1. **Коды ASCII и КОИ-8: характеристики, состав, структура.**

**ASCII** — американская стандартная кодировочная таблица для печатных символов и некоторых специальных кодов. ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.

Помимо знаков внешнего алфавита в начале кода содержатся служебные кодовые комбинации для управления аппаратурой передачи и отображения данных. Эти знаки не отображаются на устройствах ввода и вывода, но аппаратно интерпретируются ими. Структура знаковой части кода ASCII проста: вслед за пробелом идут 15 разделителей в порядке, который легко не запоминается. С 48-й позиции идут десятичные цифры в порядке возрастания. Далее после 7 разделителей непрерывно в алфавитном порядке следуют 26 заглавных латинских букв. Через 32 позиции размещается диапазон малых латинских букв. Структура кода ASCII вполне регулярна и легко запоминаема.

**КОИ-8** — восьмибитовая ASCII-совместимая кодовая страница, разработанная для кодирования букв кириллических алфавитов. Нижняя часть таблицы кодировки (латиница) полностью соответствует кодировке ASCII. Существует также семибитовая версия кодировки, не полностью совместимая с ASCII — КОИ-7.

Разработчики КОИ-8 поместили символы русского алфавита в верхней части кодовой таблицы таким образом, что позиции кириллических символов соответствуют их фонетическим аналогам в английском алфавите в нижней части таблицы. Если в тексте, написанном в КОИ-8, убирать восьмой бит каждого символа, то получается «читаемый» текст, хотя он и написан латинскими символами.

Русский алфавит описывается в кодировке KOI8-R, украинский — в KOI8-U. KOI8-R стал фактически стандартом для русской кириллицы в 1990-х годах в юникс-подобных операционных системах и электронной почте.

1. **Альтернативная и основная кодировки (в сравнении с ASCII).**

**Основная кодировка** была принята в 1987 г. взамен КОИ-8, однако использовалась мало. Основную кодировку поддерживало только оборудование и программное обеспечение, производившееся в СССР (ЕС ПЭВМ, Лексикон), а также некоторые принтеры Epson. На базе основной кодировки была создана ISO 8859-5, но и она не нашла широкого применения. Гораздо более популярной оказалась альтернативная кодировка (с тем же набором символов, но в другом порядке).

**Альтернативная кодировка** — основанная на CP437 кодовая страница, где все специфические европейские символы во второй половине заменены на кириллицу, оставляя псевдографические символы нетронутыми. Окончательным стандартом стала кодировка IBM CP866. В этой кодировке записываются имена файлов в системе FAT. Поныне является популярной стандартной кодировкой Microsoft в среде DOS и OS/2, используется в консоли русифицированных систем семейства Windows NT. Вне среды MS-DOS в Microsoft Windows заменена стандартной кодировкой CP1251, а в операционных системах Windows NT и следующих за ней - кодировкой Юникод.

Основная и альтернативная кодировки отличаются от ASCII способами добавления русских букв в расширенную часть таблицы (8-й бит равен 1) и размещением знаков псевдографики. В основной кодировке русские буквы размещены подряд и без разрывов. В альтернативной кодировке заглавные буквы размещены подряд, а малые – разбиты на два поддиапазона, между которыми для совместимости с кодировкой IBM размещена псевдографика.

1. **Кодировки ISO 8859-5 и СР 1251.**

ISO 8859-5 — 8-битная кодовая страница из семейства кодовых страниц стандарта ISO-8859 для представления кириллицы, была создана в 1988 году на базе «основной кодировки» (все русские буквы сохранили своё расположение, за исключением заглавной Ё). Имеются буквы многих языков, использующих кириллицу, однако в целом ISO 8859-5 — не очень удобная кодировка, поскольку в ней отсутствуют многие нужные символы, такие как тире (—), кавычки-ёлочки («»), градус (°), не установлены буквы нерусских алфавитов и др. Нет также буквы Ґ, используемой в украинской письменности. Порядок символов этой кодовой страницы использовался при размещении букв кириллицы в наборе символов Unicode (со сдвигом вверх на 864 позиции).

Windows-1251 является стандартной 8-битной кодировкой для всех русских версий MS Windows. Windows-1251 выгодно отличается от других 8-битных кириллических кодировок (таких как CP866, KOI8-R и ISO 8859-5) наличием практически всех символов, использующихся в русской типографике для обычного текста; она также содержит все символы для близких к русскому языку языков: украинского, белорусского, сербского и болгарского. В Windows-1251 отсутствуют символы псевдографики, имеющиеся в CP866 и KOI8.

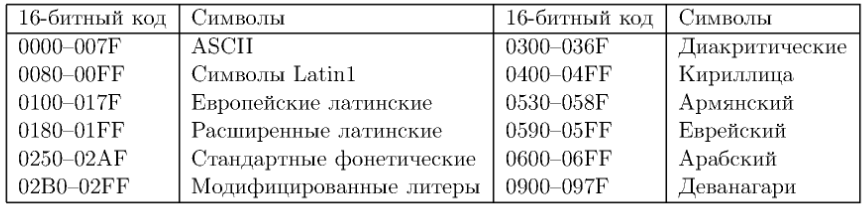
1. **Понятие о кодах EBCDIC, ДКОИ.**

**EBCDIK**(расширенный двоично-десятичный код обмена информацией) являлся первым восьмибитным кодом, разработанным корпорацией IBM для использования на мэйнфреймах собственного производства и совместимых с ними. Восьмибитные коды удобнее семибитных, там есть место и для больших, и для малых букв, и для одного национального алфавита. EBCDIC кодирует буквы латинского алфавита, арабские цифры, некоторые знаки пунктуации и управляющие символы.

**ДКОИ-8** (двоичный код для обмена информацией, 8 бит) — кириллическая кодовая страница, использовавшаяся на компьютерах серии ЕС ЭВМ. Основой для ДКОИ-8 послужила кодировка EBCDIC. Обычно русские буквы, совпадающие по начертанию с какими-либо латинскими, не использовались — вместо них ставили латинские с тем же начертанием (в таблице эти неиспользуемые буквы помечены жёлтым; остальные русские буквы, не совпадающие ни с какими латинскими, помечены зелёным). Буква Ё отсутствовала; не было также заглавной Ъ. Но потребность в локализации (поддержки многоязычия) программного обеспечения и отсутствие простых способов представления текстовых строк привели к переходу на 16-битные кодировки, наиболее известной из которых является Unicode.

1. **Кодировки Unicode, UTF-8. ISO 8859-5 и 10646.**

**Unicode разработан Apple и Xerox в 1988 г. Unicode представляет собой 16-битную кодировку и позволяет кодировать 65536 знаков вместо 256 8-битных знаков. Около 29000 кодовых позиций Unicode не заняты. 6000 - зарезервированы для использования программистами. Unicode решает проблему национальных алфавитов ценой удвоенного расхода памяти, а также позволяет единообразно представлять строки, допуская смешение алфавитов.**



UTF-8 -распространённая кодировка символов Юникода, совместимая с 8-битными форматами передачи текста. В UTF-8 можно кодировать значения кодов символов от 0 до 7FFFFFFF16 включительно . Каждый символ кодируется переменным количеством последовательных 8-битных байт. Количество может варьироваться от 1 до 6 байт включительно (реально только до 4 байт, поскольку использование кодов больше 221 не планируется) и определяется самым первым байтом. UTF-8 является самосинхронизирующейся кодировкой: при потере одного байта последующие байты будут раскодированы корректно. Текст, состоящий только из символов Юникода с номерами меньше 128, при записи в UTF-8 превращается в обычный текст ASCII. И наоборот, в тексте UTF-8 любой байт со значением меньше 128 изображает символ ASCII с тем же кодом.

ISO 8859-5 -8-битная кодовая страница из семейства кодовых страниц стандарта ISO-8859 для представления кириллицы. SO 8859-5 была создана на базе «основной кодировки» (все русские буквы сохранили своё расположение, за исключением заглавной Ё). В целом ISO 8859-5 — не очень удобная кодировка, поскольку в ней отсутствуют многие нужные символы, такие как тире (—), кавычки-ёлочки («»), градус (°) и др. Нет также буквы Ґ, используемой в украинской письменности. Порядок символов этой кодовой страницы использовался при размещении букв кириллицы в наборе символов Unicode (со сдвигом вверх на 864 позиции). В России эта кодировка почти не употребляется, тем не менее на некоторых иностранных системах для русского языка по умолчанию ставится ISO 8859-5.

Международный стандарт ISO 10646дает определение универсальному набору символов – юникоду. UCS содержит символы, необходимые для представления практически всех известных сейчас языков. Основные коды стандарта Unicode составляют первые 65,536 кодов стандарта ISO 10646 и содержат все знаки, в настоящее время определенные ISO 10646. Остальные коды ISO 10646 остаются незаполненными и зарезервированы для будущего расширения. Чтобы отразить знаки, имеющиеся в ISO 10646, стандарт Unicode включил более чем 3600 новых китайских, японских и корейских знаков и более чем 1000 других знаков.

1. **Понятие о клавиатурных раскладках. Основные принципы.**

Раскладка клавиатуры **— соглашение о соответствии типографических символов (букв, цифр, знаков препинания и т. д.) письменного языка клавишам клавиатуры компьютера, пишущей машинки или другого устройства, с помощью которого вводится текст. Раскладка устанавливает несколько соответствий между клавишами и значениями, вводимыми с их помощью.**

Механическая раскладка - стандартная компьютерная клавиатура, также называемая клавиатурой PC/AT, имеет 101 или 102 клавиши, расположение которых подчиняется единой общепринятой схеме, и спроектирована в расчёте на английский алфавит, содержащий 26 букв. По своему назначению клавиши на клавиатуре PC/AT подразделяются на шесть групп:

· функциональные клавиши ( F1 — F12 );

· алфавитно-цифровые клавиши;

· клавиши управления курсором ( Home , End , Page Up , Page Down , Delete , ←, Backspace , ← , → , ↑ , ↓ );

· клавиши цифровой панели;

· специализированные клавиши ( Esc , Print Screen , Pause , Insert и т. д.);

· клавиши-модификаторы ( ⇧ Shift , Ctrl , Alt , Alt Gr , Caps Lock , Num Lock , Scroll Lock ). Визуальная раскладка — маркировка клавиш. Если предполагается, что на клавиатуре будут работать с двумя раскладками, то на клавиши обычно наносят двойные обозначения. Например, черным цветом наносят символы латинской раскладки, а красным цветом — символы национальной раскладки. Переключение между двумя раскладками в операционных системах семейства Microsoft Windows обычно выполняется нажатием комбинации клавиш ⇧ Shift + Alt или ⇧ Shift + Ctrl .

Функциональная раскладка — значения, вводимые одиночным или совместным нажатием клавиш. Она обеспечивается минимум двумя составными частями – аппаратной и программной раскладками. Возможно существование нескольких раскладок для одного письменного языка. Например, существуют раскладки ЙЦУКЕН и фонетическая ЯВЕРТЫ для русского языка; QWERTY, Дворака и Colemak для английского языка. Одна и та же раскладка может использоваться и для нескольких языков. Например, латинская раскладка QWERTY используется в пяти языках, хотя в каждом из случаев её названия в операционных системах семейства MS Windows различаются. Одна и та же раскладка может по-разному адаптироваться для использования на компьютере. Например, в MS Windows поддерживаются два варианта адаптации русской раскладки ЙЦУКЕН для компьютерных клавиатур — «Русская» и «Русская (Машинопись)», различающихся расположением небуквенных знаков и буквы Ё ё.

1. **Раскладки QWERTY и ЙЦУКЕН и соответствие между знаками кириллицы и латинского алфавита для обычной и фонетической латино-кириллических раскладок.**

Стандартные раскладки клавиатур имеют частотно-рычажное происхождение: они распределяют клавиши по убыванию известной частоты встречаемости букв данного языка от середины к краям клавиатуры, от развитого указательного до перерытой ногой мизинца.   
Кириллические раскладки имеют в своей основе другие частотные распределения букв (QWERTY -> ЙЦУКЕН) и могут быть нанесены на клавиатуру верхним регистром. Другой способ сделать это - так называемая фонетическая раскладка - навеяна КОИ-8: русские буквы нанесены на клавиши однотипных по произношению или написанию латинских ( A-А, Б-B, В-V, Г-G, Д-D, Ф-F, K-K, O-O и так далее), за исключением трёх букв.

1. **Раскладки Дворака и Diktor.**

Клавиатура Дворака — раскладка клавиатуры, запатентованная Августом Двораком и Вильямом Дилли в 1936 для набора английских символов как альтернатива раскладки QWERTY. Является наиболее совершенной, именно на ней установлен мировой рекорд скорости печати на клавиатуре. Причину быстроты печати по Двораку является не только частотный принцип размещения, но и чередование рук. Существует Дворак для программистов. Эта раскладка рассчитана на людей, пишущих код на C, Java, Pascal, HTML, CSS и XML. Раскладка основана на размещении клавиш в упрощённой раскладке Дворака с некоторыми улучшениями специально для программистов. Существуют также расположения клавиш, предназначенные для набора одной рукой. Русской версии клавиатуры Дворака не существует, потому что стандартная русская раскладка уже достаточно оптимизирована.

В 2006 г. была предложена новая раскладка для кириллицы – Diktor. Эта раскладка рассчитана на слепое десятипальцевое мягкое печатание именно на компьютерной клавиатуре и свободна от рычажных атавизмов. К плюсам раскладки можно отнести удобное расположение точки и запятой, а так же чередование рук, то есть расположение букв и символов сбалансировано.

1. **Позиционные системы счисления.**

Позиционная система счисления - система счисления, в которой значение каждого числового знака в записи числа зависит от его позиции. Примером позиционной системы является хорошо известная десятичная система счисления. Примером непозиционной системы – римская система. Выполнение арифметических действий над числами в непозиционной системе весьма неудобно. Поэтому позиционные системы в настоящее время получили наибольшее распространение. Позиционная система счисления определяется целым числом b > 1, называемым основанием системы счисления. Система счисления с основанием b также называется b-ричной. Наиболее распространенными системами счисления являются системы с основанием 2, 3, 8, 10 и 16. Системы счисления с нецелыми основаниями практически не используются. Чем больше основание системы счисления, тем более экономна запись, и тем более сложная аппаратура требуется.

С позиционной системой счисления также связаны обратная и дополнительная кодировки отрицательных чисел в ЭВМ. В дополнительном коде отрицательные числа представляются большими положительными, вдвое сужая диапазон допустимых положительных чисел. Дополнительный код положительного числа совпадает с его прямым кодом. При этом, чем больше положительное значение кодового слова отрицательного числа, тем меньше абсолютная величина представляемого им числа. Обратный код легко получается из прямого, но неудобен неоднозначностью представления 0 и требует изменения способа выполнения арифметических операций.

1. **Представление целых чисел в ЭВМ.**

Целые числа могут представляться в компьтере со знаком или без знака. Целые числа без знака обычно занимают в памяти один или два байта и принимают в однобайтовом формате значения от 000000002 до 111111112, а в двухбайтовом формате от 00000000 000000002 до 11111111 111111112. Целые числа со знаком обычно занимают в памяти компьютера один два или четыре байта, при этом самый левый (старший) разряд содержит информацию о знаке числа. Знак "+" кодируется нулём, а "-" — единицей.

В компьютерной технике применяется три формы записи (кодирования) целых чисел со знаком: прямой код, обратный код, дополнительный код. Последние две формы применяются особенно широко, так как позволяют упростить конструкцию арифметико-логического устройства компьютера путём замены разнообразных арифметических операций операцией сложения.

Положительные числа в прямом, обратном и дополнительном кодах изображаются одинаково — двоичными кодами с цифрой 0 в знаковом разряде.

Обычно отрицательные десятичные числа при вводе в машину автоматически преобразуются в обратный или дополнительный код и в таком виде перемещаются и участвуют в операциях. При вводе таких чисел из машины происходит обратное преобразование в отрицательные десятичные числа.

1. **Перевод чисел из одной системы счисления в другую.**

Перевод в десятичную систему счисления

Если число в b-ричной системе счисления равно  то для перевода в десятичную систему вычисляем такую сумму 

1011002 = 1 · 25 + 0 · 24 + 1 · 23 + 1 · 22 + 0 · 21 + 0 · 20 = 32 + 8 + 4 + 0 = 4410

Перевод из десятичной системы счисления

Целая часть. Последовательно делим целую часть десятичного числа на основание, пока десятичное число не станет равно нулю. Полученные при делении остатки являются цифрами нужного числа. Число в новой системе записывают, начиная с последнего остатка.

Дробная часть. Дробную часть десятичного числа умножаем на основание системы, в которую требуется перевести. Отделяем целую часть. Продолжаем умножать дробную часть на основание новой системы, пока она не станет равной 0. Число в новой системе составляют целые части результатов умножения в порядке, соответствующем их получению. Перевод из двоичной в восьмеричную и шестнадцатеричную системы. Для перевода в восьмеричную систему двоичное число разбивается на триады, для перевода в шестнадцатеричную – на тетрады.

1. **Особенности целочисленной арифметики в ЭВМ.**

Целочисленная арифметика на ЭВМ имеет три очень существенных преимущества по сравнению с вещественной арифметикой:

* целые числа всегда представимы своими точными значениями;
* операции целочисленной арифметики дают точные результаты;
* операции целочисленной арифметики выполняются быстрее, чем операции вещественной («плавающей») арифметики.

Недостатком целого типа данных является сравнительно узкий диапазон допустимых значений (для типа Integer — от -32768 до 32767). При исполнении программы автоматически не контролируется выход значения целой величины за эти границы. В этом случае получается ошибочный результат. Если такая опасность существует, то программист должен сам предусматривать в своей программе предупреждение целочисленного переполнения. Чаще всего целый тип используется для представления счетчиков, номеров, индексов и других целочисленных величин.

Вещественный тип данных не является упорядоченным. Вещественные числа в памяти ЭВМ представляются в формате с плавающей точкой, т.е. в виде совокупности пары чисел — целого порядка и нормализованной мантиссы. Поскольку размер ячейки памяти ограничен, в большинстве случаев мантисса оказывается «обрезанной», иными словами, приближенной. Точное представление в памяти имеет лишь дискретное конечное множество вещественных значений.

1. **Научная (экспоненциальная) форма записи числа. Машинное представление с плавающей точкой.**

**Экспоненциальная запись** — представление действительных чисел в виде мантиссы и порядка. Удобна при представлении очень больших и очень малых чисел, а также для унификации их написания.

 , где

§ N — записываемое число;

§ M — мантисса;

§ n — основание показательной функции;

§ p (целое) — порядок;

§  — характеристика числа.

На компьютере (в частности в тексте компьютерных программ) экспоненциальную запись записывают в виде **MEp**, где:

M — мантисса,

E (exponent) — буква E, означающая «\*10^» («…умножить на десять в степени…»).

p — порядок,

Например:

 (это элементарный заряд)

1. **Различия представлений числовых текстовых данных в ЭВМ.**

Вся информация в ЭВМ хранится в виде наборов бит, то есть комбинаций 0 и 1. Числа представляются двоичными комбинациями в соответствии с числовыми форматами, принятыми для работы в данной ЭВМ, а символьный код устанавливает соответствие букв и других символов двоичным комбинациям.

Для чисел имеется три числовых формата:

* двоичный с фиксированной точкой;
* двоичный с плавающей запятой;
* двоично-кодированный десятичный (BCD).

Числовая информация, как и текстовая, представляется в виде последовательности символов, но основное отличие заключается не в том, что числовые данные используют более узкий набор символов (цифры, знаки «+», «—» и др.), а в том, что с числовыми данными можно выполнять арифметические операции, а для работы с текстовыми данными используется совсем другой набор операций. Первые вычислительные машины обрабатывали лишь числовую информацию, что нашло отражение в их названии.

В настоящее время большая часть обрабатываемых данных — текстового типа. Графические данные: рисунки, схемы, чертежи, фотографии — наиболее наглядны и доступны для восприятия и интерпретации человеком, так как сразу передают необходимый образ, а числовые и текстовые данные требуют мысленного воссоздания образа. В последнее время в компьютерах все шире используется звуковая и видеоинформация, а также мультимедийная форма представления информации, в которой комбинируются данные всех вышеперечисленных типов.

1. **Использование калькуляторов OC UNIX (bc) и MS Windows для операций с числами в различных системах счисления.**

**bc** (**b**asic **c**alculator) — интерактивный интерпретатор Си-подобного языка, позволяет выполнять вычисления с произвольно заданной точностью. Часто используется как калькулятор в командной строке UNIX-подобных операционных систем. Программа **bc** позволяет производить арифметические вычисления с произвольной точностью над числами произвольной величины, а также содержит некоторые языковые возможности. Имеются средства установки основания системы счисления для входных и выходных данных, а так же для перевода чисел из одной системы счисления в другую.

**Калькулятор** (calc.exe) — компонент Microsoft Windows, имитирующий работу обычного карманного калькулятора. Калькулятор вычисляет арифметические операции и операции со скобками, тригонометрические функции, натуральные, двоичные, десятичные и произвольные логарифмы, степенные и экспоненциальные функции, модуль, факториал. Как выражение, так и  
результат могут быть представлены в любой системе исчисления от двоичной до шестнадцатеричной. В отличие от большинства других калькуляторов, в недесятичных системах исчисления могут быть записаны не только целые, но и дробные числа.