

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

ВЕЩЕСТВЕННЫЕ ЧИСЛА
ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

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

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	5
ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА ВЕЩЕСТВЕННЫХ ЧИСЕЛ.....	5
Дробные числа в позиционных системах счисления.....	5
Формат с фиксированной запятой.....	8
Формат с плавающей запятой.....	10
ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ	16
8-битные кодировки.....	16
Unicode.....	18
Представление UTF-16	19
Представление UTF-8	21
РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАБОТЫ.....	23
ИНДИВИДУАЛЬНЫЕ ВАРИАНТЫ	39
Задача № 1	39
Задача № 2.....	40
Задача № 3	41
Задача № 4.....	42
ТРЕБОВАНИЯ К ОТЧЕТУ.....	43
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА.....	44

ВВЕДЕНИЕ

В языках программирования манипулирование числами разного формата реализуется через различные числовые типы данных, при этом компьютер обычно предоставляет программисту возможность выбора из нескольких числовых форматов наиболее подходящего для конкретной задачи. Вещественные числа в компьютерах различных типов записываются с использованием четырех, шести, восьми или десяти байт, одним из способов представления - с фиксированной и плавающей запятой. Для обработки текстовой информации необходимо представить ее в двоичной знаковой системе, при этом каждому знаку ставится в соответствие уникальный 8-битовый двоичный код.

В домашнем задании по теоретической информатике рассматриваются способы представления: вещественных чисел в цифровых вычислительных машинах и основы машинной арифметики, а так же алгоритмы кодирования текстовой информации с помощью двоичного кода. 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 ВЕЩЕСТВЕННЫХ ЧИСЕЛ

Дробные числа в позиционных системах счисления

Перевод дробного числа из некоторой произвольной позиционной системы счисления с основанием q в десятичную осуществляется посредством записи числа в виде взвешенной суммы цифр.

Алгоритм 1

1. Пронумеровать разряды целой части числа справа налево, присвоив младшему разряду (разряду единиц) номер «0», следующему разряду – номер «1» и т. д.
2. Пронумеровать разряды дробной части числа слева направо, присвоив разряду после десятичной запятой номер «-1», следующему разряду – номер «-2» и т.д.
3. Составить взвешенную сумму, каждым слагаемым которой является произведение очередной цифры числа на ее вес, равный основанию q в степени номера разряда цифры.
4. Вычислить значение составленной суммы (вычисления осуществлять по правилам десятичной системы счисления).

Для перевода дробного числа из десятичной системы счисления в позиционную систему счисления с основанием q используется следующий алгоритм.

Алгоритм 2

1. *Целая часть* числа переводится по алгоритму:
 - 1.1. «В столбик» поделить число на основание системы счисления q . Записать остаток от деления в виде цифры в системе счисления с основанием q и запомнить ее.
 - 1.2. Частное поделить на основание системы счисления q . Записать остаток от деления в виде цифры в системе счисления с основанием q и запомнить ее.
 - 1.3. Продолжать выполнять шаг 2, пока в результате не получится частное, меньшее q . Записать это частное в виде цифры в системе счисления с основанием q и запомнить ее.
 - 1.4. Записать все полученные цифры в порядке от последней (последнего частного) до первой (первого остатка).
2. «В столбик» умножить *дробную часть числа* число на основание системы счисления q . Записать целую часть произведения в виде цифры в системе счисления с основанием q и запомнить ее.
3. Дробную часть произведения умножить на основание системы счисления q . Записать целую часть произведения в виде цифры в системе счисления с основанием q и запомнить ее.
4. Продолжать выполнять шаг 3, пока в результате не получится целое число, либо не будет достигнута требуемая точность.

5. Записать все полученные цифры в порядке от первой (целой части первого произведения – шаг 2) до последней.

Следует помнить, что не все конечные дроби в одной системе счисления могут быть представлены конечными дробями в другой системе счисления, поэтому при переводе обычно останавливаются на приближенном результате с некоторой требуемой точностью.

Перевод чисел из двоичной системы счисления в систему счисления с основанием $r = 2^k$, где k – некоторое натуральное число ($k > 1$) осуществляется следующим образом.

Алгоритм 3

1. Начиная с десятичного разделителя отметить в двоичном числе группы по k двоичных цифр.
2. Если в самой правой группе менее k цифр, то дополнить группу справа незначащими нулями до k цифр.
3. Каждой группе соответствует одна цифра r -ичной системы счисления. Пользуясь таблицей перевода ([табл. 1](#) и [2](#) для $r = 16$ и $r = 8$ соответственно), заменить все отмеченные двоичные группы на r -ичные цифры.

Перевод чисел из системы счисления с основанием $r = 2^k$ (k – некоторое натуральное число, $k > 1$) в двоичную осуществляется по обратному алгоритму.

Алгоритм 4

1. Пользуясь таблицей перевода, каждую цифру r -ичной системы счисления заменить на группу k двоичных цифр.
2. Незначащие нули (слева в целой части, справа в дробной части) можно отбросить.

Таблица 1 Соответствие шестнадцатеричных цифр и двоичных тетрад

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

Таблица 2 Соответствие восьмеричных цифр и двоичных триад

8-ичная цифра	Двоичная триада
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Формат с фиксированной запятой

В формате с *фиксированной запятой* (fixed point format) представляются вещественные числа с точностью до заданного количества знаков после запятой. В памяти числа с

фиксированной запятой хранятся в рассмотренных ранее целочисленных форматах, но при этом предполагается, что часть битов числа представляют собой его дробную часть.

Для обозначения таких форматов используется Q-нотация.

Символ $Q_{m.n}$ обозначает число с фиксированной запятой, хранящееся в виде $(m + n + 1)$ -битного знакового целого числа, причем старший бит представляет собой знак числа, следующие m бит – его целую часть, следующие n бит – его дробную часть. Величину n обычно называют масштабом (*scale*).

Если m не указано, то предполагается, что $m = 0$.

Если $n = 0$, то имеем обычное $(m + 1)$ -битное целое число.

В формате $Q_{m.n}$ целое число N представляет вещественное число R , определяемое по формуле:

$$R = N \cdot 2^{-n} \quad (1)$$

И наоборот, вещественное число R хранится в виде целого числа N , определяемого по формуле:

$$N = R \cdot 2^n \quad (2)$$

В формате $Q_{m.n}$ существует 2^{m+n+1} целых значений из диапазона $[-2^{m+n}; 2^{m+n}-1]$, а значит могут быть представлены 2^{m+n+1} значений с фиксированной запятой из диапазона $[-2^m; 2^m - 2^{-n}]$ с шагом 2^{-n} .

Для получения суммы/разности двух чисел в формате $Qm.n$ достаточно просто сложить/вычесть их целочисленные представления, т. к.:

$$R_1 \pm R_2 = N_1 \cdot 2^{-n} \pm N_2 \cdot 2^{-n} = (N_1 \pm N_2) \cdot 2^{-n} \quad (3)$$

Для получения произведения/частного необходимо выполнить умножение/деление целочисленных представлений и скорректировать результат (поделить на 2^n при умножении и домножить на 2^n при делении):

$$R_1 \times R_2 = N_1 \cdot 2^{-n} \times N_2 \cdot 2^{-n} = (N_1 \times N_2 \times 2^{-n}) \cdot 2^{-n}; \quad (4)$$

$$\frac{R_1}{R_2} = \frac{N_1 \cdot 2^{-n}}{N_2 \cdot 2^{-n}} = \left(\frac{N_1}{N_2} \times \frac{1}{2^{-n}} \right) \cdot 2^{-n}. \quad (5)$$

Замечание: корректировку результата можно свести к арифметическому сдвигу вправо на n разрядов после умножения и к арифметическому сдвигу влево на n разрядов после деления.

Формат с плавающей запятой

В формате *с плавающей запятой* (floating point format) вещественное число R в системе счисления q представляются в экспоненциальном виде:

$$R = M \cdot q^E \quad (6)$$

Для двоичной системы имеем запись:

$$R = M \cdot 2^E \quad (6')$$

Величину M называют *мантиссой* (significand), а величину E – *порядком* (exponent).

Например:

$$\begin{aligned}-7,625 &= -111,101_2 = -1,11101_2 \cdot 2^2 = -11,1101_2 \cdot 2^1 = \\ &= -1111,01_2 \cdot 2^{-1}.\end{aligned}$$

Нормальной формой называют форму записи (6) или (6'), в которой

$$0 \leq M < 1 \quad (7)$$

В нормальной форме у одного числа может быть множество различных представлений.

Нормализованной формой называют форму записи (4.6), в которой

$$1 \leq M < q \quad (8)$$

Или для двоичной системы счисления:

$$1 \leq M < 2 \quad (8')$$

В нормализованной форме у числа единственное представление, но число «нуль» не может быть представлено вообще. Кроме того, в двоичной системе счисления в нормализованной записи целая часть мантииссы всегда равна 1.

Нормализованная запись для рассмотренного выше примера: $-7,625 = -1,11101_2 \cdot 2^2$.

Формат с плавающей запятой хранит в памяти значение мантииссы M со знаком и значение порядка со знаком E .

Форматы с плавающей запятой, представления особых значений, правила выполнения арифметических операций в этих форматах, правила округления и исключительные

ситуации при выполнении операций описаны в стандарте **IEEE 754**.

Форматы вещественных чисел с плавающей запятой, определенные в стандарте IEEE 754, описаны в таблице 3. Диапазоны представимых значений приведены в таблице 4.

Таблица 3 Структура форматов вещественных чисел по IEEE 754

Наименование	Общая длина, бит	Структура			
		<i>K</i>	<i>S</i>	<i>E</i> *	<i>M</i> *
Binary16 Число половинной точности (half precision)	16	15	1	5	10
Binary32 Число одинарной точности (single precision) Тип Pascal: Single	32	127	1	8	23
Binary64 Число двойной точности (double precision) Тип Pascal: Double	64	1 023	1	11	52
Binary128 Число четверной точности (quadruple precision)	128	16 383	1	15	112

Таблица 4 Диапазоны значений вещественных чисел по IEEE 754

Формат	Тип Pascal	Мин. значение денормал. (положит.)	Мин. значение нормал. (положит.)	Макс. значение нормал. (положит.)
Binary16	–	$5,9 \cdot 10^{-8}$	$6,1 \cdot 10^{-5}$	$65\,504$
Binary32	Single	$1,5 \cdot 10^{-45}$	$1,2 \cdot 10^{-38}$	$3,4 \cdot 10^{38}$
Binary64	Double	$5,0 \cdot 10^{-324}$	$2,3 \cdot 10^{-308}$	$1,7 \cdot 10^{308}$
Binary128	–	$10^{-4\,965}$	$3,4 \cdot 10^{-4\,932}$	$1,2 \cdot 10^{4\,932}$

Все форматы хранят вещественные числа, записанные в нормализованной форме – см. (6') и (8'), в следующем виде:

S	E^*	M^*
-----	-------	-------

Поле « S » (обычно 1 бит) хранит знак мантиисы: 0 – мантииса положительная, 1 – мантииса отрицательная.

Поле « E^* » (длину для разных форматах см. в [табл. 3](#)) хранит экспоненту в смещенном виде:

$$E^* = E + K \quad (9)$$

Здесь, K – смещение порядка (exponent bias), определяемое форматом (см. [таблицу 3](#)).

Поле « M^* » (длину для разных форматах см. в [табл. 3](#)) хранит биты дробной части мантиисы (т. к. в нормализованной записи целая часть всегда равна 1).

Сложение/вычитание чисел с плавающей запятой осуществляется по следующему алгоритму.

Алгоритм 5 сложения чисел в формате с плавающей запятой

1. Привести числа к общему порядку – наибольшему из порядков слагаемых. Для этого денормализуем слагаемые, порядок которых меньше требуемого: сдвигаем в мантиисе запятую на количество разрядов влево, равное разнице между текущим порядком и требуемым.

2. Выполнить сложение/вычитание мантиис, т. к.

$$R_1 \pm R_2 = M_1 \cdot 2^{E^*} \pm M_2 \cdot 2^{E^*} = (M_1 \pm M_2) \cdot 2^{E^*}$$

3. При необходимости, нормализовать полученную сумму/разность.
4. Округлить мантиссу результата, если в ее дробной части получено больше разрядов, чем позволяет сохранить формат.

При умножении/делении выполняется умножение/деление мантисс и сложение/вычитание порядков с последующей нормализацией результата.

Ситуация, когда в результате выполнения операции получается значение по модулю большее, чем максимальное представимое (см. [табл. 4](#)), называется *переполнением* (overflow). Это может произойти, например, при сложении больших чисел.

При переполнении обычно возвращается особое значение – *положительная* или *отрицательная бесконечность* ($\pm\text{Infinity}$). Это значение представляется следующим образом: все биты порядка выставлены в 1, все биты мантиссы сброшены в 0, а в поле знака – бит знака бесконечности.

S	E^*	M^*
\pm	1111...1	0000...0

Так, например, для формата Binary32 положительная и отрицательная бесконечности имеют вид соответственно: 7F 80 00 00₁₆ и FF 80 00 00₁₆.

Ситуация, когда в результате выполнения операции получается значение по модулю меньшее, чем минимальное

представимое нормализованное (см. [табл. 4](#)), называется *потерей точности* (underflow). Это может произойти, например, при делении на большое число.

При потере точности результат обычно возвращается в *денормализованном* виде (denormalized, subnormal). Такие значения представляется следующим образом: все биты порядка сброшены в 0, в поле мантииссы хранятся биты дробной части денормализованной мантииссы, а в поле знака – бит знака мантииссы.

S	E^*	M^*
\pm	0000...0	$m_1 m_2 m_3 \dots m_n$

Целая часть денормализованной мантииссы равна 0, т. е., мантиисса имеет вид:

$$M^* = 0, m_1 m_2 m_3 \dots m_n$$

У денормализованных чисел предполагается порядок, равный минимальному допустимому в рамках данного формата: по [табл. 3](#):

$$E_D = -K + 1 \quad (10)$$

К денормализованным значениям можно также отнести и значения положительного и отрицательного нуля. Это значение представляется следующим образом: все биты порядка сброшены в 0, все биты мантииссы сброшены в 0, а в поле знака – бит знака нуля.

S	E^*	M^*
\pm	0000...0	0000...0

Так, например, для формата Binary32 положительный и отрицательный нуль имеют вид соответственно: 00 00 00 00₁₆ и 80 00 00 00₁₆.

ПРЕДСТАВЛЕНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ

8-битные кодировки

При представлении текста каждому символу алфавита ставится в соответствии некоторая последовательность бит (*bit pattern*) – двоичный код. Перечень символов и их двоичных кодов называется *кодировкой* (*encoding*).

В 1963 г. была разработана, а в 1967 г. доработана стандартная кодировка ASCII (American Standard Code for Information Interchange).

Данная кодировка использует 7-битные кодовые последовательности (00₁₆–7F₁₆) и определяет коды для следующих символов ([Приложение 1](#)):

- латинских букв;
- цифр;
- знаков препинания;
- некоторых математических символов;
- некоторых специальных символов;
- т. н. управляющих символов (например, CR и LF – символы возврата каретки и перевода строки; HT – горизонтальная табуляция, SP – пробел, ESC –

альтернативный регистр, DEL – удаление, BEL – звуковой сигнал).

Следует знать, в кодировке ASCII коды латинских букв образуют непрерывную возрастающую последовательность: «А» – 41_{16} , «В» – 42_{16} , ..., «Z» – $5A_{16}$; аналогично и для строчных букв: «а» – 61_{16} , «b» – 62_{16} , ..., «z» – $7A_{16}$. Непрерывную возрастающую последовательность образуют и коды цифр: «0» – 30_{16} , «1» – 31_{16} , ..., «9» – 39_{16} .

В дальнейшем ASCII стала международным стандартом под именем ISO/IEC 646:1991. В этом стандарте часть символов не определены и должны быть заданы в национальных реализациях (в [Приложении 1](#) выделены серым цветом).

8-битные кодировки каждый символ текста кодируют 8-битным целым числом. Для кодирования управляющих символов, символов латинского алфавита, цифр, знаков препинания и некоторых других символов используется таблица ASCII

Существуют различные варианты дополнения кодировки ASCII до 8-битной.

Семейство стандартов ISO 8859 используется в операционных системах Unix. Стандарт ISO 8859-5 (см. [Приложение 2](#)) этого семейства определяет символы кириллицы и других славянских языков.

В операционных системах MS-DOS и Windows различные варианты наборов символов для 8-битных кодов 80_{16} – FF_{16}

называют *кодowymi страницами* (code page). В MS-DOS помимо символов национального алфавита кодовая страница содержала *символы псевдографики*, использовавшиеся для отображения графического интерфейса в текстовом режиме. С выходом Windows необходимость в псевдографике отпала, поэтому стали использоваться новые наборы кодовых страниц.

Кодовые страницы, содержащие символы кириллицы: MS-DOS cp866 ([см. Приложение 3](#)) и Windows-1251 ([см. Приложение 4](#)).

Unicode

В настоящее время для кодирования, представления и обработки текста используется стандарт Unicode. Этот стандарт каждому символу (*code point*) ставит в соответствие 21-битное числовое значение из диапазона 000000_{16} – $10FFFF_{16}$.

В пространстве Unicode можно выделить *блоки* (blocks) символов — непрерывные последовательности символов, относящиеся к одной «письменности». Например, блок «Controls and Basic Latin» с кодами 0000_{16} – $007F_{16}$ (табл. 5.5) эквивалентен таблице ASCII (ср. [Приложение 1](#)), а блок «Cyrillic» с кодами 0400_{16} – $04FF_{16}$ ([Приложение 6](#)) содержит символы кириллицы.

Символы Unicode будем записывать в виде «U+xxxx», где xxxx — шестнадцатеричное значение кода символа.

Существуют различные варианты представления 21-битных кодов Unicode в памяти (в оперативной или во внешней – файлах на диске). Рассмотрим следующие: UTF-8, UTF-16, UTF-32.

Представление UTF-16

Каждый код Unicode представляется в виде 2 байт или 4 байт.

Символы от U+0000 до U+D7FF и от U+E000 до U+FFFF кодируются 2 байтными значениями («как есть», т. е. код равен численному значению code point).

Символы от U+D800 до U+DFFF – зарезервированы (см. ниже), они не должны соответствовать реальным символам.

Символы от U+010000 до U+10FFFF кодируются 4 байтными значениями по следующему алгоритму.

Алгоритм 1. Получение кода UTF-16 для символов от U+010000 до U+10FFFF

1. Из 21-битного кода символа вычитается значение 10000_{16} . В результате получается 20-битное значение от 0_{16} до $FFFFF_{16}$.
2. К старшим 10 битам (значению от 0_{16} до $3FF_{16}$) результата, полученного в п. 1, прибавляется значение $D800_{16}$. Полученное 16-битное значение (из диапазона $D800_{16}$ – $DBFF_{16}$) становится первой половиной кода. Оно

называется *старшим суррогатом* (high surrogate или leading surrogate).

3. К младшим 10 битам (значению от 0_{16} до $3FF_{16}$) результата, полученного в п. 1, прибавляется значение $DC00_{16}$. Полученное 16-битное значение (из диапазона $DC00_{16}$ – $DFFF_{16}$) становится второй половиной кода. Оно называется *младшим суррогатом* (low surrogate или trailing surrogate).
4. 2-байтные значения, полученные в пунктах 2 и 3, образуют 4-байтный код символа.

В зависимости от архитектуры аппаратного и программного обеспечения, UTF-16 может записывать байты кода в разном порядке: Big-endian или Little-endian; соответствующие варианты представления принято обозначать UTF-16BE и UTF-16LE.

При сохранении текста в представлении UTF-16 в файл, в начало файла может добавляться т. н. *маркер порядка байт* (byte order mark, BOM) –символ U+FEFF «ZERO WIDTH NON-BREAKING SPACE». Этот символ позволяет определить используемый порядок байт: если в начала файла находятся байты FE FF, то при сохранении использовался UTF-16BE, если же байты FF FE, то использовался UTF-16LE. Если BOM отсутствует, то стандарт предполагает использование UTF-16BE (однако многие Windows-приложения

предполагают UTF-16LE, т. к. данная кодировка используется в операционной системе по умолчанию).

Представление UTF-8

Каждый код Unicode представляется в виде 1, 2, 3 или 4 байт.

Алгоритм 2. Получение кода UTF-8 (см. [Приложение 7](#))

1. Для символа из диапазона от U+0000 до U+007F (соответствующего 7-битной кодировке ASCII) используется 1-байтный код, содержащий числовое значение символа.
2. Для символа из диапазона от U+0080 до U+07FF (в этот диапазон попадают и символы блока Cyrillic) используется 2-байтный код. Числовое значение из рассматриваемого диапазона является 11-битным. Запишем его следующим образом: $00000xxx\ xхуууууу_2$. Тогда 2 байта кода будут иметь следующий вид: $110xxxxx\ 10уууууу_2$.
3. Для символа из диапазона от U+0800 до U+FFFF используется 3-байтный код. Числовое значение из рассматриваемого диапазона является 16-битным. Запишем его следующим образом: $xxxxуууу\ ууzzzzzz_2$. Тогда 3 байта кода будут иметь следующий вид: $1110xxxx\ 10уууууу\ 10zzzzzz_2$.
4. Для символа из диапазона от U+10000 до U+10FFFF используется 4-байтный код. Числовое значение из рассматриваемого диапазона является 21-битным. Запишем

его следующим образом: 000xxxуу ууууzzzz zzvvvvvvv₂.

Тогда 4 байта кода будут иметь следующий вид:

11110xxx 10уууууу 10zzzzzz 10vvvvvvv₂.

Первый байт последовательности называется leading byte, а последующие (в случае многобайтного кода) – continuation bytes.

Замечание. UTF-8 представляет собой префиксный код: байты с префиксом «0» являются 1-байтными кодами; байты с префиксом «110» – первые байты в 2-байтном коде; байты с префиксом «1110» – первые байты в 3-байтном коде; байты с префиксом «11110» – первые байты в 4-байтном коды; байты с префиксом «10» – байты-продолжения (второй и последующий байты) в многобайтном коде. Можно заметить, что количество единичных битов в начале первого байта равно общему количеству байт в коде (исключение – однобайтные коды).

При хранении и передаче информации в представлении UTF-8 байты всегда следуют в порядке, который определен алгоритмом 2, т.е. вариаций BE и LE для данного представления не существует. Однако, иногда может использоваться маркер порядка байт EF BB BF, обозначающий кодировку UTF-8 (этот маркер является представлением символа U+FEFF в UTF-8).

РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАБОТЫ

Задача 1. Покажите, как в формате Q4.3 представляются числа: 10,75 и $-7,625$. Ответ приведите в двоичном и шестнадцатеричном виде.

Решение, 1-й способ

В формате Q4.3 число $R_1 = 10,75$ представляется как целое число N_1 , определяемое формулой (4.2):

$$N_1 = R_1 \cdot 2^n = 10,75 \cdot 2^3 = 86.$$

Выполняя перевод в двоичную систему, получаем:

$86 = 1010110_2 = 56_{16}$. Положительные числа хранятся в прямом коде, поэтому число R_1 представляется как двоичное значение $01010110_2 = 56_{16}$.

Число $R_2 = -7,625$ представляется как целое число N_2 , определяемое формулой (4.2):

$$N_2 = R_2 \cdot 2^n = -7,625 \cdot 2^3 = -61.$$

Выполняя перевод в двоичную систему, получаем:

$-61 = -111101_2 = -3D_{16}$. Отрицательные числа хранятся в дополнительном коде, поэтому число R_2 представляется как двоичное значение $11000011_2 = C3_{16}$.

Решение, 2-й способ

Переведем число 10,75 в двоичную систему счисления:

$$10,75 = 1010,11_2.$$

Для формата Q4.3 требуются 3 знака после запятой и 7 знаков всего, поэтому дополним число незначащим нулем справа:

$$10,75 = 1010,110_2.$$

Т. о., заданное вещественное число хранится в виде: $01010110_2 = 56_{16} = 86$.

Переведем число $-7,625$ в двоичную систему счисления:

$$-7,625 = -111,101_2.$$

Незначащими нулями слева дополняем число до 7 знаков:

$$-7,625 = -0111,101_2.$$

Если бы число было бы положительным, оно хранилось бы в виде: 00111101_2 , но т. к. число отрицательное, найдем его дополнительный код: $11000011_2 = C3_{16} = -61$.

Ответ: $01010110_2, 56_{16}; 11000011_2, C3_{16}$.

Задача 2. Покажите, как в формате Q4.3 представляются числа: $-4,125$ и $-13,625$. Выполните сложение чисел в соответствии с правилами формата и укажите, произошло ли при этом переполнение. Представьте полученный результат в виде десятичного вещественного числа.

Решение

В формате Q4.3 число $R_1 = -4,125$ представляется как целое число N_1 , определяемое [формулой \(2\)](#):

$$N_1 = R_1 \cdot 2^n = -4,125 \cdot 2^3 = -33.$$

Выполняя перевод в двоичную систему

$(-33 = -00100001_2)$, а также учитывая, что отрицательные значения хранятся в дополнительном коде, получаем, что R_1 представляется как двоичное значение $11011111_2 = DF_{16}$.

Число $R_2 = -13,625$ представляется как целое число N_2 , определяемое [формулой \(2\)](#):

$$N_2 = R_2 \cdot 2^n = -13,625 \cdot 2^3 = -109.$$

Выполняя перевод в двоичную систему

$(-109 = -01101101_2)$, а также учитывая, что отрицательные значения хранятся в дополнительном коде, получаем, что R_2 представляется как двоичное значение $10010011_2 = 93_{16}$.

Для того, чтобы получить сумму вещественных чисел в формате с фиксированной запятой, достаточно выполнить сложение из целочисленных представлений – [по формуле \(3\)](#).

Выполняем сложение полученных знаковых целочисленных значений:

$$\begin{array}{rcccccccc}
 & (*) & & * & * & * & * & * \\
 & & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
 + & & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
 \hline
 & (1) & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0
 \end{array}$$

Был перенос из разрядной сетки, а переноса в знаковый (старший) разряд не было. Значит, произошло переполнение. В результате операции получен **неверный** результат

$01110010_2 = 72_{16}$ (выходящая за пределы разрядной сетки цифра отбрасывается).

Выполняя перевод в десятичную систему счисления, получаем целое значение $N_3 = 114$, по [формуле \(1\)](#) находим представленное таким образом вещественное число:

$$R_3 = N_3 \cdot 2^{-n} = 114 \cdot 2^{-3} = 14,25.$$

Ответ: 72_{16} ; 14,25; результат неверный

Задача 3. Покажите, как в формате IEEE 754 Binary16 представляются числа:

а) $-10,1875$; б) $0,17578125$;

Ответ приведите в шестнадцатеричном виде.

Решение задания «а»

Выполним перевод числа в двоичную систему счисления по [алгоритму 2](#) (здесь сам перевод опущен).

$$-10,1875 = -1010,0011_2.$$

Запишем число в нормализованном виде, т. е. в виде $(4.6')$, так чтобы выполнялось [условие \(8'\)](#):

$$-10,1875 = -1010,0011_2 = -1,0100011_2 \cdot 2^3.$$

Обозначим мантиссу числа $M_1 = -1,0100011_2$, а порядок числа $-E_1 = 3$.

Знаковый бит «S» будет содержать значение 1, т. к. мантисса M_1 отрицательна.

Биты смещенного порядка « E^* » (5 бит) хранят значение:

$$E_1^* = E_1 + K = 3 + 15 = 18 = 10010_2.$$

Биты мантииссы « M^* » (10 бит) хранят значение дробной части мантииссы, т. е. двоичные цифры после запятой: 0100011000_2 . Дробная часть дополняется до нужного количества разрядов незначащими нулями справа.

15				8				7	0					
1	1	0	0	1	0	0	1	0	0	0	1	1	0	0
S		E^*					M^*							

Т. о., первое число хранится в виде: $11001001\ 00011000_2 = \text{CD } 18_{16}$.

Ответ: C9 18₁₆.

Решение задания «б»

Выполним перевод числа в двоичную систему счисления по [алгоритму 2](#) (здесь сам перевод опущен).

$$0,17578125 = 0,00101101_2.$$

Запишем число в нормализованном виде, т. е. в виде $(6')$, так чтобы выполнялось [условие \(8'\)](#):

$$0,17578125 = 0,00101101_2 = 1,01101_2 \cdot 2^{-3}.$$

Обозначим мантииссу числа $M_2 = 1,01101_2$, а порядок числа – $E_2 = -3$.

Знаковый бит « S » будет содержать значение 0, т. к. мантиисса M_2 положительна.

Биты смещенного порядка « E^* » (5 бит) хранят значение:

$$E_2^* = E_2 + K = -3 + 15 = 12 = 01100_2.$$

Как и все целые значения, это число дополняется до 5 требуемых разрядов нулями слева.

Биты мантиисы « M^* » (10 бит) хранят значение дробной части мантиисы, т. е. двоичные цифры после запятой: 0110100000_2 . Дробная часть дополняется до нужного количества разрядов незначащими нулями справа.

15			8						7	0					
0	0	1	1	0	0	0	1	1	0	1	0	0	0	0	0
S		E^*					M^*								

Т. о., число хранится в виде: $00110001\ 10100000_2 = 31\ A0_{16}$.

Ответ: $31\ A0_{16}$.

Задача 4. Покажите, как в формате IEEE 754 Binary16 представляются указанные числа. Выполните сложение чисел в соответствии с правилами формата и укажите, произошло ли при этом переполнение. Покажите, как полученный результат представляется в формате IEEE 754 Binary16. Представьте полученный результат в виде десятичного вещественного числа.

а) 71,625 и 15,5625; б) $-10,859375$ и $61,6875$

Решение задания «а»

Покажем, как в заданном формате представляется число 71,625.

$$71,625 = 1000111,101_2 = 1,000111101_2 \cdot 2^6.$$

Мантииса положительна, поэтому знаковый бит « S » равен 0.

Смещенный порядок: $E_1^* = 6 + 15 = 21 = 10101_2$.

Дробная часть мантиисы: $M_1^* = 0001111010_2$.

Получаем следующее представление:

$$01010100\ 01111010_2 = 54\ 7A_{16}.$$

Покажем, как в заданном формате представляется число 15,5625.

$$15,5625 = 1111,1001_2 = 1,1111001_2 \cdot 2^3.$$

Мантисса положительна, поэтому знаковый бит «S» равен 0.

Смещенный порядок: $E_2^* = 3 + 15 = 18 = 10010_2$.

Дробная часть мантиссы: $M_2^* = 1111001000_2$.

Получаем следующее представление:

$$01001011\ 11001000_2 = 4B\ C8_{16}.$$

Сложение осуществляется в соответствии с [алгоритмом 5](#).

1. Приводим числа к общему порядку – наибольшему из порядков слагаемых. В данном случае – это $E_1 = 6$. Денормализуем второе слагаемое: сдвигаем в мантиссе запятую на 3 разряда влево (разница между текущим порядком $E_2 = 3$ и требуемым $E_1 = 6$). Получаем числа:

$$R_1 = 1,000111101_2 \cdot 2^6$$

$$R_2 = 0,0011111001_2 \cdot 2^6$$

2. Выполняем сложение мантисс, при этом порядок результата будет равен порядку, полученному на шаге 1 ($E_1 = 6$).

			*	*	*	*	*				
	1,	0	0	0	1	1	1	1	0	1	
+	0,	0	0	1	1	1	1	1	0	0	1
	1,	0	1	0	1	1	1	0	0	1	1

3. Полученная сумма уже нормализована.

4. В дробной части результата 10 разрядов, округление не требуется.

Т. о., получено значение: $R_s = 1,0101110011_2 \cdot 2^6$. Порядок не превышает максимально допустимый, значит переполнения нет.

В формате Binary16 данное число представляется следующим образом: $01010101\ 01110011_2 = 55\ 73_{16}$.

Получим десятичное представление числа R_s .

$$1,0101110011_2 \cdot 2^6 = 1010111,0011_2 = 87,1875.$$

Ответ: $54\ 7A_{16}$; $4B\ C8_{16}$; переполнения нет; $55\ 73_{16}$; 87,1875.

Решение задания «б»

Покажем, как в заданном формате представляется число $-10,859375$.

$$-10,859375 = -1010,110111_2 = -1,010110111_2 \cdot 2^3.$$

Мантисса отрицательна, поэтому знаковый бит «S» равен 1.

Смещенный порядок: $E_1^* = 3 + 15 = 18 = 10010_2$.

Дробная часть мантиссы: $M_1^* = 0101101110_2$.

Получаем следующее представление:

$$11001001\ 01101110_2 = C9\ 6E_{16}.$$

Покажем, как в заданном формате представляется число 61,6875.

$$61,6875 = 111101,1011_2 = 1,111011011_2 \cdot 2^5.$$

Мантисса положительна, поэтому знаковый бит «S» равен 0.

Смещенный порядок: $E_2^* = 5 + 15 = 20 = 10100_2$.

Дробная часть мантииссы: $M_2^* = 1110110110_2$.

Получаем следующее представление:

$$01010011 \ 10110110_2 = 53 \text{ В}_{16}.$$

Сложение осуществляется в соответствии с [алгоритмом 5](#).

1. Приводим числа к общему порядку – наибольшему из порядков слагаемых. В данном случае – это $E_2 = 5$. Денормализуем первое слагаемое: сдвигаем в мантиссе запятую на 2 разряда влево (разница между текущим порядком $E_1 = 3$ и требуемым $E_1 = 5$). Получаем числа:

$$R_1 = -0,01010110111_2 \cdot 2^5$$

$$R_2 = 1,111011011_2 \cdot 2^5$$

2. Выполняем вычитание мантиисс, при этом порядок результата будет равен порядку, полученному на шаге 1 ($E_2 = 5$).

					*		*	*		*	*	*
	1,	1	1	1	0	1	1	0	1	1	0	0
-	0,	0	1	0	1	0	1	1	0	1	1	1
	1,	1	0	0	1	0	1	1	0	1	0	1

3. Полученная разность уже нормализована.
4. В дробной части результата – 11 разрядов. Требуется округление до 10 разрядов. Стандарт определяет несколько методов округления, продемонстрируем самый простой – *округление к нулю* (round toward zero, truncating), заключающийся в том, что «лишние» разряды просто отбрасываются. В нашем случае округленная мантисса:

$$M_s = 1,1001011010_2.$$

Т. о., получено значение: $R_s = 1,100101101_2 \cdot 2^5$. Порядок не превышает максимально допустимый, значит переполнения нет.

В формате Binary16 данное число представляется следующим образом: $01010010\ 01011010_2 = 52\ 5A_{16}$.

Получим десятичное представление числа R_s .

$$1,100101101_2 \cdot 2^5 = 110010,1101_2 = 50,8125.$$

Ответ: C9 6E₁₆; 53 B6₁₆; переполнения нет; 52 5A₁₆; 50,8125.

Решение задания «в»

Покажем, как в заданном формате представляется число 42 560.

$$42\ 560 = 1010011001000000_2 = 1,010011001_2 \cdot 2^{15}.$$

Мантисса положительна, поэтому знаковый бит «S» равен 0.

Смещенный порядок: $E_1^* = 15 + 15 = 30 = 11110_2$.

Дробная часть мантиссы: $M_1^* = 0100110010_2$.

Получаем следующее представление:

$$01111001\ 00110010_2 = 79\ 32_{16}.$$

Покажем, как в заданном формате представляется число 34 273.

$$34\ 273 = 1000010111100001_2 = 1,000010111100001_2 \cdot 2^{15}.$$

Мантисса положительна, поэтому знаковый бит «S» равен 0.

Смещенный порядок: $E_2^* = 15 + 15 = 30 = 11110_2$.

Дробная часть мантииссы: $M_2^* = 000010111100001_2$. Полученное значение содержит 15 разрядов, в то время как формат Binary16 хранит только 10. Рассматриваемое число не может быть представлено в формате Binary16, поэтому используется ближайшее представимое:

$$1,000010111100000_2 \cdot 2^{15} = 34\,272.$$

Получаем следующее представление:

$$01111000\,00101111_2 = 78\,2F_{16}.$$

Сложение осуществляется в соответствии с [алгоритмом 5](#).

1. У чисел одинаковый порядок $E_1 = E_2 = 15$.
2. Выполняем сложение мантиисс, при этом порядок результата будет равен порядку, полученному на шаге 1 ($E_1 = E_2 = 15$).

					*	*	*	*	*		
	1,	0	1	0	0	1	1	0	0	1	0
+	1,	0	0	0	0	1	0	1	1	1	1
	1	0,	0	1	0	1	1	0	0	0	1

3. Полученная сумма денормализована. Нормализуем:

$$10,0101100001_2 \cdot 2^{15} = 1,00101100001_2 \cdot 2^{16}.$$

Однако, полученный порядок $E_s = 16$ непредставим в формате Binary16. Т.о., при выполнении сложения происходит переполнение. По умолчанию, при переполнении в качестве результата возвращается значение бесконечности, в нашем случае – положительной, т.к. полученная сумма положительна. В формате Binary16 положительная бесконечность имеет вид: $7C\,00_{16}$.

Ответ: 79 32₁₆; 78 2F₁₆; переполнение; 7C 00₁₆; +∞.

Задача 5. Покажите, как текст «Я выучил C++14!» представляется с использованием 8-битных кодировок:

а) cp866; б) Windows-1251.

В ответе приведите шестнадцатеричные коды символов.

Решение задания «а»

Найдем код для каждого символа текста. Коды русских букв будем искать по таблице [Приложение 3](#) для «верхней» части кодовой страницы cp866, коды латинских букв («C»), знаков (пробел, «+», «!») и цифр – по таблице [Приложение 1](#) для «нижней» части таблицы ASCII.

Символ	Я	[SP]	в	ы	у	ч	и	л	[SP]
Табл.	3	1	3	3	3	3	3	3	1
Код	9F	20	A2	EB	E3	E7	A8	AB	20

Символ	С	+	+	1	4	!
Табл.	1	1	1	1	1	1
Код	43	2B	2B	31	34	21

Т. о., символы заданного текста кодируется следующими числами: 9F 20 A2 EB E3 E7 A8 AB 20 43 2B 2B 31 34 21.

Длина строки – 15 символов, длина кода – 15 байт.

Ответ: 9F 20 A2 EB E3 E7 A8 AB 20 43 2B 2B 31 34 21.

Решение задания «б»

Найдем код для каждого символа текста. Коды русских букв будем искать по таблице [Приложение 4](#) для «верхней» части кодовой страницы Windows-1251, коды латинских букв («C»),

знаков (пробел, «+», «!») и цифр – по таблице [Приложение 1](#) для «нижней» части таблицы ASCII.

Символ	Я	[SP]	в	ы	у	ч	л	л	[SP]
Табл.	3	1	3	3	3	3	3	3	1
Код	DF	20	E2	FB	F3	F7	E8	EB	20

Символ	С	+	+	1	4	!
Табл.	1	1	1	1	1	1
Код	43	2B	2B	31	34	21

Т. о., символы заданного текста кодируется следующими числами: DF 20 E2 FB F3 F7 E8 EB 20 43 2B 2B 31 34 21.

Длина строки – 15 символов, длина кода – 15 байт.

Ответ: DF 20 E2 FB F3 F7 E8 EB 20 43 2B 2B 31 34 21.

Задача 5. Покажите, какие символы Unicode соответствуют символам заданного текста:

«Я выучил C++14!»;

Каждый код укажите в формате U+xxxx, где xxxx – шестнадцатеричное представление кода.

Покажите, как эти символы представляются с использованием UTF-8, UTF-16, UTF-32. Ответ приведите в виде шестнадцатеричных кодов, без маркера порядка байт (BOM).

Решение

Найдем символы текста в таблицах Unicode. Коды русских букв будем искать в блоке Cyrillic (символы 0400₁₆ – 04FF₁₆, по таблице [Приложение 6](#)). Коды латинских букв, цифр и знаков

будем искать в блоке Controls and Basic Latin (символы 0000_{16} – $007F_{16}$, по таблице [Приложение 5](#)).

Символ	Таблица	Символ Unicode
Я	6	042F
[SP]	5	0020
в	6	0432
ы	6	044B
у	6	0443
ч	6	0447
и	6	0438
л	6	043B
[SP]	5	0020
С	5	0043
+	5	002B
+	5	002B
1	5	0031
4	5	0034
!	5	0021

Т. о., символы заданного текста представляют собой следующие символы Unicode: U+042F U+0020 U+0432 U+044B U+0443 U+0447 U+0438 U+043B U+0020 U+0043 U+002B U+002B U+0031 U+0034 U+0021.

Получим представление в UTF-8.

Символы U+0020, U+0043, U+002B, U+0031, U+0034 и U+0021 будут иметь 1-байтные коды, т. к. они принадлежат диапазону от U+0000 до U+007F ([алгоритма 2](#) и строку 1 по таблице [Приложение 1](#)).

«Пробел (SP)»: $U+0020 \rightarrow 00100000_2 = 20_{16}$.

«С»: $U+0043 \rightarrow 01000011_2 = 43_{16}$.

«+»: $U+002B \rightarrow 00101011_2 = 2B_{16}$.

«1»: $U+0031 \rightarrow 00110001_2 = 31_{16}$.

«4»: $U+0034 \rightarrow 00110100_2 = 34_{16}$.

«!»: $U+0021 \rightarrow 00100001_2 = 21_{16}$.

Остальные символы (в рассматриваемом примере – русские буквы) будут иметь 2-байтные коды, т. к. они принадлежат диапазону от U+0080 до U+07FF.

Построим код UTF-8 для символа U+042F (буква «Я»). Запишем численное значение символа в двоичном виде: 00000100 00101111₂. Оставим 11 значащих бит ([алгоритма 2](#) и строку 2 по таблице [Приложение 7](#)): 10000101111₂.

Разобьем на группы из 5 бит и 6 бит: 10000-101111₂.

Старший байт кода символа имеет вид: 110xxxxx₂, а младший байт – 10yyyyyy₂. xxxxx = 10000₂ (старшие 5 бит числового значения символа); yyyyyy = 101111₂ (младшие 6 бит числового значения символа).

Т. о., код символа U+042F имеет вид:

$$11010000 \ 10101111_2 = D0 \ AF_{16}.$$

Аналогично получаем коды остальных русских букв.

«в»: $U+0432 = 0000010000110010_2 = 10000110010_2 =$
 $10000-110010_2 \rightarrow 11010000 \ 10110010_2 = D0 \ B2_{16}.$

«ы»: $U+044B = 0000010001001011_2 = 10001001011_2 =$
 $10001-001011_2 \rightarrow 11010001 \ 10001011_2 = D1 \ 8B_{16}.$

«у»: $U+0443 = 0000010001000011_2 = 10001000011_2 =$
 $10001-000011_2 \rightarrow 11010001 \ 10000011_2 = D1 \ 83_{16}.$

«Ч»: $U+0447 = 0000010001000111_2 = 10001000111_2 =$
 $10001-000111_2 \rightarrow 11010001\ 10000111_2 = D1\ 87_{16}.$

«И»: $U+0438 = 0000010000111000_2 = 10000111000_2 =$
 $10000-111000_2 \rightarrow 11010000\ 10111000_2 = D0\ B8_{16}.$

«Л»: $U+043B = 0000010000111011_2 = 10000111011_2 =$
 $10000-111011_2 \rightarrow 11010000\ 10111011_2 = D0\ BB_2.$

Т. о., в UTF-8 заданный текст имеет вид:

D0 AF 20 D0 B2 D1 8B D1 83 D1 87 D0 B8 D0 BB 20 43 2B 2B
31 34 21.

ИНДИВИДУАЛЬНЫЕ ВАРИАНТЫ

Задача № 1

Даны два вещественных числа.

- Покажите, как эти числа представляются в формате с фиксированной запятой $Q8.7$ с порядком байт *Big-endian* (приведите двоичное и шестнадцатеричное представления).
- Выполните сложение чисел в соответствии с правилами формата.
- Представьте результат в виде десятичной дроби (вещественного числа в десятичной системе счисления).

Вариант	Задание
1.	$-83,8125$ и $52,8125$
2.	$-99,390625$ и $29,640625$
3.	$-92,59375$ и $20,78125$
4.	$-112,578125$ и $60,640625$
5.	$-91,40625$ и $32,65625$
6.	$-96,8125$ и $53,34375$
7.	$-97,765625$ и $48,640625$
8.	$-66,171875$ и $52,640625$.
9.	$-103,4375$ и $33,390625$
10.	$-64,65625$ и $30,203125$

Задача № 2

Даны представления двух вещественных чисел в формате с плавающей запятой с одинарной точностью.

- Запишите эти числа в виде десятичной дроби (вещественного числа в десятичной системе счисления).
- Выполните сложение чисел в соответствии с правилами формата. Покажите, как результат представляется в формате с плавающей запятой с одинарной точностью.
- Представьте результат в виде десятичной дроби (вещественного числа в десятичной системе счисления).

Вариант	Задание ¹
1.	42B31800 и C1FB8000
2.	42855000 и C214E000
3.	42A96000 и C1AF0000
4.	42953000 и C272A000
5.	4280A800 и C21EA000
6.	42835000 и C2131000
7.	428AA800 и C24CB000
8.	42A45800 и C1A70000
9.	429AD000 и C19D8000
10.	42A74800 и C27C7000

¹ приведена шестнадцатеричная запись, байты записаны от старшего к младшему.

Задача № 3

Дана строка, состоящая из русских букв, цифр и знаков.

- Определите последовательность кодов символов строки в рамках 8-битной кодировки cp1251 (коды приведите в шестнадцатеричном виде).
- Покажите, как полученная последовательность кодов хранится в памяти компьютера, если используется «насколевский» формат (приведите шестнадцатеричные представления байт памяти).
- Покажите, как полученная последовательность кодов хранится в памяти компьютера, если используется формат *нуль-терминированной строки* (приведите шестнадцатеричные представления байт памяти).
- Каков будет размер текстового файла, если сохранить в него заданную строку с использованием *представления UTF-8* и не записывать в начало файла *BOM*?

Указание: При ответе на этот вопрос нет необходимости выписывать саму последовательность байт файла.

Вариант	Задание
1.	Вё5\$шя8\$эвгаг=
2.	Юз3\$лнюккд=
3.	Хтщ,тшерк2!
4.	Ож7#ешсеиц5?
5.	Зз7;Юуцюзфщ3~

6.	Юу@Зжасяв?
7.	Шт@шютбвю?
8.	Есю@вйехр~
9.	Н4;аплятё4.
10.	Жгё,еёгушюя=

Задача № 4

Текстовый файл содержит последовательность байт.

Зная, что текст был сохранен с использованием представления *UTF-8*, определите последовательность кодов Unicode символов текста (каждый код укажите в формате U+xxxx, где xxxx – шестнадцатеричное представление кода). Укажите тест, соответствующий полученной последовательности кодов *Unicode*.

Вариант	Задание
1.	D0 9B D0 BB D1 8D D0 B4 40 57 66 64 71 2E
2.	D0 96 D0 B2 D0 B6 73 25 76 66 64 7A 7E
3.	4D 77 66 72 2C D0 94 D1 86 D0 B4 D0 BF D1 86 3F
4.	4D 67 72 66 25 D0 B5 D0 B3 D1 8D D1 82 21
5.	D0 9F D0 B7 D1 89 D0 B4 3A 73 62 77 68 77 6A 7E
6.	48 77 6D 74 40 D0 B7 D1 80 D1 89 D1 8D 6B 7E
7.	D0 A6 D0 B2 D1 88 6A 3A 69 74 7A 77 71 3D
8.	D0 81 D1 83 D0 B5 7A 3A 6A 74 76 68 71 6D 7E
9.	D0 A1 D0 BB D0 B3 D1 8F 40 62 77 6A 77 D1 8F 3F
10.	54 72 73 6B 23 D0 B5 D0 B9 D0 BE 77 3F

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,
- представление строки;
- последовательность кодов.

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>.

ПРИЛОЖЕНИЯ




ПРИЛОЖЕНИЕ 1. Кодировка ASCII-1967

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL 00	SOH 01	STX 02	ETX 03	EOT 04	ENQ 05	ACK 06	BEL 07	BS 08	HT 09	LF 0A	VT 0B	FF 0C	CR 0D	SO 0E	SI 0F
1	DLE 10	DC1 11	DC2 12	DC3 13	DC4 14	NAK 15	SYN 16	ETB 17	CAN 18	EM 19	SUB 1A	ESC 1B	FS 1C	GS 1D	RS 1E	US 1F
2	SP 20	! 21	" 22	# 23	\$ 24	% 25	& 26	' 27	(28) 29	* 2A	+ 2B	, 2C	- 2D	. 2E	/ 2F
3	0 30	1 31	2 32	3 33	4 34	5 35	6 36	7 37	8 38	9 39	: 3A	; 3B	< 3C	= 3D	> 3E	? 3F
4	@ 40	A 41	B 42	C 43	D 44	E 45	F 46	G 47	H 48	I 49	J 4A	K 4B	L 4C	M 4D	N 4E	O 4F
5	P 50	Q 51	R 52	S 53	T 54	U 55	V 56	W 57	X 58	Y 59	Z 5A	[5B	\ 5C] 5D	^ 5E	_ 5F
6	` 60	a 61	b 62	c 63	d 64	e 65	f 66	g 67	h 68	i 69	j 6A	k 6B	l 6C	m 6D	n 6E	o 6F
7	p 70	q 71	r 72	s 73	t 74	u 75	v 76	w 77	x 78	y 79	z 7A	{ 7B	 7C	} 7D	~ 7E	DEL 7F

ПРИЛОЖЕНИЕ 2. Кодировка ISO 8859-5 (Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	PAD 80	HOP 81	BPH 82	NBH 83	IND 84	NEL 85	SSA 86	ESA 87	HTS 88	HTJ 89	VTs 8A	PLD 8B	PLU 8C	RI 8D	SS2 8E	SS3 8F
9	DCS 90	PU1 91	PU2 92	STS 93	CCH 94	MW 95	SPA 96	EPA 97	SOS 98	SGCI 99	SCI 9A	CSI 9B	ST 9C	OCS 9D	PM 9E	APC 9F
A	NBSP A0	Ё A1	Ђ A2	Ѓ A3	Є A4	S A5	I A6	Ї A7	J A8	Љ A9	Њ AA	Ћ AB	Ќ AC	SHY AD	Ў AE	Џ AF
B	A B0	Б B1	В B2	Г B3	Д B4	Е B5	Ж B6	З B7	И B8	Й B9	К BA	Л BB	М BC	Н BD	О BE	П BF
C	P C0	С C1	Т C2	У C3	Ф C4	Х C5	Ц C6	Ч C7	Ш C8	Щ C9	Ъ CA	Ы CB	Ь CC	Э CD	Ю CE	Я CF
D	a D0	б D1	в D2	г D3	д D4	е D5	ж D6	з D7	и D8	й D9	к DA	л DB	м DC	н DD	о DE	п DF
E	p E0	с E1	т E2	у E3	ф E4	х E5	ц E6	ч E7	ш E8	щ E9	ъ EA	ы EB	ь EC	э ED	ю EE	я EF
F	№ F0	ё F1	ђ F2	ѓ F3	є F4	s F5	i F6	ï F7	j F8	љ F9	њ FA	ћ FB	ќ FC	§ FD	ў FE	џ FF

ПРИЛОЖЕНИЕ 3. Кодовая страница ср866 (MS-DOS Cyrillic)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	А 80	Б 81	В 82	Г 83	Д 84	Е 85	Ж 86	З 87	И 88	Й 89	К 8A	Л 8B	М 8C	Н 8D	О 8E	П 8F
9	Р 90	С 91	Т 92	У 93	Ф 94	Х 95	Ц 96	Ч 97	Ш 98	Щ 99	Ъ 9A	Ы 9B	Ь 9C	Э 9D	Ю 9E	Я 9F
A	a A0	б A1	в A2	г A3	д A4	е A5	ж A6	з A7	и A8	й A9	к AA	л AB	м AC	н AD	о AE	п AF
B	 B0	 B1	 B2	 B3	┐ B4	┌ B5	 B6	π B7	└ B8	 B9	 BA	┐ BB	└ BC	└ BD	└ BE	└ BF
C	└ C0	┐ C1	└ C2	┐ C3	— C4	┐ C5	└ C6	 C7	└ C8	└ C9	└ CA	└ CB	└ CC	= CD	└ CE	└ CF
D	 D0	└ D1	π D2	└ D3	└ D4	└ D5	π D6	 D7	└ D8	└ D9	└ DA	■ DB	■ DC	■ DD	■ DE	■ DF
E	р E0	с E1	т E2	у E3	ф E4	х E5	ц E6	ч E7	ш E8	щ E9	ъ EA	ы EB	ь EC	э ED	ю EE	я EF
F	Ё F0	ё F1	Є F2	є F3	İ F4	ï F5	Ÿ F6	ÿ F7	° F8	· F9	· FA	√ FB	№ FC	¤ FD	■ FE	NBSP FF

ПРИЛОЖЕНИЕ 4. Кодовая страница Windows-1251 (cp1251)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ђ 80	Ѓ 81	Ѕ 82	Ї 83	„ 84	… 85	† 86	‡ 87	€ 88	‰ 89	Љ 8A	‹ 8B	Њ 8C	Ќ 8D	Ќ 8E	Џ 8F
9	ђ 90	‘ 91	’ 92	“ 93	” 94	• 95	– 96	— 97		™ 99	љ 9A	› 9B	њ 9C	ќ 9D	ћ 9E	џ 9F
A	NBSP A0	Ў A1	ў A2	Ј A3	Ѡ A4	Ѓ A5	Ѕ A6	§ A7	Ё A8	© A9	Є AA	« AB	¬ AC	SHY AD	® AE	Ї AF
B	° B0	± B1	І B2	і B3	г B4	μ B5	¶ B6	· B7	ё B8	№ B9	є BA	» BB	ј BC	Ѕ BD	ѕ BE	ї BF
C	А C0	Б C1	В C2	Г C3	Д C4	Е C5	Ж C6	З C7	И C8	Й C9	К CA	Л CB	М CC	Н CD	О CE	П CF
D	Р D0	С D1	Т D2	У D3	Ф D4	Х D5	Ц D6	Ч D7	Ш D8	Щ D9	Ъ DA	Ы DB	Ь DC	Э DD	Ю DE	Я DF
E	а E0	б E1	в E2	г E3	д E4	е E5	ж E6	з E7	и E8	й E9	к EA	л EB	м EC	н ED	о EE	п EF
F	р F0	с F1	т F2	у F3	ф F4	х F5	ц F6	ч F7	ш F8	щ F9	ъ FA	ы FB	ь FC	э FD	ю FE	я FF

ПРИЛОЖЕНИЕ 5. Unicode: блок Controls and Basic Latin

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
001	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
002	SP 0020	! 0021	" 0022	# 0023	\$ 0024	% 0025	& 0026	' 0027	(0028) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
003	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
004	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
005	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C] 005D	^ 005E	_ 005F
006	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
007	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	DEL 007F

ПРИЛОЖЕНИЕ 6. Unicode: блок Cyrillic

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
040	Ё 0400	Ё 0401	Ђ 0402	Ѓ 0403	Є 0404	Ѕ 0405	І 0406	Ї 0407	Ј 0408	Љ 0409	Њ 040A	Ћ 040B	Ќ 040C	И 040D	Ў 040E	Џ 040F
041	А 0410	Б 0411	В 0412	Г 0413	Д 0414	Е 0415	Ж 0416	З 0417	И 0418	Й 0419	К 041A	Л 041B	М 041C	Н 041D	О 041E	П 041F
042	Р 0420	С 0421	Т 0422	У 0423	Ф 0424	Х 0425	Ц 0426	Ч 0427	Ш 0428	Щ 0429	Ъ 042A	Ы 042B	Ь 042C	Э 042D	Ю 042E	Я 042F
043	а 0430	б 0431	в 0432	г 0433	д 0434	е 0435	ж 0436	з 0437	и 0438	й 0439	к 043A	л 043B	м 043C	н 043D	о 043E	п 043F
044	р 0440	с 0441	т 0442	у 0443	ф 0444	х 0445	ц 0446	ч 0447	ш 0448	щ 0449	ъ 044A	ы 044B	ь 044C	э 044D	ю 044E	я 044F
045	ё 0450	ё 0451	ђ 0452	ѓ 0453	є 0454	ѕ 0455	і 0456	ї 0457	ј 0458	љ 0459	њ 045A	ќ 045B	ќ 045C	и 045D	ў 045E	џ 045F
046	Ґ 0460	Ƶ 0461	Ђ 0462	Ѓ 0463	Є 0464	Ѕ 0465	І 0466	Ї 0467	Ј 0468	Љ 0469	Њ 046A	Ћ 046B	Ќ 046C	И 046D	Ў 046E	Џ 046F
047	Ψ 0470	ψ 0471	Θ 0472	θ 0473	Υ 0474	υ 0475	Ϝ 0476	ϝ 0477	Ου 0478	ου 0479	Ο 047A	ο 047B	Ϟ 047C	ϟ 047D	Ϡ 047E	ϡ 047F
048	Ѕ 0480	ѕ 0481	ѐ 0482	ѓ 0483	ѐ 0484	ё 0485	ё 0486	ѐ 0487	ѐ 0488	ѐ 0489	Й 048A	й 048B	Ь 048C	ь 048D	Р 048E	р 048F
049	Ѓ 0490	ѓ 0491	Ѓ 0492	ѓ 0493	Ђ 0494	Ђ 0495	Ж 0496	ж 0497	З 0498	з 0499	К 049A	к 049B	К 049C	к 049D	К 049E	к 049F
04A	К 04A0	к 04A1	Ѓ 04A2	ѓ 04A3	Ѓ 04A4	Ѓ 04A5	Ѓ 04A6	Ѓ 04A7	Ѓ 04A8	Ѓ 04A9	Ѓ 04AA	Ѓ 04AB	Ѓ 04AC	Ѓ 04AD	Ѓ 04AE	Ѓ 04AF

04B	Ү 04B0	ү 04B1	Х 04B2	х 04B3	Ц 04B4	ц 04B5	Ч 04B6	ч 04B7	Ч 04B8	ч 04B9	Һ 04BA	һ 04BB	Ө 04BC	ө 04BD	Ө 04BE	ө 04BF
04C	І 04C0	Ӣ 04C1	ӓ 04C2	Ѓ 04C3	ѓ 04C4	Љ 04C5	л 04C6	Њ 04C7	њ 04C8	Џ 04C9	ң 04CA	Ч 04CB	ч 04CC	М 04CD	м 04CE	І 04CF
04D	Ӑ 04D0	ӑ 04D1	Ӓ 04D2	ӓ 04D3	Ӕ 04D4	ӕ 04D5	Ӗ 04D6	ӗ 04D7	Ә 04D8	ә 04D9	Ӧ 04DA	ӧ 04DB	Ө 04DC	ө 04DD	Ӫ 04DE	ӫ 04DF
04E	З 04E0	з 04E1	Й 04E2	й 04E3	Й 04E4	й 04E5	Ӧ 04E6	ӧ 04E7	Ө 04E8	ө 04E9	Ӧ 04EA	ӧ 04EB	Ө 04EC	ө 04ED	Ӫ 04EE	ӫ 04EF
04F	Ӱ 04F0	ӱ 04F1	Ӳ 04F2	ӳ 04F3	Ӵ 04F4	ӵ 04F5	Ӷ 04F6	ӷ 04F7	Ӹ 04F8	ӹ 04F9	Ӻ 04FA	ӻ 04FB	Ӽ 04FC	ӽ 04FD	Ӿ 04FE	ӿ 04FF

ПРИЛОЖЕНИЕ 7. Кодировка UTF-8

Начало	Конец	Символ	Байт 1	Байт 2	Байт 3	Байт 4
U+0000	U+007F	0xxxxxxx	0xxxxxxx			
U+0080	U+07FF	00000xxx ххуууууу	110xxxxx	10уууууу		
U+0800	U+FFFF	xxxxуууу ууzzzzzz	1110xxxx	10уууууу	10zzzzzz	
U+10000	U+10FFFF	000xxxxуу ууууzzzz zzvvvvvv	11110xxx	10уууууу	10zzzzzz	10vvvvvv