Приложение 1
ПРАКТИКУМ К ГЛАВЕ 2 «ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ В КОМПЬЮТЕРЕ»

Практическая работа к п. 2.1

Пример 2.1. Представьте в виде разложения по степеням основания числа $2466{,}675_{10}$, $1011{,}11_2$.

Решение

Для десятичного числа $2466{,}675_{10}$, $k = 3$, $n = 3$, разложение имеет вид:

$$2466{,}675_{10} = 2 \cdot 10^3 + 4 \cdot 10^2 + 6 \cdot 10^1 + 6 \cdot 10^0 + 6 \cdot 10^{-1} + \\ + 7 \cdot 10^{-2} + 5 \cdot 10^{-3}.$$

Для двоичного числа $1011{,}11_2$, $k = 3$, $n = 2$, разложение имеет вид:

$$1011{,}11_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2}.$$

Обратите внимание! В разложении по степеням надо записывать все элементы в этой системе счисления, которую рассматриваем. Однако мы использовали десятичную запись всех элементов, так как она для человека более привычна.

Значение степени основания указывает на разряд цифры в числе. Основание системы счисления показывает, во сколько раз изменяется значение цифры при переносе её в соседний разряд. Например, в десятичном числе 33 цифра 3, стоящая в 0-й позиции, означает 3 единицы, а цифра 3, стоящая в 1-й позиции, означает 3 десятка, т. е. 30. Таким образом, значения цифр, стоящих в двух соседних позициях, различаются в 10 раз.

Пример 2.2. Определите, во сколько раз различаются значения цифры 6 в десятичном числе 165063,6.

Решение

Записываем над цифрами номера позиций справа налево:

Номер позиции	5	4	3	2	1	0	–1
Цифра	1	6	5	0	6	3	6

При переносе цифры 6 из (−1)-го разряда в 0-й её значение увеличивается в 10 раз. При переносе цифры 6 из 0-го разряда в 1-й её значение увеличивается ещё в 10 раз. Таким образом, значение цифры 6 в 1-м разряде отличается в 100 раз от значения этой же цифры в −1-м разряде.

Рассуждая аналогично, определяем, что значение цифры 6, стоящей в 4-м разряде, в 1000 раз больше значения такой же цифры, стоящей в 1-м разряде.

Пример 2.3. Какое десятичное число соответствует шестнадцатеричному числу $5D8{,}AC1_{16}$?

Решение

Представим число $5D8{,}AC1_{16}$ в виде разложения по степеням числа 16, а затем выполним их сложение с точностью до 3 знаков после запятой:

$$5D8{,}AC1_{16} = 5 \cdot 16^2 + 13 \cdot 16^1 + 8 \cdot 16^0 + 10 \cdot 16^{-1} + 12 \cdot 16^{-2} + 1 \cdot 16^{-3} = 1280 + 208 + 8 + 0{,}625 + 0{,}046875 + 0{,}000244140625 \approx \approx 1496{,}672_{10}.$$

Таким образом: $5D8{,}AC1_{16} \approx 1496{,}672_{10}$.

Пример 2.4. Перевести десятичное число $136{,}4_{10}$ в пятеричную систему счисления.

Решение

1. Перевод целой части числа:
   136/5 = 27 (остаток 1);
   27/5 = 5 (остаток 2);
   5/5 = 1 (остаток 0).
   Деление заканчивается, так как последнее частное меньше 5.
   Результат формируется из последнего частного (1) и остатков в обратном порядке: целая часть равна 1021.
2. Перевод дробной части числа:
   $0{,}4 \cdot 5 = 2{,}0$ (целая часть числа равна 2).
   Умножение заканчивается, так как дробная часть последнего числа равна 0.
   Дробная часть равна 2.
   Ответ: $1021{,}2_5$.

Пример 2.5. Укажите через запятую в порядке возрастания все основания систем счисления, в которых запись числа 51_{10} оканчивается на 3.

Решение

1. Поскольку число в некоторой системе счисления оканчивается на 3, основание искомой системы счисления должно быть больше 3.
2. Обозначим искомое основание системы счисления как X.
3. Обратимся к алгоритму перевода десятичного числа в другую систему счисления. Последняя цифра в записи числа в новой системе счисления получается как остаток от деления исходного числа на основание. Таким образом, задача сводится к тому, чтобы найти все натуральные числа[1)] X, для которых остаток от деления $51/X$ равен 3. Это условие можно записать в виде:

$$(51 - 3) = M \cdot X,$$

где M — результат деления нацело на X числа 51.

4. Перепишем выражение в виде: $48 = M \cdot X$.
5. Как видим, задача сводится к тому, чтобы найти делители числа 48, бóльшие 3.
6. Число 48 делится на 1, 2, 3, 4, 6, 8, 12, 16, 24, 48. Из этих делителей в качестве X подходят числа 4, 6, 8, 12, 16, 24, 48.

Пример 2.6. Вычислите сумму чисел X и Y, если $X = 1010111_2$, $Y = 152_8$. Результат представьте в двоичном виде.

Решение

1. Поскольку ответ надо представить в двоичном виде, наиболее удобно и быстро решить данную задачу путём перевода всех чисел в двоичную систему счисления.
2. Число X уже представлено в двоичной системе счисления. Представим число Y в двоичной системе счисления: $Y = 152_8 = 1101010_2$.
3. Результат сложения чисел X и Y в столбик представлен далее:

Разряд	7	6	5	4	3	2	1	0
Единица переноса		1	1	1	1	1		
Число *X*	+	1	0	1	0	1	1	1
Число *Y*		1	1	0	1	0	1	0
Сумма	1	1	0	0	0	0	0	1

Поясним последовательность сложения.

1. Сначала складываем младшие биты (0-й разряд): $1 + 0 = 1$. Записываем 1 в 0-й разряд.

1) Натуральные числа — это целые положительные числа, бóльшие 0: 1, 2, 3...

2. Складываем биты 1-го разряда: 1 + 1 = 10. Записываем 0 в 1-й разряд и единицу переноса — в старший 2-й разряд.
3. Складываем биты 2-го разряда и единицу переноса 1 + 1 + 0 = 10. Записываем 0 во 2-й разряд и единицу переноса — в старший 3-й разряд.
4. Аналогично проводим сложение в 4-м и 5-м разрядах. Каждый раз записываем единицу переноса.
5. В 6-м (последнем) разряде складываем биты и единицу переноса: 1 + 1 + 1 = 11. Записываем 1 в 6-й разряд и переносим единицу в старший 7-й разряд. На этом сложение заканчивается.

Ответ: $X + Y = 11000001_2$.

Практическая работа к п. 2.2

Пример 2.7. Получите разные представления целочисленного формата положительного числа $+135_{10}$.

Решение

1. Переводим число 135_{10} в двоичное: 10000111_2. Длина двоичного числа равна 8 бит.
2. Поскольку исходное число положительно, для представления его в компьютере можно использовать 1, 2 и 4 байта.
3. В однобайтовом представлении число имеет вид 10000111. При этом знаковый разряд отсутствует.
4. В двухбайтовом представлении число имеет вид

$$0\ 0000000\ 10000111.$$

5. В четырёхбайтовом представлении число имеет вид

$$0\ 0000000\ 00000000\ 00000000\ 10000111.$$

В двух- и четырёхбайтовом представлении крайний левый бит имеет значение 0, что указывает на положительное число.

Пример 2.8. Получите двухбайтовое представление целочисленного формата отрицательного числа -135_{10}, если это возможно.

Решение

1. Переводим число 135_{10} в двоичное: 10000111_2.
2. Так как число отрицательное и длина двоичного кода числа без знака равна 8 бит, данное число попадает в диапазон для представления чисел в двухбайтовом формате.

3. Добавляем слева нули для получения 16-битового кода:

0 0000000 10000111.

4. Получаем обратный код числа:

1 1111111 01111000.

5. Прибавляем 1. Получается число

1 1111111 01111001.

Это дополнительный код.

6. Для проверки складываем противоположные числа:

0 0000000 10000111 и 1 1111111 01111001.

Получаем:

(1) 0 0000000 00000000.

Единицу в старшем (17-м) бите (записана в скобках) отбрасываем.

Для получения дополнительного кода числа можно также пользоваться формулой

$$2^n - |B|,$$

где n — количество разрядов в числе; B — двоичная запись исходного числа, представленная в n-разрядной форме.

Пример 2.9. В двух байтах представлено целое отрицательное число в формате с фиксированной точкой:

1010 1101 0100 1101.

Запишите это число в в десятичной форме.

Решение

1. Прежде всего, представим число –1 в двоичной форме в двухбайтовом целочисленном формате. Число 1 — это

0000 0000 0000 0001.

2. Обратный код числа 1 — это

1111 1111 1111 1110.

3. Прибавляем 1 и получаем дополнительный код числа:

1111 1111 1111 1111.

4. Заданное число — это *дополнительный код* искомого числа. Чтобы получить *обратный код*, надо вычесть из дополнительного кода 1. Операцию вычитания в двоичной заменим сложением с числом –1, записанным в двухбайтовом целочисленном формате. Единицу переполнения (17-й бит), полученную при таком сложении, отбросим:

$$\begin{array}{l} 1010\ 1101\ 0100\ 1101 \\ + \\ 1111\ 1111\ 1111\ 1111 \\ = \\ 1010\ 1101\ 0100\ 1100 \end{array}$$

5. Для получения искомого числа в двоичной форме меняем 1 на 0, а 0 на 1:

$$0101\ 0010\ 1011\ 0011.$$

6. Переводим число в десятичную форму. Получаем: 21 171. Искомое число: –21 171.

Пример 2.10. Получите отрицательное число $-341{,}375_{10}$ в четырёхбайтовом машинном представлении и запишите его в виде шестнадцатеричного числа.

Решение

1. Переводим целую часть числа (без знака) в двоичную форму:

$$341_{10} = 101010101_2.$$

2. Переводим дробную часть числа:

$$0{,}375_{10} = 0{,}011_2.$$

Таким образом:

$$-341{,}375_{10} = -101010101{,}011_2.$$

3. В полученном числе 12 знаков, поэтому добавляем справа ещё 12 нулей.
4. Представляем число в нормализованной форме:

$$-0{,}101010101011000000000000_2 \cdot 10_2^{1001}.$$

В этой записи степень 1001 означает 9. Порядок числа в машинном представлении будет соответствовать числу

$$64 + 9 = 73 = 1001001_2.$$

5. Записываем искомое машинное представление числа:

$$1\ \ 1001001\ \ 101010101011000000000000.$$

В этой записи пробелами отделены части числа: знак, порядок, мантисса.
6. Переведём полученное представление в шестнадцатеричное число. Для этого разобьём его на тетрады и заменим каждую тетраду шестнадцатеричной цифрой:

$$1011\ 0100\ 1010\ 1010\ 1011\ 0000\ 0000\ 0000 = B4AAB000_{16}.$$

Практическая работа к п. 2.3

Пример 2.11. В таблице представлена часть кодовой таблицы ASCII:

Символ	6	8	R	S	D	r	s
Десятичный код	54	56	82	83	68	114	115
Шестнадцатеричный код	36	38	52	53	44	72	73

Каков шестнадцатеричный код символа «d»?

Решение

1. Анализируем приведённые десятичные коды символов. Это позволяет сделать следующие выводы:
   - коды цифр меньше, чем коды букв;
   - коды прописных английских букв меньше, чем коды строчных английских букв;
   - коды соседних цифр и соседних букв различаются на 1.
2. По таблице также определяем, что разность между десятичными кодами прописных и строчных букв составляет 114 – 82 = 32. Следовательно, десятичный код буквы «d» будет равен 68 + 32 = 100.
3. Переводим десятичный код в шестнадцатеричный делением на 16:

   100/16 = 6 (4 в остатке).
4. Записываем шестнадцатеричный код символа «d», состоящий из последнего частного от деления 6 и остатка 4, т. е. 64.
5. Проверяем по таблице ASCII-кодов: ответ правильный.

Пример 2.12. Определите информационный объём в килобайтах 10 страниц текста, записанного в кодировке Unicode, если на странице 40 строк, а в каждой строке по 64 символа.

Решение

1. По условию определяем значения параметров документа: $b = 2$, $p = 10$, $r = 40$, $s = 64$. Вычисляем объём в байтах: $V = 2 \cdot 10 \cdot 40 \cdot 64$ (перемножать числа не обязательно). Для определения информационного объёма в килобайтах полученный результат делим на 2^{10}, так как 1 Кбайт = 210 байт.
2. В числителе выделяем простые сомножители, в том числе и степени числа 2, и сокращаем дробь:

$$(2 \cdot 10 \cdot 400 \cdot 64)/2^{10} = 2 \cdot 2 \cdot 5 \cdot 5 \cdot 2^3 \cdot 2^6/2^{10} = 50 \text{ Кбайт}.$$

Обратите внимание: при решении мы разложили числа 10 и 40 на простейшие множители и выделили степени 2 для того, чтобы можно было их сократить в числителе и знаменателе. Такой подход позволяет избежать ошибок при умножении и делении больших чисел.

Практическая работа к п. 2.4

Пример 2.13. Объём растрового изображения размером 256 × 256 пикселей составляет 32 Кбайт. Каково максимально возможное число цветов в палитре, используемой для создания изображения?

Решение

1. Представим объём памяти, занимаемый растровым изображением в байтах и битах, в виде степеней числа 2:

$$V = 2^5 \cdot 2^{10} = 2^{15} \text{ байт} = 2^{15} \cdot 2^3 = 2^{18} \text{ бит}.$$

2. Вычислим общее количество пикселей в изображении:

$$K = 256 \cdot 256 = 2^8 \cdot 2^8 = 2^{16}.$$

3. Найдём количество бит, используемых для кодирования цвета:

$$n = V \ / \ K = 2^{18} \ / \ 2^{16} = 2^2 = 4 \text{ бита}.$$

С помощью 4 бит можно закодировать 24 = 16 цветов.

Пример 2.14. Изображение размером 210 × 270 мм^2 было отсканировано с разрешением 150 dpi при глубине цвета 8 бит. Определите в наиболее подходящих единицах измерения информационный объём полученного растрового изображения.

Решение

1. Переведём длину и ширину изображения сначала в сантиметры, а затем в дюймы, при условии что 1 дюйм = 2,5 см. Ширина: 210 / 10 / 2,5 = 8,4 дюйма. Длина: 270 / 10 / 2,5 = = 10,8 дюйма.
2. Вычислим число пикселей на 1 квадратный дюйм при заданном разрешении: 150 · 150.
3. Вычислим общее количество пикселей в изображении: 8,4 · 10,8 · 150 · 150.
4. По условию глубина цвета равна 8 бит, т. е. каждая точка кодируется 8 битами, поэтому объём графического изображения в битах будет равен: 8,4 · 10,8 · 150 · 150 · 8.

5. Чтобы перевести объём в килобайты, надо разделить результат на 1024: $(8{,}4 \cdot 10{,}8 \cdot 150 \cdot 150 \cdot 8)/1024 = 15\ 946{,}\ 875$ Кбайт.
6. Если число получилось большое, то следует перевести объём в мегабайты: $15\ 946{,}\ 875\ /\ 1024 \approx 15{,}58$ Мбайт.

Ответ округлён до 2 знаков после запятой в большую сторону.

Пример 2.15. Матрица цифровой камеры мобильного телефона имеет разрешение 1152×864. Какое разрешение будет иметь фотоснимок размером 20×30 см? Будет ли качественной отпечатанная фотография заданного размера? Какой размер должен иметь фотоснимок в сантиметрах, если требуется качественная печать (с разрешением 300 dpi)?

Решение

1. Переведём размер фотоснимка в дюймы: $20\ /\ 2{,}5 = 8$; $30\ /\ 2{,}5 = 12$.
2. Вычислим разрешение в dpi по длине (большей стороне): $1152\ /\ 12 = 96$, по ширине (меньшей стороне): $864\ /\ 8 = 108$. Полученное разрешение ни по ширине, ни по длине снимка не подходит для качественной печати снимка размером 20×30 см^2.
3. Оценим размер качественного отпечатка фотоснимка. По длине он должен быть не более $1152\ /\ 300 = 3{,}84$ дюйма, или 9,6 см, по ширине — не более $864\ /\ 300 = 2{,}88$ дюйма, или 7,2 см.

Практическая работа к п. 2.5

Пример 2.16. Вычислите в мегабайтах с точностью до двух знаков после запятой объём звуковых данных без сжатия, полученных в результате одноканальной записи (моно). Значение сигнала фиксируется 44 032 раза в секунду и записывается 16-битовым кодом. Общая длительность записи 1 минута.

Решение

1. Определяем по условию значения параметров звукозаписи: $D = 44\ 032$ Гц, $T = 60$ с, $N = 16$ бит, $K = 1$. Тогда объём звуковых данных:

$$V = 44\ 032 \cdot 60\ 16 \cdot 1 \text{ (бит).}$$

2. Чтобы перевести полученное значение в байты делим его на 8, а чтобы перевести байты в мегабайты, делим на 2^{20}:

$$V = 44\ 032 \cdot 60 \cdot 16 \cdot 1/(8 \cdot 2^{20}) = 5{,}04 \text{ Мбайт.}$$

Практическая работа к п. 2.7

Пример 2.17. В чемпионате России по футболу в высшей лиге участвуют 16 команд. Каждая команда в течение сезона играет со всеми прочими командами по 2 раза — один раз на своём поле и один раз на поле соперника. В файл вносятся данные матча: дата (день и месяц кодируются отдельно, год не кодируется) и двоичные коды команд участников. Все эти данные кодируются минимально необходимым количеством бит. Кроме того, кодируется количество забитых голов — для результата каждой команды отводится по 4 бита. Для простоты кодирования месяцев будем считать, что футбольный сезон длится все 12 месяцев (хотя фактически это не так). Каков информационный объём файла в байтах после того, как прошла половина сезона — сыграна половина всех матчей?

Решение

1. Определим минимальное количество бит, необходимых для кодирования команды. Так как команд 16, находим степень 2, ближайшую к 16 (или равную). Это будет число $16 = 2^4$. Следовательно, для кодирования команды нужны 4 бита.

2. Определяем количество бит, необходимых для кодирования даты:

Элемент данных	Число значений, подлежащих кодированию	Степень 2, ближайшая к числу значений, но больше него	Число битов, необходимых для кодирования
День месяца	31	$32 = 2^5$	5
Месяц	12	$16 = 2^4$	4

3. Определяем, сколько бит содержит запись результатов одного матча. Для кодирования забитых голов по условию отводится по 4 бита для каждой команды. Всего надо сложить:

- 5 бит (код дня месяца);
- 4 бита (код месяца);
- 4 бита (код одной команды);
- 4 бита (код другой команды);
- 4 бита (код числа голов, забитых одной командой);
- 4 бита (код числа голов, забитых другой командой).

Таким образом, одна запись занимает 25 бит.

4. Определяем, сколько всего матчей играют команды в сезоне. Удобно представить сетку матчей в виде таблицы, как это обычно и делается:

	$K1$	$K2$	...	$K16$
$K1$				
$K2$				
...				
$K16$				

5. Вычислим количество матчей, сыгранных за половину сезона.

В нижней части таблицы указываются матчи 1-й половины сезона, в верхней — 2-й половины сезона. Фоном выделены клетки, которые не заполняются, так как команда сама с собой не играет.

В таблице 16 столбцов и 16 строк с результатами матчей за вычетом закрашенных ячеек — их тоже 16.

Таким образом, всего матчей за сезон $16 \cdot 16 - 16 = 256 - 16 = 240$. За половину сезона играется 120 матчей.

6. Вычислим в битах информационный объем файла с результатами 120 сыгранных матчей:

$$V = 120 \cdot 25 \text{ (бит)}.$$

7. Для перевода в байты надо это число разделить на 8:

$$120 \cdot 25/8 = 15 \cdot 25 = 375 \text{ (байт)}.$$

Пример 2.18. Информационный объём сообщения, содержащего 512 символов некоторого алфавита, закодированных двоичным кодом, составляет 384 байта. Найдите мощность алфавита, с помощью которого записано сообщение.

Решение

1. Определим информационный объём сообщения в битах, учитывая, что 1 байт = 8 бит:

$$384 \cdot 8 \text{ бит} = 3072 \text{ бита}.$$

2. Определим длину кода одного символа в сообщении:

$$3072 / 512 = 6 \text{ бит}.$$

3. Мощность алфавита, каждый символ которого кодируется 6 битами, равна:

$$2^6 = 64 \text{ символа}.$$

Пример 2.19. Для 5 символов заданы их двоичные коды (для некоторых символов из двух бит, для некоторых — из трёх). Эти коды представлены в таблице:

0	1	А	Б	Г
000	001	01	10	110

Определите, какой набор букв закодирован двоичной строкой 110011010001.

Решение

Анализируем код слева направо. Первым символом может быть только Г, так как последовательность начинается с цепочки 110. Следующим символом может быть только А, так как код 01 задан в таблице, а код 010 — нет. Рассуждая дальше, аналогичным образом получаем ГАББ1.

Пример 2.20. Для составления смайликов используются последовательности длиной 3 символа, состоящие из символов «:» — двоеточие, «;» — точка с запятой, «–» — минус, «(» — открывающая скобка, «)» — закрывающая скобка. Сколько различных последовательностей можно составить из указанных символов?

Решение

1. Для кодирования используется алфавит из 5 символов, поэтому мощность алфавита $P = 5$.
2. Коды (последовательности символов алфавита) имеют по условию задачи одинаковую длину, равную 3.
3. По формуле $N = P^K$ определяем число возможных комбинаций: $N = 5^3 = 125$. Следовательно, из указанных символов можно составить 125 различных последовательностей длиной 3 символа.

Примечание: надо понимать, что не все последовательности из 3 символов, оставленные из указанного алфавита, реально используются как смайлики.