

ТЕОРИЯ ИНФОРМАЦИИ

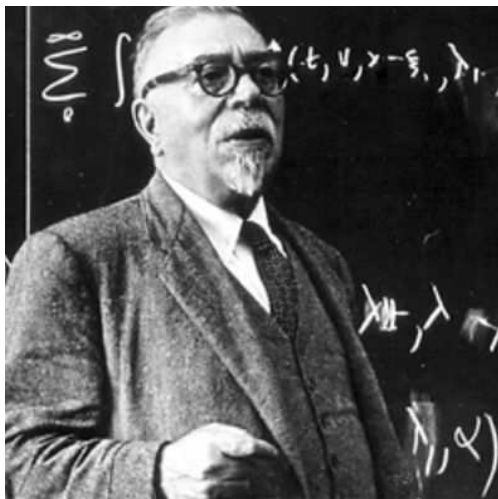
Литература

1. Лидовский В. В. Теория информации: Учебное пособие. — М.: Компания Спутник+, 2004. — 111 с.
http://globalteka.ru/books/doc_details/9415----.html
2. Яглом А., Яглом И. Вероятность и информация — М.: Наука 1973.

Основные понятия теории информации

Информация (от лат. informatio, разъяснение, изложение, осведомленность) — сведения о чем-либо, независимо от формы их представления.

«информация — это обозначение содержания, полученное нами из внешнего мира в процессе приспособления к нему нас и наших чувств» — Норберт Винер



Норберт Винер (*Norbert Wiener*) американский учёный, выдающийся математик и философ, основоположник кибернетики и теории искусственного интеллекта. 1894-1964

Теория информации — комплексная, в основном математическая теория, включающая в себя описание и оценки методов извлечения, передачи, хранения и классификации информации.

Получение, создание, сбор, обработка, накопление, хранение, поиск, распространение и использование информации называется *информационным процессом*.

Математическое представление информационного процесса называется *математической моделью*.

Для извлечения, передачи, хранения и классификации информации требуется формальное представление информации, при этом, каждому описываемому объекту или понятию ставится в соответствие некоторый *числовой код*.

Формализованный вид информации, представленный математическими величинами (числовым кодом), называют *данными*.

Данные, являющиеся результатом фиксации некоторой информации, сами могут выступать как источник информации. Информация, извлекаемая из данных, может подвергаться обработке, и результаты обработки фиксируются в виде новых данных.

Форма представления информации, имеющая признаки начала и конца, предназначенная для передачи через среду связи называется *сообщением*. Сообщением также является форма предоставления информации как совокупность первичных сигналов или знаков, содержащих информацию.

Знак (символ) представляет собой соглашение (явное или неявное) о приписывании чему-либо (означающему) какого-либо определённого смысла. Знаком также называют конкретный случай использования такого соглашения для передачи информации. Знак может быть *составным*, то есть состоять из нескольких других знаков. Например, цифры являются знаками чисел. Буквы являются знаками звуков и, вместе со словами, являются знаками человеческого языка.

Материальным носителем информации, используемым для передачи сообщений, является *сигнал*.

Сигнал может генерироваться, но его приём не обязателен, в отличие от сообщения, которое должно быть принято принимающей стороной, иначе оно не является сообщением. Сигналом может быть любой физический процесс, параметры которого изменяются в соответствии с передаваемым сообщением.

Теория информации оперирует с математическими моделями. Рассматривает носители информации как элементы абстрактного (математического) множества, а взаимодействия между носителями как способ расположения элементов в этом множестве. Такой подход дает возможность формально описать код информации, то есть определить абстрактный код и исследовать его математическими методами. Для этих исследований применяет методы теории вероятностей, математической статистики, линейной алгебры, теории игр и других математических теорий.

Основные разделы теории информации — кодирование источника (сжимающее или экономное кодирование) и канальное (помехоустойчивое) кодирование. *Кодированием* называется процесс преобразования сообщения в комбинацию символов в соответствии с кодом. Процесс восстановления сообщения из комбинации символов называется *декодированием*.

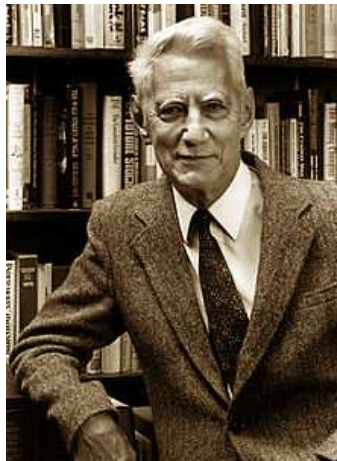
Теория информации тесно связана с криптографией и другими смежными дисциплинами. *Криптография* — наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства, а также невозможности отказа от авторства) информации.

Основы этой теории заложил американский учёный Р. Хартли в 1928 г., который определил меру количества информации для некоторых задач связи. Позднее теория была существенно развита американским учёным К. Шенноном, российскими учёными А.Н. Колмогоровым, В.М Глушковым и др.

Современная теория информации включает в себя как разделы *теорию кодирования, теорию алгоритмов, теорию цифровых автоматов, криптографию* и *криптоанализ, теорию оценок, теорию обнаружения* и некоторые другие.



Ральф Винтон Лайон Хартли (*Ralph Vinton Lyon Hartley*) американский учёный-электронщик, 1888-1970.



Клод Элвуд Шеннон (*Claude Elwood Shannon*) американский инженер и математик 1916-2001



Андрей Николаевич Колмогоров советский математик, один из основоположников современной теории вероятностей 1903-1987



Виктор Михайлович Глушков, выдающийся советский математик и кибернетик **1923-1982**

Альтернативная теория информации:

"Качественная теория информации", предложенная польским учёным М. Мазуром (в противовес «Количественной теории информации» которой по сути является теория, разработанная К. Шенноном)



Мариан Мазур, ученый, польский специалист по теории информации, участвующих в elektrotermia и кибернетики, создатель польской школы кибер-преступников. 1909-1983.

Виды информации

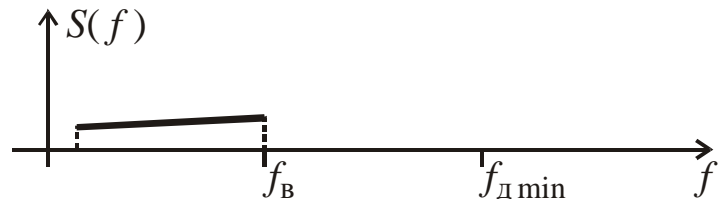


Дискретная информация характеризуется последовательными точными значениями некоторой величины.

Непрерывная информация характеризуется непрерывным процессом изменения некоторой величины.

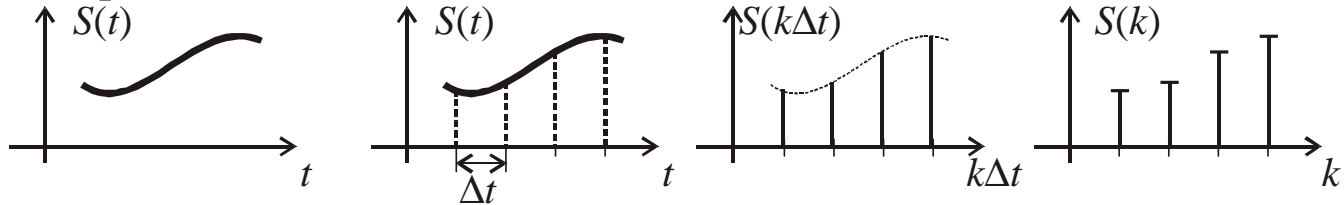
Теорема Котельникова: частота дискретизации для непрерывного процесса должна быть не менее чем в два раза выше верхней частоты спектра непрерывного процесса.

$$f_{\text{д}} = \frac{1}{\Delta t} \geq 2 f_{\text{в}}$$



Δt — интервал дискретизации.

Таким образом, из непрерывного сигнала можно получать его дискретный аналог.



Аппроксимация по Котельникову позволяет осуществлять обратное действие. Однако, в этом случае, практически всегда получается приближённый непрерывный процесс, если сравнивать его с исходным непрерывным процессом до дискретизации.

Спектр непрерывного процесса можно получить с помощью преобразования Фурье от временной зависимости, описывающей непрерывный процесс.

$$S(j\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

Формальное представление информации

Для перевода информации в формальный, цифровой вид используются специальные таблицы, сопоставляющие кодируемым сущностям их коды и называемые *таблицами кодировки (кодowymi страницами, character set)*. Такой процесс сопоставления называют *примитивным кодированием*.

Пример: ACSII (American Standard Code for Information Interchange — Американский стандартный код для обмена информацией)

| Код | | Полное имя кода в Unicode (краткое имя в ASCII) |
|--|---------------------|--|
| 10-й | 16-й Клавиатурный | |
| <i>Перевод имени кода</i> — описание использования кода. | | |

| | | | | |
|----|----|----|-----------------------------|---|
| 0 | 00 | ~0 | NULL (NUL) | <i>Пусто</i> — этот код используется как завершающий в представлении строк многими системами программирования, например, Си, поэтому его использование в текстовых файлах крайне нежелательно. |
| 1 | 01 | ^A | START OF HEADING (SOH) | <i>Начало заголовка</i> — практически не используется. |
| 2 | 02 | ^B | START OF TEXT (STX) | <i>Начало текста</i> — практически не используется. |
| 3 | 03 | ^C | END OF TEXT (ETX) | <i>Конец текста</i> — в Unix и MS-DOS ввод этого символа с клавиатуры служит сигналом для прекращения выполнения программы. |
| 4 | 04 | ^D | END OF TRANSMISSION (EOT) | <i>Конец передачи</i> — в Unix и PostScript означает конец вводимых данных. |
| 5 | 05 | ^E | ENQUIRY (ENQ) | <i>Кто там?</i> — практически не используется. |
| 6 | 06 | ^F | ACKNOWLEDGE (ACK) | <i>Подтверждение, да</i> — практически не используется. |
| 7 | 07 | ^G | BELL (BEL) | <i>Звонок</i> — при его печати на консоли MS-DOS или Unix должен произойти звуковой сигнал. |
| 8 | 08 | ^H | BACKSPACE (BS) | <i>Возврат на шаг</i> — означает, что следующий символ следует печатать с предшествующей позиции. |
| 9 | 09 | ^I | HORIZONTAL TABULATION (TAB) | <i>Горизонтальная таблица</i> — переход на следующую позицию табуляции. |
| 10 | 0A | ^J | LINE FEED (LF) | <i>Подача новой строки</i> — переход на новую строку. В текстовых файлах MS-DOS и Microsoft Windows с сохранением текущей горизонтальной позиции. В текстовых файлах Unix с переходом на первую горизонтальную позицию. |

| | | | | |
|----|----|----|---------------------------------|---|
| 11 | 0B | ^K | VERTICAL TABULATION (VT) | <i>Вертикальная таблица</i> — используется очень редко, как правило, принтерами. |
| 12 | 0C | ^L | FORM FEED (FF) | <i>Подача новой формы</i> — для консоли, как правило, означает очистку экрана, для принтера — завершение печати на текущем листе и выход на новый. |
| 13 | 0D | ^M | CARRIAGE RETURN (CR) | <i>Возврат каретки</i> — переход на первую горизонтальную позицию строки. В текстовых файлах MS-DOS и Microsoft Windows с сохранением текущей строки, а в текстовых файлах Macintosh OS с переходом на новую строку. В текстовых файлах Unix не используется. |
| 14 | 0E | ^N | SHIFT OUT (SO) | <i>Выход</i> — используется очень редко, как правило, принтерами. |
| 15 | 0F | ^O | SHIFT IN (SI) | <i>Вход</i> — используется очень редко, как правило, принтерами. |
| 16 | 10 | ^P | DATA LINK ESCAPE (DLE) | <i>Адресация 1</i> — практически не используется. |
| 17 | 11 | ^Q | DEVICE CONTROL ONE (DC1) | Используется некоторыми телекоммуникационными протоколами как байт X-ON. |
| 18 | 12 | ^R | DEVICE CONTROL TWO (DC2) | Практически не используется. |
| 19 | 13 | ^S | DEVICE CONTROL THREE (DC3) | Используется некоторыми телекоммуникационными протоколами как байт X-OFF. |
| 20 | 14 | ^T | DEVICE CONTROL FOUR (DC4) | Практически не используется. |
| 21 | 15 | ^U | NEGATIVE ACKNOWLEDGE (NAK) | <i>Нет</i> — практически не используется. |
| 22 | 16 | ^V | SYNCHRONOUS IDLE (SI) | <i>Синхронизация</i> — практически не используется. |
| 23 | 17 | ^W | END OF TRANSMISSION BLOCK (ETB) | <i>Конец блока</i> — практически не используется. |
| 24 | 18 | ^X | CANCEL (CAN) | <i>Аннулирование</i> — используется очень редко, как правило, принтерами. |
| 25 | 19 | ^Y | END OF MEDIUM (EM) | <i>Конец носителя</i> — практически не используется. |
| 26 | 1A | ^Z | SUBSTITUTE (SUB) | <i>Замена</i> — в MS-DOS, Macintosh OS и CP/M — это маркер конца текстового файла. |

| | | | | |
|-----|----|-----|-----------------------|---|
| 27 | 1B | ^[_ | ESCAPE (ESC) | <i>Адресатор 2</i> — указывает на то, что некоторое количество кодов после него и он сам образуют группу, рассматриваемую как один код. |
| 28 | 1C | ^[\ | FILE SEPARATOR (FS) | <i>Разделитель файлов</i> — практически не используется. |
| 29 | 1D | ^]` | GROUP SEPARATOR (GS) | <i>Разделитель групп</i> — практически не используется. |
| 30 | 1E | ^_ | RECORD SEPARATOR (RS) | <i>Разделитель записей</i> — практически не используется. |
| 31 | 1F | ^` | UNIT SEPARATOR (US) | <i>Разделитель элементов</i> — практически не используется. |
| 127 | 7F | ^~ | DELETE (DEL) | <i>Забой</i> — удаление последнего видимого знака печатаемой строки. |

| Код | | Символ | Имя символа в Unicode 3.2 |
|------|------|--------|------------------------------|
| 10-й | 16-й | | |
| 32 | 20 | | SPACE |
| 33 | 21 | ! | EXCLAMATION MARK |
| 34 | 22 | " | QUOTATION MARK |
| 35 | 23 | # | NUMBER SIGN |
| 36 | 24 | \$ | DOLLAR SIGN |
| 37 | 25 | % | PERCENT SIGN |
| 38 | 26 | & | AMPERSAND |
| 39 | 27 | ' | APOSTROPHE |
| 40 | 28 | (| LEFT PARENTHESIS |
| 41 | 29 |) | RIGHT PARENTHESIS |
| 42 | 2A | * | ASTERISK |
| 43 | 2B | + | PLUS SIGN |
| 44 | 2C | , | COMMA |
| 45 | 2D | - | HYPHEN-MINUS |
| 46 | 2E | . | FULL STOP |
| 47 | 2F | / | SOLIDUS |
| 48 | 30 | 0 | DIGIT ZERO |
| 49 | 31 | 1 | DIGIT ONE |
| 50 | 32 | 2 | DIGIT TWO |
| 51 | 33 | 3 | DIGIT THREE |
| 52 | 34 | 4 | DIGIT FOUR |
| 53 | 35 | 5 | DIGIT FIVE |
| 54 | 36 | 6 | DIGIT SIX |
| 55 | 37 | 7 | DIGIT SEVEN |
| 56 | 38 | 8 | DIGIT EIGHT |
| 57 | 39 | 9 | DIGIT NINE |
| 58 | 3A | : | COLON |
| 59 | 3B | ; | SEMICOLON |
| 60 | 3C | < | LESS-THAN SIGN |
| 61 | 3D | = | EQUALS SIGN |
| 62 | 3E | > | GREATER-THAN SIGN |
| 63 | 3F | ? | QUESTION MARK |

| Код | | Символ | Имя символа в Unicode 3.2 | Код | | Символ | Имя символа в Unicode 3.2 |
|------|------|--------|------------------------------|------|------|--------|------------------------------|
| 10-й | 16-й | | | 10-й | 16-й | | |
| 64 | 40 | @ | COMMERCIAL AT | 96 | 60 | ˆ | GRAVE ACCENT |
| 65 | 41 | A | LATIN CAPITAL LETTER A | 97 | 61 | a | LATIN SMALL LETTER A |
| 66 | 42 | B | LATIN CAPITAL LETTER B | 98 | 62 | b | LATIN SMALL LETTER B |
| 67 | 43 | C | LATIN CAPITAL LETTER C | 99 | 63 | c | LATIN SMALL LETTER C |
| 68 | 44 | D | LATIN CAPITAL LETTER D | 100 | 64 | d | LATIN SMALL LETTER D |
| 69 | 45 | E | LATIN CAPITAL LETTER E | 101 | 65 | e | LATIN SMALL LETTER E |
| 70 | 46 | F | LATIN CAPITAL LETTER F | 102 | 66 | f | LATIN SMALL LETTER F |
| 71 | 47 | G | LATIN CAPITAL LETTER G | 103 | 67 | g | LATIN SMALL LETTER G |
| 72 | 48 | H | LATIN CAPITAL LETTER H | 104 | 68 | h | LATIN SMALL LETTER H |
| 73 | 49 | I | LATIN CAPITAL LETTER I | 105 | 69 | i | LATIN SMALL LETTER I |
| 74 | 4A | J | LATIN CAPITAL LETTER J | 106 | 6A | j | LATIN SMALL LETTER J |
| 75 | 4B | K | LATIN CAPITAL LETTER K | 107 | 6B | k | LATIN SMALL LETTER K |
| 76 | 4C | L | LATIN CAPITAL LETTER L | 108 | 6C | l | LATIN SMALL LETTER L |
| 77 | 4D | M | LATIN CAPITAL LETTER M | 109 | 6D | m | LATIN SMALL LETTER M |
| 78 | 4E | N | LATIN CAPITAL LETTER N | 110 | 6E | n | LATIN SMALL LETTER N |
| 79 | 4F | O | LATIN CAPITAL LETTER O | 111 | 6F | o | LATIN SMALL LETTER O |
| 80 | 50 | P | LATIN CAPITAL LETTER P | 112 | 70 | p | LATIN SMALL LETTER P |
| 81 | 51 | Q | LATIN CAPITAL LETTER Q | 113 | 71 | q | LATIN SMALL LETTER Q |
| 82 | 52 | R | LATIN CAPITAL LETTER R | 114 | 72 | r | LATIN SMALL LETTER R |
| 83 | 53 | S | LATIN CAPITAL LETTER S | 115 | 73 | s | LATIN SMALL LETTER S |
| 84 | 54 | T | LATIN CAPITAL LETTER T | 116 | 74 | t | LATIN SMALL LETTER T |
| 85 | 55 | U | LATIN CAPITAL LETTER U | 117 | 75 | u | LATIN SMALL LETTER U |
| 86 | 56 | V | LATIN CAPITAL LETTER V | 118 | 76 | v | LATIN SMALL LETTER V |
| 87 | 57 | W | LATIN CAPITAL LETTER W | 119 | 77 | w | LATIN SMALL LETTER W |
| 88 | 58 | X | LATIN CAPITAL LETTER X | 120 | 78 | x | LATIN SMALL LETTER X |
| 89 | 59 | Y | LATIN CAPITAL LETTER Y | 121 | 79 | y | LATIN SMALL LETTER Y |
| 90 | 5A | Z | LATIN CAPITAL LETTER Z | 122 | 7A | z | LATIN SMALL LETTER Z |
| 91 | 5B | [| LEFT SQUARE BRACKET | 123 | 7B | { | LEFT CURLY BRACKET |
| 92 | 5C | \ | REVERSE SOLIDUS | 124 | 7C | | VERTICAL LINE |
| 93 | 5D |] | RIGHT SQUARE BRACKET | 125 | 7D | } | RIGHT CURLY BRACKET |
| 94 | 5E | ˘ | CIRCUMFLEX ACCENT | 126 | 7E | ˜ | TILDE |
| 95 | 5F | _ | LOW LINE | | | | |

Для русского языка имеется несколько разных таблиц кодировки символов кириллицы: KOI8-R, IBM-866, CP-1251, ISO-8551-5, все они одинаково изображают символы первой половины таблицы (от 0 до 127) и различаются представлением символов русского алфавита и псевдографики.

Кодировки символов не определяют изображения символов, а только состав набора символов и способ его представления в компьютере. Кроме того очень важен порядок перечисления символов в наборе, так как он влияет самым существенным образом на алгоритмы сортировки.

Единицы измерения информации

Бит (англ. *binary digit*; также игра слов: англ. *bit* — немного) — один двоичный разряд в двоичной системе счисления. Обозначается по ГОСТ 8.417-2002. Для образования кратных единиц применяется с приставками СИ и с двоичными приставками, принятыми МЭК (Международная электротехническая комиссия — международная некоммерческая организация по стандартизации в области электрических, электронных и смежных технологий).

| Приставка | Аналогичная десятичная | приставка Сокращения по МЭК для битов, байтов | Значение, на которое умножается исходная величина |
|-----------|------------------------|---|--|
| киби | кило (10^3) | Кибит, КиБ | $2^{10} = 1\ 024$ |
| меби | мега (10^6) | Мибит, МиБ | $2^{20} = 1\ 048\ 576$ |
| гиби | гига (10^9) | Гибит, ГиБ | $2^{30} = 1\ 073\ 741\ 824$ |
| теби | тера (10^{12}) | Тибит, ТиБ | $2^{40} = 1\ 099\ 511\ 627\ 776$ |
| пеби | пета (10^{15}) | Пибит, ПиБ | $2^{50} = 1\ 125\ 899\ 906\ 842\ 624$ |
| эксби | экса (10^{18}) | Эибит, ЭиБ | $2^{60} = 1\ 152\ 921\ 504\ 606\ 846\ 976$ |
| зеби | зетта (10^{21}) | Зибит, ЗиБ | $2^{70} = 1\ 180\ 591\ 620\ 717\ 411\ 303\ 424$ |
| йоби | йотта (10^{24}) | Йибит, ЙиБ | $2^{80} = 1\ 208\ 925\ 819\ 614\ 629\ 174\ 706\ 176$ |

Двоичные логарифмы других оснований

Нат

Замена логарифма 2 на e , приводит к единицам *нат* ($\log_2 e \approx 1,443$ бит.), 1 бит = $\ln 2 \approx 0,693$ нат,

Трит

Один трит (*trinary digit*), как один троичный разряд, может принимать три возможных значения (состояния, кода): 0, 1 и 2. 1 трит (**троп**) равен троичному логарифму 3-х **возможных состояний** (кодов) одного троичного разряда

Трит является подмножеством *троичного разряда*, который является одним троичным числовым разрядом троичного представления числа в троичной системе счисления. Один троичный числовой разряд (одна позиция, одно знакоместо) троичного представления числа в троичной системе счисления Троичный разряд может принимать только три взаимоисключающих значения (три троичных величины), которым соответствуют три троичные цифры (три троичных знака), которые могут иметь разные обозначения:

Состояния в троичных системах счисления

| | | | | | | | | | | | | |
|---|---|----|-----------|---|---|---|---|------------|---------------|-------|---------|-------|
| 1 | 0 | -1 | $\bar{1}$ | - | i | N | N | ложь | ложно | false | false | false |
| 2 | 1 | 0 | 0 | 0 | 0 | O | Z | неизвестно | Не определено | null | unknown | Fail |
| 3 | 2 | +1 | 1 | + | 1 | P | P | истина | истинно | true | true | True |

Децит

Один децит (*decimal digit*) (дит — *dit*, Хартли — *Hart*, бан — *ban*) как один десятичный разряд, может принимать десять возможных значения (состояния, кода): 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9.

Байт (англ. *byte*)

Единица хранения и обработки цифровой информации, это совокупность битов, обрабатываемая компьютером одновременно. В современных вычислительных системах байт считается равным восьми битам, в этом случае он может принимать одно из 256 (2^8) различных значений (состояний, кодов). Однако в истории компьютеров известны решения с другим размером байта, например 6 битов, 36 битов в PDP-10. Поэтому иногда в компьютерных стандартах и официальных документах для однозначного обозначения 8-битного слова используется термин «октет» (лат. *octet*).

В большинстве вычислительных архитектур минимальный независимо адресуемый набор данных.

Название «байт» (слово *byte* представляет собой сокращение словосочетания *Binary Term* — «двоичный терм») было впервые использовано в 1956 году В. Бухгольцем при проектировании первого суперкомпьютера IBM 7030 для пучка одновременно передаваемых в устройствах ввода-вывода шести битов. Позже, в рамках того же проекта, байт был расширен до восьми бит.

Байтовая адресация памяти была впервые применена в системе IBM System/360. В более ранних компьютерах адресовать можно было только целиком машинное слово, состоявшее из нескольких байтов, что затрудняло обработку текстовых данных.

Восьмибитные байты были приняты в System/360, вероятно, из-за использования BCD-формата представления чисел: одна десятичная цифра (0-9) требует 4 бита (тетраду) для хранения; один 8-битный байт может представлять две десятичные цифры. 6-битные байты могут хранить только по одной десятичной цифре, два бита остаются незадействованными.

По другой версии, 8-битный размер байта связан с 8-битным же числовым представлением символов в кодировке EBCDIC.

По третьей версии, из-за двоичной системы кодирования в компьютерах наиболее выгодными для аппаратной реализации и удобными для обработки данных являются длины слов кратные степеням 2, в

том числе и 1 Байт = $2^3 = 8$ битов, системы и компьютеры с длинами слов не кратными степеням 2 отпали из-за невыгодности и неудобства.

Постепенно 8-битные байты стали стандартом де-факто и с начала 1970-х в большинстве компьютеров байты состоят из 8 бит и размер машинного слова кратен 8 битам.

Из соображений удобства единицы нетекстовых типов данных также делают кратными 8 битам, например:

- размер одного сэмпла в звуковых файлах равен 8, 16 или 24 битам;
- размер пикселя в системе RGB равен 24 битам (по 8 бит на цвет).

В таких обозначениях как **байт** (русское) или **B** (английское) под байтом (B) подразумевается именно 8 бит, хотя сам термин «байт» не вполне корректен с точки зрения теории.

Октет

Восемь двоичных разрядов (8 битов). В русском языке октет обычно называют байтом. Октет имеет 256 **возможных состояний** (кодов, значений). Во французском языке используются обозначения **o**, **Ko**, **Mo** и т. д. (от слова *octet*) дабы подчеркнуть, что речь идёт именно о 8 битах.

Ниббл (англ. *nibble*, *nybble*)

Ниббл или полубайт — единица измерения информации, равная четырём двоичным разрядам (битам), удобна тем, что представима одной шестнадцатеричной цифрой, то есть является одним шестнадцатеричным разрядом. Переменная размера «ниббл» может принимать $2^4=16$ различных значений. В русском языке используется синоним «тетрада».

Трайт

Один трайт = 6 трит = $6\log_3 3$;

Базовые понятия теории информации

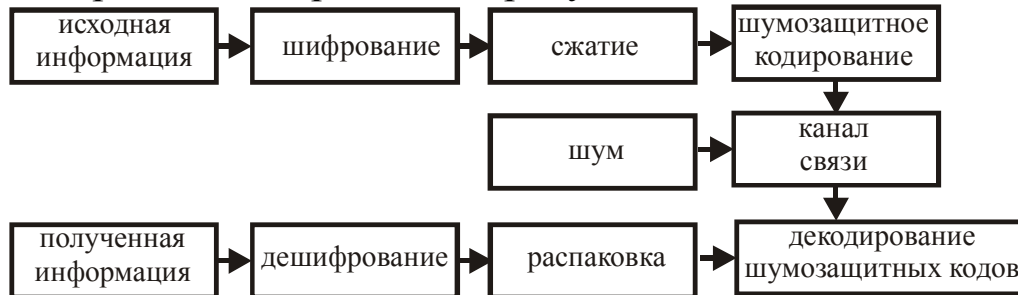
Информация — нематериальная сущность, при помощи которой с любой точностью можно описывать реальные (материальные), виртуальные и понятийные сущности.

Канал связи — это среда передачи информации, которая характеризуется в первую очередь максимально возможной для неё скоростью передачи данных (ёмкостью канала связи)

Шум — случайные искажения сигнала в канале связи при передаче информации.

Кодирование — преобразование дискретной информации из одного вида в другой.

Общая схема передачи изображена на рисунке:



Вероятностный подход к измерению дискретной и непрерывной информации.

Для дискретных случайных величин X и Y , заданных законами распределения $P(X = X_i) = p_i$, $P(Y = Y_j) = q_j$ и совместным распределением $P(X = X_i, Y = Y_j) = p_{ij}$, количество информации, содержащейся в X относительно Y , равно

$$I(X, Y) = \sum_{i, j} p_{ij} \log_2 \frac{p_{ij}}{p_i p_j}.$$

Для непрерывных случайных величин X и Y , заданных плотностями распределения вероятностей $p_X(t_1)$, $p_Y(t_2)$ и $p_{XY}(t_1, t_2)$, аналогичная формула имеет вид

$$I(X, Y) = \iint_{\mathbb{R}^2} p_{XY}(t_1, t_2) \log_2 \frac{p_{XY}(t_1, t_2)}{p_X(t_1) p_Y(t_2)} dt_1 dt_2$$

Так как

$$P(X = X_i, Y = Y_i) = \begin{cases} 0, & \text{при } i \neq j, \\ P(X = X_i), & \text{при } i = j. \end{cases}$$

следовательно

$$I(X, X) = \sum_i p_i \log_2 \frac{p_i}{p_i p_i} = -\sum_i p_i \log_2 p_i.$$

То есть энтропия дискретной случайной величины X в теории информации определяется формулой $H(X) = HX = I(X, X)$.

Смысл энтропии Шеннона: энтропия дискретной случайной величины — это минимум среднего количества бит на 1 символ, которое нужно передавать по каналу связи о текущем значении данной дискретной случайной величины.

Пусть некоторая дискретная случайная величина X , имеет четыре значения, вероятность которых одинакова:

$$P(X = 0) = P(X = 1) = P(X = 2) = P(X = 3) = \frac{1}{4}.$$

Закодируем эти величины:

$\text{code}(X = 0) = 00$, $\text{code}(X = 1) = 01$, $\text{code}(X = 2) = 10$, $\text{code}(X = 3) = 11$.

Здесь функция $\text{code}(X)$ каждому значению X ставит однозначно в соответствие некоторый битовый код.

Введём функцию, возвращающую длину сообщения, кодирующего заданное значение X : $L(X) = \text{len}(\text{code}(X))$.

Тогда $ML(X)$ будет означать среднюю длину сообщения, кодирующего X .

Для данного распределения X справедливо $ML(X) = HX$.

Пусть $P(X = 0) = \frac{3}{4}$, $P(X = 1) = \frac{1}{8}$, $P(X = 2) = P(X = 3) = \frac{1}{16}$

Тогда $HX = \frac{3}{4} \log_2 \frac{4}{3} + \frac{1}{8} \log_2 8 + \frac{1}{8} \log_2 16 = \frac{19}{8} - \frac{3}{4} \log_2 3 \approx 1,186 \frac{\text{бит}}{\text{симв.}}$.

Закодируем величины следующим образом

$\text{code}(X = 0) = 0$, $\text{code}(X = 1) = 10$, $\text{code}(X = 2) = 110$, $\text{code}(X = 3) = 111$.

Средняя длина сообщения $\frac{3}{4} \cdot 1 + \frac{1}{8} \cdot 2 + \frac{1}{16} \cdot 3 + \frac{1}{16} \cdot 3 = \frac{22}{16} \approx 1,375 \frac{\text{бит}}{\text{симв.}}$.

Или $L(X)$ имеет распределение

$$P(L(X) = 1) = \frac{3}{4}, P(L(X) = 2) = \frac{1}{8}, P(L(X) = 3) = \frac{1}{8}.$$

$$ML(X) = 1 \cdot \frac{3}{4} + 2 \cdot \frac{1}{8} + 3 \cdot \frac{1}{8} = 1,375 \frac{\text{бит}}{\text{симв.}}$$

В итоге $ML(X) > HX$. То есть среднее количество бит приходящихся на одно кодируемое значение не меньше, чем величина энтропии.

Свойства меры информации и энтропии:

1) $I(X, Y) \geq 0$, $I(X, Y) = 0 \Leftrightarrow X$ и Y независимы;

2) $I(X, Y) = I(Y, X)$;

3) $HX = 0 \Leftrightarrow X$ – константа;

4) $I(X, Y) = HX + HY - H(X, Y)$, где $H(X, Y) = \sum_{i, j} p_{ij} \log_2 p_{ij}$;

5) $I(X, Y) \leq I(X, X)$. Если $I(X, Y) = I(X, X)$, то X – функция от Y .

При независимости случайных величин X и Y одна из них ничем не описывает другую, что и отражается в том, что для таких случайных величин $I(X, Y) = 0$.

Пример измерения количества информации при подбрасывании двух игральных костей. X_1 и X_2 количества очков, выпавших соответственно на 1-й и 2-й игральной кости. $Y = X_1 + X_2$.

Необходимо найти $I(Y, X_1)$, $I(X_1, X_1)$, $I(Y, Y)$.

Можно считать, что законы распределения вероятностей для дискретных случайных величин X_1 и X_2 совпадают.

| | | | | | | |
|-------|---------------|---------------|---------------|---------------|---------------|---------------|
| X_1 | 1 | 2 | 3 | 4 | 5 | 6 |
| p | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{1}{6}$ | $\frac{1}{6}$ |

То есть при $j = 1 \dots 6$: $q_j = P(X_1 = j) = \frac{1}{6}$.

Закон распределения вероятностей для дискретной случайной величины Y , $P(Y = i) = P(X_1 + X_2 = i)$, $i = 2 \dots 12$,

вследствие того, что X_1, X_2 — независимы и поэтому

$$P(X_1 = n, X_2 = m) = P(X_1 = n)P(X_2 = m),$$

будет

$$p_i = P(X_1 + X_2 = i) = \sum_{\substack{n+m=i \\ 1 \leq n, m \leq 6}} P(X_1 = n)P(X_2 = m) = \sum_{\substack{n+m=i \\ 1 \leq n, m \leq 6}} \frac{1}{36}.$$

Таблицы, определяющие Y :

| $X_2 \backslash X_1$ | 1 | 2 | 3 | 4 | 5 | 6 |
|----------------------|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| | | | | | | | | | | | |
|-----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| $Y = X_1 + X_2$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| p | $\frac{1}{36}$ | $\frac{2}{36}$ | $\frac{3}{36}$ | $\frac{4}{36}$ | $\frac{5}{36}$ | $\frac{6}{36}$ | $\frac{5}{36}$ | $\frac{4}{36}$ | $\frac{3}{36}$ | $\frac{2}{36}$ | $\frac{1}{36}$ |

то есть при $i = 2...12$, $p_i = P(Y = i) = \frac{6 - |7 - i|}{36}$.

Закон совместного распределения вероятностей дискретной случайной величины X_1 и Y будет

$p_{ij} = P(Y = i, X_1 = j) = P(Y = i | X_1 = j)P(X_1 = j)$, например,

$$P(Y = 2, X_1 = 1) = P(Y = 2 | X_1 = 1)P(X_1 = 1) = P(X_2 = 1)P(X_1 = 1) = \frac{1}{36}.$$

В общем случае $p_{ij} = P(Y = i, X_1 = j) = \begin{cases} \frac{1}{36}, & \text{при } 1 \leq i - j \leq 6, \\ 0, & \text{иначе.} \end{cases}$

$$\begin{aligned}
I(Y, X_1) &= \sum_{j=1}^6 \sum_{1 \leq i-j \leq 6} p_{ij} \log_2 \frac{p_{ij}}{p_i q_j} = \sum_{j=1}^6 \sum_{1 \leq i-j \leq 6} p_{ij} \log_2 \frac{p_{ij}}{p_i q_j} = \frac{1}{36} \sum_{j=1}^6 \sum_{1 \leq i-j \leq 6} \log_2 \frac{1}{6 p_i} = \\
&= \frac{1}{36} \left(\sum_{i=2}^7 \log_2 \frac{1}{6 p_i} + \sum_{i=3}^8 \log_2 \frac{1}{6 p_i} + \sum_{i=4}^9 \log_2 \frac{1}{6 p_i} + \sum_{i=5}^{10} \log_2 \frac{1}{6 p_i} + \sum_{i=6}^{11} \log_2 \frac{1}{6 p_i} + \sum_{i=7}^{12} \log_2 \frac{1}{6 p_i} \right) = \\
&= \frac{1}{36} (10 + 24 \log_2 3 - 10 \log_2 5) \approx 0,69 \frac{\text{бит}}{\text{симв.}}.
\end{aligned}$$

$$I(X_1, X_1) = I(X_2, X_2) = - \sum_{i=2}^6 q_i \log_2 q_i = 1 + \log_2 3 \approx 2,58 \frac{\text{бит}}{\text{симв.}}.$$

$$I(Y, Y) = - \sum_{i=2}^{12} p_i \log_2 p_i = \frac{1}{36} (46 + 60 \log_2 3 - 10 \log_2 5) \approx 3,27 \frac{\text{бит}}{\text{симв.}}.$$

Расчёт и моделирование на ЭВМ

```

class dice // игральная кость
{ long g;
public:
dice(long a):g(a) {}
~dice() {}
long get() { return 1+rand()%g; }
};

```

```

int main()
{ const long G=6;
  srand(time(NULL));
  float summ,summX,summY;
  long i,j,x1,x2,H,min=G+G,max=0,N=1000000L;
  dice a(G),b(G);
  long* x=new long[G];
  memset(x,0,G*sizeof(long));
  for(i=0;i<N;i++) x[a.get()-1]++;
  cout<<"№\tвероятность выпадений\tк-во выпадений\tчастость появления"<<endl;
  for(i=0;i<G;i++)
  cout<<i+1<<":\t1/"<<G<<""<<1./G<<'\t'<<x[i]<<'\t\t'<<x[i]/float(N)<<endl;
  // таблица результатов суммы очков для двух игральных костей
  for(i=0;i<G;i++) cout<<" "<<char(32+92*!i)<<'\t'<<i+1; cout<<endl;
  for(i=1;i<=(G+1)*(G+2);i++) cout<<char(45-i*(2==i)); cout<<endl;
  for(i=1;i<=G;i++,cout<<endl)
    for(cout<<i<<'|',j=1;j<=G;cout<<'\t'<<i+j++)
      { long k=i+j;
        if(min>k) min=k;
        if(max<k) max=k;
      }
  cout<<"min="<<min<<"\tmax="<<max<<endl; H=max-min+1;
  float* p=new float[H];
  long* m=new long[H];
  long* px1=new long[G];
  long* px2=new long[G];
  // расчёт вероятностей для Y
  for(i=0;i<H;i++) p[i]=(G-abs(G-1-i))/float(G*G);
  for(summ=i=0;i<G;i++) for(j=0;j<G;j++) summ+=log(1./G/p[i+j])/log(2.);
  summ/=G*G;

```

```

for(summX=i=0;i<G;i++) summX+=log(1./G)/log(2.);
summX/=-G;
for(summY=i=0;i<H;i++) summY-=p[i]*log(p[i])/log(2.);
memset(m,0,H*sizeof(long));
memset(px1,0,G*sizeof(long));
memset(px2,0,G*sizeof(long));
for(i=0;i<N;i++) x1=a.get(),x2=b.get(),px1[x1-1]++,px2[x2-1]++,m[x1+x2-min]++;
cout<<"№\твероятность\тчастость"<<endl;
for(i=0;i<H;i++)
    cout<<i+min<<": "<<G-abs(G-1-i)<<"/"<<G*G<<="<<(G-abs(G-1-
i))/float(G*G)<<' \т '<<(p[i]=m[i]/float(N))<<endl;
cout<<"I(Y,X1)в="<<summ<<" бит/симв."<<endl;
cout<<"I(X1,X1)в="<<summX<<" бит/симв."<<endl;
cout<<"I(Y,Y)в="<<summY<<" бит/симв."<<endl;
for(summ=i=0;i<G;i++) for(j=0;j<G;j++) summ+=log(1./G/p[i+j])/log(2.);
summ/=G*G;
cout<<"I(Y,X1)ч="<<summ<<" бит/симв."<<endl;
for(summX=i=0;i<G;i++) summX-=px1[i]/float(N)*log(px1[i]/float(N))/log(2.);
cout<<"I(X1,X1)ч="<<summX<<" бит/симв."<<endl;
for(summX=i=0;i<G;i++) summX-=px2[i]/float(N)*log(px2[i]/float(N))/log(2.);
cout<<"I(X2,X2)ч="<<summX<<" бит/симв."<<endl;
for(summY=i=0;i<H;i++) summY-=p[i]*log(p[i])/log(2.);
cout<<"I(Y,Y)ч="<<summY<<" бит/симв."<<endl;
delete[]x;
delete[]m;
delete[]p;
delete[]px1;
delete[]px2;
return 0;
}

```

Зависимость между последовательностью выбираемыми символами также уменьшает неопределённость и ведёт к дальнейшему уменьшению энтропии. Действительно, чем больше вероятностные связи между соседними символами сообщения, тем меньше свобода выбора последующего символа, а следовательно, тем меньше информации содержится в среднем в каждом вновь выбираемом символе источника.

Обозначим через $P(a_i|a_j)$ вероятность того, что источник выберет символ a_i , при условии, что ранее он выбрал символ a_j . Тогда при фиксированном a_j энтропия источника

$$H(A|a_j) = -\sum_{i=1}^K P(a_i|a_j) \log P(a_i|a_j).$$

Усредняя по всем возможным значениям предшествующего символа a_j , получаем

$$H(A|A') = \sum_{j=1}^K H(A|a_j) P(a_j) = \sum_{i=1}^K \sum_{j=1}^K P(a_j) P(a_i|a_j) \log P(a_i|a_j).$$

Совместная вероятность $P(a_i, a_j) = P(a_j) P(a_i|a_j)$.

Тогда энтропия источника с памятью, простирающейся лишь на соседний символ (*марковского источника*): $H(A|A') = -\sum_{i=1}^K \sum_{j=1}^K P(a_i, a_j) \log P(a_i|a_j)$.

Энтропия $H(A|A')$ называется условной. Условная энтропия не может превышать безусловную:

$$H(A|A') \leq H(A)$$

Типичным примером источника с памятью и неравновероятным выбором символов является текст, написанный на естественном языке. Неравные вероятности символов (букв) и связи между ними обусловлены структурой языка. Так в русском языке вероятность буквы «о» составляет примерно 0,09, а буквы «ф» — лишь 0,002. Наибольшую вероятность имеет символ «пробел» — примерно 0,125.

Формула для определения энтропии источника с памятью всё более усложняется по мере дальнейшего учёта вероятностных связей между символами. Однако, энтропию источника можно определить и опытным путём. Согласно многочисленным экспериментальным данным для текста на русском языке энтропия составляет 1,5 бита на букву. Если считать, что русский алфавит содержит 32 буквы, то максимальная энтропия $H_{\max}(A) = \log_2 32 = 5 \frac{\text{бит}}{\text{симв.}}$.

Таким образом, средняя информация, содержащаяся в одной букве «абсолютно хаотического русского текста» в $\frac{5}{1,5} \approx 3,3$ раза больше, чем в русском тексте, соответствующем нормальной речи.

Энтропия источника связана с понятием его *избыточности*: $\rho_{\text{и}} = 1 - \frac{H(A)}{H_{\max}(A)}$.

Избыточность представляет собой безразмерную величину, которая может принимать значения от 0 до 1. Избыточность равна 0, если энтропия источника максимальна, то есть выдаваемые источником символы равновероятны и независимы. Такой источник называют источником без избыточности.

Смысл избыточности можно пояснить следующим образом. Для передачи или хранения некоторого объёма информации I от источника с энтропией $H(A)$ требуется $n = \frac{I}{H(A)}$ символов. Если же теперь

каким-то образом полностью устранить избыточность источника, то для передачи этого же объёма информации потребуется $n_0 = \frac{I}{H_{\max}(A)}$ символов. Поскольку $H(A) < H_{\max}(A)$, то $n > n_0$.

Избыточность

$$\rho_{\text{и}} = 1 - \frac{I \cdot n_0}{n \cdot I} = 1 - \frac{n_0}{n}.$$

Избыточность характеризует относительное удлинение сообщения по сравнению с сообщением от источника без избыточности ($\rho_{\text{и}} = 0$). Избыточность источника русского текста $\rho_{\text{и}} = 1 - \frac{1,5}{5} = 0,7$ (70%).

Это означает, что устранив избыточность, можно было бы сократить сообщение в $\frac{n}{n_0} = \frac{1}{1 - \rho_{\text{и}}} = \frac{10}{3} \approx 3,3$ раза.

Ещё одной важной характеристикой источника является его *производительность*. Если скорость выдачи символов источником постоянна и равняется $v_{\text{и}}$, то его производительность

$$H'(A) = v_{\text{и}} H(A).$$

Это выражение определяет среднее количество информации, которое может выдавать (создавать) источник в единицу времени. Если $H(A)$ определяется в битах, то $H'(A)$ имеет размерность $\frac{\text{бит}}{\text{с}}$.

Источник называется *стационарным*, если его вероятностные, а следовательно, и информационные характеристики не зависят от времени.

Задача Память двоичного стационарного источника с символами 0 и 1 простирается лишь на соседний символ и, следовательно, дискретная последовательность символов, выдаваемых источником, описывается простой цепью Маркова с матрицей переходных вероятностей:

$$\begin{vmatrix} P(1|\hat{1}) & P(1|\hat{0}) \\ P(0|\hat{1}) & P(0|\hat{0}) \end{vmatrix}$$

где $P(a_i|\hat{a}_j)$ — вероятность символа a_i при условии, что ему предшествует символ \hat{a}_j .

Полагая, что $P(1|\hat{1}) = 0,9$; $P(1|\hat{0}) = 0,7$, найти энтропию источника и его избыточность. Найти энтропию и избыточность двоичного источника без памяти, но с теми же значениями вероятностей передачи символов.

Решение:

Прежде всего, найдем $P(0|\hat{1}) = 1 - 0,9 = 0,1$; $P(0|\hat{0}) = 1 - 0,7 = 0,3$ и безусловные вероятности передачи символов.

По формулу полной вероятности $P(0) = P(\hat{0})P(0|\hat{0}) + (1 - P(\hat{0}))P(0|\hat{1})$.

В силу однородности цепи (последовательности символов) $P(0) = P(\hat{0})$, тогда

$$P(0) = P(0)P(0|\hat{0}) + (1 - P(0))P(0|\hat{1}).$$

Имеем

$$P(0) = P(0) \cdot 0,3 + (1 - P(0)) \cdot 0,1;$$

$$0,8 \cdot P(0) = 0,1;$$

Отсюда $P(0) = 0,125$, $P(1) = 0,875$. Можно убедиться в справедливости соотношения

$$P(1) = P(0)P(1|\hat{0}) + (1 - P(0))P(1|\hat{1}).$$

Энтропия источника

$$H(A|\hat{A}) = -P(0)\left(P(0|\hat{0})\log_2 P(0|\hat{0}) + P(1|\hat{0})\log_2 P(1|\hat{0})\right) - \\ -P(1)\left(P(0|\hat{1})\log_2 P(0|\hat{1}) + P(1|\hat{1})\log_2 P(1|\hat{1})\right) = 0,51 \frac{\text{бит}}{\text{симв.}}$$

Избыточность источника

$$\varepsilon_{\text{и}} = 1 - \frac{H(A|\hat{A})}{H_{\text{max}}(A)} = 0,49.$$

Для источника без памяти при тех же безусловных вероятностях передачи символов

$$H(A) = -P(1)\log_2 P(1) - P(0)\log_2 P(0) = 0,541 \frac{\text{бит}}{\text{симв.}}$$

Избыточность источника $\varepsilon_{\text{и}} = 1 - \frac{H(A)}{H_{\text{max}}(A)} = 0,459$

Программа, приведённая ниже моделирует описанный источник. Вероятности переходов задаются в виде матрицы в текстовом файле.

```
void ExeFile(char*& n)
{ int j,k=0; for(;n[k];k++); for(k--;k>=0;--k) if(n[k]!='\\') break; for(j=0;n[+k];n[j++]=n[k]); n[j]=0; return; }
int main(int mn,char* nm[])
{ int N,size,i,j,k,l,n,row,col,err,max,pmax,ed,e,kp;
char *p,**v,g,shift,buf[32]; float *a,*b,r,R; int *Pr,**pr,*rr,*ss;
unsigned long long seed,C,A; A=6364136223846793005; C=1442695040888963407;
ExeFile(nm[0]);
cout<<"Генератор символов по заданной матрице NxN условных вероятностей\n\n";
if(mn!=5) cerr<<nm[0]<<" matrix_cond_ver.txt simv.ext shift N\n",
    cout<<"\nmatrix_cond_ver.txt - числа в текстовом файле в виде прямоугольной таблицы\n",
    cout<<" либо сумма построчно равна 1, либо одно значение в строчке вне диапазона (0..1),\n",
    cout<<"simv.ext - результирующий текст,\n",
    cout<<"shift - добавка к кодам символов для переноса в текстовую часть (изначально коды отсчитываются с 00h,\n",
    cout<<"N - размер синтезируемого файла.\n",exit(1);
ifstream in(nm[1],ios::binary); if(!in) cerr<<"file \""<nm[1]<<"\" not open!\n",exit(1);
in.seekg(0,ios::end);size=in.tellg();in.seekg(0,ios::beg);
p=new char[size+2];if(!p) cerr<<"Error memory!\n",exit(1);in.read((char*)p,size);in.close();
for(i=0;i-size;+i) if(p[i]==9||p[i]==13) p[i]=32; p[i+]=0x0A;p[i+]=0;
for(k=i=1;i-size-3;+i) if(p[i]==32&&p[i-1]==32); else p[k+]=p[i]; p[k]=0;
if(p[0]==32) for(i=0;p[i];+i) p[i]=p[i+1]; p[i]=0;
for(i=1;p[i];+i) if(p[i]==10&&p[i-1]==32) p[i-1]=10;
for(i=1;p[i];+i) if(p[i]==32&&p[i-1]==10) p[i]=10;
for(k=i=1;p[i];+i) if(p[i]==10&&p[i-1]==10); else p[k+]=p[i]; p[k]=0;
for(row=i=0;p[i];) if(p[i+]=10) row++;
shift=atoi(nm[3]); N=atoi(nm[4]);
for(j=col=i=0;p[i];i++)
{ if(p[i]==10) if(col+1!=(err=row)) break; else col=err=0,j++;
if(p[i]==32) col++;
}
```

```

if(err) cerr<<"Ошибка количества данных в строке "<<j<<" err="<<err<<" col="<<col<<endl,exit(1); else col=row;
for(j=max=0,i=0;p[i],i++,j++) if(p[i]==32||p[i]==10) { if(j>max) max=j; j=-1; }
for(k=i=0;p[i],++i) { if(p[i]=='.') if(++k>1)
    cerr<<"Ошибка более одной десятичной точки в числе!",exit(1); if(p[i]==32||p[i]==10) k=0; }
for(pmax=j=k=i=0;p[i],++i) { if(p[i]==10||p[i]==32) { if(k>pmax) pmax=k; j=k=0; } if(j) k++; if(p[i]=='.') j=1; }
if(pmax>6) cerr<<"Точность слишком высокая "<<endl,exit(1);
Pr=new int[col*row];
for(n=k=i=0;p[i];buf[k++]=p[i++])
    if(p[i]==10||p[i]==32)
        { buf[k]=0;
          for(j=0;buf[j],j++) if(buf[j]=='.') break;
          for(l=0;l<pmax;l++,j++) { if(!buf[j]) break; else buf[j]=buf[j+1]; if(!buf[j]) break; }
          for(;l<pmax;l++) buf[j++]=48;
          buf[j]=k=0;
          Pr[n++]=atoi(buf);
          if(!p[++i]) break;
        }
for(ed=1,i=0;i<pmax;i++,ed*=10);
for(i=0;i<row;i++)
    {
    for(k=l=j=0;j<row;j++) if(Pr[i*row+j]>ed||Pr[i*row+j]<0) l=j,k++;
    if(k>l) cerr<<"Ошибка вне диапазона больше 1 числа"<<endl,exit(1);
    if(!k) { for(j=0;j<row;j++) k+=Pr[i*row+j]; if(k!=ed) cerr<<"Ошибка сумма вероятностей в строке > 1"<<endl,exit(1); }
    else { for(k=j=0;j<row;j++) if(l!=j) k+=Pr[i*row+j]; Pr[i*row+1]=ed-k; }
    }
cout<<"Исходная таблица условных вероятностей:"<<endl;
for(i=0;i<row;i++,cout<<endl) for(j=0;j<row;j++) cout<<Pr[i*row+j]/float(ed)<<'\\t';
row--;
a=new float[row*row];
b=new float[row];
for(i=0;i<row;i++) for(j=0;j<row;j++) a[j*row+i]=(i==j)+float(Pr[(row+1)*row+j]-Pr[i*(row+1)+j])/ed;
for(i=0;i<row;i++) b[i]=float(Pr[(row+1)*row+i])/ed;

```

```

matrix<float> Y(row,row),Z(row,row),W(row,1),X(row,1);
for(i=0;i<row;i++) for(j=0;j<row;j++) Y(i,j)=a[i*row+j];
for(i=0;i<row;i++) X(i,0)=b[i];
Z=Y.inv(); W.Mul(Z,X);
vect<float> P(row+1);
for(r=i=0;i<row;i++) r+=(P[i]=W(i,0));
P[P.up()]=1-r;
cout<<"Рассчитанные безусловные вероятности:"<<endl;
for(i=0;i<row;i++) cout<<"P("<<i<<")="<<P[i]<<'\\t'; cout<<endl;
cout<<"Энтропия и избыточность для рассчитанных безусловных вероятностей:"<<endl;
for(r=i=0;i<=row;i++) if(P[i]>1e-9) r+=-P[i]*log(P[i])/log(2.);
cout<<"H(A)="<<r<<endl;
cout<<"x(A)="<<1-r/(log(row+1.)/log(2.))<<endl;
cout<<"Максимальная энтропия:"<<endl;
cout<<"Hmax(A)="<<log(row+1.)/log(2.)<<endl;
cout<<"Оценка возможности сжатия исходного текста без учёта связей между символами:"<<endl;
if((1-r/(log(row+1.)/log(2.)))==1) cout<<"n/n0=oo"<<endl;
else cout<<"n/n0="<<1/(1-(1-r/(log(row+1.)/log(2.))))<<endl;
for(R=i=0;i<=row;R+=-P[i+]*r)
    for(r=j=0;j<=row;j++) if(Pr[i*(row+1)+j]) r+=Pr[i*(row+1)+j]/float(ed)*log(Pr[i*(row+1)+j]/float(ed))/log(2.);
cout<<"Энтропия и избыточность для рассчитанных условных вероятностей:"<<endl;
cout<<"H(A|A^)="<<R<<endl;
cout<<"x(A|A^)="<<1-R/(log(row+1.)/log(2.))<<endl;
cout<<"Оценка возможности сжатия исходного текста при учёте связи между соседними символами:"<<endl;
if(1-R/(log(row+1.)/log(2.)))==1) cout<<"n/n0=oo"<<endl;
else cout<<"n/n0="<<1/(1-(1-R/(log(row+1.)/log(2.))))<<endl;
seed=time(NULL),seed=A*seed+C,kp=k=(seed>>32)%(row+1);
ofstream ou(nm[2],ios::binary);
pr=new int*[row+1];
rr=new int[row+1];
ss=new int[row+1];
for(i=0;i<=row;i++) pr[i]=new int[row+1];

```

```

for(i=0;i<=row;i++) for(ss[i]=j=0;j<=row;j++) pr[i][j]=0;
for(j=0;j<N;g=kp=k,ss[k]++,ou.put(g+shift),j++) { for(seed=A*seed+C,e=(seed>>32)%ed,k=i=0;i<row;i++)
k+=!(e-Pr[kp*(row+1)+i])>>(sizeof(int)*8-1)&1; if(j) pr[kp][k]++; }
ou.close();
cout<<"Количество выпадений символов согласно таблице условных вероятностей:"<<endl;
for(n=i=0;i<=row;i++,cout<<endl) for(j=0;j<=row;j++) cout<<pr[i][j]<<'\\t',n+=pr[i][j];
cout<<"Объём выборки:"<<+n<<endl;
for(i=0;i<=row;i++) for(rr[i]=j=0;j<=row;j++) rr[i]+=pr[i][j];
cout<<"Таблица условных частотей:"<<endl;
for(i=0;i<=row;i++,cout<<endl) for(j=0;j<=row;j++)
if(rr[i]) cout<<"P("<<j<<"|"<<i<<")="<<pr[i][j]/float(rr[i])<<'\\t'; else cout<<"P("<<j<<"|"<<i<<")=-0\\t";
cout<<"Безусловные частоты выпадений символов:"<<endl;
for(i=0;i<=row;i++) cout<<"P("<<i<<")="<<ss[i]/float(n)<<'\\t'; cout<<endl;
cout<<"Энтропия и избыточность для найденных безусловных частотей:"<<endl;
for(r=i=0;i<=row;i++) if(ss[i]) r+=-ss[i]/float(n)*log(ss[i]/float(n))/log(2.);
cout<<"H~(A)="<<r<<endl;
cout<<"x(A)="<<1-r/(log(row+1.)/log(2.))<<endl;
cout<<"Возможность сжатия синтезированного текста без учёта связи между соседними символами:"<<endl;
if(1-r/(log(row+1.)/log(2.))=1) cout<<"n/n0=oo"<<endl;
else cout<<"n/n0="<<1/(1-(1-r/(log(row+1.)/log(2.))))<<endl;
for(R=i=0;i<=row;R+=-ss[i+]/float(n)*r)
for(r=j=0;j<=row;j++) if(pr[i][j]) if(rr[i]) r+=pr[i][j]/float(rr[i])*log(pr[i][j]/float(rr[i]))/log(2.);
cout<<"Энтропия и избыточность для найденных условных частотей:"<<endl;
cout<<"H~(A|A^)="<<R<<endl;
cout<<"x(A|A^)="<<1-R/(log(row+1.)/log(2.))<<endl;
cout<<"Возможность сжатия синтезированного текста при учёте связи между соседними символами:"<<endl;
if(1-R/(log(row+1.)/log(2.))=1) cout<<"n/n0=oo"<<endl;
else cout<<"n/n0="<<1/(1-(1-R/(log(row+1.)/log(2.))))<<endl;
for(i=0;i<=row;i++) delete [] pr[i];
delete [] p;delete [] a;delete [] b;delete [] pr;delete [] rr;delete [] ss; return 0;
}

```