, для 8-ми разрядной ячейки $2^k = 2^8 = 256 = 100000000_2$.

1	0	0	0	0	0	0	0	0

В таком случае говорят, что единица вышла за пределы разрядной сетки.

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Восстановление числа по его дополнительному коду

Пусть $k = 8$

Нужно восстановить исходное число по дополнительному коду:

1	1	0	0	0	1	1	0

2 способ.

Заменяем единицы на нули и нули на единицы.

0	0	1	1	1	0	0	1

$$\begin{array}{r} +\begin{array}{r} 00111001 \\ 1 \end{array} \\ \hline 00111010 \end{array}$$

0	0	1	1	1	0	1	0

$111010_2 = 58 = |-58|$

Искомое число: -58

Иногда требуется по имеющемуся дополнительному коду числа восстановить исходное число. Модуль искомого числа в таком случае можно получить двумя способами:

1. Провести обратную цепочку преобразований: вычесть единицу из дополнительного кода числа, инвертировать полученный результат и перевести его в десятичную систему счисления.

2. Построить дополнительный код для имеющегося дополнительного кода и перевести результат в десятичную систему счисления.


Проверь себя
-1 -1 -1 -1
-2 -1 0 1 2
00111010
00111001
+
1
00111010
00111

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Почему отрицательные числа в компьютере представляются в дополнительном коде?

- Это позволяет заменить операцию вычитания операцией сложения
- Это удобно для программистов
- Это позволяет представить больший диапазон чисел в k разрядах

Сбросить

Подтвердить ответ

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Почему отрицательные числа в компьютере представляются в дополнительном коде?

- ○ Это позволяет представить больший диапазон чисел в k разрядах
- ◉ Это позволяет заменить операцию вычитания операцией сложения
- ○ Это удобно для программистов

Верно

Закрыть

Сбросить

Подтвердить ответ

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Как будет выглядеть дополнительный код для числа -52 в 8-ми разрядной ячейке?

- 0100 1100
- 1100 1110
- 1100 1100
- 0011 0011

Сбросить

Подтвердить ответ

RU

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Как будет выглядеть дополнительный код для числа -52 в 8-ми разрядной ячейке?

- ○ 1100 1110
- ◉ 1100 1100
- ○ 0100 1100
- ○ 0011 0011

Верно

Закрыть

Сбросить

Подтвердить ответ

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Как будет выглядеть дополнительный код для числа -52 в 16-ти разрядной ячейке?

- 1111 1111 1100 1011
- 1111 1111 1100 1100
- 000 0000 0011 0011
- 1000 0000 1100 1100

Сбросить

Подтвердить ответ

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

Выберите правильный ответ

Как будет выглядеть дополнительный код для числа -52 в 16-ти разрядной ячейке?

- (•) 1111 1111 1100 1100
- ( ) 1111 1111 1100 1011
- ( ) 000 0000 0011 0011
- ( ) 1000 0000 1100 1100

Верно

Закрыть

Сбросить

Подтвердить ответ

Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа

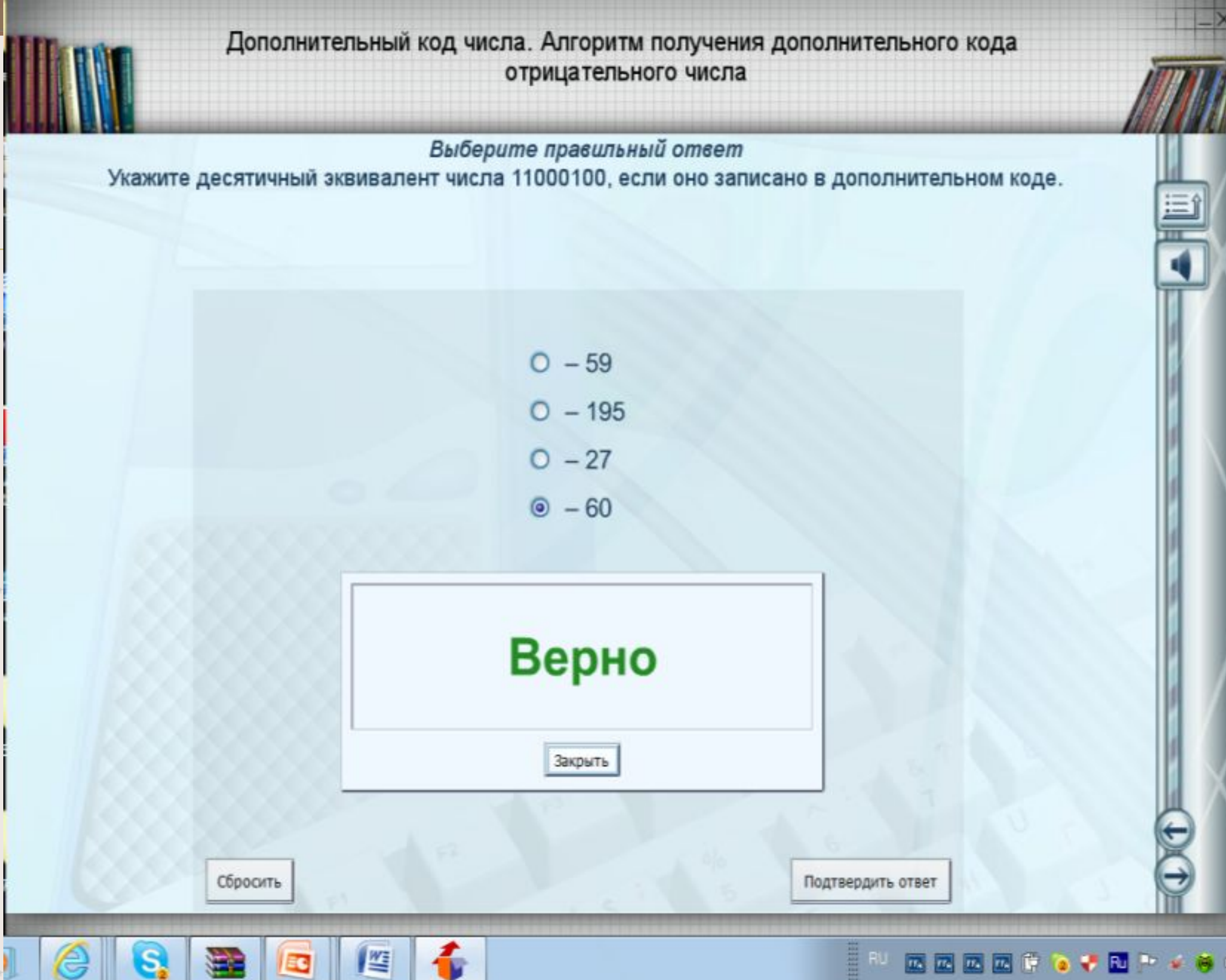
Выберите правильный ответ

Укажите десятичный эквивалент числа 11000100, если оно записано в дополнительном коде.

- – 27
- – 60
- – 195
- – 59

Сбросить

Подтвердить ответ


Дополнительный код числа. Алгоритм получения дополнительного кода отрицательного числа
Выберите правильный ответ
Укажите десятичный эквивалент числа 11000100, если оно записано в дополнительном коде.
– 59
– 195
– 27
– 60
Верно
Закрыть
Сбросить
Подтвердить ответ
RU

Задания на дом:

1. **Угринович Н.Д. п. 2.9., стр.103-105.**
2. **Заполнить карточки.**