

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования Оренбургской области

Управление образования администрации г. Оренбурга

МОАУ "Лицей №7"

РАССМОТРЕНО

на заседании
Методического совета
Протокол № 1
от «22» 08 2024 г.

СОГЛАСОВАНО

Заместитель директора
 Алексеева Э.Н.
«27» 08 2024 г.

УТВЕРЖДЕНО

Директор
 Пушкарёва Н.Г.
Приказ № 238
от «29» 08 2024 г.



РАБОЧАЯ ПРОГРАММА

(ID 1965132)

учебного предмета «Информатика (углублённый уровень)»

для обучающихся 10 – 11 классов

Составитель:

Рыжова Л.А., учитель математики

г. Оренбург 2024 год

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа по информатике (углублённый уровень) на уровне среднего общего образования разработана на основе требований к результатам освоения основной образовательной программы среднего общего образования, представленных в ФГОС СОО, а также федеральной рабочей программы воспитания.

Программа по информатике даёт представление о целях, общей стратегии обучения, воспитания и развития обучающихся средствами учебного предмета «Информатика» на углублённом уровне, устанавливает обязательное предметное содержание, предусматривает его структурирование по разделам и темам курса, определяет распределение его по классам (годам изучения), даёт примерное распределение учебных часов по тематическим разделам курса и рекомендуемую (примерную) последовательность их изучения с учётом межпредметных и внутрипредметных связей, логики учебного процесса, возрастных особенностей обучающихся.

Программа по информатике определяет количественные и качественные характеристики учебного материала для каждого года изучения, в том числе для содержательного наполнения разного вида контроля (промежуточной аттестации обучающихся, всероссийских проверочных работ, государственной итоговой аттестации). Программа по информатике является основой для составления авторских учебных программ и учебников, поурочного планирования курса учителем.

Информатика в среднем общем образовании отражает:

сущность информатики как научной дисциплины, изучающей закономерности протекания и возможности автоматизации информационных процессов в различных системах;

основные области применения информатики, прежде всего информационные технологии, управление и социальную сферу;

междисциплинарный характер информатики и информационной деятельности.

Курс информатики для уровня среднего общего образования является завершающим этапом непрерывной подготовки обучающихся в области информатики и информационно-коммуникационных технологий, опирается на содержание курса информатики уровня основного общего образования и опыт постоянного применения информационно-коммуникационных технологий, даёт теоретическое осмысление, интерпретацию и обобщение этого опыта.

Результаты углублённого уровня изучения учебного предмета «Информатика» ориентированы на получение компетентностей для

последующей профессиональной деятельности как в рамках данной предметной области, так и в смежных с ней областях. Они включают в себя:

овладение ключевыми понятиями и закономерностями, на которых строится данная предметная область, распознавание соответствующих им признаков и взаимосвязей, способность демонстрировать различные подходы к изучению явлений, характерных для изучаемой предметной области;

умение решать типовые практические и теоретические задачи, характерные для использования методов и инструментария данной предметной области;

наличие представлений о данной предметной области как целостной теории (совокупности теорий), основных связях со смежными областями знаний.

В рамках углублённого уровня изучения информатики обеспечивается целенаправленная подготовка обучающихся к продолжению образования в организациях профессионального образования по специальностям, непосредственно связанным с цифровыми технологиями, таким как программная инженерия, информационная безопасность, информационные системы и технологии, мобильные системы и сети, большие данные и машинное обучение, промышленный интернет вещей, искусственный интеллект, технологии беспроводной связи, робототехника, квантовые технологии, системы распределённого реестра, технологии виртуальной и дополненной реальностей.

Основная цель изучения учебного предмета «Информатика» на углублённом уровне среднего общего образования – обеспечение дальнейшего развития информационных компетенций обучающегося, его готовности к жизни в условиях развивающегося информационного общества и возрастающей конкуренции на рынке труда. В связи с этим изучение информатики в 10–11 классах должно обеспечить:

сформированность мировоззрения, основанного на понимании роли информатики, информационных и коммуникационных технологий в современном обществе;

сформированность основ логического и алгоритмического мышления;

сформированность умений различать факты и оценки, сравнивать оценочные выводы, видеть их связь с критериями оценивания и связь критериев с определённой системой ценностей, проверять на достоверность и обобщать информацию;

сформированность представлений о влиянии информационных технологий на жизнь человека в обществе, понимание социального, экономического, политического, культурного, юридического, природного,

эргономического, медицинского и физиологического контекстов информационных технологий;

принятие правовых и этических аспектов информационных технологий, осознание ответственности людей, вовлечённых в создание и использование информационных систем, распространение информации;

создание условий для развития навыков учебной, проектной, научно-исследовательской и творческой деятельности, мотивации обучающихся к саморазвитию.

В содержании учебного предмета «Информатика» выделяются четыре тематических раздела.

Раздел **«Цифровая грамотность»** посвящён вопросам устройства компьютеров и других элементов цифрового окружения, включая компьютерные сети, использованию средств операционной системы, работе в сети Интернет и использованию интернет-сервисов, информационной безопасности.

Раздел **«Теоретические основы информатики»** включает в себя понятийный аппарат информатики, вопросы кодирования информации, измерения информационного объёма данных, основы алгебры логики и компьютерного моделирования.

Раздел **«Алгоритмы и программирование»** направлен на развитие алгоритмического мышления, разработку алгоритмов и оценку их сложности, формирование навыков реализации программ на языках программирования высокого уровня.

Раздел **«Информационные технологии»** посвящён вопросам применения информационных технологий, реализованных в прикладных программных продуктах и интернет-сервисах, в том числе в задачах анализа данных, использованию баз данных и электронных таблиц для решения прикладных задач.

В приведённом далее содержании учебного предмета «Информатика» курсивом выделены дополнительные темы, которые не входят в обязательную программу обучения, но могут быть предложены для изучения отдельным мотивированным и способным обучающимся.

Углублённый уровень изучения информатики рекомендуется для технологического профиля, ориентированного на инженерную и информационную сферы деятельности. Углублённый уровень изучения информатики обеспечивает: подготовку обучающихся, ориентированных на специальности в области информационных технологий и инженерные специальности, участие в проектной и исследовательской деятельности, связанной с современными направлениями отрасли информационно-

коммуникационных технологий, подготовку к участию в олимпиадах и сдаче Единого государственного экзамена по информатике.

Последовательность изучения тем в пределах одного года обучения может быть изменена по усмотрению учителя при подготовке рабочей программы и поурочного планирования.

Общее число часов, рекомендованных для изучения информатики – 272 часа: в 10 классе – 136 часов (4 часа в неделю), в 11 классе – 136 часов (4 часа в неделю).

СОДЕРЖАНИЕ ОБУЧЕНИЯ

10 КЛАСС

Цифровая грамотность

Требования техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения.

Принципы работы компьютеров и компьютерных систем. Архитектура фон Неймана. Автоматическое выполнение программы процессором. Оперативная, постоянная и долговременная память. Обмен данными с помощью шин. Контроллеры внешних устройств. Прямой доступ к памяти.

Основные тенденции развития компьютерных технологий. Параллельные вычисления. Многопроцессорные системы. Суперкомпьютеры. Распределённые вычислительные системы и обработка больших данных. Мобильные цифровые устройства и их роль в коммуникациях. Встроенные компьютеры. Микроконтроллеры. Роботизированные производства.

Программное обеспечение компьютеров и компьютерных систем. Виды программного обеспечения и их назначение. Особенности программного обеспечения мобильных устройств. Параллельное программирование. Системное программное обеспечение. Операционные системы. Утилиты. Драйверы устройств. Установка и деинсталляция программного обеспечения.

Файловые системы. Принципы размещения и именования файлов в долговременной памяти. Шаблоны для описания групп файлов.

Программное обеспечение. Лицензирование программного обеспечения и цифровых ресурсов. Проприетарное и свободное программное обеспечение. Коммерческое и некоммерческое использование программного обеспечения и цифровых ресурсов. Ответственность, устанавливаемая законодательством Российской Федерации за неправомерное использование программного обеспечения и цифровых ресурсов.

Принципы построения и аппаратные компоненты компьютерных сетей. Сетевые протоколы. Сеть Интернет. Адресация в сети Интернет. Протоколы стека TCP/IP. Система доменных имён.

Разделение IP-сети на подсети с помощью масок подсетей. Сетевое администрирование. Получение данных о сетевых настройках компьютера. Проверка наличия связи с узлом сети. Определение маршрута движения пакетов.

Виды деятельности в сети Интернет. Сервисы Интернета. Геоинформационные системы. Геолокационные сервисы реального времени

(например, локация мобильных телефонов, определение загруженности автомагистралей), интернет-торговля, бронирование билетов и гостиниц.

Государственные электронные сервисы и услуги. Социальные сети – организация коллективного взаимодействия и обмена данными. Сетевой этикет: правила поведения в киберпространстве. Проблема подлинности полученной информации. Открытые образовательные ресурсы.

Техногенные и экономические угрозы, связанные с использованием информационно-коммуникационных технологий. Общие проблемы защиты информации и информационной безопасности. Средства защиты информации в компьютерах, компьютерных сетях и автоматизированных информационных системах. Правовое обеспечение информационной безопасности.

Предотвращение несанкционированного доступа к личной конфиденциальной информации, хранящейся на персональном компьютере, мобильных устройствах. Вредоносное программное обеспечение и способы борьбы с ним. Антивирусные программы. Организация личного архива информации. Резервное копирование. Парольная защита архива.

Шифрование данных. Симметричные и несимметричные шифры. Шифры простой замены. Шифр Цезаря. Шифр Виженера. Алгоритм шифрования RSA.

Теоретические основы информатики

Информация, данные и знания. Информационные процессы в природе, технике и обществе.

Непрерывные и дискретные величины и сигналы. Необходимость дискретизации информации, предназначенной для хранения, передачи и обработки в цифровых системах.

Двоичное кодирование. Равномерные и неравномерные коды. Декодирование сообщений, записанных с помощью неравномерных кодов. Условие Фано. Построение однозначно декодируемых кодов с помощью дерева. Единицы измерения количества информации. Алфавитный подход к оценке количества информации.

Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления. Алгоритм перевода целого числа из P -ичной системы счисления в десятичную. Алгоритм перевода конечной P -ичной дроби в десятичную. Алгоритм перевода целого числа из десятичной системы счисления в P -ичную. Перевод конечной десятичной дроби в P -ичную. Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними. Арифметические

операции в позиционных системах счисления. Троичная уравновешенная система счисления. Двоично-десятичная система счисления.

Кодирование текстов. Кодировка ASCII. Однобайтные кодировки. Стандарт UNICODE. Кодировка UTF-8. Определение информационного объёма текстовых сообщений.

Кодирование изображений. Оценка информационного объёма графических данных при заданных разрешении и глубине кодирования цвета. Цветовые модели. Векторное кодирование. Форматы графических файлов. Трёхмерная графика. Фрактальная графика.

Кодирование звука. Оценка информационного объёма звуковых данных при заданных частоте дискретизации и разрядности кодирования.

Алгебра логики. Понятие высказывания. Высказывательные формы (предикаты). Кванторы существования и всеобщности.

Логические операции. Таблицы истинности. Логические выражения. Логические тождества. Доказательство логических тождеств с помощью таблиц истинности. Логические операции и операции над множествами.

Законы алгебры логики. Эквивалентные преобразования логических выражений. Логические уравнения и системы уравнений.

Логические функции. Зависимость количества возможных логических функций от количества аргументов. Полные системы логических функций.

Канонические формы логических выражений. Совершенные дизъюнктивные и конъюнктивные нормальные формы, алгоритмы их построения по таблице истинности.

Логические элементы в составе компьютера. Триггер. Сумматор. Многоразрядный сумматор. Построение схем на логических элементах по заданному логическому выражению. Запись логического выражения по логической схеме.

Представление целых чисел в памяти компьютера. Ограниченность диапазона чисел при ограничении количества разрядов. Переполнение разрядной сетки. Беззнаковые и знаковые данные. Знаковый бит. Двоичный дополнительный код отрицательных чисел.

Побитовые логические операции. Логический, арифметический и циклический сдвиги. Шифрование с помощью побитовой операции «исключающее ИЛИ».

Представление вещественных чисел в памяти компьютера. Значащая часть и порядок числа. Диапазон значений вещественных чисел. Проблемы хранения вещественных чисел, связанные с ограничением количества разрядов. Выполнение операций с вещественными числами, накопление ошибок при вычислениях.

Алгоритмы и программирование

Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.

Этапы решения задач на компьютере. Инструментальные средства: транслятор, отладчик, профилировщик. Компиляция и интерпретация программ. Виртуальные машины.

Интегрированная среда разработки. Методы отладки программ. Использование трассировочных таблиц. Отладочный вывод. Пошаговое выполнение программы. Точки останова. Просмотр значений переменных.

Язык программирования (Python, Java, C++, C#). Типы данных: целочисленные, вещественные, символьные, логические. Ветвления. Сложные условия. Циклы с условием. Циклы по переменной. Взаимозаменяемость различных видов циклов. Инвариант цикла. Составление цикла с использованием заранее определённого инварианта цикла.

Документирование программ. Использование комментариев. Подготовка описания программы и инструкции для пользователя.

Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления: разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры.

Нахождение всех простых чисел в заданном диапазоне. Представление числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень.

Обработка данных, хранящихся в файлах. Текстовые и двоичные файлы. Файловые переменные (файловые указатели). Чтение из файла. Запись в файл.

Разбиение задачи на подзадачи. Подпрограммы (процедуры и функции). Рекурсия. Рекурсивные объекты (фракталы). Рекурсивные процедуры и функции. Использование стека для организации рекурсивных вызовов.

Использование стандартной библиотеки языка программирования. Подключение библиотек подпрограмм сторонних производителей. Модульный принцип построения программ.

Численные методы. Точное и приближённое решения задачи. Численные методы решения уравнений: метод перебора, метод половинного деления. Приближённое вычисление длин кривых. Вычисление площадей фигур с помощью численных методов (метод прямоугольников, метод трапеций). Поиск максимума (минимума) функции одной переменной методом половинного деления.

Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно.

Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию). Линейный поиск заданного значения в массиве.

Сортировка одномерного массива. Простые методы сортировки (метод пузырька, метод выбора, сортировка вставками). Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Двоичный поиск в отсортированном массиве.

Двумерные массивы (матрицы). Алгоритмы обработки двумерных массивов: заполнение двумерного числового массива по заданным правилам, поиск элемента в двумерном массиве, вычисление максимума (минимума) и суммы элементов двумерного массива, перестановка строк и столбцов двумерного массива.

Информационные технологии

Текстовый процессор. Редактирование и форматирование. Проверка орфографии и грамматики. Средства поиска и автозамены в текстовом процессоре. Использование стилей. Структурированные текстовые документы. Сноски, оглавление. Коллективная работа с документами. Инструменты рецензирования в текстовых процессорах. Облачные сервисы. Деловая переписка. Реферат. Правила цитирования источников и оформления библиографических ссылок. Оформление списка литературы. Знакомство с компьютерной вёрсткой текста. Технические средства ввода текста. Специализированные средства редактирования математических текстов.

Анализ данных. Основные задачи анализа данных: прогнозирование, классификация, кластеризация, анализ отклонений. Последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов. Программные средства и интернет-сервисы для обработки и представления данных. Большие данные. Машинное обучение. Интеллектуальный анализ данных.

Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Вычисление коэффициента корреляции двух рядов данных. Построение столбчатых, линейчатых и круговых диаграмм. Построение графиков функций. Подбор линии тренда, решение задач прогнозирования.

Численное решение уравнений с помощью подбора параметра. Оптимизация как поиск наилучшего решения в заданных условиях. Целевая функция, ограничения. Локальные и глобальный минимумы целевой функции. Решение задач оптимизации с помощью электронных таблиц.

11 КЛАСС

Теоретические основы информатики

Теоретические подходы к оценке количества информации. Закон аддитивности информации. Формула Хартли. Информация и вероятность. Формула Шеннона.

Алгоритмы сжатия данных. Алгоритм RLE. Алгоритм Хаффмана. Алгоритм LZW. Алгоритмы сжатия данных с потерями. Уменьшение глубины кодирования цвета. Основные идеи алгоритмов сжатия JPEG, MP3.

Скорость передачи данных. Зависимость времени передачи от информационного объёма данных и характеристик канала связи. Причины возникновения ошибок при передаче данных. Коды, позволяющие обнаруживать и исправлять ошибки, возникающие при передаче данных. Расстояние Хэмминга. Кодирование с повторением битов. Коды Хэмминга.

Системы. Компоненты системы и их взаимодействие. Системный эффект. Управление как информационный процесс. Обратная связь.

Модели и моделирование. Цель моделирования. Соответствие модели моделируемому объекту или процессу, цели моделирования. Формализация прикладных задач.

Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики).

Графы. Основные понятия. Виды графов. Описание графов с помощью матриц смежности, весовых матриц, списков смежности. Решение алгоритмических задач, связанных с анализом графов (построение оптимального пути между вершинами графа, определение количества различных путей между вершинами ориентированного ациклического графа).

Деревья. Бинарное дерево. Деревья поиска. Способы обхода дерева. Представление арифметических выражений в виде дерева. Дискретные игры двух игроков с полной информацией. Построение дерева перебора вариантов,

описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции. Выигрышные стратегии.

Средства искусственного интеллекта. Сервисы машинного перевода и распознавания устной речи. Когнитивные сервисы. Идентификация и поиск изображений, распознавание лиц. Самообучающиеся системы. Искусственный интеллект в компьютерных играх. Использование методов искусственного интеллекта в обучающих системах. Использование методов искусственного интеллекта в робототехнике. Интернет вещей. Перспективы развития компьютерных интеллектуальных систем. Нейронные сети.

Алгоритмы и программирование

Формализация понятия алгоритма. Машина Тьюринга как универсальная модель вычислений. Тезис Чёрча–Тьюринга.

Оценка сложности вычислений. Время работы и объём используемой памяти, их зависимость от размера исходных данных. Оценка асимптотической сложности алгоритмов. Алгоритмы полиномиальной сложности. Переборные алгоритмы. Примеры различных алгоритмов решения одной задачи, которые имеют различную сложность.

Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена».

Многоразрядные целые числа, задачи длинной арифметики.

Словари (ассоциативные массивы, отображения). Хэш-таблицы. Построение алфавитно-частотного словаря для заданного текста.

Стеки. Анализ правильности скобочного выражения. Вычисление арифметического выражения, записанного в постфиксной форме.

Очереди. Использование очереди для временного хранения данных.

Алгоритмы на графах. Построение минимального остовного дерева взвешенного связного неориентированного графа. Количество различных путей между вершинами ориентированного ациклического графа. Алгоритм Дейкстры.

Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения. Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева.

Динамическое программирование как метод решения задач с сохранением промежуточных результатов. Задачи, решаемые с помощью динамического программирования: вычисление рекурсивных функций, подсчёт количества вариантов, задачи оптимизации.

Понятие об объектно-ориентированном программировании. Объекты и классы. Свойства и методы объектов. Объектно-ориентированный анализ.

Разработка программ на основе объектно-ориентированного подхода. Инкапсуляция, наследование, полиморфизм.

Среды быстрой разработки программ. Проектирование интерфейса пользователя. Использование готовых управляемых элементов для построения интерфейса.

Обзор языков программирования. Понятие о парадигмах программирования.

Информационные технологии

Этапы компьютерно-математического моделирования: постановка задачи, разработка модели, тестирование модели, компьютерный эксперимент, анализ результатов моделирования.

Дискретизация при математическом моделировании непрерывных процессов. Моделирование движения. Моделирование биологических систем. Математические модели в экономике. Вычислительные эксперименты с моделями.

Обработка результатов эксперимента. Метод наименьших квадратов. Оценка числовых параметров моделируемых объектов и процессов. Восстановление зависимостей по результатам эксперимента.

Вероятностные модели. Методы Монте-Карло. Имитационное моделирование. Системы массового обслуживания.

Табличные (реляционные) базы данных. Таблица – представление сведений об однотипных объектах. Поле, запись. Ключ таблицы. Работа с готовой базой данных. Заполнение базы данных. Поиск, сортировка и фильтрация данных. Запросы на выборку данных. Запросы с параметрами. Вычисляемые поля в запросах.

Многотабличные базы данных. Типы связей между таблицами. Внешний ключ. Целостность базы данных. Запросы к многотабличным базам данных.

Интернет-приложения. Понятие о серверной и клиентской частях сайта. Технология «клиент – сервер», её достоинства и недостатки. Основы языка HTML и каскадных таблиц стилей (CSS). Сценарии на языке JavaScript. Формы на веб-странице.

Размещение веб-сайтов. Услуга хостинга. Загрузка файлов на сайт.

Ввод изображений с использованием различных цифровых устройств (цифровых фотоаппаратов и микроскопов, видеокамер, сканеров и других устройств). Графический редактор. Разрешение. Кадрирование. Исправление перспективы. Гистограмма. Коррекция уровней, коррекция цвета. Обесцвечивание цветных изображений. Ретушь. Работа с областями. Фильтры.

Многослойные изображения. Текстовые слои. Маска слоя. Каналы. Сохранение выделенной области. Подготовка иллюстраций для веб-сайтов. Анимированные изображения.

Векторная графика. Прimitives. Изменение порядка элементов. Выравнивание, распределение. Группировка. Кривые. Форматы векторных рисунков. Использование контуров. Векторизация растровых изображений.

Принципы построения и редактирования трёхмерных моделей. Сеточные модели. Материалы. Моделирование источников освещения. Камеры. Аддитивные технологии (3D-принтеры). Понятие о виртуальной реальности и дополненной реальности.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ ПО ИНФОРМАТИКЕ (УГЛУБЛЁННЫЙ УРОВЕНЬ) НА УРОВНЕ СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

ЛИЧНОСТНЫЕ РЕЗУЛЬТАТЫ

Личностные результаты отражают готовность и способность обучающихся руководствоваться сформированной внутренней позицией личности, системой ценностных ориентаций, позитивных внутренних убеждений, соответствующих традиционным ценностям российского общества, расширение жизненного опыта и опыта деятельности в процессе реализации средствами учебного предмета основных направлений воспитательной деятельности.

В результате изучения информатики на уровне среднего общего образования у обучающегося будут сформированы следующие личностные результаты:

1) гражданского воспитания:

осознание своих конституционных прав и обязанностей, уважение закона и правопорядка, соблюдение основополагающих норм информационного права и информационной безопасности;

готовность противостоять идеологии экстремизма, национализма, ксенофобии, дискриминации по социальным, религиозным, расовым, национальным признакам в виртуальном пространстве;

2) патриотического воспитания:

ценностное отношение к историческому наследию, достижениям России в науке, искусстве, технологиях, понимание значения информатики как науки в жизни современного общества;

3) духовно-нравственного воспитания:

сформированность нравственного сознания, этического поведения;

способность оценивать ситуацию и принимать осознанные решения, ориентируясь на морально-нравственные нормы и ценности, в том числе в сети Интернет;

4) эстетического воспитания:

эстетическое отношение к миру, включая эстетику научного и технического творчества;

способность воспринимать различные виды искусства, в том числе основанного на использовании информационных технологий;

5) физического воспитания:

сформированность здорового и безопасного образа жизни, ответственного отношения к своему здоровью, в том числе за счёт соблюдения

требований безопасной эксплуатации средств информационных и коммуникационных технологий;

б) трудового воспитания:

готовность к активной деятельности технологической и социальной направленности, способность инициировать, планировать и самостоятельно выполнять такую деятельность;

интерес к сферам профессиональной деятельности, связанным с информатикой, программированием и информационными технологиями, основанными на достижениях науки информатики и научно-технического прогресса, умение совершать осознанный выбор будущей профессии и реализовывать собственные жизненные планы;

готовность и способность к образованию и самообразованию на протяжении всей жизни;

7) экологического воспитания:

осознание глобального характера экологических проблем и путей их решения, в том числе с учётом возможностей информационно-коммуникационных технологий;

8) ценности научного познания:

сформированность мировоззрения, соответствующего современному уровню развития науки, достижениям научно-технического прогресса и общественной практики, за счёт понимания роли информационных ресурсов, информационных процессов и информационных технологий в условиях цифровой трансформации многих сфер жизни современного общества;

осознание ценности научной деятельности, готовность осуществлять проектную и исследовательскую деятельность индивидуально и в группе.

В процессе достижения личностных результатов освоения программы по информатике у обучающихся совершенствуется эмоциональный интеллект, предполагающий сформированность:

саморегулирования, включающего самоконтроль, умение принимать ответственность за своё поведение, способность адаптироваться к эмоциональным изменениям и проявлять гибкость, быть открытым новому;

внутренней мотивации, включающей стремление к достижению цели и успеху, оптимизм, инициативность, умение действовать, исходя из своих возможностей;

эмпатии, включающей способность понимать эмоциональное состояние других, учитывать его при осуществлении коммуникации, способность к сочувствию и сопереживанию;

социальных навыков, включающих способность выстраивать отношения с другими людьми, заботиться, проявлять интерес и разрешать конфликты.

МЕТАПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

В результате изучения информатики на уровне среднего общего образования у обучающегося будут сформированы метапредметные результаты, отраженные в универсальных учебных действиях, а именно – познавательные универсальные учебные действия, коммуникативные универсальные учебные действия, регулятивные универсальные учебные действия, совместная деятельность.

Познавательные универсальные учебные действия

1) базовые логические действия:

самостоятельно формулировать и актуализировать проблему, рассматривать её всесторонне;

устанавливать существенный признак или основания для сравнения, классификации и обобщения;

определять цели деятельности, задавать параметры и критерии их достижения;

выявлять закономерности и противоречия в рассматриваемых явлениях;

разрабатывать план решения проблемы с учётом анализа имеющихся материальных и нематериальных ресурсов;

вносить коррективы в деятельность, оценивать соответствие результатов целям, оценивать риски последствий деятельности;

координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия;

развивать креативное мышление при решении жизненных проблем.

2) базовые исследовательские действия:

владеть навыками учебно-исследовательской и проектной деятельности, навыками разрешения проблем, способностью и готовностью к самостоятельному поиску методов решения практических задач, применению различных методов познания;

осуществлять различные виды деятельности по получению нового знания, его интерпретации, преобразованию и применению в различных учебных ситуациях, в том числе при создании учебных и социальных проектов;

формировать научный тип мышления, владеть научной терминологией, ключевыми понятиями и методами;

ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;

выявлять причинно-следственные связи и актуализировать задачу, выдвигать гипотезу её решения, находить аргументы для доказательства своих утверждений, задавать параметры и критерии решения;

анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях;

давать оценку новым ситуациям, оценивать приобретённый опыт;

осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду;

уметь переносить знания в познавательную и практическую области жизнедеятельности;

уметь интегрировать знания из разных предметных областей;

выдвигать новые идеи, предлагать оригинальные подходы и решения, ставить проблемы и задачи, допускающие альтернативные решения.

3) работа с информацией:

владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления;

создавать тексты в различных форматах с учётом назначения информации и целевой аудитории, выбирая оптимальную форму представления и визуализации;

оценивать достоверность, легитимность информации, её соответствие правовым и морально-этическим нормам;

использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;

владеть навыками распознавания и защиты информации, информационной безопасности личности.

Коммуникативные универсальные учебные действия

1) общение:

осуществлять коммуникации во всех сферах жизни;

распознавать невербальные средства общения, понимать значение социальных знаков, распознавать предпосылки конфликтных ситуаций и смягчать конфликты;

владеть различными способами общения и взаимодействия, аргументированно вести диалог, уметь смягчать конфликтные ситуации;

развёрнуто и логично излагать свою точку зрения с использованием языковых средств.

2) совместная деятельность:

понимать и использовать преимущества командной и индивидуальной работы;

выбирать тематику и методы совместных действий с учётом общих интересов и возможностей каждого члена коллектива;

принимать цели совместной деятельности, организовывать и координировать действия по их достижению: составлять план действий, распределять роли с учётом мнений участников, обсуждать результаты совместной работы;

оценивать качество своего вклада и каждого участника команды в общий результат по разработанным критериям;

предлагать новые проекты, оценивать идеи с позиции новизны, оригинальности, практической значимости;

осуществлять позитивное стратегическое поведение в различных ситуациях, проявлять творчество и воображение, быть инициативным.

Регулятивные универсальные учебные действия

1) самоорганизация:

самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;

самостоятельно составлять план решения проблемы с учётом имеющихся ресурсов, собственных возможностей и предпочтений;

давать оценку новым ситуациям;

расширять рамки учебного предмета на основе личных предпочтений;

делать осознанный выбор, аргументировать его, брать ответственность за решение;

оценивать приобретённый опыт;

способствовать формированию и проявлению широкой эрудиции в разных областях знаний, постоянно повышать свой образовательный и культурный уровень.

2) самоконтроль:

давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям;

владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и

оснований, использовать приёмы рефлексии для оценки ситуации, выбора верного решения;

оценивать риски и своевременно принимать решения по их снижению;

принимать мотивы и аргументы других при анализе результатов деятельности.

3) принятия себя и других:

принимать себя, понимая свои недостатки и достоинства;

принимать мотивы и аргументы других при анализе результатов деятельности;

признавать своё право и право других на ошибку;

развивать способность понимать мир с позиции другого человека.

ПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

В процессе изучения курса информатики углублённого уровня *в 10 классе* обучающимися будут достигнуты следующие предметные результаты:

владение представлениями о роли информации и связанных с ней процессов в природе, технике и обществе, понятиями «информация», «информационный процесс», «система», «компоненты системы», «системный эффект», «информационная система», «система управления»;

владение методами поиска информации в сети Интернет, умение критически оценивать информацию, полученную из сети Интернет;

умение характеризовать большие данные, приводить примеры источников их получения и направления использования, умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений), понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация данных, интерпретация результатов;

понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров, тенденций развития компьютерных технологий;

владение навыками работы с операционными системами, основными видами программного обеспечения для решения учебных задач по выбранной специализации;

наличие представлений о компьютерных сетях и их роли в современном мире, о базовых принципах организации и функционирования компьютерных сетей, об общих принципах разработки и функционирования интернет-приложений;

понимание угроз информационной безопасности, использование методов и средств противодействия этим угрозам, соблюдение мер безопасности, предотвращающих незаконное распространение персональных данных, соблюдение требований техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения, понимание правовых основ использования компьютерных программ, баз данных и работы в сети Интернет;

понимание основных принципов дискретизации различных видов информации, умение определять информационный объём текстовых, графических и звуковых данных при заданных параметрах дискретизации, умение определять среднюю скорость передачи данных, оценивать изменение времени передачи при изменении информационного объёма данных и характеристик канала связи;

умение использовать при решении задач свойства позиционной записи чисел, алгоритма построения записи числа в позиционной системе счисления с заданным основанием и построения числа по строке, содержащей запись этого числа в позиционной системе счисления с заданным основанием, умение выполнять арифметические операции в позиционных системах счисления;

умение выполнять преобразования логических выражений, используя законы алгебры логики, умение строить логическое выражение в дизъюнктивной и конъюнктивной нормальных формах по заданной таблице истинности, исследовать область истинности высказывания, содержащего переменные, решать несложные логические уравнения и системы уравнений;

понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, нахождение всех простых чисел в заданном диапазоне, обработка многоразрядных целых чисел, анализ символьных строк и других), алгоритмов поиска и сортировки, умение определять сложность изучаемых в курсе базовых алгоритмов (суммирование элементов массива, сортировка массива, переборные алгоритмы, двоичный поиск) и приводить примеры нескольких алгоритмов разной сложности для решения одной задачи;

владение универсальным языком программирования высокого уровня (Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных, умение использовать основные управляющие конструкции, умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных, определять, при каких исходных данных возможно получение указанных результатов, выявлять данные, которые могут привести к ошибке в работе программы, формулировать предложения по улучшению программного кода;

умение создавать структурированные текстовые документы и демонстрационные материалы с использованием возможностей современных программных средств и облачных сервисов;

умение использовать электронные таблицы для анализа, представления и обработки данных (включая вычисление суммы, среднего арифметического, наибольшего и наименьшего значений, решение уравнений, выбор оптимального решения, подбор линии тренда, решение задач прогнозирования).

В процессе изучения курса информатики углублённого уровня *в 11 классе* обучающимися будут достигнуты следующие предметные результаты:

умение строить неравномерные коды, допускающие однозначное декодирование сообщений (префиксные коды), использовать простейшие коды, которые позволяют обнаруживать и исправлять ошибки при передаче данных, строить код, обеспечивающий наименьшую возможную среднюю длину сообщения при известной частоте символов, пояснять принципы работы простых алгоритмов сжатия данных;

умение решать алгоритмические задачи, связанные с анализом графов (задачи построения оптимального пути между вершинами графа, определения количества различных путей между вершинами ориентированного ациклического графа), умение использовать деревья при анализе и построении кодов и для представления арифметических выражений, при решении задач поиска и сортировки, умение строить дерево игры по заданному алгоритму, разрабатывать и обосновывать выигрышную стратегию игры;

умение разрабатывать и реализовывать в виде программ базовые алгоритмы, умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья), использовать базовые операции со структурами данных, применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк, использовать при разработке программ библиотеки подпрограмм, знать функциональные возможности инструментальных средств среды разработки, умение использовать средства отладки программ в среде программирования, умение документировать программы;

умение создавать веб-страницы;

владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними, умение использовать табличные (реляционные) базы данных (составлять запросы в базах данных, выполнять сортировку и поиск записей в базе данных, наполнять разработанную базу данных) и справочные системы;

умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования, оценивать соответствие модели моделируемому объекту или процессу, представлять результаты моделирования в наглядном виде;

умение организовывать личное информационное пространство с использованием различных средств цифровых технологий, понимание возможностей цифровых сервисов государственных услуг, цифровых образовательных сервисов;

понимание основных принципов работы, возможностей и ограничения применения технологий искусственного интеллекта в различных областях, наличие представлений о круге решаемых задач машинного обучения (распознавания, классификации и прогнозирования) наличие представлений об использовании информационных технологий в различных профессиональных сферах.

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

10 КЛАСС

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Раздел 1. Цифровая грамотность					
1.1	Компьютер - универсальное устройство обработки данных	6	1		https://kpolyakov.spb.ru/ https://resh.edu.ru/
1.2	Программное обеспечение	7			https://kpolyakov.spb.ru/ https://resh.edu.ru/
1.3	Компьютерные сети	7			https://kpolyakov.spb.ru/ https://resh.edu.ru/
1.4	Информационная безопасность	8		2	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		28			
Раздел 2. Теоретические основы информатики					
2.1	Представление информации в компьютере	18		2	https://kpolyakov.spb.ru/ https://resh.edu.ru/
2.2	Основы алгебры логики	15		1	https://kpolyakov.spb.ru/ https://resh.edu.ru/
2.3	Компьютерная арифметика	8	1	1	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		41			
Раздел 3. Алгоритмы и программирование					

3.1	Введение в программирование	16		0.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.2	Вспомогательные алгоритмы	10		2	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.3	Численные методы	6	0	3	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.4	Алгоритмы обработки символьных данных	7		1	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.5	Алгоритмы обработки массивов	11		3.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		50			
Раздел 4. Информационные технологии					
4.1	Обработка текстовых документов	8	0	2.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
4.2	Анализ данных	9		3	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		17			
Резервное время		0	0		
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		136	2	21.5	

11 КЛАСС

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Раздел 1. Теоретические основы информатики					
1.1	Информация и информационные процессы	13	1	2.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
1.2	Моделирование	10		2	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		23			
Раздел 2. Алгоритмы и программирование					
2.1	Элементы теории алгоритмов	7		1	https://kpolyakov.spb.ru/ https://resh.edu.ru/
2.2	Алгоритмы и структуры данных	30		10	https://kpolyakov.spb.ru/ https://resh.edu.ru/
2.3	Основы объектно-ориентированного программирования	17	1	4.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		54			
Раздел 3. Информационные технологии					
3.1	Компьютерно-математическое моделирование	11		2	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.2	Базы данных	13		4	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.3	Веб-сайты	15		4	https://kpolyakov.spb.ru/ https://resh.edu.ru/

3.4	Компьютерная графика	10		3.5	https://kpolyakov.spb.ru/ https://resh.edu.ru/
3.5	3D-моделирование	10		3	https://kpolyakov.spb.ru/ https://resh.edu.ru/
Итого по разделу		59			
Резервное время		0	0		
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		136	2	36.5	

ПОУРОЧНОЕ ПЛАНИРОВАНИЕ

10 КЛАСС

№ п/п	Тема урока	Количество часов			Дата изучения	Электронные цифровые образовательные ресурсы
		Всего	Контрольные работы	Практические работы		
1	Требования техники безопасности и гигиены при работе с компьютерами и другими компонентами цифрового окружения	1			1 неделя сентября	https://resh.edu.ru/subject/lesson/6471/start/51669/
2	Принципы работы компьютеров и компьютерных систем. Архитектура фон Неймана. Автоматическое выполнение программы процессором. Оперативная, постоянная и долговременная память.	1			1 неделя сентября	https://resh.edu.ru/subject/lesson/4715/start/325090/
3	Обмен данными с помощью шин. Контроллеры внешних устройств. Прямой доступ к памяти.	1			1 неделя сентября	https://resh.edu.ru/subject/lesson/5425/start/15091/
4	Автоматическое выполнение программы процессором. Основные тенденции развития компьютерных технологий. Параллельные вычисления. Многопроцессорные системы. Суперкомпьютеры.	1			1 неделя сентября	https://resh.edu.ru/subject/lesson/5425/start/15091/
5	Оперативная, постоянная и долговременная память. Контроллеры внешних устройств. Прямой доступ к памяти. Входная работа.	1	1		2 неделя сентября	https://resh.edu.ru/subject/lesson/5425/start/15091/
6	Современные компьютерные технологии. Распределённые вычислительные системы и обработка больших данных. Мобильные цифровые устройства и их роль в коммуникациях. Встроенные компьютеры. Микроконтроллеры. Роботизированные производства.	1			2 неделя сентября	https://resh.edu.ru/subject/lesson/5421/start/35815/
7	Программное обеспечение компьютеров, компьютерных систем и мобильных	1			2 неделя сентября	https://resh.edu.ru/subject/lesson/5421/start/35815/

	устройств. Виды программного обеспечения и их назначение. Особенности программного обеспечения мобильных устройств. Параллельное программирование.					
8	Системное программное обеспечение. Операционные системы	1			2 неделя сентября	https://resh.edu.ru/subject/lesson/5421/start/35815/
9	Утилиты. Драйверы устройств. Параллельное программирование	1			3 неделя сентября	https://resh.edu.ru/subject/lesson/5421/start/35815/
10	Инсталляция и деинсталляция программного обеспечения	1			3 неделя сентября	https://resh.edu.ru/subject/lesson/5421/start/35815/
11	Файловые системы. Принципы размещения и именования файлов в долговременной памяти. Шаблоны для описания групп файлов	1			3 неделя сентября	
12	Законодательство Российской Федерации в области программного обеспечения и данных. Программное обеспечение. Лицензирование программного обеспечения и цифровых ресурсов. Проприетарное и свободное программное обеспечение. Коммерческое и некоммерческое использование программного обеспечения и цифровых ресурсов. Ответственность, устанавливаемая законодательством Российской Федерации за неправомерное использование программного обеспечения и цифровых ресурсов.	1			3 неделя сентября	
13	Принципы построения и аппаратные компоненты компьютерных сетей. Сетевые протоколы.	1			4 неделя сентября	
14	Сеть Интернет. Адресация в сети Интернет. Протоколы стека TCP/IP. Система доменных имён.	1			4 неделя сентября	
15	Разделение IP-сети на подсети с помощью масок подсетей	1			4 неделя сентября	

16	Сетевое администрирование. Получение данных о сетевых настройках компьютера. Проверка наличия связи с узлом сети. Определение маршрута движения пакетов.	1			4 неделя сентября	
17	Виды деятельности в сети Интернет. Сервисы Интернета. Государственные электронные сервисы и услуги. Геоинформационные системы. Геолокационные сервисы реального времени (например, локация мобильных телефонов, определение загруженности автомагистралей), интернет-торговля, бронирование билетов и гостиниц.	1			1 неделя октября	
18	Государственные электронные сервисы и услуги. Социальные сети – организация коллективного взаимодействия и обмена данными. Сетевой этикет: правила поведения в киберпространстве. Проблема подлинности полученной информации. Открытые образовательные ресурсы.	1			1 неделя октября	
19	Техногенные и экономические угрозы, связанные с использованием информационно-коммуникационных технологий. Общие проблемы защиты информации и информационной безопасности. Средства защиты информации в компьютерах, компьютерных сетях и автоматизированных информационных системах. Правовое обеспечение информационной безопасности.	1			1 неделя октября	
20	Информационная безопасность	1			1 неделя октября	
21	Предотвращение несанкционированного доступа к личной конфиденциальной информации, хранящейся на персональном компьютере, мобильных устройствах. Вредоносное программное	1			2 неделя октября	

	обеспечение и способы борьбы с ним. Антивирусные программы.					
22	Вредоносные программное обеспечение и методы борьбы с ним	1			2 неделя октября	
23	Практическая работа по теме "Антивирусные программы"	1		0.5	2 неделя октября	
24	Организация личного архива информации. Резервное копирование. Парольная защита архива	1			2 неделя октября	
25	Шифрование данных	1			3 неделя октября	
26	Симметричные и несимметричные шифры. Шифры простой замены. Шифр Цезаря. Шифр Виженера.	1			3 неделя октября	
27	Алгоритм шифрования RSA. Стеганография	1			3 неделя октября	
28	Практическая работа по теме "Шифрование данных"	1		1	3 неделя октября	
29	Информация, данные и знания. Информационные процессы в природе, технике и обществе	1			4 неделя октября	
30	Непрерывные и дискретные величины и сигналы. Необходимость дискретизации информации, предназначенной для хранения, передачи и обработки в цифровых системах	1			4 неделя октября	
31	Двоичное кодирование. Равномерные и неравномерные коды. Декодирование сообщений, записанных с помощью неравномерных кодов	1			4 неделя октября	
32	Условие Фано. Построение однозначно декодируемых кодов с помощью дерева. Граф Ал. А. Маркова	1			4 неделя октября	
33	Единицы измерения количества информации. Алфавитный подход к оценке количества информации	1				https://resh.edu.ru/subject/lesson/6469/start/15059/
34	Системы счисления. Развёрнутая запись целых и дробных чисел в позиционной	1				https://resh.edu.ru/subject/lesson/5620/start/15124/

	системе счисления. Свойства позиционной записи числа: количество цифр в записи, признак делимости числа на основание системы счисления.					
35	Перевод чисел из одной системы счисления в другую. Перевод конечной десятичной дроби в Р-ичную.	1				https://resh.edu.ru/subject/lesson/5620/start/15124/
36	Двоично-десятичная система счисления	1				https://resh.edu.ru/subject/lesson/5620/start/15124/
37	Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними	1				https://resh.edu.ru/subject/lesson/5620/start/15124/
38	Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними	1				
39	Двоичная, восьмеричная и шестнадцатеричная системы счисления, связь между ними	1				
40	Двоично-десятичная система счисления.	1				
41	Арифметические операции в позиционных системах счисления	1				https://resh.edu.ru/subject/lesson/5423/start/35985/
42	Троичная уравновешенная система счисления	1	1			https://resh.edu.ru/subject/lesson/5423/start/35985/
43	Кодирование текстов. Кодировка ASCII. Однобайтные кодировки. Стандарт UNICODE. Кодировка UTF-8. Определение информационного объёма текстовых сообщений.	1				
44	Растровое кодирование изображений. Кодирование изображений. Оценка информационного объёма графических данных при заданных разрешении и глубине кодирования цвета.	1				https://resh.edu.ru/subject/lesson/5556/start/166550/
45	Практическая работа по теме "Дискретизация графической информации"	1		0.5		https://resh.edu.ru/subject/lesson/5556/start/166550/
46	Цветовые модели. Векторное кодирование. Форматы файлов.	1				https://resh.edu.ru/subject/lesson/5556/start/166550/

	Трёхмерная графика. Фрактальная графика					
47	Кодирование звука. Оценка информационного объёма звуковых данных при заданных частоте дискретизации и разрядности кодирования	1				https://resh.edu.ru/subject/lesson/5556/start/166550/
48	Практическая работа по теме "Дискретизация звуковой информации"	1		0.5		https://resh.edu.ru/subject/lesson/5556/start/166550/
49	Основы алгебры логики.	1				https://resh.edu.ru/subject/lesson/5426/start/163620/
50	Логические операции. Таблицы истинности	1				https://resh.edu.ru/subject/lesson/5426/start/163620/
51	Логические выражения. Логические тождества. Доказательство логических тождеств с помощью таблиц истинности	1				https://resh.edu.ru/subject/lesson/5426/start/163620/
52	Практическая работа по теме «Построение и анализ таблиц истинности в табличном процессоре»	1		0.5		https://resh.edu.ru/subject/lesson/5426/start/163620/
53	Логические операции и операции над множествами	1				
54	Логические операции и операции над множествами	1				
55	Законы алгебры логики. Эквивалентные преобразования логических выражений	1				
56	Логические уравнения и системы уравнений	1				https://resh.edu.ru/subject/lesson/4713/start/202991/
57	Логические функции. Зависимость количества возможных логических функций от количества аргументов. Полные системы логических функций	1				https://resh.edu.ru/subject/lesson/4713/start/202991/
58	Канонические формы логических выражений. Совершенные дизъюнктивные и конъюнктивные нормальные формы, алгоритмы их построения по таблице истинности	1				https://resh.edu.ru/subject/lesson/4713/start/202991/
59	Логические элементы в составе компьютера	1				

60	Триггер. Сумматор. Многоразрядный сумматор	1				
61	Построение схем на логических элементах. Запись логического выражения по логической схеме	1				
62	Микросхемы и технология их производства	1				
63	Представление целых чисел в памяти компьютера. Ограниченность диапазона чисел при ограничении количества разрядов. Переполнение разрядной сетки	1				
64	Беззнаковые и знаковые данные. Знаковый бит. Двоичный дополнительный код отрицательных чисел	1				
65	Побитовые логические операции. Логический, арифметический и циклический сдвиги	1				
66	Шифрование с помощью побитовой операции «исключающее ИЛИ»	1				
67	Представление и хранение в памяти компьютера вещественных чисел	1		1		
68	Выполнение операций с вещественными числами, накопление ошибок при вычислениях	1				
69	Практическая работа по теме «Изучение поразрядного машинного представления целых и вещественных чисел»	1		1		
70	Анализ алгоритмов. Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат.	1				
71	Этапы решения задач на компьютере. Инструментальные средства: транслятор, отладчик, профилировщик	1				

72	Среда программирования. Компиляция и интерпретация программ. Виртуальные машины. Интегрированная среда разработки	1				
73	Методы отладки программ	1				
74	Язык программирования Python. Типы переменных в языке программирования. Типы данных: целочисленные, вещественные, символьные, логические.	1				
75	Обработка целых чисел	1				
76	Обработка вещественных чисел	1				
77	Случайные и псевдослучайные числа	1				
78	Ветвления. Сложные условия	1				
79	Циклы с условием	1				
80	Циклы по переменной. Взаимозаменяемость различных видов циклов	1		0		
81	Обработка натуральных чисел с использованием циклов	1				
82	Нахождение всех простых чисел в заданном диапазоне Практическая работа по теме «Решение задач методом перебора»	1		1		
83	Инвариант цикла	1				
84	Документирование программ. Использование комментариев. Подготовка описания программы и инструкции для пользователя.	1				
85	Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления: разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры.	1				
86	Нахождение всех простых чисел в заданном диапазоне. Представление	1				

	числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень.					
87	Обработка данных, хранящихся в файлах. Текстовые и двоичные файлы. Файловые переменные (файловые указатели).	1				
88	Чтение из файла. Запись в файл.	1				
89	Разбиение задачи на подзадачи	1				
90	Использование стандартной библиотеки языка программирования. Подключение библиотек подпрограмм сторонних производителей	1				
91	Подпрограммы (процедуры и функции)	1				
92	Подпрограммы (процедуры и функции)	1				
93	Практическая работа по теме "Разработка подпрограмм"	1		1		
94	Рекурсия. Рекурсивные объекты (фракталы). Рекурсивные процедуры и функции. Использование стека для организации рекурсивных вызовов	1				
95	Практическая работа по теме "Рекурсивные подпрограммы"	1		1		
96	Использование стандартной библиотеки языка программирования. Подключение библиотек подпрограмм сторонних производителей.	1				
97	Модульный принцип построения программ	1				
98	Численные методы. Точное и приближённое решения задачи. Численные методы решения уравнений: метод перебора, метод половинного деления. Приближённое вычисление длин кривых.	1				
99	Практическая работа по теме «Численное решение уравнений»	1		1		

100	Использование дискретизации в вычислительных задачах	1				
101	Практическая работа по теме «Приближённое вычисление длин кривых и площадей фигур»	1		1		
102	Практическая работа по теме «Поиск максимума (минимума) функции»	1		1		
103	Обработка символьных данных. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке. Встроенные функции языка программирования для обработки символьных строк.	1				
104	Алгоритмы обработки символьных строк: разбиение строки на слова по пробельным символам	1				
105	Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно.	1				
106	Алгоритмы обработки символьных строк: поиск подстроки внутри данной строки; замена найденной подстроки на другую строку	1				
107	Практическая работа по теме "Обработка строк с использованием функций стандартной библиотеки языка программирования"	1		1		
108	Генерация слов в заданном алфавите	1				
109	Массивы и последовательности чисел. Практическая работа по теме "Заполнение массива"	1		0.5		

110	Обобщённые характеристики массива	1				
111	Линейный поиск заданного значения в массиве. Практическая работа по теме "Линейный поиск заданного значения в массиве"	1		1		
112	Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию).	1				
113	Практическая работа по теме "Поиск минимального (максимального) элемента в числовом массиве"	1		1		
114	Сортировка одномерного массива. Простые методы сортировки. Практическая работа по теме "Простые методы сортировки массива"	1		0.5		
115	Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Практическая работа по теме "Быстрая сортировка массива"	1		0.5		
116	Двоичный поиск в отсортированном массиве. Практическая работа по теме "Двоичный поиск"	1		0.5		
117	Двумерные массивы (матрицы)	1				
118	Алгоритмы обработки матриц	1				
119	Алгоритмы обработки двумерных массивов: заполнение двумерного числового массива по заданным правилам, поиск элемента в двумерном массиве, вычисление максимума (минимума) и суммы элементов двумерного массива, перестановка строк и столбцов двумерного массива.	1				
120	Решение задач анализа данных	1				

121	Проверка орфографии и грамматики. Средства поиска и автозамены в текстовом процессоре. Использование стилей. Структурированные текстовые документы. Сноски, оглавление. Коллективная работа с документами. Инструменты рецензирования в текстовых процессорах. Облачные сервисы. Деловая переписка. Реферат. Правила цитирования источников и оформления библиографических ссылок. Оформление списка литературы. Знакомство с компьютерной вёрсткой текста.	1				https://resh.edu.ru/subject/lesson/5225/start/203084/
122	Средства текстового процессора	1				https://resh.edu.ru/subject/lesson/5225/start/203084/
123	Технические средства ввода текста. Специализированные средства редактирования математических текстов.	1				
124	Компьютерная вёрстка текста	1				
125	Практическая работа по теме "Вёрстка документов с математическими формулами"	1		1		
126	Инструменты рецензирования	1				
127	Практическая работа по теме "Многостраничные документы"	1		1		
128	Облачные сервисы. Коллективная работа с документами. Практическая работа по теме "Коллективная работа с документами"	1		0.5		
129	Анализ данных. Большие данные. сновные задачи анализа данных: прогнозирование, классификация, кластеризация, анализ отклонений. Последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и/или построение модели, преобразование данных, визуализация	1				

	данных, интерпретация результатов. Программные средства и интернет-сервисы для обработки и представления данных.					
130	Машинное обучение. Интеллектуальный анализ данных.	1				
131	Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Вычисление коэффициента корреляции двух рядов данных.	1				
132	Практическая работа по теме "Анализ данных с помощью электронных таблиц". Построение столбчатых, линейчатых и круговых диаграмм. Промежуточная аттестация. Комплекс заданий стандартизированной формы.	1		1		
133	Построение графиков функций. Практическая работа по теме "Наглядное представление результатов статистической обработки данных в виде диаграмм средствами редактора электронных таблиц"	1		0.5		
134	Линии тренда. Практическая работа по теме "Подбор линии тренда, прогнозирование"	1		0.5		
135	Подбор параметра. Практическая работа по теме "Численное решение уравнений с помощью подбора параметра"	1		0.5		
136	Оптимизация как поиск наилучшего решения в заданных условиях. Практическая работа по теме "Решение задач оптимизации с помощью электронных таблиц"	1		0.5		
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		136	2	21.5		

11 КЛАСС

№ п/п	Тема урока	Количество часов			Дата изучения	Электронные цифровые образовательные ресурсы
		Всего	Контрольные работы	Практические работы		
1	Количество информации. Теоретические подходы к оценке количества информации.	1				https://resh.edu.ru/subject/lesson/6469/start/15059/
2	Количество информации. Теоретические подходы к оценке количества информации.	1				https://resh.edu.ru/subject/lesson/6469/start/15059/
3	Закон аддитивности информации. Формула Хартли. Информация и вероятность. Формула Шеннона.	1				https://resh.edu.ru/subject/lesson/6455/start/10503/
4	Алгоритмы сжатия данных. Алгоритм RLE.	1				
5	Алгоритм Хаффмана	1				
6	Практическая работа по теме "Сжатие данных с помощью алгоритма Хаффмана"	1		1		
7	Алгоритм LZW	1				
8	Алгоритмы сжатия данных с потерями. Практическая работа по теме "Сжатие данных с потерями (алгоритмы JPEG, MP3)"	1		0.5		

9	Скорость передачи данных. Зависимость времени передачи от информационного объёма данных и характеристик канала связи.	1				
10	Скорость передачи данных. Зависимость времени передачи от информационного объёма данных и характеристик канала связи.	1				
11	Причины возникновения ошибок при передаче данных. Коды, позволяющие обнаруживать и исправлять ошибки, возникающие при передаче данных. Расстояние Хэмминга. Кодирование с повторением битов. Коды Хэмминга.	1				
12	Помехоустойчивые коды	1				
13	Практическая работа по теме "Помехоустойчивые коды"	1		1		
14	Системы. Компоненты системы и их взаимодействие. Системный эффект. Управление как информационный процесс. Обратная связь	1				

15	Модели и моделирование. Цель моделирования. Соответствие модели моделируемому объекту или процессу, цели моделирования. Формализация прикладных задач.	1				
16	Представление результатов моделирования в виде, удобном для восприятия человеком. Графическое представление данных (схемы, таблицы, графики).	1				
17	Графы. Основные понятия. Виды графов. Описание графов с помощью матриц смежности, весовых матриц, списков смежности. Решение алгоритмических задач, связанных с анализом графов (построение оптимального пути между вершинами графа, определение количества различных путей между вершинами ориентированного ациклического графа)	1				
18	Решение задач с помощью графов	1				
19	Деревья. Бинарное дерево. Деревья поиска. Способы	1				

	обхода дерева. Представление арифметических выражений в виде дерева. Дискретные игры двух игроков с полной информацией.					
20	Основы теории игр. Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции.	1				
21	Выигрышные стратегии.	1				
22	Построение дерева перебора вариантов, описание стратегии игры в табличной форме. Выигрышные и проигрышные позиции.	1				
23	Практическая работа по теме "Поиск выигрышной стратегии в игре с полной информацией"	1		1		
24	Средства искусственного интеллекта. Сервисы машинного перевода и распознавания устной речи. Когнитивные сервисы. Идентификация и поиск изображений, распознавание лиц. Самообучающиеся системы.	1				

25	Искусственный интеллект в компьютерных играх. Использование методов искусственного интеллекта в обучающих системах. Использование методов искусственного интеллекта в робототехнике. Интернет вещей. Перспективы развития компьютерных интеллектуальных систем. Нейронные сети.	1				
26	Практическая работа по теме "Средства искусственного интеллекта"	1		1		
27	Формализация понятия алгоритма. Машина Тьюринга как универсальная модель вычислений. Тезис Чёрча—Тьюринга	1				
28	Практическая работа по теме "Составление простой программы для машины Тьюринга"	1		1		
29	Машина Поста	1				
30	Нормальные алгорифмы Маркова	1				
31	Алгоритмически неразрешимые задачи. Задача останова.	1				

	Невозможность автоматической отладки программ					
32	Сложность вычислений. Оценка сложности вычислений. Время работы и объём используемой памяти, их зависимость от размера исходных данных. Оценка асимптотической сложности алгоритмов. Алгоритмы полиномиальной сложности. Переборные алгоритмы. Примеры различных алгоритмов решения одной задачи, которые имеют различную сложность.	1				
33	Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена»	1				
34	Практическая работа по теме "Поиск простых чисел в заданном диапазоне"	1		0.5		
35	Многоразрядные целые числа, задачи длинной арифметики	1				
36	Практическая работа по теме "Реализация вычислений с многоразрядными числами"	1		0.5		

37	Словари (ассоциативные массивы, отображения). Хэш-таблицы. Построение алфавитно-частотного словаря для заданного текста	1				
38	Практическая работа по теме "Построение алфавитно-частотного словаря для заданного текста"	1		0.5		
39	Анализ текста на естественном языке. Выделение последовательностей по шаблону. Регулярные выражения. Частотный анализ	1				
40	Практическая работа по теме "Анализ текста на естественном языке"	1		0.5		
41	Стеки. Анализ правильности скобочного выражения	1				
42	Вычисление арифметического выражения, записанного в постфиксной форме	1				
43	Практическая работа по теме "Вычисление арифметического выражения, записанного в постфиксной форме"	1		1		
44	Очереди. Использование очереди для временного хранения данных	1				

45	Практическая работа по теме "Использование очереди"	1		1		
46	Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения. Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева.	1				
47	Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения	1				
48	Практическая работа по теме "Использование деревьев для вычисления арифметических выражений"	1		1		
49	Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева	1				
50	Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева	1				
51	Алгоритмы на графах. Построение минимального	1				

	остовного дерева взвешенного связного неориентированного графа					
52	Обход графа в глубину. Обход графа в ширину	1				
53	Количество различных путей между вершинами ориентированного ациклического графа	1				
54	Алгоритм Дейкстры.	1				
55	Практическая работа по теме "Вычисление длины кратчайшего пути между вершинами графа (алгоритм Дейкстры)"	1		1		
56	Алгоритм Флойда—Уоршалла	1				
57	Задачи, решаемые с помощью динамического программирования: вычисление рекурсивных функций	1				
58	Практическая работа по теме "Вычисление рекурсивных функций с помощью динамического программирования"	1		1		
59	Задачи, решаемые с помощью динамического	1				

	программирования: подсчёт количества вариантов					
60	Практическая работа по теме "Подсчёт количества вариантов с помощью динамического программирования"	1		1		
61	Задачи, решаемые с помощью динамического программирования: задачи оптимизации	1				
62	Понятие о парадигмах программирования. Обзор языков программирования	1				
63	Понятие об объектно-ориентированном программировании	1				
64	Объекты и классы. Свойства и методы объектов	1				
65	Объектно-ориентированный анализ	1				
66	Практическая работа по теме "Использование готовых классов в программе"	1		1		
67	Разработка программ на основе объектно-ориентированного подхода	1				
68	Практическая работа "Разработка простой	1		1		

	программы с использованием классов"					
69	Инкапсуляция. Практическая работа по теме "Разработка класса, использующего инкапсуляцию"	1		0.5		
70	Наследование. Полиморфизм	1				
71	Практическая работа по теме "Разработка иерархии классов"	1		1		
72	Среды быстрой разработки программ. Проектирование интерфейса пользователя	1				
73	Проектирование интерфейса пользователя	1				
74	Использование готовых управляемых элементов для построения интерфейса	1				
75	Практическая работа по теме "Разработка программы с графическим интерфейсом"	1		1		
76	Обзор языков программирования. Понятие о парадигмах программирования. Изучение второго языка программирования	1				
77	Изучение второго языка программирования	1				

78	Этапы компьютерно-математического моделирования	1				
79	Дискретизация при математическом моделировании непрерывных процессов. Моделирование движения	1				
80	Практическая работа по теме "Моделирование движения"	1		1		
81	Моделирование биологических систем. Практическая работа по теме "Моделирование биологических систем"	1		0.5		
82	Математические модели в экономике. Вычислительные эксперименты с моделями	1				
83	Вероятностные модели. Практическая работа по теме "Имитационное моделирование с помощью метода Монте-Карло". Системы массового обслуживания.	1		0.5		
84	Компьютерное моделирование систем управления	1				
85	Обработка результатов эксперимента	1				

86	Табличные (реляционные) базы данных	1				
87	Таблица – представление сведений об однотипных объектах. Поле, запись. Ключ таблицы. Работа с готовой базой данных. Заполнение БД	1				
88	Поиск, сортировка и фильтрация данных. Запросы на выборку данных. Запросы с параметрами. Вычисляемые поля в запросах	1				
89	Практическая работа по теме "Работа с готовой базой данных"	1		1		
90	Многотабличные базы данных. Типы связей между таблицами. Внешний ключ. Целостность базы данных	1				
91	Практическая работа по теме "Разработка многотабличной базы данных"	1		1		
92	Запросы к многотабличным базам данных	1				
93	Практическая работа по теме "Запросы к многотабличной базе данных"	1		1		
94	Язык управления данными SQL	1				

95	Практическая работа по теме "Управление данными с помощью языка SQL"	1		1		
96	Нереляционные базы данных. Экспертные системы	1				
97	Интернет-приложения	1				
98	Понятие о серверной и клиентской частях сайта. Технология «клиент — сервер», её достоинства и недостатки	1				
99	Основы языка HTML. Каскадные таблицы стилей (CSS). Сценарии на языке JavaScript. Формы на веб-странице.	1				
100	Практическая работа по теме "Создание текстовой веб-страницы"	1		1		
101	Основы языка HTML	1				
102	Основы языка HTML	1				
103	Практическая работа по теме "Создание веб-страницы, включающей мультимедийные объекты (рисунки, звуковые данные, видео)"	1		1		
104	Основы каскадных таблиц стилей (CSS)	1				

105	Практическая работа по теме "Оформление страницы с помощью каскадных таблиц стилей"	1		1		
106	Сценарии на языке JavaScript	1				
107	Сценарии на языке JavaScript	1				
108	Формы на веб-странице	1				
109	Практическая работа по теме "Обработка данных форм"	1		1		
110	Размещение веб-сайтов. Услуга хостинга. Загрузка файлов на сайт	1				
111	Кадрирование. Исправление перспективы. Гистограмма. Коррекция уровней, коррекция цвета. Обесцвечивание цветных изображений	1				https://resh.edu.ru/subject/lesson/5348/start/15186/
112	Ввод изображений с использованием различных цифровых устройств. Практическая работа по теме "Обработка цифровых фотографий"	1		0.5		https://resh.edu.ru/subject/lesson/5348/start/15186/
113	Ретушь. Работа с областями. Фильтры. Практическая работа по теме "Ретушь цифровых фотографий"	1		1		https://resh.edu.ru/subject/lesson/5348/start/15186/

114	Многослойные изображения. Текстовые слои. Маска слоя. Каналы. Сохранение выделенной области	1				https://resh.edu.ru/subject/lesson/5348/start/15186/
115	Практическая работа по теме "Многослойные изображения". Текстовые слои. Маска слоя. Каналы. Сохранение выделенной области. Подготовка иллюстраций для веб-сайтов.	1		1		https://resh.edu.ru/subject/lesson/5348/start/15186/
116	Подготовка иллюстраций для веб-сайтов. Практическая работа по теме "Анимированные изображения"	1		1		https://resh.edu.ru/subject/lesson/5556/start/166550/
117	Векторная графика. Векторизация растровых изображений. Прimitives. Изменение порядка элементов. Выравнивание, распределение. Группировка. Кривые. Форматы векторных рисунков. Использование контуров.	1				https://resh.edu.ru/subject/lesson/5348/start/15186/
118	Практическая работа по теме "Векторная графика"	1		1		https://resh.edu.ru/subject/lesson/5556/start/166550/
119	Принципы построения и редактирования трёхмерных моделей	1		1		https://resh.edu.ru/subject/lesson/5556/start/166550/

120	Сеточные модели. Материалы. Моделирование источников освещения. Камеры. Аддитивные технологии (3D-принтеры). Понятие о виртуальной реальности и дополненной реальности.	1				
121	Практическая работа по теме "Создание простых трёхмерных моделей"	1		1		
122	Сеточные модели. Материалы	1				
123	Практическая работа по теме "Сеточные модели"	1		1		
124	Моделирование источников освещения. Камеры	1				
125	Практическая работа по теме "Рендеринг"	1		1		
126	Аддитивные технологии (3D-принтеры)	1				
127	Понятие о виртуальной реальности и дополненной реальности	1				
128	Обобщение по теме "Системы счисления"	1				
129	Обобщение по теме "Системы счисления"	1				
130	Обобщение по теме "Алгебра логики"	1				

131	Обобщение по теме "Алгебра логики"	1				
132	Промежуточная аттестация. Комплекс заданий стандартизированной формы.	1	1			
133	Обобщение по теме "Алгоритмизация и программирование"	1				
134	Обобщение по теме "Алгоритмизация и программирование"	1				
135	Обобщение по теме "Электронные таблицы"	1				
136	Обобщение по теме "Электронные таблицы"	1				
ОБЩЕЕ КОЛИЧЕСТВО ЧАСОВ ПО ПРОГРАММЕ		136	1	36.5		

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

ОБЯЗАТЕЛЬНЫЕ УЧЕБНЫЕ МАТЕРИАЛЫ ДЛЯ УЧЕНИКА

- Информатика (в 2 частях); углубленное обучение 10 класс/ Поляков К.Ю., Еремин Е.А., Общество с ограниченной ответственностью «БИНОМ. Лаборатория знаний»; Акционерное общество «Издательство «Просвещение»
- Информатика (в 2 частях); углубленное обучение 11 класс/ Поляков К.Ю., Еремин Е.А., Общество с ограниченной ответственностью «БИНОМ. Лаборатория знаний»; Акционерное общество «Издательство «Просвещение»
-

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ УЧИТЕЛЯ

На сайте «Единое содержание общего образования»

Раздел Методические материалы / Методические пособия и рекомендации –

<https://edsoo.ru/mr-informatika/> 6

1. Информатика (базовый уровень). Реализация ФГОС основного общего образования : методическое пособие для учителя. – ФГБНУ «ИСРО РАО», 2022. – 142 с.: ил.
2. Информатика (углублённый уровень). Реализация ФГОС основного общего образования : методическое пособие для учителя. – ФГБНУ «ИСРО РАО», 2022. – 211 с.: ил.
3. Система оценки достижений планируемых предметных результатов освоения учебного предмета «Информатика» : методические рекомендации. – ФГБНУ «ИСРО», 2023. – 83 с.: ил.
4. Информатика (углублённый уровень). Реализация требований ФГОС среднего общего образования : методическое пособие для учителя. – ФГБНУ «ИСРО», 2023. – 226 с. : ил.
5. Реализация профильного обучения технологической (инженерной) направленности на уровне среднего общего образования : методические рекомендации. – ФГБНУ «ИСРО», 2023. – 56 с.
6. Смешанное обучение в условиях цифровой трансформации образования (для учебных предметов «Математика», «Информатика») : методические рекомендации. – ФГБНУ «ИСРО РАО», 2022. – 43 с.

Раздел Методические семинары / Информатика – <https://edsoo.ru/metodicheskie-seminary/ms-informatika/>

Раздел Методические интерактивные кейсы – https://edsoo.ru/metodicheskie_kejsy/

Раздел Всероссийская олимпиада школьников / Информатика –

<https://vserosolimp.edsoo.ru/informatic>

Вопросы, связанные с подготовкой обучающихся к Всероссийской олимпиаде по искусственному интеллекту, размещены на сайте <https://ai.edu.gov.ru/>.

ЦИФРОВЫЕ ОБРАЗОВАТЕЛЬНЫЕ РЕСУРСЫ И РЕСУРСЫ СЕТИ

<https://kpolyakov.spb.ru/>

<https://resh.edu.ru/>

Контрольные работы 10 класс

1. Компьютер - универсальное устройство обработки данных
2. Компьютерная арифметика

Контрольные работы 11 класс

1. Информация и информационные процессы
2. Основы объектно-ориентированного программирования

Общие критерии оценивания результатов учебной деятельности:

Отметка «5» (отлично) - ученик обнаруживает усвоение обязательного уровня и уровня повышенной сложности учебных программ; выделяет главные положения в учебном материале и не затрудняется при ответах на видоизмененные вопросы; свободно применяет полученные знания на практике; не допускает ошибок в воспроизведении изученного материала, а также в письменных и практических работах, которые выполняет уверенно и аккуратно.

Отметка «4» (хорошо) - ученик обнаруживает усвоение обязательного и частично повышенного уровня сложности учебных программ, отвечает без особых затруднений на вопросы учителя; умеет применять полученные знания на практике; в устных ответах не допускает серьезных ошибок, легко устраняет отдельные неточности с помощью дополнительных вопросов учителя, в письменных и практических работах делает незначительные ошибки.

Отметка «3» (удовлетворительно) - ученик обнаруживает усвоение обязательного уровня учебных программ, но испытывает затруднения при его самостоятельном воспроизведении и требует дополнительных уточняющих вопросов учителя; предпочитает отвечать на вопросы воспроизводящего характера и испытывает затруднение при ответах на видоизмененные вопросы; допускает ошибки в письменных и практических работах.

Отметка «2» (неудовлетворительно) - у обучающегося имеются отдельные представления об изученном материале, но все же большая часть обязательного уровня учебных программ не усвоена, в письменных и практических работах ученик допускает грубые ошибки. При оценке результатов деятельности учащихся учитывается характер допущенных ошибок: существенных, несущественных.

Критерии и нормы оценивания письменных контрольных работ

- Отметка «5»: ставится за работу, выполненную полностью без ошибок и недочетов.

- Отметка «4»: ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета, не более трех недочетов.

- Отметка «3»: ставится, если ученик правильно выполнил не менее 2/3 всей работы или допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более трех негрубых ошибок, одной негрубой ошибки и трех недочетов, при наличии 4-5 недочетов.

- Отметка «2»: ставится, если число ошибок и недочетов превысило норму для оценки «3» или правильно выполнено не менее 2/3 всей работы.

Критерии и нормы оценивания практической работы

- Отметка «5»: а) выполнил работу в полном объеме с соблюдением необходимой последовательности ее проведения; б) самостоятельно и рационально выбрал и загрузил необходимое программное обеспечение, все задания выполнил в условиях и режимах, обеспечивающих получение результатов и выводов с наибольшей точностью; в) в представленном отчете правильно и аккуратно выполнил все записи, таблицы, рисунки, чертежи, графики, вычисления и сделал выводы.

- Отметка «4»: работа выполнена правильно, с учетом 2-3 несущественных ошибок, исправленных самостоятельно по требованию учителя.

- Отметка «3»: работа выполнена правильно не менее чем наполовину, или допущена существенная ошибка.

- Отметка «2»: в ходе работы допущены две (и более) существенные ошибки, которые учащийся не может исправить по учителя.

Если работа носит тестовый характер:

- Отметка «5» выставляется, если правильно выполнено 86-100% заданий теста;

- Отметка «4» выставляется, если правильно выполнено 71-85% заданий теста;

- Отметка «3» выставляется, если правильно выполнено 50-70% заданий теста;

- Отметка «2» выставляется, если правильно выполнено 11-49% заданий теста.

Оценочные материалы

10 класс

Входная работа

1. В кодировке КОИ-8 каждый символ кодируется 8 битами. Лена написала текст (в нем нет лишних пробелов):

«Ява, Куба, Лусон, Маражо, Суматра, Сулавеси, Эспаньола — острова».

Ученица вычеркнула из списка название одного из островов. Заодно она вычеркнула ставшие лишними запятые и пробелы — два пробела не должны идти подряд.

При этом размер нового предложения в данной кодировке оказался на 9 байтов меньше, чем размер исходного предложения. Напишите в ответе вычеркнутое название острова.

Решение. Поскольку один символ кодируется одним байтом, из текста удалили 9 символов. Заметим, что лишние запятая и пробел занимают два байта. Значит, название острова, которое удалили из списка, должно состоять из 7 букв. Из всего списка только одно название острова состоит из 7 букв — Суматра.

Ответ: Суматра.

2. Ваня шифрует русские слова, записывая вместо каждой буквы ее номер в алфавите (без пробелов). Номера букв даны в таблице:

А 1	Й 11	У 21	Э 31
Б 2	К 12	Ф 22	Ю 32
В 3	Л 13	Х 23	Я 33
Г 4	М 14	Ц 24	
Д 5	Н 15	Ч 25	
Е 6	О 16	Ш 26	
ё 7	П 17	Щ 27	
Ж 8	Р 18	Ъ 28	
З 9	С 19	Ы 29	
И 10	Т 20	Ь 30	

Некоторые шифровки можно расшифровать несколькими способами. Например, 311333 может означать «ВАЛЯ», может — «ЭЛЯ», а может — «ВААВВВ». Даны четыре шифровки:

3135420

2102030

1331320

2033510

Только одна из них расшифровывается единственным способом. Найдите ее и расшифруйте. Получившееся слово запишите в качестве ответа.

Решение. Проанализируем каждый вариант ответа:

- 1) «3135420» может означать как «ВЛДГТ» так и «ВАВДГТ».
- 2) «2102030» может означать только «БИТЬ».
- 3) «1331320» может означать как «ЛЭВТ» так и «АВЭВТ».
- 4) «2033510» может означать как «ТЯДИ» так и «ТВВДИ».

Следовательно, ответ «БИТЬ».

3. Напишите наименьшее число X , для которого истинно высказывание:

$(X > 16)$ **И** **НЕ** (X нечетное).

Решение. Логическое «И» истинно только тогда, когда истинны оба высказывания.

Запишем выражение в виде

$(X > 16)$ **И** (X четное).

Значит, наименьшее число, для которого высказывание будет истинным — 18.

Ответ: 18.

Приведем другое решение на языке Python.

```
for x in range(1, 100):
    if (x > 16) and not(x % 2 != 0):
        print(x)
        break
```

4. Между населенными пунктами A, B, C, D, E построены дороги, протяженность которых (в километрах) приведена в таблице:

	A	B	C	D	E
A		5	3		
B	5		1	4	
C	3	1		6	
D		4	6		1
E				1	

Определите длину кратчайшего пути между пунктами A и E. Передвигаться можно только по дорогам, протяженность которых указана в таблице.

Решение. Найдём все варианты маршрутов из A в E и выберем самый короткий.

Из пункта A можно попасть в пункты B, C.

Из пункта B можно попасть в пункты C, D.

Из пункта C можно попасть в пункты B, D.

Из пункта D можно попасть в пункт E.

A–B–C–D–E: длина маршрута 13 км.

A–C–D–E: длина маршрута 10 км.

A–B–D–E: длина маршрута 10 км.

A–C–B–D–E: длина маршрута 9 км.

Самый короткий путь: A–C–B–D–E. Длина маршрута 9 км.

Ответ: 9.

5. У исполнителя Делитель две команды, которым присвоены номера:

1. раздели на 2

2. вычти 3

Первая из них уменьшает число на экране в 2 раза, вторая уменьшает его на 3. Исполнитель работает только с натуральными числами. Составьте алгоритм получения из числа 76 числа 5, содержащий не более 5 команд. В ответе запишите только номера команд.

(Например, 21211 – это алгоритм:

вычти 3

раздели на 2

вычти 3

раздели на 2

раздели на 2,

который преобразует число 33 в 3.)

Если таких алгоритмов более одного, то запишите любой из них.

Решение. Поскольку Делитель работает только с натуральными числами и число 76 — четное, первая команда может быть 1, при этом получится число 38. Оно тоже четное, тогда вторая команда тоже может быть 1, получится число 19, из которого число 5 можно получить последовательностью команд 212. Следовательно, из числа 76 число 5 можно получить последовательностью команд 11212.

Ответ: 11212.

Приведем другое решение на языке Python.

```
def f(flag, x):
    if flag == 1:
        return x / 2
    if flag == 2:
        return x - 3

x = 76
for flag in range(1, 3):
    for flag1 in range(1, 3):
        for flag2 in range(1, 3):
            for flag3 in range(1, 3):
                for flag4 in range(1, 3):
                    if f(flag, f(flag1, f(flag2, f(flag3, f(flag4, x))))
== 5:
                        print(flag4, flag3, flag2, flag1, flag)
```

6. Ниже приведена программа, записанная на пяти языках программирования.

Бейсик	Python
<pre>DIM s, t AS INTEGER INPUT s INPUT t IF s > 10 AND t > 10 THEN PRINT 'YES' ELSE PRINT 'NO' ENDIF</pre>	<pre>s = int(input()) t = int(input()) if s > 10 and t > 10: print("YES") else: print("NO")</pre>
Паскаль	Алгоритмический язык
<pre>var s, t: integer; begin readln(s); readln(t); if (s > 10) and (t > 10) then writeln('YES') else writeln('NO') end.</pre>	<pre>алг нач цел s, t ввод s ввод t если s > 10 и t > 10 то вывод "YES" иначе вывод "NO" все кон</pre>
C++	


```
#include <iostream>
using namespace std;
int main() {
    int s, t;
    cin >> s;
    cin >> t;
    if (s > 10 && t > 10)
        cout << "YES";
    else
        cout << "NO";
    return 0;
}
```

Было проведено 9 запусков программы, при которых в качестве значений переменных s и t вводились следующие пары чисел:

(10, 10); (10, 6); (−4, 8); (2, 9); (12, 7); (−11, 4); (−8, 13); (10, 9); (11, 11).

Сколько было запусков, при которых программа напечатала «NO»?

Решение. Заметим, что программа напечатает «NO», если одна из введенных переменных s и t будет меньше или равна 10. Значит, было 8 запусков, при которых программа напечатала «NO». В качестве значений переменных s и t в этих случаях вводились следующие пары чисел:

(10, 10); (10, 6); (−4, 8); (2, 9); (12, 7); (−11, 4); (−8, 13); (10, 9).

Ответ: 8.

Примечание.

Другой способ решения и правила замены логических условий на противоположные приведены в задании [10955](#).

Приведем другое решение на языке Python.

```
def f(s, t):
    if s > 10 and t > 10:
        return 0
    else:
        return 1
a = ((10, 10), (10, 6), (−4, 8), (2, 9), (12, 7), (−11, 4), (−8, 13), (10, 9), (11, 11))
kol = 0
for i in range(9):
    kol += f(a[i][0], a[i][1])
print(kol)
```

7. На месте преступления были обнаружены четыре обрывка бумаги. Следствие установило, что на них записаны фрагменты одного IP-адреса. Криминалисты обозначили эти фрагменты буквами А, Б, В и Г:

2.12	22	.30	5.121
А	Б	В	Г

Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.

Решение. IP-адрес представляет собой четыре разделенных точками числа, каждое из которых не больше 255.

Под буквой В указано «.30». Так как числа в IP-адресе не могут быть больше 255, нельзя добавить в конце этого числа еще один разряд. Следовательно, этот фрагмент — последний.

Рассмотрим фрагмент под буквой Б. Так как числа в IP-адресе не могут быть больше 255, фрагмент Б должен быть на первом месте.

В конце фрагмента Г — число 121, отделенное точкой. Так как в IP-адресе не может быть числа, большего 255, то за фрагментом Г должен следовать фрагмент, начинающийся с точки. Значит, фрагмент Г идет перед фрагментом В.

Следовательно, ответ БАГВ.

8. В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для логической операции «И» — символ «&».

Запрос	Количество страниц(тыс.)
Сербия & Хорватия	500
Сербия Хорватия	3000
Сербия	2000

Какое количество страниц (в тысячах) будет найдено по запросу *Хорватия*? Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

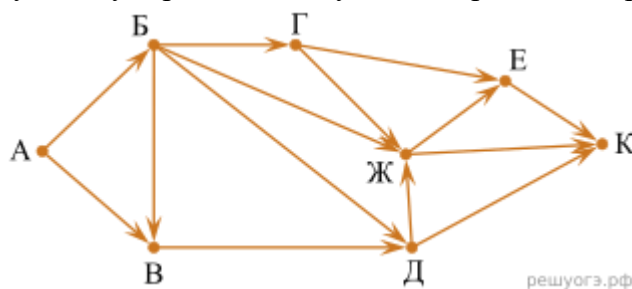
Решение. По формуле включений и исключений имеем:

$$m(\text{Сербия}|\text{Хорватия}) = m(\text{Сербия}) + m(\text{Хорватия}) - m(\text{Сербия}\&\text{Хорватия}).$$

Тогда $3000 = 2000 + m(\text{Хорватия}) - 500$, откуда $m(\text{Хорватия}) = 1500$.

Ответ: 1500.

9. На рисунке — схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж и К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К, проходящих через город Ж?



Решение. Количество путей до города X = количество путей добраться в любой из тех городов, из которых есть дорога в X.

При этом если путь должен не проходить через какой-то город, нужно просто не учитывать этот город при подсчете сумм. А если город, наоборот, обязательно должен лежать на пути, тогда для городов, в которые из нужного города идут дороги, в суммах нужно брать только этот город.

С помощью этого наблюдения посчитаем последовательно количество путей до каждого из городов:

$$A = 1$$

$$B = A = 1$$

$$V = A + B = 2$$

$$Г = B = 1$$

$$Д = B + V = 3$$

$$Ж = B + Г + Д = 5$$

$$E = Ж = 5 \text{ (Г не учитываем, поскольку путь должен проходить через город Ж)}$$

$$K = E + Ж = 5 + 5 = 10 \text{ (Д не учитываем, поскольку путь должен проходить через город Ж)}.$$

Приведем другое решение.

Количество путей из города А в город К, проходящих через город Ж, равно произведению количества путей из города А в город Ж и количества путей из города Ж в город К.

Найдем количество путей из города А в город Ж:

$$A = 1$$

$$B = A = 1$$

$$B = A + B = 2$$

$$\Gamma = B = 1$$

$$D = B + B = 3$$

$$Ж = B + \Gamma + D = 5.$$

Найдем количество путей из города Ж в город К (при этом Ж - исходный пункт):

$$Ж = 1$$

$$E = Ж = 1$$

$$K = Ж + E = 2.$$

Тогда количество путей из города А в город К, проходящих через город Ж, равно $5 \cdot 2 = 10$.

Ответ: 10.

10. Среди приведенных ниже трех чисел, записанных в различных системах счисления, найдите максимальное и запишите его в ответе в десятичной системе счисления. В ответе запишите только число, основание системы счисления указывать не нужно.

$$59_{16}, 126_8, 1011100_2.$$

Решение. Переведем все числа в десятичную систему счисления:

$$1. 59_{16} = 89_{10};$$

$$2. 126_8 = 86_{10};$$

$$3. 1011100_2 = 92_{10}.$$

Таким образом, наибольшим среди этих трех чисел является число 92.

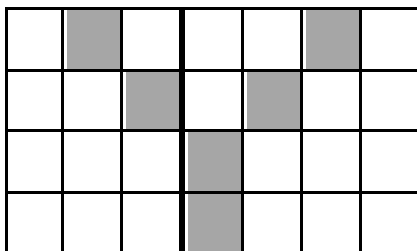
Ответ: 92.

Приведем другое решение на языке Python.

```
def f(x, i):  
    x = int(str(x), i)  
    return x  
print(max(f(59, 16), f(126, 8), f(1011100, 2)))
```

Контрольная работа «Компьютер - универсальное устройство обработки данных»

1. Текст длиной 32768 символов закодирован с помощью алфавита, содержащего 64 символа. Сколько килобайт занимает в памяти этот текст?
2. Сообщение длиной 28672 символа занимает в памяти 21 Кбайт. Найдите мощность алфавита, который использовался при кодировании.
3. Рисунок размером 92 на 2048 пикселей закодирован с палитрой 2 цвета. Сколько килобайт занимает в памяти рисунок без учёта сжатия?
4. Рисунок размером 512 на 384 пикселей занимает в памяти 168 Кбайт (без учёта сжатия). Найдите максимально возможное количество цветов в палитре изображения.
5. Дан черно-белый растровый рисунок:



Запишите шестнадцатеричный код, соответствующий этому рисунку.

6. Звук длительностью 2 минуты оцифрован с частотой 11000 Гц.

Разрядность кодирования - 24 бита. Определите размер полученного файла в Мбайтах. Результат округлите до двух цифр после запятой.

Ответы.

	1	2	3	4	5	6
1	24	64	23	128	4450408	3,78

Контрольная работа «Компьютерная арифметика»

Системы счисления

- 1) Перевести число из двоичной системы в системы с основанием 8, 10, 16.
- 2) Перевести число из восьмеричной системы в системы с основанием 2, 10, 16.
- 3) Перевести число из десятичной системы в системы с основанием 2, 8, 16.
- 4) Перевести число из шестнадцатеричной системы в системы с основанием 2, 8, 10.
- 5) Сложить двоичные числа.
- 6) Сложить восьмеричные числа.
- 7) Сложить шестнадцатеричные числа.
- 8) Вычесть двоичные числа.
- 9) Вычесть восьмеричные числа.
- 10) Вычесть шестнадцатеричные числа.

----- Вариант 1 -----

- 1) 10001100_2
- 2) 205_8
- 3) 155
- 4) 89_{16}
- 5) $10010110_2 + 11001011_2$
- 6) $6271_8 + 7024_8$
- 7) $B06_{16} + C64_{16}$
- 8) $1011010_2 - 11111001_2$
- 9) $1521_8 - 7041_8$
- 10) $401_{16} - E1A_{16}$

Ответы к контрольной работе

----- Вариант 1 -----

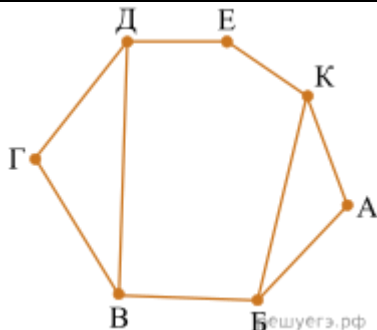
- 1) 214_8 $1408C_{16}$
- 2) 10000101_2 13385_{16}
- 3) 10011011_2 233_8 $9B_{16}$
- 4) 10001001_2 211_8 137
- 5) 101100001_2
- 6) 15315_8
- 7) 176_{16}
- 8) -10011111_2
- 9) -5320_8
- 10) -19_{16}

Промежуточная аттестация

1. На рисунке схема дорог изображена в виде графа, в таблице звёздочкой обозначено наличие дороги между населёнными пунктами. Отсутствие звёздочки означает, что такой дороги нет.

	п1	п2	п3	п4	п5	п6	п7
п1						*	*

П2			*	*		*	
П3		*		*			
П4		*	*		*		
П5				*			*
П6	*	*					*
П7	*				*	*	



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе.

Определите, какие номера населённых пунктов соответствуют населённым пунктам *Б* и *В*. В ответе запишите эти два номера в порядке возрастания без пробелов и знаков препинания.

Пример. Пусть населённым пунктам *Д* и *Е* соответствуют номера *П1* и *П2*. Тогда в ответе нужно написать «12».

Решение. Заметим, что *Е* — единственная вершина степени 2, которая связана с вершинами третьей степени *Д* и *К*, связанными с остальными вершинами степени 2. Значит, *Е* соответствует *П5*. Далее рассмотрим два варианта.

1. Пусть *Д* соответствует *П4*, а *К* соответствует *П7*. *В* — единственная вершина степени 3, в которую есть дорога из *Д*, следовательно, *В* соответствует *П2*. *Б* — единственная вершина степени 3, в которую есть дорога из *К*, следовательно, *Б* соответствует *П6*.

2. Пусть *Д* соответствует *П7*, а *К* соответствует *П4*. *В* — единственная вершина степени 3, в которую есть дорога из *Д*, следовательно, *В* соответствует *П6*. *Б* — единственная вершина степени 3, в которую есть дорога из *К*, следовательно, *Б* соответствует *П2*.

Таким образом, населённым пунктам *Б* и *В* соответствуют *П2* и *П6*.

Ответ: 26.

Приведём решение Артёма Гридина на языке Python.

```
from itertools import permutations
table = '67 346 24 235 47 127 156'.split()
graph = 'ГД ГВ ВД АК АВ КБ КЕ ДЕ ВВ'.split()
print('1 2 3 4 5 6 7')
for p in permutations('АБВГДЕК'):
    if all(str(p.index(c2)+1) in table[p.index(c1)] for c1, c2 in graph):
        print(*p)
        break
```

2. Логическая функция *F* задаётся выражением $(x \vee y) \rightarrow (z \equiv x)$.

Дан частично заполненный фрагмент, содержащий **неповторяющиеся** строки таблицы истинности функции *F*.

Определите, какому столбцу таблицы истинности соответствует каждая из переменных *x*, *y*, *z*.

Переменная 1	Переменная 2	Переменная 3	Функция
--------------	--------------	--------------	---------

???	???	???	F
	0	0	0
	0		0

В ответе напишите буквы x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала — буква, соответствующая первому столбцу; затем — буква, соответствующая второму столбцу, и т. д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Пусть задано выражение $x \rightarrow y$, зависящее от двух переменных x и y , и фрагмент таблицы истинности:

Переменная 1	Переменная 2	Функция
???	???	F
0	1	0

Тогда первому столбцу соответствует переменная y , а второму столбцу соответствует переменная x . В ответе нужно написать: yx .

Решение. Данная импликация принимает значение 0 тогда и только тогда, когда

$$\begin{cases} x + y = 1, \\ x \neq z. \end{cases} \quad (*)$$

Пусть $x = 0$, тогда $y = z = 1$. В первой строке нет двух единиц, значит, $x = 1$, и эта переменная находится в первом столбце. Тогда первая строка имеет вид 1 0 0.

Вторая строка должна отличаться от первой, поэтому она имеет вид 1 0 1. Рассмотрим два варианта:

x	y	z
1	0	0
1	0	1
x	z	y
1	0	0
1	0	1

Первый вариант не удовлетворяет системе (*), а второй — удовлетворяет.

Ответ: xzy .

Приведем другое решение.

Составим таблицу истинности для выражения $(x \vee y) \rightarrow (z \equiv x)$ вручную или при помощи языка Python:

```
print("x y z")
for x in range(0, 2):
    for y in range(0, 2):
        for z in range(0, 2):
            if not((x or y) <= (z == x)):
                print(x, y, z)
```

Далее выпишем те наборы переменных, при которых данное выражение равно 0. В наборах переменные запишем в порядке x, y, z . Получим следующие наборы:

Получим следующие наборы:

(0, 1, 1)

(1, 0, 0),

(1, 1, 0).

Сопоставим эти наборы с приведенным в задании фрагментом таблицы истинности.

Первая строка таблицы может соответствовать только набору (1, 0, 0). Следовательно, первый столбец таблицы соответствует переменной x , и в первом столбце первой строки стоит 1.

Второй столбец таблицы может соответствовать только переменной z , поскольку переменная y принимает нулевое значение только в одном наборе. Тогда третий столбец соответствует переменной y .

Другая программа Артёма Гридина для построения таблицы истинности.

Составим таблицу истинности для выражения $(x \vee y) \rightarrow (z \equiv x)$ вручную или при помощи языка Python:

```
import itertools
print('x y z')
for x, y, z in itertools.product((0, 1), repeat = 3):
    if not((x or y) <= (z == x)):
        print(x, y, z)
```

4. По каналу связи передаются сообщения, содержащие только семь букв: А, Б, Г, И, М, Р, Я. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: А — 010, Б — 011, Г — 100. Какое **наименьшее** количество двоичных знаков потребуется для кодирования слова МАГИЯ?

Примечание. Условие Фано означает, что ни одно кодовое слово не является началом другого кодового слова.

Решение. Следующая буква должна кодироваться как 11, поскольку 10 мы взять не можем. 100 взять не можем из-за Г, значит, следующая буква должна быть закодирована кодом 101. Следующая буква должна кодироваться как 000, поскольку 00 взять не можем, иначе не останется кодовых слов для оставшейся буквы, которые удовлетворяют условию Фано. Значит, последняя буква будет кодироваться как 001. Тогда наименьшее количество двоичных знаков, которые потребуются для кодирования слова МАГИЯ, равно $2 + 3 + 3 + 3 + 3 = 14$.

Ответ: 14.

Примечание.

Заметим, что после кодирования всех букв, входящих в слово МАГИЯ, должен остаться хотя бы один свободный код для кодирования буквы Р, которая не входит в данное слово, но может передаваться по каналу связи. Проверить наличие свободного кода можно, построив дерево кодов, как показано в задаче [18553](#).

5. Автомат обрабатывает натуральное число N ($0 \leq N \leq 255$) по следующему алгоритму.

1. Строится восьмибитная двоичная запись числа N .
2. Все цифры двоичной записи заменяются на противоположные (0 на 1, 1 на 0).
3. Полученное число переводится в десятичную запись.
4. Из нового числа вычитается исходное, полученная разность выводится на экран.

Пример. Дано число $N = 13$. Алгоритм работает следующим образом.

1. Восьмибитная двоичная запись числа N : 00001101.
2. Все цифры заменяются на противоположные, новая запись: 11110010.
3. Десятичное значение полученного числа 242.
4. На экран выводится число $242 - 13 = 229$.

Какое число нужно ввести в автомат, чтобы в результате получилось 111?

Решение. Заметим, что инверсия двоичной восьмибитной записи числа в сумме с исходным числом дает 11111111, то есть 255. (В исходном примере: $00001101 + 11110010 = 11111111$.) Следовательно, если исходное число равно N , то инвертированное число равно $255 - N$. Затем автомат осуществляет вычитание, вычисляя $255 - 2N$.

Поэтому, чтобы найти число, которое нужно ввести в автомат для получения 111, нужно решить уравнение $255 - 2N = 111$. Тем самым, искомое число равно 72.

Ответ: 72.

Примечание.

Заметим, что по условию из нового числа вычитается исходное, следовательно, для получения разности 111 новое число должно быть больше исходного, а значит, исходное число должно быть не более 127.

Приведём другое решение на языке Python.

```
for n in range(0, 256):
    s = bin(n)[2:] # перевод в двоичную систему
    s = str(s)
    if len(s) < 8:
        s = '0' * (8 - len(s)) + s
    s = s.replace('1', '*')
    s = s.replace('0', '1')
    s = s.replace('*', '0')
    r = int(s, 2) # перевод в десятичную систему
    if r - n == 111:
        print(n)
```

Приведём другое решение на языке Python.

```
a = str.maketrans('01', '10')
for N in range(0, 256):
    R = bin(N)[2:].zfill(8)
    R = str.translate(R, a)
    if int(R, 2) - N == 111:
        print(N)
```

7. Музыкальный фрагмент был оцифрован и записан в виде файла без использования сжатия данных. Получившийся файл был передан в город *A* по каналу связи за 15 секунд. Затем тот же музыкальный фрагмент был оцифрован повторно с разрешением в 2 раза выше и частотой дискретизации в 1,5 раза меньше, чем в первый раз. Сжатие данных не производилось. Полученный файл был передан в город *B*; пропускная способность канала связи с городом *B* в 2 раза выше, чем канала связи с городом *A*. Сколько секунд длилась передача файла в город *B*? В ответе запишите только целое число, единицу измерения писать не нужно.

Решение. Объём файла прямо пропорционален разрешению файла и его частоте дискретизации. Следовательно, объём файла во втором случае в $2/1,5 = 4/3$ раза больше. Длительность передачи обратно пропорциональна пропускной способности канала связи, откуда получаем, что длительность передачи файла во второй раз равна: $15 \cdot (1/2) \cdot (4/3) = 10$.

Ответ: 10.

8. Шифр кодового замка представляет собой последовательность из пяти символов, каждый из которых является цифрой от 1 до 5. Сколько различных вариантов шифра можно задать, если известно, что цифра 1 встречается ровно три раза, а каждая из других допустимых цифр может встречаться в шифре любое количество раз или не встречаться совсем?

Решение. Количество способов поставить три 1 на пять позиций —

$$C_5^3 = \frac{5!}{3!2!} = \frac{120}{12} = 10.$$

После того, как определили позиции трёх 1, на оставшиеся две позиции можем поставить любое из четырёх чисел, это можно сделать $4^2 = 16$ способами. Итого всего $10 \cdot 16 = 160$ кодов.

Приведём другое решение на языке Python.

```
from itertools import product
alphabet = '12345'
count=0
for i in product(alphabet, repeat=5):
    if i.count('1') == 3:
        count += 1
print(count)
```

11. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 6 символов и содержащий только символы из 7-буквенного набора Н, О, Р, С, Т, У, Х. В базе данных для хранения сведений о каждом пользователе отведено одинаковое целое число байт, при этом для хранения сведений о 100 пользователях используется 1400 байт. Для каждого пользователя хранятся пароль и дополнительные сведения. Для хранения паролей используют посимвольное кодирование, все символы кодируются одинаковым и минимально возможным количеством бит. Сколько бит отведено для хранения дополнительных сведений о каждом пользователе?

$$\frac{1400}{100} = 14$$

Решение. Для хранения сведений об одном пользователе используется 14 байт = 112 бит.

Согласно условию, в пароле могут быть использованы 7 букв. Известно, что с помощью N бит можно закодировать 2^N различных вариантов. Поскольку $2^2 < 7 < 2^3$, то для записи каждой из 7 букв необходимо 3 бита. Для записи шести букв необходимо $3 \cdot 6 = 18$ бит.

Тогда для хранения дополнительных сведений используется $112 - 18 = 94$ бит.

Ответ: 94.

12. Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w . Например, выполнение команды **заменить** (111, 27) преобразует строку 05111150 в строку 0527150. Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА условие

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ условие

ТО команда1

ИНАЧЕ команда2

КОНЕЦ ЕСЛИ

выполняется команда1 (если условие истинно) или команда2 (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 68 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (222) ИЛИ нашлось (888)

ЕСЛИ нашлось (222)

ТО заменить (222, 8)

ИНАЧЕ заменить (888, 2)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение. Данный алгоритм сначала заменит 9 первых восьмерок на три двойки а затем заменит эти три двойки обратно на одну восьмерку. То есть за четыре повторения цикла из строки убирается 8 восьмерок. За восемь таких групп по четыре повторения, то есть за $8 \cdot 4 = 32$ повторения цикла из строки уберется $8 \cdot 8 = 64$ восьмерки, и останется 4 восьмерки. Первые три из них будут заменены на двойку.

Ответ: 28.

Приведем решение Михаила Глинского.

Для нахождения ответа используется программа на языке Питон, выполняющая заданные преобразования строки:

```
s='8'*68
while ('222' in s) or ('888' in s):
    if ('222' in s):
        s=s.replace ('222','8',1)
    else:
        s=s.replace ('888','2',1)
print(s)
```

13. Петя записал IP-адрес школьного сервера на листке бумаги и положил его в карман куртки. Петина мама случайно постирала куртку вместе с запиской. После стирки Петя обнаружил в кармане четыре обрывка с фрагментами IP-адреса. Эти фрагменты обозначены буквами А, Б, В и Г. Восстановите IP-адрес. В ответе укажите последовательность букв, обозначающих фрагменты, в порядке, соответствующем IP-адресу.



Решение. IP-адрес представляет собой числа, разьединенные точками, причем числа эти не больше 255.

Посмотрим внимательнее на данные фрагменты: под буквой В мы видим «.61». Число, на которое указывает этот фрагмент, начинается с 61. Поскольку числа в IP-адресе не могут быть больше 255, мы не можем добавить в конце этого числа еще один разряд, а фрагментов, начинающихся с точки, больше нет. Следовательно, этот фрагмент — последний.

Посмотрим на фрагмент под буквой Г. В нем стоит число без точек, значит, это либо последний фрагмент, либо первый. Место последнего фрагмента уже занято, значит, фрагмент Г на первом месте.

В конце фрагмента А — число 162, отделенное точкой. Поскольку в IP-адресе не может быть числа, большего 255, то за фрагментом А должен следовать фрагмент, начинающийся с точки. Значит, фрагмент А идет перед фрагментом В.

Ответ: ГБАВ.

14. Значение арифметического выражения $9^8 + 3^5 - 9$ записали в системе счисления с основанием 3. Сколько цифр 2 содержится в этой записи?

Решение. Преобразуем выражение:

$$(3^2)^8 + 3^5 - 3^2;$$

$$3^{16} + 3^5 - 3^2;$$

$$3^{16} + 3^5 = 100...00100000;$$

$$100...00100000 - 3^2 = 100...00022200.$$

В полученном числе три двойки.

Ответ: 3.

Приведём другое решение на языке Python.

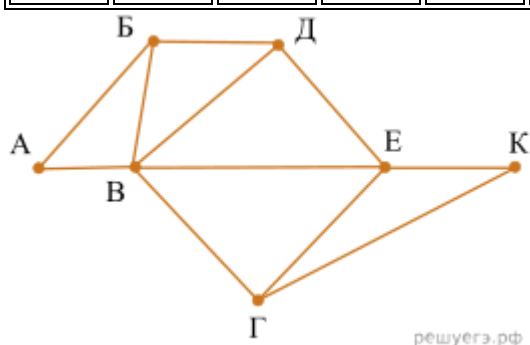
```
x = 9**8 + 3**5 - 9
s = ''
while x != 0:
    s += str(x % 3)
    x //= 3
s = s[::-1]
print(s.count("2"))
```

11 класс.

Входная работа

1. На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		40		15			
п2	40			35		50	
п3					10	65	8
п4	15	35				22	33
п5			10			50	
п6		50	65	22	50		40
п7			8	33		40	



Так как таблицу и схему рисовали независимо друг от друга, нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе.

Определите длину дороги из пункта Б в пункт Д. В ответе запишите целое число.

Решение. Есть только один пункт, из которого ведёт 5 дорог, — это В, а в таблице — П6.

Из А ведёт две дороги, и одна из них — в В. В таблице такому соответствует П5.

Из Б ведёт три дороги, причём есть дороги в А и в В, в таблице под такое подходит только П3.

Из Д три дороги, две из которых — в Б и в В, в таблице только один пункт такому соответствует — П7.

Таким образом, Б — это П3, а Д — П7. Длина дороги между П3 и П7 — 8.

Ответ: 8.

2. Логическая функция F задаётся выражением $(x \wedge \neg y) \vee (y \equiv z) \vee \neg w$. На рисунке приведён фрагмент таблицы истинности функции F , содержащий все наборы аргументов, при которых функция F ложна. Определите, какому столбцу таблицы истинности функции F соответствует каждая из переменных w, x, y, z . Все строки в представленном фрагменте разные.

Перем. 1	Перем. 2	Перем. 3	Перем. 4
???	???	???	???
	0		
1	0		0
1		0	0

В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (без разделителей).

Решение. Рассмотрим данное выражение. Преобразуем логическое выражение $(x \wedge \neg y) \vee (y \equiv z) \vee \neg w$ и получим систему, при которой оно ложно:

$$\begin{cases} x = 0, \\ y = 1, \\ y \neq z, \\ w = 1. \end{cases}$$

Сразу видно, что первый столбец — это w , поскольку w всегда должна равняться единице. Также ясно, что x — это переменная 4, так как $y \neq z$. Из первого выражения $x \wedge \neg y$ и последней строчки таблицы видно, что переменная 3 — это y , а вторая переменная — это z .

Примечание.

Рассмотрим, как будет выглядеть полная таблица истинности. Переменная w всегда должна принимать значение 1, поэтому в первом столбце во всех строках будет стоять единица. Исходя из условия $y \neq z$ можно заключить, что во втором столбце в последней строке будет стоять единица, и в первых двух строках третьего столбца тоже будут стоять единицы. В первой четвертой строке должна стоять единица, поскольку строки в таблице истинности должны быть разными.

Перем. 1	Перем. 2	Перем. 3	Перем. 4
???	???	???	???
1	0	1	1
1	0	1	0
1	1	0	0

Вариант $wuzx$ не подходит, поскольку в первой строке функция F окажется истинной.

Приведем другое решение.

Составим таблицу истинности функции F вручную или при помощи языка Python:

```
print("x y z w")
for x in range(0, 2):
    for y in range(0, 2):
        for z in range(0, 2):
            for w in range(0, 2):
                if not((x and not(y)) or (y == z) or not(w)):
                    print(x, y, z, w)
```

Далее выпишем те наборы переменных, при которых данное выражение равно 0. В наборах переменные запишем в порядке x, y, z, w .

(0, 0, 1, 1),

(0, 1, 0, 1),

(1, 1, 0, 1).

Заметим, что переменная w всегда должна быть равна 1, поэтому ей соответствует первый столбец заданной таблицы.

Заметим, что вторая и третья строки заданной таблицы, содержащие по два нуля, соответствуют наборам переменных (0, 0, 1, 1) или (0, 1, 0, 1), тогда первая строка соответствует набору (1, 1, 0, 1). Значит, в первой строке $z = 0$, а все остальные переменные равны 1, и переменной z соответствует второй столбец заданной таблицы.

Тогда вторая строка заданной таблицы, в которой переменная z также равна 0, соответствует набору (0, 1, 0, 1), в котором $x = 0$, а остальные переменные равны 1, поэтому переменной x соответствует четвертый столбец таблицы.

Тогда переменной y соответствует третий столбец.

Ответ: wzyx.

4. По каналу связи передаются сообщения, содержащие только семь букв: А, Б, Г, И, М, Р, Я. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: А — 010, Б — 011, И — 10. Какое **наименьшее** количество двоичных знаков потребуется для кодирования слова ГРАММ?

Примечание. Условие Фано означает, что ни одно кодовое слово не является началом другого кодового слова.

Решение. Для трёх букв кодовые слова уже известны, осталось подобрать для оставшихся четырёх букв такие кодовые слова, которые обеспечат наименьшее количество двоичных знаков для кодирования слова ГРАММ.

Закодируем букву М кодовым словом 00, поскольку буква М повторяется в слове ГРАММ два раза. Для буквы Г возьмём кодовое слово 110. Кодовое слово 111 взять не можем, поскольку для остальных букв не останется кодовых слов, удовлетворяющих условию Фано. Оставшиеся две буквы закодируем кодовыми словами длины 4. Таким образом, наименьшее количество двоичных знаков, которые потребуются для кодирования слова ГРАММ, равно $3 + 4 + 3 + 2 + 2 = 14$.

Ответ: 14.

Примечание.

Заметим, что после кодирования всех букв, входящих в слово ГРАММ, должен остаться хотя бы один свободный код для кодирования буквы Я, которая не входит в данное слово, но может передаваться по каналу связи. Проверить наличие свободного кода можно, построив дерево кодов, как показано в задаче [18553](#).

5. На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.

1. Строится двоичная запись числа N .

2. К этой записи дописываются справа ещё два разряда по следующему правилу:

а) складываются все цифры двоичной записи, и остаток от деления суммы на 2 дописывается в конец числа (справа). Например, запись 10000 преобразуется в запись 100001;

б) над этой записью производятся те же действия — справа дописывается остаток от деления суммы цифр на 2.

Полученная таким образом запись (в ней на два разряда больше, чем в записи исходного числа N) является двоичной записью искомого числа R .

Укажите такое наименьшее число N , для которого результат работы алгоритма больше 97. В ответе это число запишите в десятичной системе счисления.

Решение. Если изначально сумма разрядов была чётная, то в конец запишется 00, что эквивалентно $N \rightarrow N \cdot 4$.

Если же сумма была нечётная, то запишется 10, что эквивалентно $N \rightarrow N \cdot 4 + 2$. В обоих случаях число получается чётным.

Посмотрим на чётные числа, превосходящие 97.

$98_{10} = 1100010_2$ — на конце 10, но сумма остальных разрядов чётна, не подходит ни под один из случаев.

$100_{10} = 1100100_2$ — на конце 00, но сумма остальных разрядов нечётна, не подходит.

$102_{10} = 1100110_2$ — на конце 10, а сумма остальных разрядов нечётна. Число подходит под второй случай, значит, число, из которого оно было получено, равно $\frac{102 - 2}{4} = 25$.

Ответ: 25.

Приведём другое решение на языке Python.

```
def f(s):
    summa = 0
    for i in range(len(s)):
        summa += int(s[i])
    return summa
for n in range(1, 100):
    s = bin(n)[2:] # перевод в двоичную систему
    s = str(s)
    summa = f(s)
    s = s + str(summa % 2)
    summa = f(s)
    s = s + str(summa % 2)
    r = int(s, 2) # перевод в десятичную систему
    if r > 97:
        print(n)
        break
```

Приведём другое решение Силина Николая на языке Python.

```
for n in range(0, 1000000):
    r=bin(n)[2:]
    r+= str(r.count('1')%2)
    r+= str(r.count('1')%2)
    if int(r,2)>97:
        print (n)
        break
```

7. Производится звукозапись музыкального фрагмента в формате стерео (двухканальная запись) с частотой дискретизации 32 кГц и 32-битным разрешением. Результаты записываются в файл, сжатие данных не производится; размер полученного файла 40 Мбайт. Затем производится повторная запись этого же фрагмента в формате моно (одноканальная запись) с частотой дискретизации 16 кГц и 16-битным разрешением. Сжатие данных не производилось.

Укажите размер файла в Мбайт, полученного при повторной записи. В ответе запишите только целое число, единицу измерения писать не нужно.

Решение. Получаем систему:

$40 \text{ (Мбайт)} = 2 \text{ (количество каналов)} \cdot 32000 \text{ (частота)} \cdot 4 \text{ (разрешение в байтах)} \cdot X \text{ (время в секундах)}.$

$V \text{ (Мбайт)} = 1 \text{ (количество каналов)} \cdot 16000 \text{ (частота)} \cdot 2 \text{ (разрешение в байтах)} \cdot X \text{ (время в секундах)}.$

Тогда $v = 40 : (2 \cdot 2 \cdot 2) = 5 \text{ Мбайт}.$

Ответ: 5.

8. Шифр кодового замка представляет собой последовательность из пяти символов, каждый из которых является цифрой от 1 до 4. Сколько различных вариантов шифра можно задать, если известно, что цифра 1 встречается ровно два раза, а каждая из других допустимых цифр может встречаться в шифре любое количество раз или не встречаться совсем?

$$5 \cdot \frac{4}{2} = 10.$$

Решение. Количество способов поставить две 1 на пять позиций — 10. После того, как определили позиции двух 1, на оставшиеся позиции можем поставить любое из трёх чисел, это можно сделать $3^3 = 27$ способами. Итого всего $10 \cdot 27 = 270$ кодов.

Ответ: 270.

Приведём другое решение на языке Python.

```
from itertools import product
alphabet = '1234'
ap=[]
for i in product(alphabet, repeat=5):
    if i.count('1') == 2:
        ap.append(i)
print(len(ap))
```

11. При регистрации в компьютерной системе каждому пользователю выдаётся пароль, состоящий из 9 символов. Из соображений информационной безопасности каждый пароль должен содержать хотя бы 1 десятичную цифру, как прописные, так и строчные латинские буквы (в латинском алфавите 26 букв), а также не менее 1 символа из 6-символьного набора: «&», «#», «\$», «*», «!», «@». В базе данных для хранения сведений о каждом пользователе отведено одинаковое и минимально возможное целое число байт. При этом используют посимвольное кодирование паролей, все символы кодируют одинаковым и минимально возможным количеством бит. Кроме собственно пароля, для каждого пользователя в системе хранятся дополнительные сведения, для чего выделено целое число байт; это число одно и то же для всех пользователей. Для хранения сведений о 20 пользователях потребовалось 500 байт. Сколько байт выделено для хранения дополнительных сведений об одном пользователе? В ответе запишите только целое число – количество байт.

Решение. Согласно условию, в пароле могут быть использованы 10 цифр (0..9), 52 буквы (строчные и прописные) и ещё 6 символов из спецнабора, всего $10 + 52 + 6 = 68$ символов. Известно, что с помощью N бит можно закодировать 2^N различных вариантов. Поскольку $2^6 < 68 < 2^7$, то для записи каждого из 68 символов необходимо 7 бит.

Для хранения всех 9 символов номера нужно $9 \cdot 7 = 63$ бит, а так как для записи используется целое число байт, то берём ближайшее не меньшее значение, кратное восьми, это число $64 = 8 \cdot 8$ бит (8 байт).

Для хранения всех сведений об одном пользователе используется $500 : 20 = 25$ байт. Следовательно, для хранения дополнительных сведений выделено $25 - 8 = 17$ байт.

Ответ: 17.

12. Исполнитель Редактор получает на вход строку цифр и преобразовывает её.

Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w .

Например, выполнение команды **заменить** (555, 63) преобразует строку 12555550 в строку 1263550.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) **нашлось** (v).

Эта команда проверяет, встречается ли цепочка v в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 1000 идущих подряд цифр 8? В ответе запишите полученную строку.

НАЧАЛО

ПОКА нашлось (999) ИЛИ нашлось (888)

ЕСЛИ нашлось (888)

ТО заменить (888, 9)

ИНАЧЕ заменить (999, 8)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

Решение. Данный алгоритм сначала заменит все триады из восьмёрок на девятки.

Заметим, что 1000 восьмёрок образуют 333 триады и остаётся ещё одна восьмёрка. То есть получится 333 девятки и одна восьмёрка.

Затем алгоритм заменит 9 первых девяток на три восьмёрки, а затем заменит эти три восьмёрки обратно на одну девятку. То есть за четыре прохода цикла из начала строки убирается 8 девяток.

Таким образом, из строки, содержащей 333 девятки, уберётся $8 \cdot 41 = 328$ девяток, и останется 5 девяток и одна восьмёрка. Три первых девятки будут заменены на одну восьмёрку, и останется строка 8998.

Ответ: 8998.

Примечание.

Рекомендуем сравнить эту задачу с задачей [10504](#). Исходные данные в этих задачах «симметричные» (строка из 1000 девяток или из 1000 восьмёрок), но ответы получаются «несимметричные», что связано с порядком замены строк внутри цикла: в первую очередь заменяются строки из восьмёрок.

Приведём другое решение на языке Python.

```
s = '8' * 1000
while ('999' in s) or ('888' in s):
    if '888' in s:
        s = s.replace('888', '9', 1)
    else:
        s = s.replace('999', '8', 1)
print(s)
```

13. Доступ к файлу index.html, размещённому на сервере www.ftp.ru, осуществляется по протоколу http. В таблице приведены фрагменты адреса этого файла, обозначенные буквами от А до З. Запишите последовательность этих букв, соответствующую адресу данного файла.

А	.html
Б	www.
В	/
Г	ftp
Д	.ru
Е	http
Ж	index
З	://

Решение. Адрес файла начинается с протокола, после этого ставятся знаки «://», имя сервера, каталог и имя файла. Здесь протокол — под буквой Е, «://» — под буквой З, имя сервера — под буквами БГД, далее идет разделитель «/» (В), затем — имя файла (ЖА).

Ответ: ЕЗБГДВЖА.

14. Значение арифметического выражения $49^{10} + 7^{30} - 49$ записали в системе счисления с основанием 7. Сколько цифр 6 содержится в этой записи?

Решение. Последовательно рассмотрим данное выражение.

Рассмотрим первое слагаемое: $49^{10} = 7^{20} = 100...00_7$. — на конце 20 нулей.

Рассмотрим второе слагаемое: $7^{30} = 100...00_7$ — на конце 30 нулей.

Представим число 49 в системе счисления с основанием 7: $7^2 = 100_7$.

Результат сложения первых двух чисел: $7^{30} + 7^{20} = 100...00100...00_7$ — на конце 20 нулей, а в середине $30 - 20 - 1 = 9$ нулей.

На последнем шаге получаем: $7^{30} + 7^{20} - 7^2 = 100...006...6600_7$ — на конце два нуля, затем 18 шестёрок, а в середине 10 нулей.

Ответ: 18.

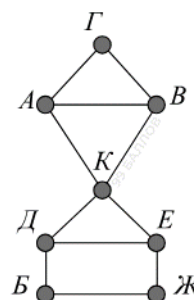
Приведём другое решение на языке Python.

```
x = 49**10 + 7**30 - 49
s = ''
while x != 0:
    s += str(x % 7)
    x //= 7
s = s[::-1]
print(s.count("6"))
```

Контрольная работы «Информация и информационные процессы»

- На рисунке схема дорог Н-ского района изображена в виде графа, в таблице содержатся данные о протяженности дорог между населёнными пунктами (в километрах). Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе.

Определите длину кратчайшей дороги между пунктами Д и Ж при условии, что дорога ДБ длиннее дороги ЕЖ.



		Номер пункта							
		1	2	3	4	5	6	7	8
Номер пункта	1		11			20			4
	2	11				5			5
	3						20	10	2
	4						1	3	
	5	20	5						
	6			20	1				
	7			10	3				1
	8	4	5	2				1	

- Ассистент Быбыл узнал, что по секретному телеграм каналу передаются сообщения, содержащие шесть букв: Б, Л, М, Ы, Р, К. Для передачи используется двоичный код, допускающий однозначное декодирование. Кодовые слова для некоторых букв ассистент смог узнать: М - 00, Р - 10, Ы - 110. Какое наименьшее кол-во символов потребуется, чтобы закодировать сообщение БЫБЫЛ?

3. Ваня записывает лучший вебинар по информатике с частотой 60 кадров в секунду, при этом изображения используют палитру, содержащую 2048 цветов, с разрешением 1080*1920 пикселей. Аудиодорожка записывается в формате квадрато, с частотой дискретизации 64 кГц и 32-битным разрешением. Какой минимальный объем памяти в Мбайтах нужно зарезервировать, чтобы записать 2 минуты такого вебинара?
4. Каждый сотрудник компании имеет электронную карту с личным кодом, номером подразделения(целое число от 1 до 2000) и дополнительной информацией. Личный код состоит из 15 символов, которые могут быть латинскими буквами (включая заглавные и строчные), десятичными цифрами и специальными символами из набора @#№&?. Для хранения кода используется посимвольное кодирование, все символы кодируются одинаковым минимально возможным количеством битов, для записи кода отводится минимально возможное целое число байтов. Номер подразделения кодируется отдельно и занимает минимально возможное целое число байтов. Известно, что на одной карте хранится всего 30 байтов данных. Сколько байтов занимает дополнительная информация?
5. Все 5-буквенные слова, составленные из букв С, А, Н, Я. Вот начало списка:
1. ССССС
 2. ССССА
 3. ССССН
 4. ССССЯ
- ...
- Запишите слово, которое стоит на $1010000_2 \cdot 11_2 - 11100_2$ -м месте от начала списка.
6. Директор решил занести всех учеников его школы в базу данных. Для каждого ученика составляется логин и пароль. Пароль содержит 7 символов и составляется из шестнадцатеричных цифр и 241 символа из специального набора. Логин создается автоматически и должен допускать однозначное декодирование логина каждого ученика (для каждого ученика создается свой отдельный логин). В базе данных для хранения сведений о каждом логине и пароле отдельно отведено минимально возможное целое число байт. Все пароли (и логины, отдельно) должны занимать одинаковое число байт. При этом используют посимвольное кодирование логина и пароля, все символы кодируются минимальным и одинаковым количеством бит. Определите, сколько дисков объемом 1 МБ нужно для хранения в базе данных 132000 учеников.

Контрольная работы «Основы объектно-ориентированного программирования»

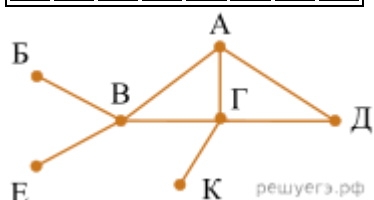
1. Найти длину строки.
2. Строка содержит последовательность из строчных и заглавных букв английского алфавита и цифр. Определите длину наибольшей убывающей подпоследовательности.

3. Строка содержит последовательность из символов «(»и «)». Определить количество пар скобок «()» в этой строке
4. На вход алгоритма подаётся натуральное число N . Алгоритм строит по нему новое число R следующим образом.
 1. Строится двоичная запись числа N .
 2. Складываются все цифры двоичной записи числа N . Если полученная сумма чётна, из числа убирают ведущую единицу (а также ставшие незначащими нули). В противном случае слева приписывается 1, а справа – два нуля.
 3. Над новой записью снова производятся действия, описанные в пункте 2.
 4. Результат переводится в десятичную систему и выводится на экран.
 Например, $N = 5_{10} = 101_2 \Rightarrow 1 \Rightarrow 1100_2 = 12_{10} = R$
 Укажите такое наименьшее число N , для которого результат работы данного алгоритма больше 100. В ответе это число запишите в десятичной системе счисления.
5. Назовём маской числа последовательность цифр, в которой также могут встречаться следующие символы:
 - символ «?» означает ровно одну произвольную цифру;
 - символ «*» означает любую последовательность цифр произвольной длины; в том числе «*» может задавать и пустую последовательность.
 Среди натуральных чисел, не превышающих 10^8 , найдите все числа, соответствующие маске $12*4?65$, делящиеся на 161 без остатка. В ответе запишите в первом столбце таблицы все найденные числа в порядке возрастания, а во втором столбце – соответствующие им результаты деления этих чисел на 161.

Промежуточная аттестация

1. На рисунке справа схема дорог Н-ского района изображена в виде графа; в таблице слева содержатся сведения о протяжённости каждой из этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		7					
п2	7		8		3	4	
п3		8		11	6		
п4			11		5		
п5		3	6	5			9
п6		4					
п7					9		



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта A в пункт $Г$. В ответе запишите целое число — так, как оно указано в таблице.

Решение. Сопоставим населённые пункты графа и населённые пункты в таблице. Из A ведут три дороги. Только из пункта $ПЗ$ ведут три дороги, следовательно, пункт $ПЗ$ — это A .

Из K ведёт одна дорога — в Γ . Заметим из таблицы, что из пункта $П7$ ведёт одна дорога в пункт $П5$, следовательно, $П7$ — это пункт K , а $П5$ — пункт Γ . Таким образом, длина дороги из A в Γ равна длине дороги из $П3$ в $П5$ — 6.

Ответ: 6.

2. Миша заполнял таблицу истинности логической функции F

$$\neg(y \rightarrow (x \equiv w)) \wedge (z \rightarrow x),$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

Переменная 1	Переменная 2	Переменная 3	Переменная 4	Функция
	1	1		1
0			0	1
	0	1	0	1

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z . В ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т. д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Решение. Составим таблицу истинности для выражения $\neg(y \rightarrow (x \equiv w)) \wedge (z \rightarrow x)$ вручную или при помощи языка Python:

```
print("x y z w")
for x in range(0, 2):
    for y in range(0, 2):
        for z in range(0, 2):
            for w in range(0, 2):
                if not(y <= (x == w)) and (z <= x):
                    print(x, y, z, w)
```

Далее выпишем те наборы переменных, при которых данное выражение равно 1. В наборах переменные запишем в порядке x, y, z, w . Получим следующие наборы:

(0, 1, 0, 1),

(1, 1, 0, 0),

(1, 1, 1, 0).

Сопоставим эти наборы с приведенным в задании фрагментом таблицы истинности. Первая строка таблицы может соответствовать только набору (1, 1, 1, 0), поскольку в двух других строках точно есть два нуля. Следовательно, переменная w может соответствовать либо первому, либо четвёртому столбцу. Поскольку в оставшихся двух строках таблицы в четвёртом столбце стоит 0, а из оставшихся двух наборов только в одном переменная w равна 0, переменная w соответствует первому столбцу. Рассмотрим третью строку таблицы. Эта строка может соответствовать только набору (0, 1, 0, 1), поскольку переменная w должна быть равна 1. Следовательно, в ней $y = 1$ и $w = 1$ и y соответствует третьему столбцу.

Рассмотрим вторую строку таблицы. Эта строка может соответствовать только набору (1, 1, 0, 0). Следовательно, в ней $z = 0$ и $w = 0$ и z соответствует четвёртому столбцу. Тогда второй столбец — это переменная x .

Ответ: wxyz.

Скопируем получившуюся таблицу на новый лист.

4. Для кодирования некоторой последовательности, состоящей из букв И, К, Л, М, Н, решили использовать неравномерный двоичный код, удовлетворяющий условию Фано.

Для буквы Л использовали кодовое слово 1, для буквы М — кодовое слово 01. Какова наименьшая возможная суммарная длина всех пяти кодовых слов?

Примечание. Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Решение. Условие Фано — никакое кодовое слово не может быть началом другого кодового слова. Поскольку уже имеется кодовое слово 1, то никакое другое не может начинаться с 1. Только с 0. Также не может начинаться с 01, поскольку у нас уже есть 01. То есть любое новое кодовое слово будет начинаться с 00. Но это не может быть 00, так как иначе мы не сможем взять больше ни одного кодового слова, поскольку все более длинные слова начинаются либо с 1, либо с 00, либо с 01. Мы можем взять либо 000, либо 001. Но не оба сразу, поскольку опять же в таком случае мы больше не сможем взять ни одного нового кода. Тогда возьмём 001. И так как нам осталось всего два кода, то можем взять 0000 и 0001. Итого имеем: 1, 01, 001, 0000, 0001. Всего 14 символов.

Ответ: 14.

5. Автомат получает на вход четырёхзначное число (число не может начинаться с нуля). По этому числу строится новое число по следующим правилам.

1. Складываются отдельно первая и вторая, вторая и третья, третья и четвертая цифры заданного числа.

2. Наименьшая из полученных трёх сумм удаляется.

3. Оставшиеся две суммы записываются друг за другом в порядке неубывания без разделителей.

Пример. Исходное число: 1984. Суммы: $1 + 9 = 10$, $9 + 8 = 17$, $8 + 4 = 12$.

Удаляется 10. Результат: 1217.

Укажите **наименьшее** число, при обработке которого автомат выдаёт результат 613.

Решение. Понятно, что 613 — это числа 6 и 13. Чтобы получить наименьшее число, возьмем третью сумму минимально возможную — 1. Минимальная сумма из трех — 1. Её и разобьем, поставив в начало единицу, так как с 0 числа не начинаются. Получаем 10XY. Поскольку остается всего 2 суммы 6 и 13, ставим максимально большие слагаемые в конец. Находим исходное число — 1067.

Ответ: 1067.

Приведём другое решение на языке Python.

```
for i in range(1000, 10000):
    s = str(i)
    k1 = int(s[0]) + int(s[1])
    k2 = int(s[1]) + int(s[2])
    k3 = int(s[2]) + int(s[3])
    first = str(k1 + k2 + k3 - max(k1, k2, k3) - min(k1, k2, k3))
    second = str(max(k1, k2, k3))
    s1 = first + second
    if s1 == '613':
        print(i)
        break
```

6. Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует 6 команд: **Поднять хвост**, означающая переход к перемещению без рисования; **Опустить хвост**, означающая переход в режим рисования; **Вперёд n** (где n — целое число), вызывающая передвижение Черепахи на n единиц в том

направлении, куда указывает её голова; **Назад n** (где n — целое число), вызывающая передвижение в противоположном голове направлении; **Направо m** (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке, **Налево m** (где m — целое число), вызывающая изменение направления движения на m градусов против часовой стрелки. Запись **Повтори k [Команда1 Команда2 ... Команда S]** означает, что последовательность из S команд повторится k раз.

Черепаше был дан для исполнения следующий алгоритм:

Повтори 4 [Вперёд 10 Направо 270]

Поднять хвост

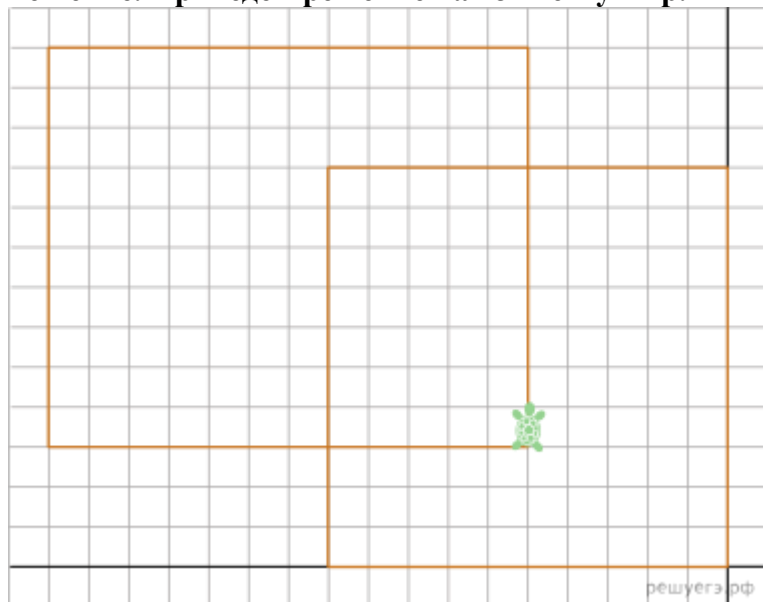
Вперёд 3 Направо 270 Вперёд 5 Направо 90

Опустить хвост

Повтори 2 [Вперёд 10 Направо 270 Вперёд 12 Направо 270].

Определите, сколько точек с целочисленными координатами будут находиться внутри объединения фигур, ограниченных заданными алгоритмом линиями, включая точки на линиях.

Решение. Приведём решение на языке КуМир.



Выберите масштаб сетки равным 1, запустите программу для исполнителя Черепаша. использовать Черепаша

```

алг
нач
опустить хвост
нц 4 раз
вперед (10)
вправо (270)
кц
поднять хвост
вперед (3)
вправо (270)
вперед (5)
вправо (90)
опустить хвост
нц 2 раз
вперед (10)
вправо (270)
вперед (12)
вправо (270)
кц
кон

```

После выполнения программы считаем точки фигуры получаем 216.

Ответ: 216.

Приведём построение рисунка на языке Python.

```
from turtle import *
lt(90)
size = 30
screensize(2000, 2000)
tracer(0)
down()
for i in range(4):
    fd(10 * size)
    rt(270)
up()
fd(3 * size)
rt(270)
fd(5*size)
rt(90)
down()
for i in range(2):
    fd(10 * size)
    rt(270)
    fd(12 * size)
    rt(270)
up()
for x in range(-20, 20):
    for y in range(-20, 20):
        setpos(x * size, y * size)
        dot(4, 'red')
done()
```

7. Графический файл с разрешением 1024 x 600 на жестком диске занимает не более 120 КБайт. Определите максимальное количество цветов, которое может использоваться для кодирования данного изображения.

Решение. Объём растрового изображения находится как произведение количества пикселей в изображении на объём памяти x , необходимый для хранения цвета одного пикселя: $1024 \cdot 600 \cdot x \leq 120 \cdot 2^{10} \cdot 2^3$ бит, откуда $x \leq 1,6$ бит, откуда находим $x = 1$ бит. Значит, в изображении можно использовать не более двух цветов.

Ответ: 2.

8. Игорь составляет таблицу кодовых слов для передачи сообщений, каждому сообщению соответствует своё кодовое слово. В качестве кодовых слов Игорь использует 5-буквенные слова, в которых есть только буквы А, В, С, Х, причём буква Х появляется ровно 1 раз. Каждая из других допустимых букв может встречаться в кодовом слове любое количество раз или не встречаться совсем. Сколько различных кодовых слов может использовать Игорь?

Решение. Пусть Х стоит в слове на первом месте. Тогда на каждое из оставшихся 4 мест можно поставить независимо одну из 3 букв. То есть всего $3 \cdot 3 \cdot 3 \cdot 3 = 81$ вариант.

Таким образом, Х можно по очереди поставить на все 5 мест, в каждом случае получая 81 вариант.

Итого получается $81 \cdot 5 = 405$ слов.

Ответ: 405.

Приведём другое решение на языке Python.

```
import itertools
alphabet = "ABCX"
ar = itertools.product(alphabet, repeat=5) #Размещение с повторением
arl = []
for i in ar:
    arl.append(list(i))
```



```
count = 0
for e in arl:
    if e.count('X') == 1:
        count += 1
print(count)
```

9. Откройте файл электронной таблицы, содержащей в каждой строке три натуральных числа.

Задание 9

Определите, сколько среди заданных троек чисел таких, которые могут быть сторонами остроугольного треугольника.

Решение. Заметим, что треугольник является остроугольным, если квадрат длины наибольшей стороны треугольника будет меньше суммы квадратов длин других двух сторон. Тогда в ячейке D1 запишем формулу $=\text{МАКС}(\text{A1:C1})^2$ и скопируем её во все ячейки диапазона D2:D5000. В ячейке E1 запишем формулу:

$$=(\text{МИН}(\text{A1:C1}))^2+(\text{СУММ}(\text{A1:C1})-\text{МИН}(\text{A1:C1})-\text{МАКС}(\text{A1:C1}))^2$$

и скопируем её во все ячейки диапазона E2:E5000. Таким образом, получим квадрат длины наибольшей стороны и сумму квадратов других двух сторон для каждой тройки чисел. После этого в ячейку F1 запишем формулу $=\text{ЕСЛИ}(\text{D1}<\text{E1};1;0)$ и скопируем её во все ячейки диапазона F2:F5000. Теперь, воспользовавшись формулой $=\text{СУММ}(\text{F1:F5000})$, получим ответ — 1074.

Ответ: 1074.

11. Каждый сотрудник предприятия получает электронный пропуск, на котором записаны личный код сотрудника, номер подразделения и некоторая дополнительная информация. Личный код состоит из 13 символов, каждый из которых может быть одной из 26 заглавных латинских букв. Для записи кода на пропуске отведено минимально возможное целое число байт, при этом используют посимвольное кодирование, все символы кодируют одинаковым минимально возможным количеством бит. Номер подразделения — целое число от 1 до 180, он записан на пропуске как двоичное число и занимает минимально возможное целое число байт. Всего на пропуске хранится 32 байта данных. Сколько байт выделено для хранения дополнительных сведений о сотруднике? В ответе запишите только целое число — количество байт.

Решение. k бит позволяют кодировать 2^k символов, поэтому для кодирования 26-символьного алфавита требуется 5 бит (ведь $2^5 = 32 \geq 26$). Для хранения 13 символов требуется $13 \cdot 5 = 65$ бит. Минимальное количество байт, вмещающее в себя 65 бит, — 9 байт (72 бита).

Номер подразделения можно закодировать 1 байтом, поскольку в 1 байт помещаются число от 0 до 255.

Для хранения данных об одном сотруднике требуется 32 байта данных. Из них 9 байт отводится на хранение личного кода, ещё 1 байт требуется для хранения номера подразделения. Следовательно, для хранения дополнительных сведений о сотруднике отводится 22 байт.

Ответ: 22.

12. Исполнитель Редактор получает на вход строку цифр и преобразовывает её.

Редактор может выполнять две команды, в обеих командах v и w обозначают цепочки цифр.

А) **заменить** (v, w).

Эта команда заменяет в строке первое слева вхождение цепочки v на цепочку w .

Например, выполнение команды **заменить** (111, 27) преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки v , то выполнение команды **заменить** (v, w) не меняет эту строку.

Б) **нашлось** (*v*).

Эта команда проверяет, встречается ли цепочка *v* в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

последовательность команд

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно).

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (111)

заменить (111, 2)

заменить (222, 11)

КОНЕЦ ПОКА

КОНЕЦ

К исходной строке, содержащей более 60 единиц и не содержащей других символов, применили приведённую выше программу. В результате получилась строка 2211. Какое наименьшее количество единиц могло быть в исходной строке?

Решение. Заметим, что алгоритм заменяет 9 единиц на 2 единицы. То есть если в исходной строке как минимум 9 единиц, то за три прохода количество единиц уменьшится на 7.

Заметим, что строка 2211 получается из исходной строки, состоящей из восьми единиц. Наибольшее число, кратное 7 и меньшее 60 — 56. Таким образом, наименьшее количество единиц, которое могло быть в исходной строке, равно $56 + 8 = 64$.

Ответ: 64.

Приведем другое решение.

```
for n in range(61, 71):
    s = '1' * n
    while '111' in s:
        s = s.replace('111', '2', 1)
        s = s.replace('222', '11', 1)
    if s == '2211':
        break
print(n)
```

13. В терминологии сетей TCP/IP маска сети — это двоичное число, меньшее 2^{32} ; в маске сначала (в старших разрядах) стоят единицы, а затем с некоторого места нули. Маска определяет, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. Обычно маска записывается по тем же правилам, что

и IP-адрес, — в виде четырёх байт, причём каждый байт записывается в виде десятичного числа. Адрес сети получается в результате применения поразрядной конъюнкции к заданному IP-адресу узла и маске. Например, если IP-адрес узла равен 231.32.255.131, а маска равна 255.255.240.0, то адрес сети равен 231.32.240.0. Для узла с IP-адресом 224.128.114.142 адрес сети равен 224.128.96.0. Чему равен третий слева байт маски? Ответ запишите в виде десятичного числа.

Решение. Рассмотрим третий слева байт в IP-адресе узла и адресе сети, представим их в двоичном виде:

$$114_{10} = 0111\ 0010_2; \ 96_{10} = 0110\ 0000_2.$$

Найдём такое число, поразрядная конъюнкция которого с числом 114 даст число 96: $1110\ 0000_2 = 224_{10}$.

Ответ: 224.

14. Значение арифметического выражения $9^8 \cdot 3^{20} - 3^{10} - 3$ записали в системе счисления с основанием 3. Сколько цифр 2 содержится в этой записи?

Решение. Последовательно рассмотрим данное выражение.

Рассмотрим первое слагаемое: $9^8 \cdot 3^{20} = 3^{36} = 100\dots 00_3$. — на конце 36 нулей.

Рассмотрим второе слагаемое: $3^{10} = 100\dots 00_3$ — на конце 10 нулей.

Результат разности первых двух чисел: $3^{36} - 3^{10} = 22\dots 2200\dots 00_3$ — 26 двоек и 10 нулей.

На последнем шаге получаем: $22\dots 2200\dots 00_3 - 10_3 = 22\dots 210_3$ — 25 двоек, единица, 9 двоек, а на конце ноль.

Таким образом, получаем, что в троичной записи исходного выражения содержится 34 двойки.

Ответ: 34.

Приведём другое решение на языке Python.

```
x = (9**8)*(3**20) - 3**10 - 3
s = ''
while x != 0:
    s += str(x % 3)
    x //= 3
s = s[::-1]
print(s.count("2"))
```

15. Обозначим через ДЕЛ(n, m) утверждение «натуральное число n делится без остатка на натуральное число m ».

Для какого наибольшего натурального числа A формула

$$\neg \text{ДЕЛ}(x, A) \rightarrow (\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$$

тождественно истинна (то есть принимает значение 1 при любом натуральном значении переменной x)?

Решение. Рассмотрим такие x , при которых скобка $(\text{ДЕЛ}(x, 6) \rightarrow \neg \text{ДЕЛ}(x, 9))$ будет ложной. Это x , которые делятся без остатка одновременно на 6 и на 9. Наименьшее общее кратное этих чисел равно 18.

Следовательно, для $x = 18$ выражение $\neg \text{ДЕЛ}(x, A)$ должно быть ложным, то есть число 18 должно делиться на A . Наибольшим таким A является число 18. Это и будет ответ.

Ответ: 18.

Приведём другое решение на языке Python.

```
for a in range(100, 0, -1):
    k = 0
    for x in range(1, 1000):
        if (x % a != 0) <= ((x % 6 == 0) <= (x % 9 != 0)):
            k += 1
```

```

if k == 999:
    print(a)
    break

```

16. Алгоритм вычисления значения функции $F(n)$, где n — натуральное число, задан следующими рекуррентными соотношениями:

$F(n) = 1$ при $n = 1$;

$F(n) = F(n - 1) \cdot n$ при $n \geq 2$.

Чему равно значение функции $F(6)$? В ответе запишите только натуральное число.

Решение. Последовательно найдём значения F :

$F(1) = 1$;

$F(2) = 2$;

$F(3) = 6$;

$F(4) = 24$;

$F(5) = 120$;

$F(6) = 720$.

Таким образом, ответ $F(6) = 720$.

Ответ: 720.

Приведём другое решение на языке Python.

```

def F(n):
    if n == 1:
        return 1
    if n >= 2:
        return F(n - 1) * n
print(F(6))

```

17. В файле содержится последовательность из 10 000 целых положительных чисел. Каждое число не превышает 10 000. Определите и запишите в ответе сначала количество пар элементов последовательности, для которых произведение элементов не кратно 14, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

[17.txt](#)

Решение. Сначала считаем все числа из файла в массив. Для каждой пары будем проверять, кратно ли 14 произведение элементов пары. Если произведение не кратно 14, будем увеличивать значения счётчика count и проверять, больше ли сумма элементов пары текущей максимальной суммы. Если сумма элементов пары больше текущей максимальной суммы, будем обновлять значение переменной maxsum.

Приведём решение задачи на языке Pascal.

```

var
i, j: integer;
count: longint;
maxsum: integer;
arr: array[1..10000] of integer;
f: text;
begin
    assign(f, 'C:\17.txt');
    reset(f);
    maxsum := 0;
    count := 0;
    for i := 1 to 10000 do readln(f, arr[i]);
    for i := 1 to 10000 - 1 do
        for j := i + 1 to 10000 do begin
            if (arr[i] * arr[j] mod 14 <> 0) then begin
                count := count + 1;
                if arr[i] + arr[j] > maxsum then maxsum := arr[i] + arr[j];
            end;
        end;
    end;

```

```

        end;
        writeln(count, ' ', maxsum);
    end.

```

Приведём решение Николая Чуркина (Тимашевск) на языке Python.

```

count = m = 0
f = open('17.txt')
l = [int(i) for i in f]
for i in range(len(l) - 1):
    for j in range(i + 1, len(l)):
        if l[i] * l[j] % 14 != 0:
            count += 1
            m = max(m, l[i] + l[j])
print(count, m)

```

В результате работы данного алгоритма при вводе данных из файла ответ — 40436496 19999.

Примечание.

Путь к файлу необходимо указать согласно расположению файла на Вашем компьютере.

18. Робот стоит в левом нижнем углу прямоугольного поля, в каждой клетке которого записано натуральное число. За один ход робот может переместиться на одну клетку вправо или на одну клетку вверх. Выходить за пределы поля робот не может. Некоторые клетки на поле окружены границами, в эти клетки робота заходить нельзя. В начальный момент запас энергии робота составляет 3000 единиц. Проходя через каждую клетку, робот расходует энергию, при этом расход равен числу, записанному в клетке. В клетках с выделенным фоном находятся зарядные станции. При прохождении через эти клетки робот не расходует, а пополняет запас энергии. Сумма пополнения равна числу, записанному в этой клетке.

Определите максимальный и минимальный запас энергии, который может быть у робота после перехода в правую верхнюю клетку поля. В ответе запишите два числа: сначала максимально возможное значение, затем — минимальное.

Исходные данные записаны в электронной таблице. Границы отмечены утолщёнными линиями.

Задание 18

Пример входных данных (для таблицы размером 4×4):

13	8	69	50
30	35	57	17
32	90	55	32
44	12	80	43

При указанных входных данных максимальное значение получается при движении по маршруту:

$$3000 - 44 - 12 + 90 - 55 - 32 - 17 - 50 = 2880,$$

а минимальное — при движении по маршруту:

$$3000 - 44 - 12 - 80 - 55 - 32 - 17 - 50 = 2710.$$

Решение. Сначала найдём максимально возможный запас энергии. Поскольку робот не может заходить в ячейки, окружённые стенами, установим в ячейки E6, E11, K6, K11 значения 10000. В ячейках с зарядными станциями инвертируем положительные значения в отрицательные, чтобы при использовании вычитаний значения в зарядных станциях, наоборот, прибавлялись к запасу топлива.

В ячейку A31 запишем формулу **=3000-A15**. Для диапазонов **B31:O31** и **A17:A30**, при переходе в очередную ячейку диапазона, из текущего запаса энергии будем вычитать значение этой ячейки. В ячейку B31 запишем формулу **=A31-B15** и скопируем её во

все ячейки диапазона **C31:O31**. В ячейку A30 запишем формулу **=A31-A14** и скопируем её во все ячейки диапазона **A17:A29**. В ячейку B30 запишем формулу **=МАКС(A30-B14;B31-B14)** и скопируем её во все ячейки диапазона **B17:O30**. Таким образом, в ячейке O17 получим максимально возможное значение запаса энергии — 2201.

Теперь найдём минимально возможный запас энергии. Поскольку робот не может заходить в ячейки, окружённые стенами, установим в ячейки E6, E11, K6, K11 значения –10000. В ячейках с зарядными станциями инвертируем положительные значения в отрицательные, чтобы при использовании вычитаний значения в зарядных станциях, наоборот, прибавлялись к запасу топлива.

В ячейку A31 запишем формулу **=3000-A15**. Для диапазонов **B31:O31** и **A17:A30**, при переходе в очередную ячейку диапазона, из текущего запаса энергии будем вычитать значение этой ячейки. В ячейку B31 запишем формулу **=A31-B15** и скопируем её во все ячейки диапазона **C31:O31**. В ячейку A30 запишем формулу **=A31-A14** и скопируем её во все ячейки диапазона **A17:A29**. В ячейку B30 запишем формулу **=МИН(A30-B14;B31-B14)** и скопируем её во все ячейки диапазона **B17:O30**. Таким образом, в ячейке O17 получим минимально возможное значение запаса энергии — 964.

Ответ: 2201 и 964.

19. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в кучу один камень, добавить два камня или увеличить количество камней в куче в два раза**. При этом нельзя повторять ход, который только что сделал второй игрок.

Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Если Петя получил кучу из 5 камней (добавил 2 камня), то следующим ходом Ваня может получить 6 или 10 камней. Получить 7 камней Ваня не может, так как для этого нужно добавить 2 камня, а такой ход только что сделал Петя. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается, когда количество камней в куче становится не менее 34. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 34 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 33$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Укажите такое значение S , при котором Петя не может выиграть за один ход, но при любом ходе Пети Ваня может выиграть своим первым ходом.

Решение. Такое значение S — 16. Своим первым ходом Петя может получить позиции 17, 18 и 32. В первых двух случаях Ваня удваивает количество камней в куче, в третьем случае Ваня добавляет в кучу два камня. Во всех случаях Ваня выигрывает своим первым ходом.

Ответ: 16.

Приведём другое решение на языке Python.

Для контроля ходов игроков заведём переменную m .

Если $m = 1$, то это действие $x + 1$

Если $m = 2$, то это действие $x + 2$

Если $m = 3$, то это действие $x * 2$

Таким образом, можно исключать ходы, которые уже были сделаны, не включая их в возвращаемые значения функции.

```
def f(x, h, m):
    if h == 3 and x >= 34:
        return 1
    elif h == 3 and x < 34:
        return 0
    elif x >= 34 and h < 3:
```

```

        return 0
    else:
        if h % 2 == 0:
            if h == 2:
                if m == 1:
                    return f(x + 2, h + 1, m) or f(x * 2, h + 1, m)
                elif m == 2:
                    return f(x + 1, h + 1, m) or f(x * 2, h + 1, m)
                elif m == 3:
                    return f(x + 1, h + 1, m) or f(x + 2, h + 1, m)
            else:
                return f(x + 1, h + 1, 1) and f(x + 2, h + 1, 2) and f(x * 2, h
+ 1, 3)

for x in range(1, 34):
    if f(x, 1, 0) == 1:
        print("Задача 19:", x)
        break

```

20. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в кучу один камень, добавить два камня или увеличить количество камней в куче в два раза**. При этом **нельзя повторять ход, который только что сделал второй игрок**.

Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Если Петя получил кучу из 5 камней (добавил 2 камня), то следующим ходом Ваня может получить 6 или 10 камней. Получить 7 камней Ваня не может, так как для этого нужно добавить 2 камня, а такой ход только что сделал Петя. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается, когда количество камней в куче становится не менее 34. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 34 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 33$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Существует несколько таких значений S , при которых у Пети есть выигрышная стратегия, причём Петя не может выиграть первым ходом, но может выиграть своим вторым ходом независимо от того, как будет ходить Ваня. Найдите наименьшее и наибольшее из таких значений S .

В ответе запишите сначала наименьшее, затем наибольшее значение.

Решение. Рассмотрим наименьшее значение $S = 8$. Своим первым ходом Петя может получить позицию 16, удвоив количество камней в куче. После этого Ваня может получить позиции 17 и 18 (позицию 32 Ваня получить не может, поскольку нельзя повторять ход, который только что сделал второй игрок). В любой из позиций Петя может удвоить количество камней в куче и выиграть своим вторым ходом.

Наибольшее значение $S = 15$. Своим первым ходом Петя добавляет в кучу один камень и получает позицию 16. После этого Ваня может получить позиции 18 и 32 (позицию 17 Ваня получить не может, поскольку нельзя повторять ход, который только что сделал второй игрок). В первом случае Петя может удвоить количество камней в куче, во втором случае Петя может добавить в кучу два камня. В обеих ситуациях Петя побеждает своим вторым ходом.

Таким образом, ответ — 815.

Ответ: 815.

21. Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может **добавить в кучу один камень, добавить два камня или увеличить количество камней в куче в два раза**. При этом **нельзя повторять ход, который только что сделал второй игрок**.

Например, если в начале игры в куче 3 камня, Петя может первым ходом получить кучу из 4, 5 или 6 камней. Если Петя получил кучу из 5 камней (добавил 2 камня), то следующим ходом Ваня может получить 6 или 10 камней. Получить 7 камней Ваня не может, так как для этого нужно добавить 2 камня, а такой ход только что сделал Петя. Чтобы делать ходы, у каждого игрока есть неограниченное количество камней. Игра завершается, когда количество камней в куче становится не менее 34. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 34 или больше камней. В начальный момент в куче было S камней; $1 \leq S \leq 33$.

Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Найдите значение S , при котором у Вани есть выигрышная стратегия, позволяющая ему выиграть вторым ходом при любой игре Пети, но у Вани нет стратегии, которая позволяла бы ему гарантированно выиграть первым ходом.

Решение. Такое значение S — 7. При $S = 7$ Петя своим первым ходом может получить одну из трёх позиций: 8, 9 или 14. В первом и третьем случаях Ваня может получить позицию 16, удвоив количество камней в куче или добавив два камня в кучу соответственно. Тогда, вне зависимости от хода Пети, Ваня побеждает своим вторым ходом.

В ситуации, когда после первого хода Пети в куче становится 9 камней, Ваня может удвоить количество камней в куче и получить позицию 18. Тогда Петя своим вторым ходом может получить позиции 19 и 20 (позицию 36 Ваня получить не может, поскольку нельзя повторять ход, который только что сделал второй игрок). Своим вторым ходом Ваня может удвоить количество камней в куче и выиграть игру.

Ответ: 7.