

Министерство образования и науки Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)**

К. А. Амеличева, У. В. Никитенко

ЦЕЛЫЕ ЧИСЛА

Методические указания к выполнению домашней работы
по дисциплине «Теоретическая информатика»

Калуга – 2018

Оглавление

ВВЕДЕНИЕ	4
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	5
ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ЦЕЛЫХ ЧИСЕЛ	5
Форматы целых чисел	5
8-битный беззнаковый формат	6
8-битный знаковый формат.....	7
16-, 32- и 64-битные форматы.....	10
Побитовые операции	11
РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАБОТЫ	14
ИНДИВИДУАЛЬНЫЕ ВАРИАНТЫ	30
Задача № 1	30
Задача № 2	31
Задача № 3	32
Задача № 4	33
Задача № 5	34
ТРЕБОВАНИЯ К ОТЧЕТУ	40
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	41

ВВЕДЕНИЕ

Целые числа и вычисления с целыми числами в цифровых вычислительных машинах имеют очень важное значение (в подавляющем количестве приложений занимают меньше ресурсов процессора, чем арифметика с плавающей точкой). Целые числа хранятся в двух возможных видах: беззнаковом (для положительных целых чисел) и со знаком (для отрицательных чисел). В компьютерной технике применяются три формы записи (кодирования) целых чисел со знаком: прямой код, обратный код, дополнительный код. Целые числа являются самыми простыми числовыми данными, с которыми работает компьютер. Вся адресная арифметика и операции с индексами массивов основаны на целочисленных операциях.

В домашнем задании по теоретической информатике рассматриваются способы представления целых чисел в цифровых вычислительных машинах и основы машинной арифметики. Изложение материала сопровождается примерами, позволяющими студенту самостоятельно изучить соответствующий раздел курса.

Цель домашнего задания: формирование у студентов практических навыков выбора подходящих форматов хранения числовых данных, исходя из требований технического задания.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ЦЕЛЫХ ЧИСЕЛ

Форматы целых чисел

По объему памяти, форматы целых чисел обычно используются: 8-битные, 16-битные, 32-битные, 64-битные.

В 8-битном формате под число отводится 1 байт памяти. При помощи 8 двоичных цифр можно представить $2^8 = 256$ различных чисел.

В 16-битном формате под число отводится 2 байта памяти. При помощи 16 двоичных цифр можно представить $2^{16} = 65\,536$ различных чисел.

В 32-битном формате под число отводится 4 байта памяти. При помощи 32 двоичных цифр можно представить $2^{32} = 4\,294\,967\,296$ различных чисел.

В 64-битном формате под число отводится 8 байт памяти. При помощи 64 двоичных цифр можно представить $2^{64} = 18\,446\,744\,073\,709\,551\,616$ различных чисел.

Записывать числа в двоичном виде (особенно длинные числа) не очень удобно, поэтому обычно пользуются шестнадцатеричной записью: 2 цифры для 8-битных форматов, 4 цифры – для 16-битных, 8 цифр – для 32-битных, 16 цифр – для 64-битных.

Также форматы классифицируются по наличию возможности/невозможности представления отрицательных чисел: знаковые (signed) и беззнаковые (unsigned).

Диапазоны чисел, представимых в названных форматах, приведены в таблице 1.

Таблица 1 Форматы целых чисел и их наименования на языке Free Pascal

	Знаковый	Беззнаковый
8-битный	[−128; +127] ShortInt	[0; 255] Byte
16-битный	[−32 768; +32 767] SmallInt, Integer	[0; 65 535] Word
32-битный	[−2 147 483 648; +2 147 483 647] LongInt, Integer	[0; 4 294 967 295] LongWord
64-битный	[−9 223 372 036 854 775 808; +9 223 372 036 854 775 8] Int64	[0; 18 446 744 073 709 551 615] QWord

8-битный беззнаковый формат

8-битный беззнаковый формат представляет неотрицательные целые числа из диапазона [0; 255].

В этом формате числа хранятся в 1 байте памяти в том виде, в котором они представляются в двоичной системе счисления, при необходимости они дополняются до 8 разрядов незначащими нулями слева.

В качестве примера операций рассмотрим сложение беззнаковых чисел.

Сложение беззнаковых чисел выполняется в соответствии с таблицей сложения, представленной в табл. 2.

Таблица 2 Таблица сложения

Операция	Результат	Перенос
$0 + 0$	0	0
$0 + 1$	1	0
$1 + 0$	1	0
$1 + 1$	0	1
$1 + 1 + 1$	1	1

При выполнении сложения может возникнуть ситуация, когда результат операции непредставим в выбранном формате. Эта ситуация называется *переполнением* (*overflow*). Например, при сложении $255 + 1$ должно было бы получаться 256, но это значение непредставимо в 8-битном беззнаковом формате. В таком случае процессор обычно порождает неверный результат (в приведенном примере: 0) и устанавливает флаг переполнения.

Признаком переполнения при выполнении операций над беззнаковыми числами является перенос из старшего разряда числа.

8-битный знаковый формат

8-битный знаковый формат представляет целые числа из диапазона $[-128; +127]$.

Старший (самый левый) бит числа называется *знаковым* и обозначает знак числа: 0 – число неотрицательное, 1 – число отрицательное.

Неотрицательные числа хранятся в *прямом коде* (sign-magnitude): «0» в знаковом бите, в остальных битах – цифры представления числа в двоичной системе счисления; при необходимости число дополняется незначащими нулями слева.

Отрицательные числа хранятся в *дополнительном коде* (two's complement): «1» в знаковом бите, в остальных битах – цифры дополнительного кода.

Замечание. Вообще говоря, для представления отрицательных чисел существует несколько способов: прямой код (sign-magnitude), обратный код (ones' complement), дополнительный код (two's complement), «смещенное» представление (excess-K, offset binary). Здесь мы рассматриваем только дополнительный код, т. к. в настоящее время чаще всего используется именно он.

Алгоритм 1 построения дополнительного кода числа

1. Записать двоичное представление модуля числа, дополнив до используемой в формате разрядности незначащими нулями слева (в случае 8-битных форматов – до 8 разрядов, в случае 16-битных – до 16 разрядов и т. д.).

2. Инвертировать все биты полученного числа (т. е. заменить значение каждого бита на противоположное: «0» на «1», а «1» на «0»).

Замечание. Полученное значение является обратным кодом числа.

3. Сложить результат с числом «1».

Алгоритм 2 построения дополнительного кода числа

1. Записать двоичное представление модуля числа, дополнив до используемой в формате разрядности незначащими нулями слева (в случае 8-битных форматов – до 8 разрядов, в случае 16-битных – до 16 разрядов и т. д.).

2. Начиная с самого младшего бита переписывать в результат все «0» до тех пор, пока не встретится «1».

3. Переписать в результат встретившуюся «1».

4. Инвертировать все оставшиеся биты числа.

Получение числа из его дополнительного кода осуществляется по тому же алгоритму, что и само построение дополнительного кода: инвертировать и прибавить 1 ([алгоритм 1](#)) или скопировать нули и первую единицу и инвертировать остальные биты ([алгоритм 2](#)). При этом будет получен модуль числа; достаточно поставить знак «минус» и получится искомое число.

Алгоритм 3 представления числа в знаковом формате

1. Если число положительное, то оно представляется в прямом коде: дополняется нулями до требуемой разрядности, в знаковом бите должно быть значение «0».

2. Если число отрицательное, то оно представляется в дополнительном коде: см. [алгоритм 1](#) или [2](#); в знаковом бите должно получиться значение «1».

Алгоритм 4 определения числа по его представлению

1. Если в знаковом бите «0», то число – неотрицательное, представленное в прямом коде: необходимо просто выполнить перевод из двоичной системы счисления.

2. Если в знаковом бите «1», то число – отрицательное, представленное в дополнительном коде: необходимо выполнить перевод из дополнительного кода ([алгоритм 1](#) или [2](#)) и поставить знак «минус».

Сложение знаковых чисел осуществляется по обычным правилам, при этом также может происходить переполнение ($127 + 2 = 129$, но это значение непредставимо в знаковом 8-битном формате).

Признак переполнения при выполнении операций над знаковыми числами: наличие или переноса в знаковый разряд, или переноса из знакового разряда числа (но не двух сразу – если произошли оба переноса, то переполнения нет, получен верный результат).

16-, 32- и 64-битные форматы

«Многобайтные» беззнаковые и знаковые форматы по формату хранения и правилам обработки сходны с 8-битными форматами.

В этих форматах значение представлено последовательностью байт (2, 4 и 8 байт), которая в памяти может быть записана по-разному.

Распространены следующие варианты записи последовательности байт (*порядок байт*, Endianness):

- от старшего к младшему (**Big-endian, BE**) – первым следует самый старший байт, последним – самый младший;
- от младшего к старшему (**Little-endian, LE**) – первым следует самый младший байт, последним – самый старший.

Побитовые операции

Побитовая (bitwise) операция применяется к каждому биту своих операндов.

Побитовое «НЕ (NOT)» (отрицание, инвертирование) – унарная операция, результат которой определяется таблицей 3. Будем обозначать побитовое «НЕ» над значением X следующим образом: **not** X .

Фактически, данная операция заменяет бит на «противоположный».

Таблица 3 Операция побитового «НЕ»: not X

X	not X
0	1
1	0

Побитовое «И (AND)» – бинарная операция, результат которой определяется таблицей 4. Будем обозначать побитовое «И» над значениями X и Y следующим образом: $X \text{ and } Y$.

Фактически, результатом данной операции является «1», только если оба ее операнда равны «1». Из таблицы значений следуют следующие свойства операции (здесь X – некоторое значение бита):

$$X \text{ and } 1 = X; \quad (1)$$

$$X \text{ and } 0 = 0.$$

Побитовое «ИЛИ (OR)» – бинарная операция, результат которой определяется таблицей 4. Будем обозначать побитовое «ИЛИ» над значениями X и Y следующим образом: $X \text{ or } Y$.

Таблица 4 Бинарные побитовые операция

X	Y	$X \text{ and } Y$	$X \text{ or } Y$	$X \text{ xor } Y$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Фактически, результатом данной операции является «1», если хотя бы один из ее операндов равен «1». Из таблицы значений следуют следующие свойства операции (здесь X – некоторое значение бита):

$$X \text{ or } 1 = 1; \quad (2)$$

$$X \text{ or } 0 = X.$$

Побитовое «Исключающее ИЛИ (eXclusive OR)» – бинарная операция, результат которой определяется таблицей 4. Будем обозначать побитовое «Исключающее ИЛИ» над значениями X и Y следующим образом: $X \text{ xor } Y$.

Фактически, результатом данной операции является «1», если ровно один из ее операндов равен «1». Из таблицы значений следуют следующие свойства операции (здесь X – некоторое значение бита):

$$\begin{aligned} X \text{ xor } 1 &= \text{not } X; \\ X \text{ xor } 0 &= X; \\ X \text{ xor } X &= 0. \end{aligned} \tag{3}$$

Самой приоритетной операцией будем считать «НЕ», следующей по приоритету – «И», самыми низкоприоритетными – «ИЛИ» и «Исключающее ИЛИ».

Сдвиг вправо (right shift) – бинарная операция, осуществляющая смещение всех битов заданного значения (8-, 16-, 32- или 64-разрядного) вправо на заданное число позиций. При этом самые правые (младшие) биты числа, уходящие за пределы разрядной сетки, отбрасываются. *Логический* (logical) или *беззнаковый* (unsigned) сдвиг вправо освободившиеся биты слева заполняет значением «0». *Арифметический* (arithmetic) или *знаковый* (signed) сдвиг вправо освободившиеся биты слева заполняет значением знакового бита.

Арифметический сдвиг имеет смысл только для чисел со знаком.

Логический сдвиг вправо числа X на N позиций будем обозначать: $X \text{ shr } N$.

Арифметический сдвиг вправо числа X на N позиций будем обозначать: $X \text{ ash } N$.

Сдвиг влево (left shift) – бинарная операция, осуществляющая смещение всех битов заданного значения (8-, 16-, 32- или 64-разрядного) влево на заданное число позиций. При этом самые левые (старшие) биты числа, уходящие за пределы разрядной сетки, отбрасываются, а освободившиеся биты справа заполняются значением «0».

Сдвиг влево числа X на N позиций будем обозначать: $X \text{ shl } N$.

Фактически, сдвиг числа X влево на N бит представляет собой «быстрое» умножение X на 2^N . Сдвиг X вправо на N бит представляет собой «быстрое» деление X на 2^N с округлением к меньшему.

РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАБОТЫ

Задача 1. Покажите, как в 8-битном беззнаковом формате представляется число 61. Ответ приведите в двоичном и шестнадцатеричном виде.

Решение

Число 61 лежит в диапазоне чисел, представимых в 8-битном беззнаковом формате: $[0; 255]$.

$$\begin{array}{r}
 61 \mid 2 \\
 \hline
 60 \mid 30 \quad 2 \\
 \quad 1 \mid 30 \quad 15 \quad 2 \\
 \quad \quad 0 \mid 14 \quad 7 \quad 2 \\
 \quad \quad \quad 1 \mid 6 \quad 3 \quad 2 \\
 \quad \quad \quad \quad 1 \mid 2 \quad 1 \\
 \quad \quad \quad \quad \quad 1
 \end{array}$$

В беззнаковом формате числа хранятся в том виде, в котором они представляются в двоичной системе счисления, только в виде полных 8 разрядов (8 бит). В полученном двоичном числе 6 разрядов, поэтому дополним число до 8 разрядов двумя незначащими нулями слева: 00111101_2

$$00111101_2 = 3D_{16}.$$

Задача 2. Покажите, как указанные десятичные числа представляются в 8-битном формате без знака: выполните перевод в двоичную систему счисления, дополните до требуемого количества разрядов. Полученные двоичные представления чисел запишите в шестнадцатеричном виде. Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение.

15

В ответе необходимо выписать полученное число в десятичной системе счисления и указать, правильный результат получен или нет.

а) 94, 45;

б) 204, 244.

Решение задания «а»

Числа 94 и 45 лежат в диапазоне чисел, представимых в 8-битном беззнаковом формате: $[0; 255]$.

1. Выполним перевод чисел в двоичную систему счисления (здесь сам перевод опущен)

$$94_{10} = 1011110_2;$$

$$45_{10} = 101101_2.$$

2. Покажем, как числа представляются в 8-битном беззнаковом формате. Для этого просто дополним двоичные представления до 8 цифр. Также запишем шестнадцатеричные представления.

$$94_{10} = 01011110_2 = 5E_{16};$$

$$45_{10} = 00101101_2 = 2D_{16}.$$

3. Выполним сложение чисел

	*	*	*	*	*			
	0	1	0	1	1	1	1	0
+	0	0	1	0	1	1	0	1
	<hr style="width: 100%; border: 0.5px solid black;"/>							
	1	0	0	0	1	0	1	1

Переноса из разрядной сетки не было, а значит и не было переполнения. В результате операции получен верный результат.

4. Представим полученный результат в шестнадцатеричном и десятичном виде. Т. к. в 8-битном беззнаковом формате представляются только положительные числа, выполняем обычный перевод (здесь перевод в десятичную систему опущен, приведен только результат).

$$10001011_2 = 8B_{16} = 139_{10}.$$

Решение задания «б»

Числа 204 и 244 лежат в диапазоне чисел, представимых в 8-битном беззнаковом формате: $[0; 255]$.

1. Выполним перевод чисел в двоичную систему счисления (здесь сам перевод опущен)

$$204_{10} = 11001100_2;$$

$$244_{10} = 11110100_2.$$

2. Покажем, как числа представляются в 8-битном беззнаковом формате. Для этого просто дополним двоичные представления до 8 цифр (у нас уже по 8 цифр). Также запишем шестнадцатеричные представления.

$$204_{10} = 11001100_2 = CC_{16};$$

$$244_{10} = 11110100_2 = F4_{16}.$$

3. Выполним сложение чисел

	(*)	*	*	*	*	*			
		1	1	0	0	1	1	0	0
+		1	1	1	1	0	1	0	0
		1	1	1	0	0	0	0	0

Был перенос из разрядной сетки, а значит произошло переполнения. «Лишний» 9-й разряд будет отброшен, т.к. он

не «помещается» в требуемый формат. В результате операции получен неверный результат 11000000_2 .

4. Представим полученный результат в шестнадцатеричном и десятичном виде. Т.к. в 8-битном беззнаковом формате представляются только положительные числа, выполняем обычный перевод (здесь перевод в десятичную систему опущен, приведен только результат).

$$11000000_2 = C0_{16} = 192_{10}.$$

Ответ: а) 139_{10} , результат верный; б) 192_{10} , результат неверный.

Задача 3. Покажите, как в 8-битном знаковом формате представляются числа $+61$ и -61 . Ответ приведите в двоичном и шестнадцатеричном виде.

Решение

Числа $+61$ и -61 принадлежат диапазону чисел, представимых в 8-битном знаковом формате: $[-128; 127]$.

Число $+61$ – неотрицательное, а значит представляется в прямом коде.

Переведем число в двоичную систему счисления (см. [задачу 1](#)): $+61_{10} = +111101_2$.

Дополняя до 8 разрядов незначащими нулями слева, получаем представление числа $+61$: 00111101_2 , или в шестнадцатеричном виде: $3D_{16}$.

Контроль: в знаковом бите получено значение «0», указывающее на неотрицательное число, каким +61 и является. Число -61 – отрицательное, а значит представляется в дополнительном коде.

Переведем число в двоичную систему счисления (см. [задачу 1](#)): $-61_{10} = -111101_2$.

Запишем модуль числа в двоичном виде, дополнив до 8 разрядов незначащими нулями: 00111101₂.

Выполним перевод в дополнительный код по [алгоритму 1](#).

1. Инвертируем все биты числа, получаем: 11000010₂.
2. Прибавляем единицу:

$$\begin{array}{r}
 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \\
 + 1 \\
 \hline
 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1
 \end{array}$$

Получаем: 11000011₂. Это и есть искомый дополнительный код.

Выполним перевод в дополнительный код по [алгоритму 2](#).

1. Переписываем все нули, начиная с самого правого, пока не встретится единица. В данном примере таких нулей нет: *****₂.
2. Переписываем первую единицу справа: *****1₂.
3. Инвертируем все оставшиеся биты числа: 11000011₂.

Полученное число – искомый дополнительный код.

Итак, дополнительный код: 11000011₂, или в шестнадцатеричном виде: C3₁₆.

Контроль: в знаковом бите получено значение «1», указывающее на отрицательное число, каким -61 и является.

Замечание: в решении данной задачи задействованы оба алгоритма получения дополнительного кода только лишь с целью проиллюстрировать их; при решении задач достаточно использовать один из них (любой).

Ответ: $00111101_2 = 3D_{16}$; $11000011_2 = C3_{16}$.

Задача 4 Заданы представления чисел в 8-битном знаковом формате. Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение. Правильный ли результат получен?

а) 44_{16} , $B8_{16}$; б) BA_{16} , $E7_{16}$; в) DF_{16} , 93_{16} ; г) 64_{16} , 54_{16} .

Решение задания «а»

Запишем двоичные представления чисел:

$$44_{16} = 01000100_2;$$

$$B8_{16} = 10111000_2.$$

Выполним сложение чисел:

	0	1	0	0	0	1	0	0
+	1	0	1	1	1	0	0	0
	1	1	1	1	1	1	0	0

Переноса из разрядной сетки не было, переноса в знаковый (старший) разряд не было. Значит, переполнения не было. В результате операции получен верный результат $11111100_2 = FC_{16}$ (*замечание:* получено отрицательное число в дополнительном коде).

Решение задания «б»

Запишем двоичные представления чисел:

$$BA_{16} = 10111010_2;$$

$$E7_{16} = 11100111_2.$$

Выполним сложение чисел:

	(*)	(*)	*	*	*	*	*		
		1	0	1	1	1	0	1	0
+		1	1	1	0	0	1	1	1
	1	1	0	1	0	0	0	0	1

Был и перенос из разрядной сетки, и перенос в знаковый (старший) разряд. Значит, переполнения не было. В результате операции получен верный результат $10100001_2 = A1_{16}$ («лишняя» цифра отбрасывается).

Решение задания «в»

Запишем двоичные представления чисел:

$$DF_{16} = 11011111_2;$$

$$93_{16} = 10010011_2.$$

Выполним сложение чисел:

	(*)			*	*	*	*	*
		1	1	0	1	1	1	1
+		1	0	0	1	0	0	1
	1	0	1	1	1	0	0	1

Был перенос из разрядной сетки, а переноса в знаковый (старший) разряд не было. Значит, произошло переполнение. В результате операции получен **неверный** результат $01110010_2 = 72_{16}$ («лишняя» цифра отбрасывается).

Решение задания «г»

Запишем двоичные представления чисел:

$$64_{16} = 01100100_2;$$

$$54_{16} = 01010100_2.$$

Выполним сложение чисел:

			(*)						*				
			0	1	1	0	0	0	1	0	0		
+			0	1	0	1	0	0	1	0	0		
			1	0	1	1	1	1	0	0	0		

Переноса из разрядной сетки не было, а перенос в знаковый (старший) разряд был. Значит, произошло переполнение. В результате операции получен **неверный** результат $10111000_2 = B8_{16}$.

Ответ: а) FC_{16} , верно; б) $A1_{16}$, верно; в) 72_{16} , неверно; г) $B8_{16}$, неверно.

Задача 5. Покажите, как числа $2\ 040\ 325\ 676 = 79\ 9C\ E6\ 2C_{16}$ и $8\ 297 = 20\ 69_{16}$ представляются в 32-битном формате без знака. Размещение байт в памяти проиллюстрируйте как для случая использования [Big-endian](#), так и для случая [Little-endian](#).

Решение

Представим заданные числа в 32-битном формате без знака. Приведенные в задании шестнадцатеричные числа дополним нулями до 8 цифр (32 бита = 4 байта, по 2 цифры на байт):

$$2\ 040\ 325\ 676 = 79\ 9C\ E6\ 2C_{16};$$

$$8\ 297 = 00\ 00\ 20\ 69_{16}.$$

В случае использования порядка байт [Big-endian](#), старшие байты хранятся первыми, младшие – последними. Заданное число будет размещаться в памяти следующим образом (одна клетка обозначает 1 байт):

2 040 325 676		79	9C	E6	2C
8 297		00	00	20	69

В случае использования порядка байт [Little-endian](#), младшие байты числа хранятся первыми, а старшие – последними. Заданное число будет размещаться в памяти следующим образом:

2 040 325 676		2C	E6	9C	79
8 297		69	20	00	00

Ответ: 2 040 325 676: 79 9C E6 2C (BE), 2C E6 9C 79 (LE);

8 297: 00 00 20 69 (BE), 69 20 00 00 (LE).

Задача 6. Задано шестнадцатеричное представление двух чисел в памяти в 16-битном беззнаковом формате. Определите, какие это числа (в десятичной системе счисления). Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение. Покажите, как сумма (результат сложения) представляется в 16-битном беззнаковом формате в шестнадцатеричном виде. Представьте сумму в десятичной системе счисления A1 89 и 27 EE, порядок байт: [Little-endian](#);

Решение

1. Запишем заданные числа в шестнадцатеричной системе счисления, при этом учтем, что задано их представление с использованием порядка байт Little-endian.

Первое число: $89A1_{16}$;

Второе число: $EE27_{16}$.

2. Переведем полученные числа в десятичную систему счисления:

$$89A1_{16} = 8 \cdot 16^3 + 9 \cdot 16^2 + 10 \cdot 16 + 1 = 35\,233;$$

$$EE27_{16} = 14 \cdot 16^3 + 14 \cdot 16^2 + 2 \cdot 16 + 7 = 60\,967.$$

3. Запишем числа в двоичном виде:

$$89A1_{16} = 1000100110100001_2;$$

$$EE27_{16} = 1110111000100111_2.$$

Выполним сложение чисел:

(*)				*					*				*	*	*	
	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0	1
+	1	1	1	0	1	1	1	0	0	0	1	0	0	1	1	1
<hr/>																
	1	0	1	1	1	0	1	1	1	1	0	0	1	0	0	0

Был перенос из разрядной сетки, а значит произошло переполнения. «Лишний» 17-й разряд будет отброшен, т.к. он не «помещается» в требуемый формат. В результате операции получен неверный результат $0111011111001000_2 = 77C8_{16}$.

4. Представим сумму в десятичной системе счисления.

$$77C8_{16} = 7 \cdot 16^3 + 7 \cdot 16^2 + 12 \cdot 16 + 8 = 30\,664.$$

5. Т. к. порядок байт [Little-endian](#), байты в сумме следуют в «обратном» порядке: C8 77.

Ответ: 35 233, 60 967, сумма: 30 664, в памяти: C8 77 (LE), было переполнение.

Задача 7. Вычислите значение выражения: $\text{not}(43_{16})$ and $(81_{16}$ or $12_{16})$. Операции выполняются над 8-битными знаковыми числами. Результат необходимо записать в шестнадцатеричном виде.

Решение

Запишем все числа в двоичном виде:

$$43_{16} = 01000011_2;$$

$$81_{16} = 10000001_2;$$

$$12_{16} = 00010010_2.$$

Будем выполнять операции в порядке приоритета. Вычислим $A = \text{not}(43_{16})$.

<i>not</i>	0	1	0	0	0	0	1	1
	1	0	1	1	1	1	0	0

Вычислим $B = 81_{16}$ [or](#) 12_{16} .

	1	0	0	0	0	0	0	1
<i>or</i>	0	0	0	1	0	0	1	0
	1	0	0	1	0	0	1	1

Вычислим A [and](#) B .

	1	0	1	1	1	1	0	0
<i>and</i>	1	0	0	1	0	0	1	1
	1	0	0	1	0	0	0	0

Запишем результат в шестнадцатеричном виде:

$$10010000_2 = 90_{16}.$$

Ответ: 90_{16} .

Задача 8. Определите результат сдвига значения $E5_{16}$ на 2 позиции, если число хранится в знаковом 8-битном формате, и применяется операция:

- а) сдвига влево;
- б) логического сдвига вправо;
- в) арифметического сдвига вправо.

Ответ дайте в шестнадцатеричном виде.

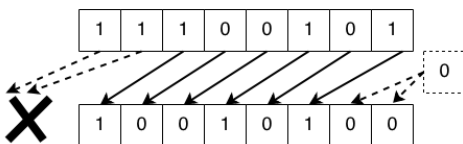
Решение задания «а»

Запишем заданное число в двоичном формате:

$$E5_{16} = 11100101_2.$$

Замечание: это представление отрицательного числа, путем перевода из дополнительного кода можно получить само число: -27 .

При сдвиге влево все биты смещаются на 2 позиции влево, 2 самых левых (старших) бита (11) теряются, а справа «освободившиеся» 2 младших бита заполняются нулями:



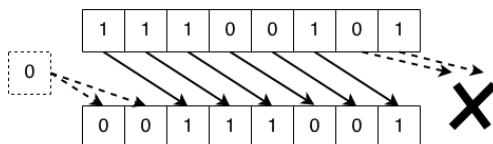
$$11100101_2 \text{ shl } 2 = 10010100_2 = 94_{16}.$$

Замечание 1: получено представление отрицательного числа, путем перевода из дополнительного кода можно получить само число: -108 . Видим, что сдвиг влево на 2 позиции оказался эквивалентен умножению на $2^2 = 4$.

Замечание 2: для сдвига влево также выделяют операции арифметического и логического сдвига, отличающиеся тем, что при арифметическом сдвиге осуществляется проверка на переполнение, а при логическом – нет. Так, в рассматриваемом примере переполнение не происходит, т. к. все отброшенные биты и знаковый бит после сдвига совпадают со знаковым битом до сдвига.

Решение задания «б»

При логическом сдвиге вправо все биты смещаются на 2 позиции вправо, 2 самых правых (младших) бита (01) теряются, а слева «освободившиеся» 2 старших бита заполняются нулями:

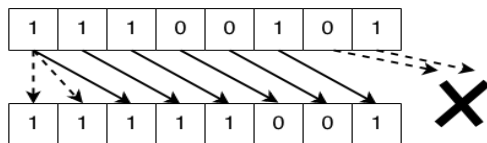


$$11100101_2 \text{ shr } 2 = 00111001_2 = 39_{16}.$$

Замечание: получено представление положительного числа, путем перевода из прямого кода можно получить само число: 57.

Решение задания «в»

При арифметическом сдвиге вправо все биты смещаются на 2 позиции вправо, 2 самых правых (младших) бита (01) теряются, а слева «освободившиеся» 2 старших бита заполняются исходным знаковым битом:



$$11100101_2 \text{ ashr } 2 = 1111001_2 = F9_{16}.$$

Замечание: получено представление отрицательного числа, путем перевода из дополнительного кода можно получить само число: -7 . Видим, что сдвиг влево на 2 позиции оказался эквивалентен делению на $2^2 = 4$ с округлением к меньшему целому:

$$-27 : 4 = -6,75 \approx -7.$$

Ответ: а) 94_{16} ; б) 39_{16} ; в) $F9_{16}$.

Задача 9. Пусть в 8-битной ячейке памяти X хранится некоторое значение. Какие операции необходимо произвести, чтобы 3-й и 4-й биты числа сохранили свое значение, а остальные биты были обнулены? (Биты в байте обычно нумеруют от младшего к старшему, младший бит считают битом 0, старший – битом 7).

Решение

Операция побитового «И» обладает свойствами, позволяющими реализовать требуемое преобразование – см. свойства (1): она позволяет сохранить значение для одних битов (нужно выполнить «and 1») и обнулить другие биты (нужно выполнить «and 0»). Т. о. решением будет операция вида:

$$X := X \text{ and } M$$

Здесь X – заданная ячейка памяти (переменная), а M – некоторая константа.

Построим значение константы M . В ее 3-й и 4-й биты необходимо поместить значение «1», тогда после операции «И» соответствующие биты X сохранят свое значение. В ее остальные биты необходимо поместить значение «0», тогда после операции «И» соответствующие биты X будут сброшены в «0».

Получаем: $M = 00011000_2 = 18_{16}$.

Рассмотрим примеры.

Пусть $X = 1100\mathbf{1}110_2$. Тогда $X \text{ and } 00011000_2 = 0000\mathbf{1}000_2$.

Пусть $X = 011\mathbf{1}1011_2$. Тогда $X \text{ and } 00011000_2 = 000\mathbf{1}1000_2$.

Значение M , определяющее, какие биты нас интересуют, называют *маской* (mask), а само рассмотренное преобразование, позволяющее выделить интересующие биты – *наложением маски*.

Ответ: $X := X \text{ and } 18_{16}$.

ИНДИВИДУАЛЬНЫЕ ВАРИАНТЫ

Задача № 1

Покажите, как указанные десятичные числа представляются в 8-битном формате со знаком. Ответ приведите в двоичном и в шестнадцатеричном виде (в решении должен быть представлен полный перевод из десятичной системы счисления).

Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение. Полученную сумму представьте в том же формате, что и слагаемые.

Найдите десятичное число, которое представляет полученная сумма.

Вариант	Задание
1.	а) -66 и -25 ; б) -114 и -63 .
2.	а) -30 и 78 ; б) 65 и 87 .
3.	а) 112 и -51 ; б) -121 и -79 .
4.	а) -126 и 106 ; б) -65 и -110 .
5.	а) -46 и 119 ; б) -127 и -49 .
6.	а) 114 и -26 ; б) -100 и -124 .
7.	а) -125 и 24 ; б) -114 и -69 .
8.	а) 99 и -21 ; б) 104 и 39 .
9.	а) -63 и 21 ; б) 80 и 73 .
10.	а) 64 и -104 ; б) 125 и 56 .

Задача № 2

Покажите, как указанные десятичные числа представляются в 16-битном формате со знаком, порядок байт: Big-endian (в скобках приведены результаты перевода чисел в шестнадцатеричную систему счисления). Ответ приведите в двоичном и в шестнадцатеричном виде.

Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение. Полученную сумму представьте в том же формате, что и слагаемые.

Найдите десятичное число, которое представляет полученная сумма.

Вариант	Задание
1.	а) 14 057 (36 E9_{16}) и 32 138 ($7\text{D }8\text{A}_{16}$); б) 2 698 ($\text{A }8\text{A}_{16}$) и $-27\,999$ ($-6\text{D }5\text{F}_{16}$).
2.	а) $-25\,028$ (-61 C4_{16}) и $-23\,695$ ($-5\text{C }8\text{F}_{16}$); б) 13 921 (36 61_{16}) и $-30\,643$ (-77 B3_{16}).
3.	а) 32 641 ($7\text{F }81_{16}$) и 23 434 ($5\text{B }8\text{A}_{16}$); б) 6 376 (18 E8_{16}) и $-22\,923$ (-59 8B_{16}).
4.	а) 19 279 ($4\text{B }4\text{F}_{16}$) и 24 887 (61 37_{16}); б) 29 742 (74 2E_{16}) и $-14\,611$ (-39 13_{16}).
5.	а) 25 544 (63 C8_{16}) и 23 965 ($5\text{D }9\text{D}_{16}$); б) $-28\,276$ ($-6\text{E }74_{16}$) и 6 862 (1A CE_{16}).
6.	а) $-17\,708$ (-45 2C_{16}) и $-28\,982$ (-71 36_{16}); б) $-17\,083$ (-42 BB_{16}) и 25 111 (62 17_{16}).

7.	а) 15 741 (3D 7D ₁₆) и 20 558 (50 4E ₁₆); б) -22 409 (-57 89 ₁₆) и 18 061 (46 8D ₁₆).
8.	а) 28 930 (71 02 ₁₆) и 8 912 (22 D0 ₁₆); б) 25 191 (62 67 ₁₆) и -2 374 (-9 46 ₁₆).
9.	а) -32 474 (-7E DA ₁₆) и -23 060 (-5A 14 ₁₆); б) -6 858 (-1A CA ₁₆) и 5 464 (15 58 ₁₆).
10.	а) -25 229 (-62 8D ₁₆) и -17 009 (-42 71 ₁₆); б) -28 805 (-70 85 ₁₆) и 30 897 (78 B1 ₁₆).

Задача № 3

Покажите, как указанные десятичные числа представляются в 16-битном формате без знака, порядок байт: Little-endian (в скобках приведены результаты перевода чисел в шестнадцатеричную систему счисления). Ответ приведите в двоичном и в шестнадцатеричном виде.

Выполните сложение двоичных представлений чисел и укажите, произошло ли при этом переполнение. Полученную сумму представьте в том же формате, что и слагаемые.

Найдите десятичное число, которое представляет полученная сумма.

Вариант	Задание
1.	а) 36 852 (8F F4 ₁₆) и 43 509 (A9 F5 ₁₆); б) 45 844 (B3 14 ₁₆) и 5 084 (13 DC ₁₆).

2.	а) 52 859 (CE 7B ₁₆) и 38 022 (94 86 ₁₆); б) 22 831 (59 2F ₁₆) и 39 665 (9A F1 ₁₆).
3.	а) 53 648 (D1 90 ₁₆) и 49 621 (C1 D5 ₁₆); б) 20 817 (51 51 ₁₆) и 2 961 (B 91 ₁₆).
4.	а) 49 761 (C2 61 ₁₆) и 39 469 (9A 2D ₁₆); б) 58 027 (E2 AB ₁₆) и 7 087 (1B AF ₁₆).
5.	а) 59 244 (E7 6C ₁₆) и 60 965 (EE 25 ₁₆); б) 44 652 (AE 6C ₁₆) и 15 105 (3B 01 ₁₆).
6.	а) 37 085 (90 DD ₁₆) и 33 884 (84 5C ₁₆); б) 8 119 (1F B7 ₁₆) и 17 911 (45 F7 ₁₆).
7.	а) 59 968 (EA 40 ₁₆) и 63 018 (F6 2A ₁₆); б) 34 794 (87 EA ₁₆) и 1 109 (4 55 ₁₆).
8.	а) 58 531 (E4 A3 ₁₆) и 57 170 (DF 52 ₁₆); б) 16 103 (3E E7 ₁₆) и 4 601 (11 F9 ₁₆).
9.	а) 36 852 (8F F4 ₁₆) и 43 509 (A9 F5 ₁₆); б) 45 844 (B3 14 ₁₆) и 5 084 (13 DC ₁₆).
10.	а) 52 859 (CE 7B ₁₆) и 38 022 (94 86 ₁₆); б) 22 831 (59 2F ₁₆) и 39 665 (9A F1 ₁₆).

Задача № 4

Вычислите значение следующего выражения, считая, что все числовые значения представляют собой 8-битные беззнаковые целые, записанные в шестнадцатеричном виде: (см. таблицу варианта)

В решении необходимо привести исходные числовые значения и результат выполнения каждой операции в двоичном виде. Ответ необходимо указать в шестнадцатеричном виде.

Вариант	Задание
1.	$((6F \text{ shr } 2) \text{ and } 3C) \text{ xor not}(43)$
2.	$((8F \text{ shr } 3) \text{ and } 3C) \text{ xor not}(60)$
3.	$(\text{not}((5F \text{ shr } 3)) \text{ and } AC) \text{ xor } 37$
4.	$(\text{not}((C8 \text{ shr } 2)) \text{ and } 84) \text{ xor } 37$
5.	$((\text{not}(44) \text{ or } 3A) \text{ shl } 2) \text{ xor } 3E$
6.	$((\text{not}(41) \text{ or } 48) \text{ shr } 2) \text{ xor } 3E$
7.	$((B7 \text{ shr } 2) \text{ and } 3C) \text{ xor not}(F2)$
8.	$((ED \text{ xor not}(8D)) \text{ and } 7C) \text{ shl } 3$
9.	$((5B \text{ xor not}(74)) \text{ and } 7C) \text{ shl } 3$
10.	$(\text{not}((2C \text{ shr } 3)) \text{ and } C6) \text{ xor } 37$

Задача № 5

Вариант 1

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
		X	X			Y	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
0	Y	X	X	1	1	0	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 2

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
X	X		Y				

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
X	X	1	0	1	0	Y	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 3

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
		X	X		Y		

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
1	1	X	X	0	1	0	Y

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 4

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
	X	X		Y			

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
0	X	X	0	1	0	Y	0

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 5

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
Y					X	X	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
0	0	1	0	Y	X	X	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 6

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
	X	X				Y	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
0	X	X	Y	0	1	0	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 7

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
	Y			X	X		

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
0	0	1	0	X	X	0	Y

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 8

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
		Y			X	X	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
Y	1	1	0	0	X	X	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 9

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
X	X					Y	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
X	X	0	1	1	0	0	Y

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R .

Вариант 10

Пусть в ячейке памяти Q размещается 8-битное беззнаковое целое. Отметим некоторые биты ячейки Q символами X и Y :

7	6	5	4	3	2	1	0
			Y		X	X	

Необходимо в ячейке памяти R сформировать 8-битное беззнаковое целое следующего вида (символами X и Y обозначены значения соответствующих битов ячейки Q):

7	6	5	4	3	2	1	0
1	1	0	1	Y	X	X	1

Напишите последовательность побитовых операций, позволяющую на основе значения Q получить значение R

ТРЕБОВАНИЯ К ОТЧЕТУ

На выполнение домашнего задания отводится 10 часов самостоятельной работы. Номер варианта студенту выдается преподавателем. Отчет на защиту предоставляется в рукописном виде.

Структура отчета:

Титульный лист.

Формулировка задания (вариант).

Результат выполнения задания:

- представление указанных чисел в форматах,
- сложение чисел, с описанием полученного результата,
- последовательность побитовых операций;

Анализ полученных результатов.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Задохина Н.В. Математика и информатика. Решение логико-познавательных задач [Электронный ресурс]: учебное пособие для студентов вузов / Н.В. Задохина.— М.: ЮНИТИ-ДАНА, 2015. <http://www.iprbookshop.ru/34474.html>
2. Начальный курс информатики. Часть 1 [Электронный ресурс]: учебное пособие/ В.А. Лопушанский [и др.]. — Воронеж: Воронежский государственный университет инженерных технологий, 2013. <http://www.iprbookshop.ru/47434.html>
3. Кудинов Ю.И. Основы современной информатики [Электронный ресурс]: учебное пособие / Ю.И. Кудинов, Ф.Ф. Пащенко. Санкт-Петербург: Лань, 2018. <https://e.lanbook.com/book/107061>.