1. **Системы счисления**

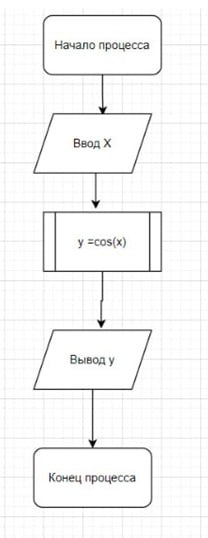
Позиционные системы счисления (десятичная, двоичная, восьмеричная, шестнадцатеричная) − это системы счисления, в которых значение цифры напрямую зависит от её положения в числе.

Двоичная система счисления-состоит всего из двух знаков 0 и 1.

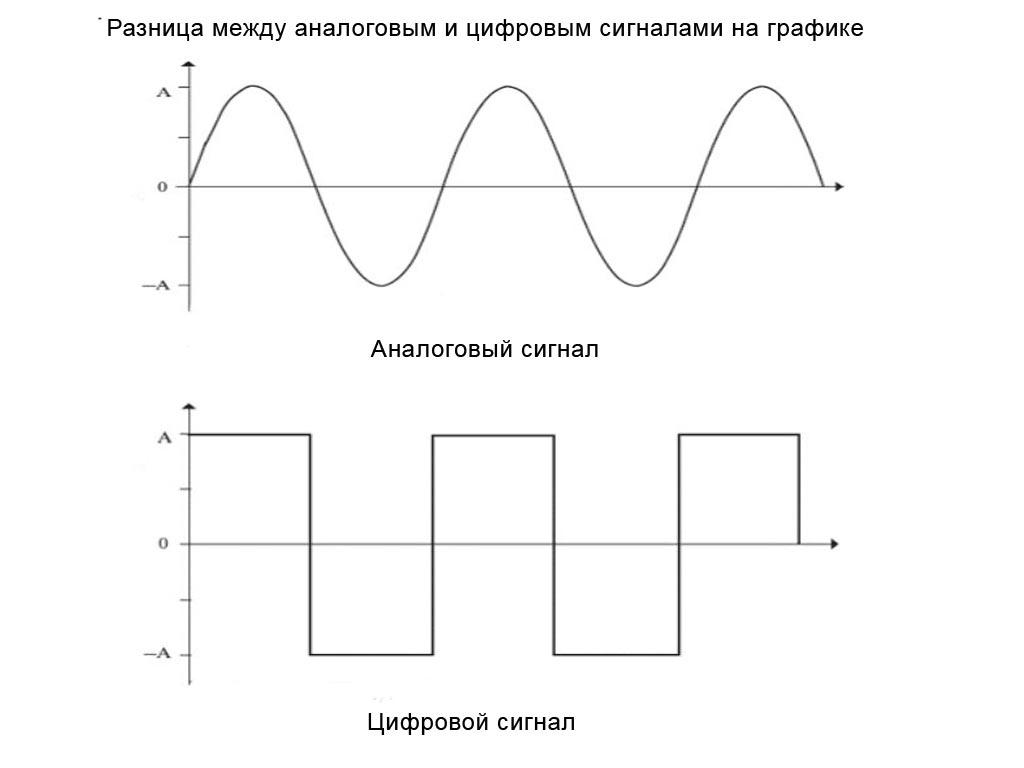
Непозиционные системы счисления-значение цифры не зависит от ее положения в числе. Примером непозиционной системы счисления служит римская система счисления.

Шестнадцатеричная система счисления- это позиционная целочисленная система счисления с основанием 16.

1. **Блок-схема функции y = cos(x)**



1. **Как представить сигнал аналоговый и цифровой?**



Аналоговые сигналы https://studfile.net/html/2706/248/html_lsgI1kedPv.BFWW/img-XQcPeG.png- описываемые как функция (чаще всего действительная от одной или нескольких переменных, например, времени, пространственных координат. При этом значение сигнала может быть определено в любой момент времени как действительное число. Если например, аналоговый сигнал задан как изменение напряжения от времениhttps://studfile.net/html/2706/248/html_lsgI1kedPv.BFWW/img-tMSxzJ.png, то размерность сигнала – вольт.

Цифровые сигналы https://studfile.net/html/2706/248/html_lsgI1kedPv.BFWW/img-5bSZgJ.png, - заданные в виде набора значений или измерений в фиксированные моменты времени. Такие значения принято называть отсчетами сигнала. Каждый отсчет задан с определенной заданной точностью, т.е является числом фиксированной разрядности.

1. **Что такое сигнал и какие сигналы существуют?**

**Сигнал** - материальное воплощение сообщения для использования при передаче, переработке и хранении информации.

**Виды**:

-Анаголовый,

- числовой,

-Детерминированный,

-случайные,

-Сигналы с ограниченной энергией,

- переодические,

-Финитные

**5. Что такое многозадачность?**

Свойство [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) или [среды выполнения](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B5%D0%B4%D0%B0_%D0%B2%D1%8B%D0%BF%D0%BE%D0%BB%D0%BD%D0%B5%D0%BD%D0%B8%D1%8F) обеспечивать возможность параллельной (или [псевдопараллельной](https://ru.wikipedia.org/w/index.php?title=%D0%9F%D1%81%D0%B5%D0%B2%D0%B4%D0%BE%D0%BF%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C&action=edit&redlink=1)) обработки нескольких [задач](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)).

**6. Какие существуют топологии сети?(5 вопрос в вопросах к 10 лк)**

Сетевая тополо́гия — это конфигурация графа, вершинам которого соответствуют конечные узлы сети (компьютеры и коммуникационное оборудование (маршрутизаторы), а рёбрам — физические или информационные связи между вершинами.

Сетевые топологии бывают:

Шина, звезда, кольцо, неполносвязанная, полносвязанная, смешанная.

1) Топология с общей шиной (англ. Bus Topology)

Одна из первых физических топологий. Суть состояла в том, что к одному длинному кабелю подсоединяли все устройства и организовывали локальную сеть. На концах кабеля требовались терминаторы. Преимущество ее было только в простоте установки. С точки зрения работоспособности была крайне не устойчивой.

2) Кольцевая топология (англ. Ring Topology)

В данной топологии каждое устройство подключается к 2-ум соседним. Создавая, таким образом, кольцо. С одного конца компьютер только принимает, а с другого только отправляет. Следующий компьютер играет роль ретранслятора сигнала. Нужда в терминаторах отпала. Соответственно, если где-то кабель повреждался, кольцо размыкалось и сеть становилась не работоспособной.

3) Топология звезда (англ. Star Topology)

Все устройства подключаются к центральному узлу, который уже является ретранслятором. В наше время данная модель используется в локальных сетях, когда к одному коммутатору подключаются несколько устройств, и он является посредником в передаче. Здесь отказоустойчивость значительно выше, чем в предыдущих двух. При обрыве, какого либо кабеля, выпадает из сети только одно устройство. Однако если откажет центральное звено, сеть станет неработоспособной.

4) Полносвязная топология (англ. Full-Mesh Topology)

Все устройства связаны напрямую друг с другом. То есть с каждого на каждый. Данная модель является самой отказоустойчивой, так как не зависит от других. Но строить сети на такой модели сложно и дорого.

5) Неполносвязная топология (англ. Partial-Mesh Topology)

Как правило, вариантов ее несколько. Однако соединение построено не с каждого на каждый, а через дополнительные узлы. То есть узел A, связан напрямую только с узлом B, а узел B связан и с узлом A, и с узлом C. Так вот, чтобы узлу A отправить сообщение узлу C, ему надо отправить сначала узлу B, а узел B в свою очередь отправит это сообщение узлу C.

6) Смешанная топология (англ. Hybrid Topology)(самая популярная)

Представляет собой древовидную структуру, которая объединяет все топологии. Одна из самых отказоустойчивых топологий, так как если у двух площадок произойдет обрыв, то парализована будет связь только между ними, а все остальные объединенные площадки будут работать безотказно.

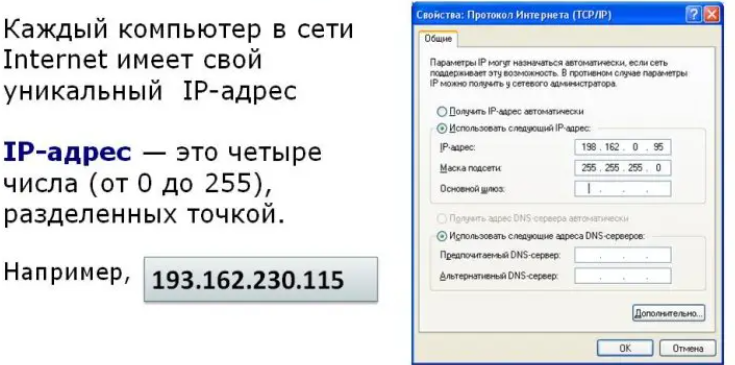
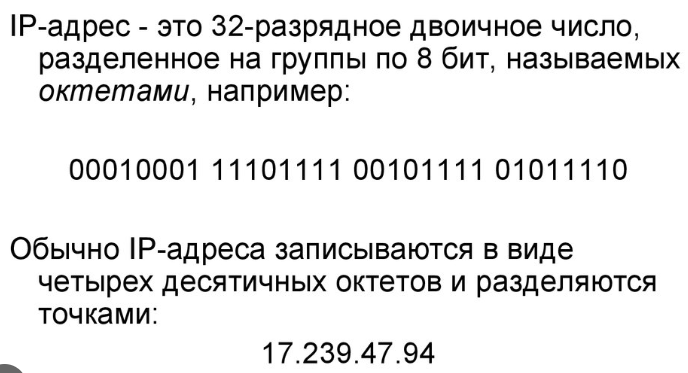
**7. Что такое отладчик в Autamation Studio? Для чего он необходим?(3 вопрос к 10 лк)**

*(именно про отладчик в Automation Studio ничего не нашла в интернете, но этот ответ, я думаю, должен подойти)*

Отладчик - это компьютерная программа, используемая для тестирования и отладки других программ ("целевая" программа). Основное применение отладчика заключается в запуске целевой программы в контролируемых условиях, которые позволяют программисту отслеживать ее выполнение и отслеживать изменения в ресурсах компьютера, которые могут указывать на неисправный код.

**8. Что такое IP-адрес? (1 вопрос к лк 11)**

IP-адрес — уникальный числовой [идентификатор](https://ru.m.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80) устройства в компьютерной [сети](https://ru.m.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C), работающей по протоколу [IP](https://ru.m.wikipedia.org/wiki/IP).

****

**9. Что такое маска сети? (2 вопрос к лк 11)**

Маска подсети является числом, и она определяет диапазон IP-адресов, которые может использовать сеть. С ее помощью сети могут делиться на небольшие подсети, которые подключаются к Интернету. Маска подсети будет обозначать эти подсети.

**Маска подсети** — 32-битное число, служащее битовой маской для разделения сетевой части (адреса подсети) и части хоста IP-адреса. Состоит из последовательности от 0 до 32 двоичных единиц, после которых остаток разрядов представляют двоичные нули. Их смешение недопустимо. Устройства в одной подсети имеют одинаковый адрес подсети и передают данные на канальном уровне.

Устройства в разных подсетях коммуницируют через маршрутизацию. Как и IP-адрес, маска может быть записана в двоично-десятичной форме (например, 255.255.0.0) или в виде префикса в CIDR-нотации — числом от 0 до 32, обозначающего длину маски в битах. Например, в подсети 192.0.2.0/24 значение /24 — это маска, равная 255.255.255.0.

**10. Протокол передачи данных. (3 вопрос к лк 11)**

Протоколы передачи данных — набор правил, который определяет порядок и особенности передачи информации для конкретных случаев.

*IP — Internet Protocol*

Он является ненадёжным, т. е. не подтверждает доставку пакетов получателю и не контролирует целостность данных. По протоколу IP передача данных осуществляется без установки соединения.

Основная задача этого протокола — маршрутизация [датаграмм](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D1%82%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), т. е. определение пути следования данных по узлам сети.

*TCP/IP — Transmission Control Protocol/Internet Protocol*

Это стек протоколов TCP и IP. Первый обеспечивает и контролирует надёжную передачу данных и следит за её целостностью. Второй же отвечает за маршрутизацию для отправки данных. Протокол TCP часто используется более комплексными протоколами.

*UDP — User Datagram Protocol*

Протокол, обеспечивающий передачу данных без предварительного создания соединения между ними. Этот протокол является ненадёжным. В нём пакеты могут не только не дойти, но и прийти не по порядку или вовсе продублироваться. Основное преимущество UDP протокола заключается в скорости доставки данных.

*FTP — File Transfer Protocol*

Протокол передачи файлов. FTP является надёжным протоколом, поэтому гарантирует передачу данных.

Этот протокол работает по принципу клиент-серверной архитектуры. Пользователь проходит аутентификацию (хотя в отдельных случаях может подключаться анонимно) и получает доступ к файловой системе сервера.

*DNS*

Это не только система доменных имён (Domain Name System), но и протокол, без которого эта система не смогла бы работать. Он позволяет клиентским компьютерам запрашивать у DNS-сервера IP-адрес какого-либо сайта, а также помогает обмениваться базами данных между серверами DNS. В работе этого протокола используются TCP и UDP.

*HTTP — HyperText Transfer Protocol*

Является протоколом клиент-серверного взаимодействия без сохранения промежуточного состояния. В роли клиента чаще всего выступает веб-браузер, хотя может быть и, например, поисковый робот. Для обмена информацией протокол HTTP в большинстве случаев использует TCP/IP.

*NTP — Network Time Protocol*

Протокол для синхронизации локальных часов устройства со временем в сети. Благодаря нему протокол выбирает более точный источник времени. NTP работает поверх UDP. Протокол достаточно устойчив к изменениям задержек в сети.

*SSH — Secure SHell*

Протокол для удалённого управления операционной системой с использованием TCP. В SSH шифруется весь трафик, причём с возможностью выбора алгоритма шифрования. В основном это нужно для передачи паролей и другой важной информации.

11. Что такое кодирование?

**Кодирование - это процесс преобразования информации из одного формата в другой, чтобы сделать ее доступной для обработки и передачи**. Это перевод информации с одного языка на другой, на запись в другой системе символов или в другом алфавите. Обычно кодированием называют перевод информации с языка людей на формальный (двоичный код), а декодирование – это переход от двоичного кода к понятному человеку набору символов.

**Первичное кодирование** - это процесс перевода информации из естественного языка в формат, который может быть обработан компьютером.

**Вторичное кодирование** - это процесс присвоения дополнительных значений или метаданных первично закодированным данным

12. Что такое помехоустойчивое кодирование?

**Помехоустойчивое (корректирующее) кодирование - это методы кодирования, которые позволяют обеспечить передачу информации через каналы связи с помехами, такими как шум, без потери данных**. Эти методы кодирования обнаруживают ошибки, что достигается путем добавления дополнительной информации в передаваемый поток данных, которая позволяет восстановить исходную информацию в случае возникновения ошибок. Одним из самых распространенных методов является **код Хэмминга**. **Он добавляет дополнительные биты в исходный поток данных, которые позволяют обнаруживать и исправлять ошибки**, возникшие в процессе передачи.

**Основной принцип – это применение избыточного кода**, причем если для кодирования используется простой код, в него вводят избыточность. Необходимость избыточности объясняется тем, что в простых кодах все кодовые комбинации являются разрешенными, **поэтому при ошибке в любом из разрядов приведет к появлению другой разрешенной комбинации, и обнаружить ошибку будет невозможно**

13. Что такое первичный и вторичный алфавиты?

**Первичный алфавит кодирования - это набор символов, которые могут быть закодированы в битовую последовательность**. Например, ASCII-кодировка, которая используется для представления латинских букв, цифр и некоторых специальных символов.

**Вторичный алфавит кодирования - это набор символов, которые могут быть закодированы в битовую последовательность, но не входят в первичный алфавит**. **Например, UTF-8**, который может кодировать символы из различных языков, включая кириллицу, китайские и японские иероглифы.

**Третичный алфавит кодирования** - **это набор символов, которые могут быть закодированы в битовую последовательность и используются для специальных целей**, например, для сжатия данных или шифрования информации.

14. Как из одного алфавита преобразуют в другой?

Для перевода из одного алфавита кодирования в другой используется процесс, называемый **транскодированием**. **Транскодирование может быть необходимо, когда данные, которые нужно передать, используют один алфавит кодирования, а получатель ожидает данные в другом алфавите**.

**Один из способов транскодирования - это использование таблицы символов**, которая соответствует каждому символу в одном алфавите символ в другом алфавите. Например, для перевода из ASCII в Unicode можно использовать таблицу символов Unicode, которая показывает соответствие между символами ASCII и символами Unicode. **Еще один способ транскодирования - это использование специальных программных библиотек и инструментов.**

15. Что такое лестничные диаграммы и как они используются?

**Лестничные диаграммы (или диаграммы временных интервалов) - это графическое представление последовательности действий во времени**. Они используются для **визуализации последовательности событий, происходящих в системе, и позволяют легко определить порядок выполнения этих событий**. Они широко применяются в сетях передачи данных для описания протоколов связи между устройствами. **Они также могут использоваться в проектах по автоматизации и управлению производственными процессами. Лестничные диаграммы нужны для программирования логических контроллеров. Они представляют собой логические схемы, ориентированы на инженеров автоматизации, обеспечивают наглядный интерфейс логики работы контроллера.** В Automation Studio используются для представления последовательности действий в системе и **позволяют создавать логические схемы управления, которые отображают порядок выполнения операций и связи между различными компонентами**.