

А. С. Есипов

Трудные темы ИНФОРМАТИКИ сдаем ЕГЭ и сессию

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.06(07)
ББК 32.973.26-018.2я7
Е83

Есипов А. С.

Е83 Трудные темы информатики. Сдаем ЕГЭ и сессию. — СПб.: БХВ-Петербург, 2010. — 208 с.: ил. — (ИиИКТ)

ISBN 978-5-9775-0603-8

Рассмотрены темы школьного курса информатики, которые из-за повышенного внимания к программированию и алгоритмизации, преобладающего в экзаменационных вопросах, отходят на второй план: кодирование, представление и обработка информации; системы счисления; булева алгебра и алгебра логики; машинная арифметика и функциональные элементы компьютера. Содержится материал по программированию в Microsoft Office — создание макросов и программ на VB и VBA. Материал подкреплен разобранными примерами и заданиями для самостоятельного решения, обращается внимание на формальные подходы, обеспечивающие наиболее быстрые решения.

Для студентов и преподавателей профильных вузов и учащихся и учителей школ

УДК 681.3.06(07)
ББК 32.973.26-018.2я7

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.10.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 16,77.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

Оглавление

Предисловие	8
Благодарности	9
Глава 1. Информация. Кодирование информации	10
1.1. Общие сведения об информации	10
Информация	10
Информационные системы	11
Свойства информации	13
1.2. Кодирование и единицы информации	14
Языки и алфавиты	14
Двоичный алфавит	15
Двоичное слово. Байт	16
Код ASCII	16
Unicode	17
Единицы информации	19
1.3. Кодирование графики и звука	20
Кодирование растровой графики	20
Кодирование векторной графики	21
Кодирование звука	22
1.4. Задания для самостоятельной работы	24
1.5. Примеры и задачи из архива ЕГЭ	24
Глава 2. Алфавитный и вероятностный подходы	29
2.1. Содержательный и алфавитный подходы	29
Формула Хартли	30
2.2. Вероятностный подход	31
Свойство аддитивности информации	33
2.3. События равновероятны	33
Использование калькулятора	34
2.4. События не равновероятны	40
Формула Шеннона	40
2.5. Общие сведения о сжатии информации	41
Алгоритм Хаффмана	42
Формула Шеннона и формула Хартли	46
Алгоритмы Зива, Лемпеля, Велча (ZLW)	47
2.6. Определение частотных характеристик	48
2.7. Задания для самостоятельной работы	52
2.8. Примеры и задачи из архива ЕГЭ	53
Глава 3. Системы счисления	55
3.1. Общие сведения о системах счисления	55
3.2. Десятичная система	56
Формула разложения числа по степеням основания	56
3.3. Системы счисления в компьютерах	57
Двоичная система счисления	57
Восьмеричная и шестнадцатеричная системы	58

3.4. Перевод чисел из одной системы в другую	59
Перевод с использованием формулы разложения	59
Задания для самостоятельной работы (часть I).....	61
Перевод целых чисел делением на основание новой системы	63
Перевод правильных дробей умножением на основание	64
Поразрядные способы перевода.....	66
Быстрый способ перевода, использующий устный счет	68
Задания для самостоятельной работы (часть II)	69
3.5. Арифметические действия в двоичной системе	69
3.6. Системы счисления в задачах ЕГЭ	71
Глава 4. Машинная арифметика.....	75
4.1. Способы представления чисел в компьютере.....	75
Прямой код.....	76
Обратный код.....	76
Дополнительный код.....	77
Выполнение арифметических операций в кодах	77
Переполнение и машинные нули.....	78
4.2. Формы представления чисел в компьютере.....	80
Естественная форма.....	80
Нормальная форма.....	81
Сравнение диапазонов представления чисел	82
Действия над числами в нормальной форме	82
Задания для самостоятельной работы.....	84
4.3. Примеры использования других систем	85
Двоично-десятичная система.....	85
Троичная система	86
Система остаточных классов	86
Римская система счисления	88
Задания для самостоятельной работы.....	88
Глава 5. Булева алгебра	89
5.1. От логики Древнего мира до наших дней.....	89
Платон и Сократ	89
Аристотель	89
Готфрид Лейбниц	90
Джордж Буль.....	91
5.2. Булева алгебра.....	91
Булевы переменные и функции	91
Булевы функции одного аргумента.....	93
Инверсия (отрицание).....	93
Булевы функции двух аргументов.....	94
Конъюнкция (логическое умножение)	95
Дизъюнкция (логическое сложение).....	95
Инверсия конъюнкции. Функция Шеффера.....	96
Инверсия дизъюнкции. Функция Пирса.....	97
Импликация (следование).....	98
Неравнозначность (сумма по модулю 2)	98
Равнозначность (эквивалентность)	99
Функции запрета.....	99

Существенная и фиктивная зависимость функций от аргументов	99
Алгебры. Сравнение по набору операций	100
Основные законы и тождества булевой алгебры	100
Правила преобразования формул	103
Правило отрицания	103
Правило свертки	104
Правило обобщенного склеивания (теорема русского математика П. С. Порецкого) ..	104
Задания для самостоятельной работы	104
5.3. Канонические формы булевых функций	105
Совершенная дизъюнктивная нормальная форма	105
Задания для самостоятельной работы	108
Совершенная конъюнктивная нормальная форма	109
Основные соотношения	112
Булева алгебра и ЕГЭ	113
Глава 6. Функциональные элементы компьютеров	118
6.1. Построение логических схем. Общие сведения	118
Переход от схемы к формуле функции	119
Минимизация сложности формул	119
Задания для самостоятельной работы	121
Пример построения схемы сумматора	122
6.2. Схемы с памятью	123
Триггеры	123
Регистры и счетчики	124
6.3. Пример построения сложной схемы	125
Дешифраторы	125
Шифраторы	126
6.4. Моделирование логических функций	128
Задания для самостоятельной работы	129
Глава 7. Алгебра логики	132
7.1. Логические высказывания и действия с ними	132
7.2. Логические операции и функции	133
Замечание о записи формул функций	134
Замечание об очередности выполнения операций	135
7.3. Логические операции в программировании	136
Задания для самостоятельной работы	139
7.4. Логические операции над высказываниями	139
Конъюнкция (логическое умножение)	139
Дизъюнкция (логическое сложение)	139
Инверсия (отрицание)	140
Импликация (следование)	140
Эквивалентность (равнозначность)	141
Неравнозначность (сумма по модулю 2)	141
7.5. Формализация высказываний	141
Логические функции и высказывания	141
Примеры формализации высказываний	143
Задания для самостоятельной работы	146
7.6. Решение текстовых логических задач	149
Задания для самостоятельной работы	156
7.7. Алгебра логики в заданиях ЕГЭ	158
Варианты решения одной текстовой задачи	158

Глава 8. Программирование в MS Office	162
8.1. Общие сведения о макросах.....	162
8.2. Создание, изменение, удаление, размещение макросов.....	163
Макрос <i>Таблица_Unicode</i>	163
Макрос <i>Выч_колич_инф</i>	164
Запуск макросов.....	167
Организация запуска макроса нажатием комбинации клавиш.....	167
Запуск макроса нажатием кнопки на панели инструментов.....	167
Редактирование кода макроса.....	168
Удаление макросов.....	168
Размещение кодов макросов.....	168
Создание макроса <i>Информация для всех</i> с помощью макрорекордера.....	168
Запуск макроса на выполнение.....	170
Просмотр кода макроса.....	170
8.3. Примеры макросов.....	171
Макрос <i>Проверка текста на палиндром</i>	171
Первая проверка правильности работы макроса.....	172
Макрос <i>Число пробелов</i>	172
Макрос <i>Один пробел</i>	173
Макрос <i>Латиница Кириллица</i>	174
Макрос <i>CC1</i> — перевод числа из двоичной системы счисления в десятичную.....	175
Макрос <i>CC2</i> — перевод числа из десятичной системы счисления в двоичную.....	175
Макрос <i>ШкольныеОценки</i>	176
Макрос <i>ТекстАсимволамиТекстаВ</i>	176
8.4. Операторы VBA.....	177
Оператор <i>If...Then...Else</i>	177
Оператор <i>Select Case</i>	178
Операторы цикла.....	179
Оператор <i>For...Next</i>	180
Оператор <i>Do Loop</i>	180
Массивы.....	181
Статические массивы.....	181
Динамические массивы.....	181
Ввод массива присваиванием значений.....	182
Ввод массива и функция <i>InputBox</i>	182
Ввод динамического массива.....	182
Использование функции <i>Array</i>	182
Использование случайных чисел.....	183
Приложение. Решения и ответы к заданиям	184
К ГЛАВЕ 1.....	184
Задание 1.1.....	184
Задание 1.2.....	184
Задание 1.3.....	184
Задание 1.4.....	185
К ГЛАВЕ 2.....	185
Задание 2.2.....	185
Задание 2.3.....	185
Задание 2.4.....	185
К ГЛАВЕ 3.....	185
Задание 3.1.....	185
Задание 3.2.....	185

Задание 3.3	186
Дополнительные задачи к заданию 3.3	186
Задание 3.4	186
Задание 3.5	186
Дополнительные задачи к заданиям 3.4 и 3.5	187
Задание 3.6	187
Задание 3.7	187
Задание 3.8	187
Задание 3.9	187
Задание 3.10	188
Задание 3.11	188
Задание 3.12	188
Задание 3.13	188
Задание 3.14	188
Задание 3.15	189
Задание 3.16	189
К ГЛАВЕ 5	189
Задание 5.1	189
Задание 5.2	189
Задание 5.3	189
Задание 5.4	190
Задание 5.5	190
Задание 5.6	190
Задание 5.7	191
Задание 5.8	191
Задание 5.9	191
Задание 5.10	192
Задание 5.11	192
Задание 5.12	192
Задание 5.13	193
Задание 5.14	194
Задание 5.15	194
К ГЛАВЕ 6	194
К ГЛАВЕ 7	195
Задание 7.1	195
Задание 7.2	195
Задание 7.3	195
Задание 7.4	196
Задание 7.5	197
Задание 7.6	197
Задание 7.7	198
Задание 7.8	199
Задание 7.9	200
Задание 7.10	202
Задание 7.11	203
Задание 7.12	203
Задание 7.13	204
Задание 7.14	205
Список литературы	206
Предметный указатель	207

Предисловие

Результаты проведения Единого государственного экзамена (ЕГЭ) по информатике показывают изменение отношения к предмету со стороны как преподавателей, так и учащихся. Пришлось согласиться, что только твердые знания теории выносимых на экзамен вопросов, а также свободное владение эффективными моделями (способами, методами, приемами) решения задач позволяют гарантировать успешную сдачу экзамена.

При обучении и в период подготовки к сдаче экзамена естественно учитывать, что ассортимент заданий в основном известен. Формулировки задач от года к году изменяются несущественно. Если и добавляются новые задачи, то количество их небольшое. Все это позволяет заранее обращать внимание на формальные подходы, обеспечивающие наиболее быстрые решения. Многие из этих подходов рассмотрены в книге достаточно подробно.

Если задачи кодирования решаются без особых затруднений (см. главу 1), то появление задач, связанных с вероятностным подходом к информации (см. главу 2), требуют особого внимания. Умение быстро решать наиболее простые задачи по системам счисления (см. главы 3 и 4) может освободить время для решения более сложных задач.

В книге много внимания уделяется булевой алгебре (см. главу 5) и алгебре логики (см. главу 6). Каждое решение типовой задачи в этих главах желательно брать как инструмент для решения подобных задач. Рене Декарт по этому поводу сказал следующее: "Каждая задача, которую я решал, становилась правилом, служившим впоследствии для решения других задач".

Книга содержит учебный материал с заданиями в различных вариантах:

- во-первых, это *примеры*, в которых рассматриваются и обсуждаются алгоритмы, приемы решения задач. Примеры, как правило, доводятся до ответов;
- во-вторых, это *задания*, где формулируется задача, а решение и ответ, как правило, выносятся в приложение;
- в-третьих, различной постановки *задания*, которые предлагается *решать самостоятельно* и на которые ответы *не даются*. Это задания для тренировки.

Желательно свободно владеть понятиями о таких объектах булевой алгебры, как наборы значений аргументов, таблицы истинности, нормальные и совершенные формы, дизъюнктивные и конъюнктивные формы и т. д. Наиболее часто в заданиях ЕГЭ встречается функция импликация. Полезно запомнить ее код в таблице истинности и ее запись в булевой алгебре.

Материал по решению текстовых логических задач в книге изложен достаточно полно. Рассмотрены различные варианты решений.

Небольшое количество листингов дано в Visual Basic 6.0 (VB) или в Visual Basic for Applications (VBA). Последняя глава содержит материал, связанный с построением макросов. Показано, что для создания программ на VB и VBA и работы с ними достаточно иметь одно из приложений Microsoft Office, например, Word.

В настоящее время Microsoft Visual Basic (VB) — самый популярный в мире инструмент разработки приложений. Более того, знание его основ необходимо для всех программистов, какими бы средствами они ни пользовались. Visual Basic for Applications (VBA) является диалектом VB и практически почти ничем от него не отличается.

Во всемирном рейтинге популярности языков программирования на конец 2009 года первые три места занимают Java, C и Visual Basic (Pascal на 18-м).

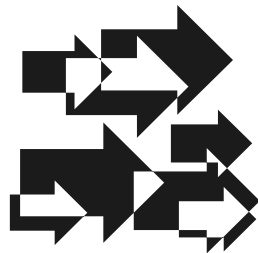
Почетное третье место языка Visual Basic в списке популярности объясняется также тем, что операционные системы от Microsoft работают почти на 97% компьютеров всего мира. За ними следуют ОС Apple — 2,3%, Linux — 0,36% (данные опроса 50 тыс. клиентов из более чем 100 стран мира). Интересно заметить, что из 97% доля ОС Windows XP составляет 86,8%, Windows 2000 — 6,9%, Windows 98 — 2,68%.

В будущем Microsoft VBA — это Visual Studio Tools Applications (VSAA) — основанный на Microsoft .NET инструментарий расширения функциональности. Десять лет назад во всем мире было не более двух миллионов программистов. Сегодня их насчитывается около десяти миллионов, из них не менее 70% используют хотя бы один из инструментов — VB или VBA.

Благодарности

Автор рад возможности выразить искреннюю благодарность Горюновой Марине Александровне — кандидату педагогических наук, доценту, зав. УМЦ дистанционных и мультимедийных технологий в образовании (ЛЮИРО) за большую помощь в апробации материала пособия на занятиях с учителями школ Ленинградской области, и Паньгиной Нине Николаевне — учителю информатики, директору Центра информационных технологий г. Сосновый Бор за доверие и помощь в привлечении автора к занятиям в Ленинградском областном центре одаренных школьников "Интеллект" (ГОУ "Интеллект", пос. Лисий Нос).

ГЛАВА 1



Информация. Кодирование информации

1.1. Общие сведения об информации

Информация

Информацией называют сведения о предметах, процессах и явлениях окружающего нас мира. Это простое определение, которое будет в дальнейшем уточняться, не исчерпывает всю глубину понятия информации.

Информацию человек воспринимает с помощью органов чувств. Зрение, слух, обоняние, вкус и осязание позволяют нам, например, получить информацию о размере, форме, цвете, твердости, аромате и вкусе яблока.

Информацию человек не только воспринимает, но и сам является ее источником с помощью речи, жестов, письма, графики. Сам образ человека несет информацию, по которой его узнают, отличают от других людей.

Источниками информации для человека служат окружающие его предметы и явления. Много информации он получает по телевидению, радио, печати, читая книги, просматривая фильмы, в разговорах с другими людьми.

Со второй половины XX века наблюдается "информационный взрыв". Развитие науки и техники, литературы и искусства привели к значительному увеличению количества информации. Одновременно существенно увеличились темп жизни и цена времени. Человек не в силах одновременно слушать и смотреть сотни и тысячи теле- и радиопередач, читать сотни и тысячи издаваемых ежедневно книг, газет и журналов. Остро встала проблема сортировки и выбора информации, разработки средств передачи и переработки информации. На помощь пришли новейшие средства телекоммуникаций, важными элементами которых служат компьютеры.

В соответствии с увеличившимся значением информации и средств телекоммуникаций, а также углубленным пониманием роли информации и информацион-

ных процессов не только в жизни человечества, но и во вселенском масштабе меняется взгляд на научную картину мира.

Известны два поля дальнего взаимодействия — электромагнитное, осуществляющее взаимодействие между электрически заряженными частицами, и гравитационное (поле тяготения), осуществляющее взаимодействие между массами. В настоящее время ученые к этим полям добавляют *информационное поле*. Современная научная картина мира строится на трех основополагающих понятиях: *вещества, энергии и информации*.

Информацию выдают, воспринимают, используют, перерабатывают все представители животного и растительного мира, все автоматически работающие механизмы, устройства.

Важно отметить, что информационные процессы в живой природе, обществе, в различных механизмах и устройствах имеют много общего и подчиняются одним и тем же законам. Это позволяет рассматривать и изучать их с одинаковых позиций, строить единую для них теорию информации.

Информационные системы

Информация не может существовать без выдающих ее объектов, передаваться без соответствующих объектов, способных ее переносить, и восприниматься без объектов, способных ее принимать, понимать и использовать. Простейшая *информационная система* состоит из источника информации, канала связи и приемника информации (рис. 1.1).



Рис. 1.1. Простейшая информационная система

Примерами информационных систем могут служить: книга и читатель, светофор и водитель (зрительные каналы связи), передающая радиостанция и радиоприемник (канал связи — радиоволны), радиоприемник и слушатель (звуковой канал связи) и т. п.

Информатика тесно связана с кибернетикой. Кибернетика — это наука о процессах управления в машинах, живых организмах, обществе. Она также занимается изучением общих законов получения, хранения, передачи и переработки информации в управляющих системах.

В кибернетике рассматриваются более сложные *информационные управляющие системы*, включающие управляющий объект, канал связи и управляемый объект (рис. 1.2). При этом управляющий и управляемый объекты могут представляться

в виде "черного ящика", что предполагает изучение работы системы независимо от внутреннего содержания и структуры этих объектов. Они могут быть любой природы: механическими, электрическими, биологическими, общественными структурами и т. д.



Рис. 1.2. Информационная управляющая система

Кроме понятия черного ящика в кибернетике используют важное понятие *обратной связи*. Функционирование управляющей системы рассматривается с разомкнутой и замкнутой обратной связью, которая может быть положительной или отрицательной.

Положительная обратная связь усиливает результаты работы управляемого объекта. Как правило, положительная обратная связь приводит к усилению управляющих сигналов, что ведет к перевозбуждению и неустойчивой работе системы. *Отрицательная обратная связь* ослабляет управляющие воздействия на управляемый объект.

Для стабильной, устойчивой, без излишних колебаний работы системы необходимо, чтобы управляющие сигналы в каждый момент времени имели соответствующую величину и знак. Это обеспечивается работой *устройства сравнения*. Оно находится на входе управляющего объекта, сравнивает передаваемые по обратной связи реальные результаты работы управляемого объекта с требуемыми результатами и вырабатывает *сигнал ошибки* соответствующего знака и величины. Это корректирует в нужном направлении управляющие сигналы и работу управляемого объекта.

Пусть управляющий объект — шофер автомашины, управляемый объект — автомашина. Управляющие сигналы — это воздействия шофера на рулевое управление. Обратная связь — зрительная. Качественная работа такой системы может определяться плавным, без колебаний вписыванием автомобиля в виражи дороги. Без обратной связи такая система не работоспособна.

Примером управляющей системы может служить система "учитель — ученик". Управляющие сигналы в такой системе — это информация, передаваемая ученику на учебных занятиях. Качество функционирования системы оценивается уровнем знаний и умений ученика, которые существенно зависят от работы обратной связи — различных видов контроля усвоения учебного материала.

Свойства информации

К свойствам информации можно отнести ее важность (ценность, полезность), достоверность (истинность, правильность), полноту, оперативность (своевременность), понятность (доступность) и некоторые другие качества.

□ Важность.

Полученная в разговоре по телефону информация может быть новой и важной для собеседников или сообщающей известные им сведения. Информация о возможных заморозках имеет различную важность для огородника, шофера и водителя трамвая. В этом проявляется субъективность в оценке информации.

Можно сказать, что цена информации определяется связанными с ней материальными или духовными приобретениями или потерями. Более ценной информации соответствует и больший выигрыш или меньший проигрыш.

□ Достоверность.

Это свойство связано с истинностью и ложностью информации. Известно, что истина объективна по содержанию (не зависит от человека), но субъективна по форме, т. к. является результатом деятельности человека.

Истина относительна. Она отражает предмет не полностью ввиду ограниченности человеческих знаний. В процессе познания человека она стремится к истине абсолютной, полностью исчерпывающей предмет познания.

Например, великий Пифагор считал, что молнии мечет на Землю разгневанный Зевс. Теперь же каждый школьник знаком с электричеством. Долгое время за истину принималась геоцентрическая система Птолемея. В настоящее время истинность, достоверность информации о гелиоцентричности Солнечной системы бесспорна.

□ Полнота.

Полнота информации о предмете, процессе, явлении зависит от ее количества, подробности, всесторонности. Понятие полноты информации о предмете так же субъективно и относительно, как и понятие истины. Информацию даже о простейшем предмете невозможно исчерпать полностью. Всегда можно что-то добавить, уточнить.

□ Оперативность.

Оперативность — это своевременность информации. С этой характеристикой информации мы встречаемся практически ежедневно. Все средства массовой информации стремятся оперативно получать информацию и передавать ее зрителям, слушателям, читателям. Устаревшая информация нас, как правило, мало интересует.

□ Доступность.

Это свойство информации связано с возможностью ее воспринимать, понимать и использовать. Сообщение, передаваемое вам, например, азбукой Морзе, если вы ее не знаете, не несет никакой информации.

Можно заметить, что все перечисленные характеристики информации взаимосвязаны, взаимозависимы. Например, ценность информации определяется ее достоверностью, полнотой и оперативностью.

1.2. Кодирование и единицы информации

Языки и алфавиты

Различают естественные и искусственные (формальные) языки. *Естественные языки* развивались веками и служат для общения людей между собой. *Формальные языки* разрабатываются для специальных применений. Примером формальных языков могут служить языки программирования, языки кодирования информации для ее передачи, хранения и т. п.

Каждый язык имеет свой *алфавит*. Под алфавитом языка понимают набор используемых символов. Под *мощностью алфавита* понимают количество составляющих алфавит символов.

Кодом называют совокупность знаков (символов), предназначенных для представления информации в соответствии с определенными правилами. Такое представление называют *кодированием*. Кодируют информацию с целью ее передачи, хранения, преобразования.

Одно и то же понятие на различных языках может кодироваться различными способами. Например, слово "стол" — это код в русском алфавите всем известного предмета мебели. В других языках, в других алфавитах этот предмет кодируется иначе.

Звук *а*, издаваемый человеком, кодируется в некоторых языках буквой *А*. Буква *А* в азбуке Морзе кодируется так: "• —" (точка, тире). В компьютере буква *А* латинского алфавита в привычной для нас десятичной системе кодируется числом 65. В свою очередь, число 65 в "привычной" для компьютера двоичной системе (цифры только 0 и 1) кодируется так: 01000001.

В естественных языках традиционно сложились некоторые неоднозначности. Например, под словом "коса" понимается и девичья коса, и речная отмель, и инструмент для скашивания травы. Подобные неоднозначности в формальных языках, как правило, недопустимы.

Различны и алфавиты языков кодирования. Это буквы *A, B, C, D, ...* и буквы *А, Б, В, Г, Д, ...* в латинском и русском алфавитах; точка и тире в азбуке Морзе;

арабские цифры 0, 1, 2, 3, ..., 9, с помощью которых записываются числа, кодирующие различные количества; красный, желтый и зеленый цвета в светофоре; цифры 0 и 1 в компьютере и т. п.

Количество и графическое изображение символов в алфавитах естественных языков определяется характерными особенностями языка, историей его развития, традициями. Например, русский алфавит имеет 33 буквы, латинский — 26, итальянский — 21, армянский — 39, арабский — 28, китайский — несколько тысяч иероглифов. Количество и изображение знаков в формальных языках могут также существенно различаться.

Наименьший по числу знаков алфавит имеет только один знак. Пусть этот знак 1 (единица). Тогда три цвета светофора можно закодировать, например, так: красный — 1, желтый — 11, зеленый — 111. Такой алфавит — самый неэкономичный по записи кодов. В этом легко убедиться, если попытаться записать в этом алфавите, например, число десять: 1111111111.

Двоичный алфавит

В информатике и вычислительной технике широко используется алфавит, имеющий два знака — 0 и 1. Этим знакам в логике и технике приводят в соответствие понятия — "нет" и "да", ложь и истина, выключено и включено. Такой алфавит называют *двоичным* или *бинарным*. От бинарного алфавита и пошло название наименьшей целой единицы информации — бит (bit, от англ. *binary* — двоичный и *digit* — знак). Клод Шеннон, американский ученый, много сделал для разработки теории информации. Он определил бит как сообщение, представляющее один из двух вариантов: например, "орел" — "решка", или "да" — "нет".

Одного бита информации достаточно, чтобы закодировать, например, состояние электрической лампочки. Кстати, на некоторых выключателях пишут 1 — включено и 0 — выключено. Взгляд на выключатель снимает для нас неопределенность в его состоянии. При этом мы получаем количество информации равно одному биту.

ПРИМЕР 1.1

Пусть требуется отгадать задуманное число из набора натуральных чисел от 1 до 8, задавая вопрос: "В первой половине находится задуманное число?" и получая в ответ "Да" или "Нет".

Задав первый вопрос и услышав первый ответ, получаем один бит информации. При этом неопределенность уменьшается в два раза (из 8 чисел остается только 4). После второго вопроса получаем еще один бит информации, уменьшающий неопределенность еще в два раза (из 4 чисел осталось только 2). Последний, третий вопрос позволяет отгадать число.

Для отгадывания потребовалось получить три ответа, три бита информации. Столько же бит требуется для оптимального кодирования двоичными кодами каждого из восьми чисел: $b_1b_2b_3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$?

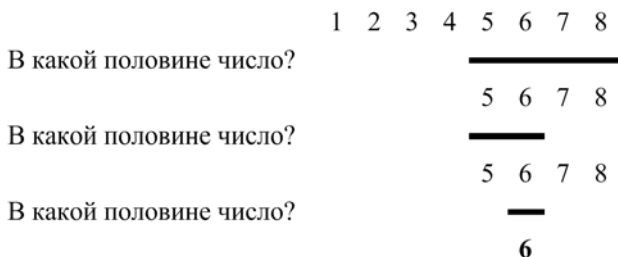


Рис. 1.3. Пример процедуры отгадывания числа

На рис. 1.3 показана последовательность шагов отгадывания числа 6. На вопрос "В какой половине находится число?" дается ответ в виде жирной линии. При решении задачи использован оптимальный по числу попыток прием деления пополам. Метод половинного деления находит широкое применение в математике, информатике и других науках.

Двоичное слово. Байт

Если нужно закодировать в двоичном алфавите красный, желтый и зеленый цвета светофора, то требуются уже два бита. Закодировать три цвета можно, например, так: 00, 01 и 10. Сообщение о том, что включен, например, красный цвет светофора, содержит информации больше одного бита.

Для кодирования четырех сторон света (север, восток, юг и запад) требуются также два бита: 00, 01, 10, 11. Поэтому сообщение о том, какая выбрана сторона света, содержит ровно два бита информации.

При кодировании восьми углов куба потребуются три бита: 000, 001, 010, 011, 100, 101, 110, 111. При кодировании от 9 до 16 объектов потребуются уже четыре бита, от 17 до 32 — 5 бит, от 33 до 64 — 6 бит и т. д.

Последовательность символов называют *словом*. Можно сделать вывод: чем больше требуется закодировать объектов, тем длиннее будет двоичное слово. Восьмибитовое двоичное слово называется *байтом*. С помощью байта можно закодировать 256 различных объектов ($2^8 = 256$).

Код ASCII

До недавнего времени байта было достаточно, чтобы закодировать все символы текста в русском и латинском алфавитах: буквы, цифры, знаки препинания,

управляющие сигналы — все то, что передавалось компьютеру с клавиатуры. Для этого использовался код *ASCII* (American Standard Coding for Information Interchange, американский стандартный код для обмена информацией).

Далее показана табл. 1.1 с примерами кодирования символов в коде ASCII.

Таблица 1.1. Примеры кодирования в ASCII

Символ	Двоичное слово	Десятичный код
Пробел	00100000	32
, (запятая)	00101100	44
. (точка)	00101110	46
...
A	01000001	65
B	01000010	66
C	01000011	67
D	01000100	68
...
A	10000000	128
Б	10000001	129
B	10000010	130
Г	10000011	131
...

Unicode

С развитием информатики байт начал сдерживать возможность увеличения количества используемых символов. В настоящее время завершён переход на двухбайтовое кодирование символов с использованием кода *Unicode*. Этот код используется в двух вариантах. Первый вариант с длиной кода в 31 бит имеет для кодирования символов, слов и команд такой запас кодов, который трудно преодолеть. Второй вариант отводит для кодирования два байта (16 бит), что даёт возможность кодировать до 65 536 объектов. Этот вариант используется в операционных системах и приложениях современных компьютеров.

На рис. 1.4 показана таблица кодирования символов в Unicode. Таблицу выводит на экран макрос, код которого представлен в листинге 1.1. О создании макроса рассказано в разд. 8.2.

Листинг 1.1. Таблица кодирования в Unicode

```
Sub Таблица_Unicode()
    ' Таблица_Unicode Макрос
    ' Макрос создан 02.01.2010 BLACK
    For i = 33 To 255
        Debug.Print " "; i; " "; Chr(i),
        If i Mod 8 = 0 Then Debug.Print
    Next
End Sub
```

Начало таблицы двухбайтового Unicode совпадает с первой половиной таблицы кода ASCII. Кодирование русского алфавита отличается от кодирования в ASCII.

Из таблицы можно получить числа, которыми кодируются в Unicode буквы русского алфавита. Прописные буквы от А до Я, включая букву Й, последовательно кодируются числами от 192 до 223. Буква Ё кодируется числом 168. Строчные буквы от а до я, включая букву й, кодируются последовательно числами от 224 до 255. Буква ё закодирована числом 184.

33	34	"	35	#	36	\$	37	¢	38	€	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	_	96	`
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	{	124		125	}	126	~	127		128	
129		130	,	131		132		133		134		135		136	
137		138		139		140		141		142		143		144	¡
145	¢	146	£	147	¤	148	¥	149	¦	150	§	151	¨	152	©
153	ª	154	«	155	¬	156	­	157	®	158	¯	159	°	160	±
161	²	162	³	163	´	164	µ	165	¶	166	·	167	¸	168	
169	º	170	»	171	¼	172	½	173	¾	174	¿	175		176	
177		178		179		180		181		182		183		184	
185		186		187		188		189		190		191		192	
193		194		195		196		197		198		199		200	
201		202		203		204		205		206		207		208	
209		210		211		212		213		214		215		216	
217		218		219		220		221		222		223		224	
225		226		227		228		229		230		231		232	
233		234		235		236		237		238		239		240	
241		242		243		244		245		246		247		248	
249		250		251		252		253		254		255			

Рис. 1.4. Таблица кодирования символов в Unicode

Единицы информации

На практике используются более объемные, производные единицы количества информации:

- 1 байт = 8 бит = 2^3 бит;
 - 1 килобайт (Кбайт) = 2^{10} байтов = 1024 байтов = $2^{10} \cdot 2^3$ бит = 2^{13} бит;
 - 1 мегабайт (Мбайт) = 2^{10} килобайтов = 1024 килобайтов = 2^{23} бит;
 - 1 гигабайт (Гбайт) = 2^{10} мегабайтов = 1024 мегабайтов = 2^{33} бит;
 - 1 терабайт (Тбайт) = 2^{10} гигабайтов = 1024 гигабайтов = 2^{43} бит
- и т. д.

ПРИМЕР 1.2

При работе с текстовым редактором Microsoft Word командой **Сервис | Статистика** можно вывести на экран диалоговое окно **Статистика** (рис. 1.5). В окне отображаются характеристики документа, загруженного в рабочее поле редактора.



Рис. 1.5. Окно Статистика

Подсчитаем объем памяти, требуемый для записи и хранения в памяти компьютера текстового документа с показанными в окне данными.

Всего знаков с пробелами 30 496. В коде ASCII требуется память 30 496 байтов или $30496/1024 = 29,78125 \approx 30$ Кбайт.

В Unicode, где каждый знак кодируется двумя байтами, потребуется память в два раза большего объема.

1.3. Кодирование графики и звука

С помощью компьютера рисуют, сочиняют музыку, компьютер понимает человеческий голос и сам может отвечать пользователю. Так как вся информация в компьютере обрабатывается и сохраняется в цифровом виде, то и графическая и звуковая информация должна быть представлена в цифровой форме. С этой целью ее кодируют числами. Различают *растровую* и *векторную* графику.

Кодирование растровой графики

Весь экран дисплея делится на точки — пиксели (pixel, от англ. *picture element* — элемент рисунка), подсвечивание которых создает видимое отображение текста и рисунков (рис. 1.6).

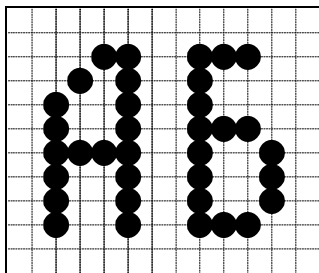


Рис. 1.6. Растровый способ отображения графики

При работе с растровыми изображениями приходится иметь дело со всеми точками рисунка. Например, если цветная фотография, полученная с помощью цифровой фотокамеры, имеет размеры 1600×1200 точек, то в памяти компьютера необходимо хранить информацию (цвет, яркость) о каждой из N ($N = 1600 \times 1200$) точек фотографии.

Информационный объем растрового изображения (Q) определяется как произведение числа входящих в изображение точек (N) на информационный объем одной точки (q), который зависит от количества возможных цветов, т. е. $Q = N \times q$.

При черно-белом изображении $q = 1$ бит (например, 1 — точка подсвечивается и 0 — точка не подсвечивается). Поэтому для хранения черно-белого (без оттенков) изображения размером 100×100 точек требуется 10 000 бит. Если между черным и белым цветами имеется еще шесть оттенков серого (всего 8), то информационный объем точки равен 3 бита ($\log_2 8 = 3$). Информационный объем такого изображения увеличивается в три раза: $Q = 30\,000$ бит.

Цветное изображение получается за счет различной яркости трех основных цветов — красного, синего и зеленого. Цветные изображения могут отображаться

в различных режимах, соответственно изменяется и информационный объем точки (табл. 1.2).

Таблица 1.2. Информационный объем точки

Режим	Информационный объем точки
2 цвета	$q = 1$ бит, черно-белое изображение
16 цветов	$q = \log_2 16 = 4$ бита
256 цветов	$q = \log_2 256 = 8$ бит = 1 байт
65 536 цветов	$q = \log_2 65536 = 16$ бит = 2 байта
16 777 216 цветов	$q = \log_2 16777216 = 24$ бит = 3 байта

Умножение информационного объема точки на количество точек дает значительную величину. Поэтому актуальной становится задача расчета объема памяти (видеопамяти), необходимой для хранения изображения (битовой карты) со всего экрана монитора. Рассмотрим два примера.

ПРИМЕР 1.3

При размерах экрана (разрешении) 640×480 точек и количестве цветов 256 ($q = 1$ байт, см. табл. 1.1) объем видеопамяти должен быть не менее $Q = 640 \times 480 \times 1 = 307200 = 300 \times 1024$ байт ≈ 300 Кбайт.

ПРИМЕР 1.4

Для экрана размером 1280×1024 точки и количестве цветов 16 777 216 ($q = 3$) имеем $Q = 1280 \times 1024 \times 3 = 3\,932\,160$ байт, $Q = 3\,932\,160 / 1024 = 3840$ Кбайт, $Q = 3840 / 1024 = 3,75$ Мбайт.

Современные компьютеры имеют объем видеопамяти 8, 16 и более мегабайт, что позволяет хранить несколько графических изображений размером во весь экран.

Кодирование векторной графики

Векторное изображение состоит из набора элементарных деталей — графических примитивов. Это линии, прямоугольники, окружности, дуги и т. п. Положение этих элементов на экране определяются координатами точек. Например, отрезок прямой задается координатами концов, окружность — координатами центра и радиусом и т. д. Кроме координат задаются цвет отображаемых деталей, толщина линий и другие характеристики.

Информация о векторном рисунке кодируется обычным способом, как хранятся тексты, формулы, числа, т. е. хранится не графическое изображение, а только

координаты и характеристики изображения его деталей. Поэтому для хранения векторных изображений требуется существенно меньше памяти, чем растровых изображений.

При запуске программы с векторным рисунком он создается каждый раз вновь, после чего в растровом виде может сохраняться в видеопамяти. При просмотре графических изображений они считываются компьютером из видеопамяти на экран дисплея с частотой 50—60 и более раз в секунду.

Кодирование звука

Человек слышит звук с частотами от 16 Гц (колебаний в секунду) до 20 кГц. До недавнего времени звуковые сигналы передавались, записывались на граммофонных пластинках и магнитных лентах и воспроизводились в виде амплитудных колебаний. При передаче радио- и телевизионных программ использовалась амплитудная или частотная модуляция так называемой несущей частоты радиосигнала низкочастотными звуковыми сигналами. На приемных устройствах звуковые сигналы усиливались и подавались на репродукторы и динамики. В процессе передачи по каналам связи на звуковые сигналы могут накладываться различной природы внешние электромагнитные воздействия, которые могут существенно эти сигналы исказить.

Значительное увеличение быстродействия компьютеров и других цифровых устройств, увеличение памяти компьютеров позволили перейти на цифровую передачу и воспроизведение аудио- и телевизионных передач, сохранять их в памяти компьютера, хранить на лазерных CD- и DVD-дисках. При этом существенно повышается надежность и качество воспроизведения.

Процесс перевода звуковых сигналов от непрерывной формы представления к дискретной, цифровой форме называют *оцифровкой*. Оцифровка звуковых сигналов заключается в их временной дискретизации и амплитудном квантовании (рис. 1.7).

Временная дискретизация состоит в том, что амплитуда звуковых сигналов измеряется только через небольшие равные промежутки времени (дискреты). Величина этих промежутков выбирается из соображений наименьших допустимых искажений звука.

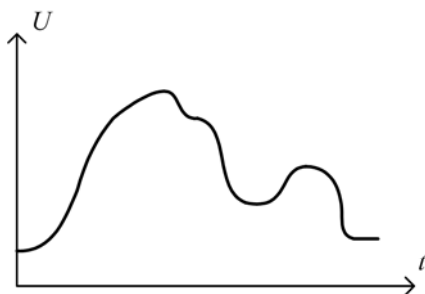
Амплитудное квантование заключается в определении в цифровом виде значений амплитуды звукового сигнала в определяемые временной дискретизацией моменты времени. При этом, ввиду ограниченной точности также возможны искажения звука, которые минимизируются выбором шага квантования амплитуды.

Преобразование сигналов из непрерывной (аналоговой) формы в цифровую выполняют аналого-цифровые преобразователи (АЦП). Преобразование сигналов

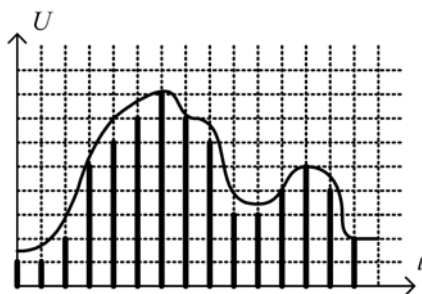
из цифровой формы в аналоговую выполняют устройства, называемые цифро-аналоговыми преобразователями (ЦАП). Например, место ЦАП и АЦП в цепочке преобразований и передаче сигнала может быть следующим:

микрофон — АЦП — компьютер — канал связи — компьютер —
ЦАП — звуковые колонки.

а) Исходный сигнал в аналоговой форме на входе АЦП



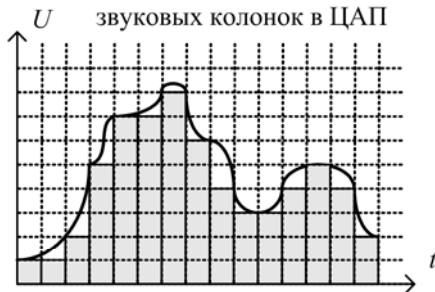
б) Дискретизация времени и квантование амплитуды в АЦП



в) Этап формирования сигнала в ЦАП



г) Формирование сигнала для звуковых колонок в ЦАП



д) Аналоговый сигнал на выходе ЦАП



Рис. 1.7. Процесс оцифровки звуковых сигналов

1.4. Задания для самостоятельной работы

1. Придумать примеры: а) информации, имеющей различную ценность для различных людей, что подтверждает субъективность в оценках важности информации; б) информации, ценность которой существенно зависит от ее оперативности; в) информации, ценность которой зависит от ее полноты.
2. Определить число бит (длину двоичного слова), потребное для кодирования в двоичном алфавите: а) всех пальцев на руках и ногах; б) всех квартир 150-квартирного дома.
3. Сколько бит информации нужно получить, чтобы отгадать загаданное число из 16 последовательно расположенных чисел натурального ряда, из 32 чисел.
4. Выбрать любую книгу и подсчитать потребный для ее записи объем памяти.

1.5. Примеры и задачи из архива ЕГЭ

ПРИМЕР 1.5

Считая, что каждый символ кодируется в Unicode двумя байтами, определите, чему равен информационный объем следующего высказывания Исаака Ньютона: "При изучении наук примеры полезнее, чем правила".

Решение

Высказывание содержит, включая знаки препинания и пробелы, 47 символов. При кодировании каждого символа двумя байтами получаем, что информационный объем высказывания равен: $47 \times 2 \text{ байта} = 47 \times 16 \text{ бит} = 752 \text{ бита}$.

Ответ: 752 бита.

Следующие три примера — на "путь" (S), "время" (t) и "скорость" (V), под которыми соответственно понимаются информационный объем сообщения, время передачи сообщения и скорость передачи данных. Используя знакомые обозначения, будем применять и известные формулы: $S = V \times t$, $V = S/t$, $t = S/V$.

ПРИМЕР 1.6

При скорости модема 56 Кбит/с передача текстового файла заняла 12 секунд. Требуется определить, сколько символов содержал переданный текст, если известно, что он закодирован в Unicode.

Решение

Известны скорость (V) и время (t). Определим информационный объем файла в битах: $S = V \times t = 56 \times 1024 \times 12 = 688\,128 \text{ бит}$. В Unicode каждый символ кодирует-

ся 16-битовым словом. Определим количество символов в тексте:
 $S/16 = 688128/16 = 43\ 008$ символов.

Ответ: 43 008 символов.

ПРИМЕР 1.7

Скорость передачи данных через ADSL-соединение равна 5 120 000 бит/с. Через данное соединение передают файл размером 5000 Кбайт. Определите время передачи файла в секундах.

Решение

Известны скорость (V) и размер файла (S). Заданную скорость удобно представить в единицах Кбайт/с: $V = (5120000/1024)/8 = 625$. Определим время передачи $t = S/V = 5000/625 = 8$ секунд.

Ответ: 8 секунд.

ПРИМЕР 1.8

Текстовый файл объемом 8000 байт передан за 20 секунд. Скорость передачи при этом составляла 200 символов в секунду. Требуется определить, сколько бит на один символ отведено при кодировании передаваемого сообщения.

Решение

Определим, сколько символов в заданном текстовом файле: $S = V \times t = 200 \times 20 = 4000$ символов. Число бит на символ получим при делении объема файла в битах на количество символов: $8000 \times 8 / 4000 = 16$ бит. Полученный ответ позволяет предположить, что текст файла закодирован в Unicode — 2 байта на символ.

ПРИМЕР 1.9

Обычный дорожный светофор без дополнительных секций подает шесть видов сигналов (непрерывные красный, желтый и зеленый, мигающие желтый и зеленый, красный и желтый одновременно). Электронное устройство управления светофором последовательно воспроизводит записанные сигналы. Подряд записано 100 сигналов светофора. Сколько байт для этого потребуется?

Решение

Для кодирования шести вариантов сигналов требуется три бита ($2^2 = 4$ — мало, $2^3 = 8$ — достаточно). 100 сигналов в сумме потребуют 300 бит памяти. Один байт равен 8-ми битам. Поэтому $300/8 = 36,5$ байта, что с небольшим избытком в 0,5 бита равно 37 байт.

Ответ: 37 байт.

ПРИМЕР 1.10

Для хранения растрового изображения размером 128×128 пикселей отвели 4 Кбайт памяти. Каково максимально возможное число цветов в палитре изображения?

Решение

Всего количество пикселей в изображении равно: $128 \times 128 = 2^7 \times 2^7 = 2^{14}$. Все свыше числа пикселей отводятся для обеспечения цветности: $4 \text{ Кбайт} = 2^2 \times 2^{10} \times 2^3 = 2^{14} \times 2^1$. Получили, что для хранения палитры цветов остался один бит — два цвета.

Ответ: 2 цвета.

ПРИМЕР 1.11

Сколько секунд потребуется модему, передающему сообщения со скоростью 28 800 бит/с, чтобы передать цветное растровое изображение размером 640×480 пикселей, при условии, что цвет каждого пиксела кодируется тремя байтами?

Решение

Определим информационный объем сообщения: $S = 640 \times 480 \times 3 \times 8 = 7\,372\,800$ бит. Для передачи требуется $t = S/V = 7372800/28800 = 256$ секунд.

Ответ: 256 секунд.

ПРИМЕР 1.12

Укажите минимальный объем памяти (в килобайтах), достаточный для хранения любого растрового изображения размером 64×64 пиксела, если известно, что в изображении используется палитра из 256 цветов. Саму палитру хранить не нужно.

1) 128; 2) 2; 3) 256; 4) 4.

Решение

Всего пикселей: $64 \times 64 = 2^6 \times 2^6 = 2^{12}$. Для задания цвета в одном пикселе требуется: 8 бит = 1 байт ($2^8 = 256$ цветов). Всего требуется памяти: $2^{12} \times 1 \text{ байт} = 2^2 \times 2^{10} \times 1 \text{ байт} = 4 \times 1024 \text{ байта} = 4 \text{ Кбайт}$.

Ответ: 4 Кбайт.

ПРИМЕР 1.13

Для составления цепочек используются бусины, помеченные буквами A, B, C, D и E . На первом месте в цепочке стоит одна из бусин A, C, E . На втором — любая

гласная, если первая буква согласная, и любая согласная, если первая буква гласная. На третьем месте — одна из бусин C, D, E , не стоящая в цепочке на первом месте.

Какая из перечисленных цепочек создана по этому правилу?

- 1) CBE ; 2) ADD ; 3) ECE ; 4) EAD .

Решение

Переборы не гарантируют быстрое получение правильного решения. Простой и наглядный подход — вспомогательная таблица. Строки таблицы — заданные варианты решения, столбцы — требования к бусинам цепочки.

Варианты ответа	1) A, C, E	2) Гласная, если 1-я согласная; согласная, если 1-я гласная	3) C, D, E , не стоит на 1-м месте	Решение
CBE	+	–	–	Нет
ADD	+	+	+	Да
ECE	+	+	–	Нет
EAD	+	–	+	Нет

Заполняется таблица по столбцам. Если ответ удовлетворяет условию построения цепочки, то ставится плюс, если не удовлетворяет — минус. После заполнения таблицы просматриваются строки. Из таблицы видно, что только цепочка ADD (вторая строка) удовлетворяет всем трем правилам.

Ответ: ADD .

ЗАДАНИЕ 1.1

Для составления цепочек используются бусины, помеченные буквами M, N, O, P, S . В середине цепочки стоит одна из бусин: M, O, S . На третьем месте — любая гласная, если первая буква согласная, и любая согласная, если первая буква гласная. На первом месте — одна из бусин O, P, S , не стоящая в середине цепочки.

Какая из перечисленных цепочек создана по этому правилу?

- 1) SMP ; 2) MSO ; 3) SNO ; 4) OSN .

Заполнение и анализ таблицы см. в примере 1.13.

ЗАДАНИЕ 1.2

Для составления цепочек разрешается использовать бусины 5 типов, обозначаемые буквами $A, B, B, И$. Каждая цепочка должна состоять из трех бусин, при этом необходимо соблюдать следующие правила:

1. На первом месте стоит одна из букв: А, Е, И.
2. После гласной буквы в цепочке не может стоять гласная, а после согласной — согласная.
3. Последней буквой не может быть А.

Какая из цепочек построена по этим правилам?

- 1) АИБ; 2) ЕБА; 3) БИВ; 4) ИБИ.

ЗАДАНИЕ 1.3

При переводе в дискретную форму аналогового сигнала длительностью 2 минуты 8 секунд использовались частота дискретизации $V = 32$ Гц и 16 уровней дискретизации. Размер полученного кода равен:

- 1) 512 байт; 2) 8 Кбайт; 3) 2 Кбайт; 4) 128 байт.

ЗАДАНИЕ 1.4

В кодовой таблице ASCII символы — заглавные буквы латинского алфавита — располагаются один за другим, в алфавитном порядке.

Латинский алфавит (для справки):

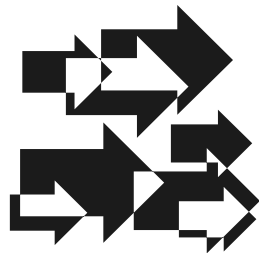
ABCDEFGHIJKLMNOPQRSTUVWXYZ.

Символ "А" имеет код 41_{16} .

Какое слово, закодированное последовательностью: $42_{16}, 58_{16}, 54_{16}, 45_{16}$?

Варианты ответов: 1) BSNE; 2) NYTO; 3) BSNE; 4) BYTE.

ГЛАВА 2



Алфавитный и вероятностный подходы

Приведенное в начале *главы 1* определение далеко не исчерпывает понятие информации, которое используется практически во всех отраслях науки и техники. В числе первых, кто занимался вопросами теории кодирования и передачи сообщений, был американский ученый Клод Шеннон. Он показал, что информацию можно измерять. Шеннону принадлежит следующее определение информации:

Информация — это снятая неопределенность.

2.1. Содержательный и алфавитный подходы

Величина неопределенности, снимаемой некоторым сообщением, и представляет собой содержащееся в сообщении количество информации.

Такой подход к определению информации называют *содержательным*. Он различает сообщения по той пользе, которую сообщение приносит его получателю. При этом количество информации в сообщении зависит от его содержания. Содержательный подход делит сообщения на информативные и сообщения, не несущие новой, полезной информации. Если в сообщении отсутствует новая, полезная, снимающая неопределенность информация, то такое сообщение несет количество информации, равное нулю. Содержательный подход можно еще назвать *субъективным*, при нем трудно организовать формальную, общую для всех получателей систему определения количества информации в сообщениях.

Алфавитный подход к определению информации, в отличие от содержательного подхода, абстрагируется от содержания сообщений. Сторонник этого подхода российский ученый А. Н. Колмогоров рассматривает сообщение как слово, записанное символами некоторого алфавита. При этом количество информации в

сообщении определяется не его содержанием, а количеством битов, потребных для кодирования данного сообщения. Алфавитный подход хорошо формализуется, его также называют *объективным*.

Формула Хартли

Было показано, что сообщение о состоянии выключателя несет один бит информации ($2 = 2^1$, $\log_2 2 = 1$). Сообщение о выборе одной из четырех сторон света несет два бита информации ($4 = 2^2$, $\log_2 4 = 2$). Сообщение о выборе одного из восьми углов куба связано с получением информации объемом в три бита ($8 = 2^3$, $\log_2 8 = 3$), и т. д.

Рассмотренные примеры показывают, что для кодирования в двоичном алфавите N альтернатив (объектов, событий, сообщений, состояний и т. п.) требуется $I = \log_2 N$ бит информации. Можно сделать вывод: чем больше требуется закодировать объектов, тем длиннее требуется двоичное слово. Его длина определяется по *формуле Хартли*, названной по имени предложившего ее американского инженера:

$$I = \log_2 N. \quad (2.1)$$

Эта формула определяет количество информации (I), которое имеем в случае получения одного из N *равновероятных* сообщений.

Напомним, что *логарифмом* называют показатель степени (I), в которую нужно возвести основание логарифма (в нашем случае — 2), чтобы получить заданное число (N). При $I = \log_2 N$ получаем $N = 2^I$.

Обозначим p — вероятность того, что произошло одно из N равновероятных событий. Так как $p = 1/N$, то $N = 1/p$, и формулу Хартли можно записать так:

$$I = \log_2 1/p. \quad (2.2)$$

Для определения целых значений величины I , как функции от N , можно воспользоваться следующей табл. 2.1.

Таблица 2.1. $I = f(N)$

N (сообщений)	2	4	8	16	32	64	128	256	512	1024
I (бит)	1	2	3	4	5	6	7	8	9	10