

Муниципальное автономное общеобразовательное учреждение города
Новосибирска
«Лицей №22 «Надежда Сибири»
Главный корпус на Советской: г. Новосибирск, ул. Советская, 63, тел. 222-35-15,
e-mail: l_22@edu54.ru
Корпус 99 на Чаплыгина: г. Новосибирск, ул. Чаплыгина, 59, тел. 223-74-15

РАССМОТРЕНО на заседании инженерной кафедры протокол № 1 от 25.08.2025  Кириленко К.А. ФИО руководителя кафедры	СОГЛАСОВАНО Заместитель директора  Н.А. Данилова от 29.08.2025
--	---

РАБОЧАЯ ПРОГРАММА
Информатика. Программирование на Пайтон
5 - 9 классов
(уровень основного общего образования)

Разработчик:
Кириленко К.А.

Новосибирск, 2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа по учебному предмету «Программирование на Пайтон» (предметная область «Математика и информатика») (далее соответственно – программирование) составлена на основе Федеральной рабочей программы по информатике, включает пояснительную записку, содержание обучения, планируемые результаты освоения программы по программированию, тематическое планирование.

Пояснительная записка отражает общие цели и задачи изучения информатики, место в структуре учебного плана, а также подходы к отбору содержания, к определению планируемых результатов.

Содержание обучения раскрывает содержательную линию «Алгоритмизация и программирование», которая предлагается для обязательного изучения в каждом классе на уровне основного общего образования.

Планируемые результаты освоения программы по предмету «Программирование на Пайтон» включают личностные, метапредметные результаты за весь период обучения на уровне основного общего образования, а также предметные достижения обучающегося за каждый год обучения.

Развитие алгоритмического мышления как необходимого условия профессиональной деятельности в современном информационном обществе, предполагающего способность обучающегося разбивать сложные задачи на более простые подзадачи, сравнивать новые задачи с задачами, решёнными ранее, определять шаги для достижения результата и так далее.

Изучение программирования оказывает существенное влияние на формирование мировоззрения обучающегося, его жизненную позицию, закладывает основы понимания принципов функционирования и использования информационных технологий как необходимого инструмента практически любой деятельности и одного из наиболее значимых технологических достижений современной цивилизации. Многие предметные знания и способы деятельности, освоенные обучающимися при изучении программирования, находят применение как в рамках образовательного процесса при изучении других предметных областей, так и в иных жизненных ситуациях, становятся значимыми для формирования качеств личности, то есть ориентированы на формирование метапредметных и личностных результатов обучения.

Цели и задачи изучения учебного предмета «Программирование на Пайтон»

Целями изучения программирования на уровне основного общего образования являются:

1. Формирование основ мировоззрения, соответствующего современному уровню развития науки информатики, достижениям научно-технического прогресса и общественной практики, за счёт развития представлений об информации как о важнейшем стратегическом ресурсе развития личности, государства, общества, понимание роли информационных процессов, информационных ресурсов и информационных технологий в условиях цифровой трансформации многих сфер жизни современного общества;
2. Развитие алгоритмического мышления как необходимого условия профессиональной деятельности в современном информационном обществе, предполагающего способность обучающегося разбивать сложные задачи на более простые подзадачи, сравнивать новые задачи с задачами, решёнными ранее, определять шаги для достижения результата и так далее;
3. Формирование и развитие компетенций обучающихся в области использования информационно-коммуникационных технологий, в том числе знаний, умений и навыков работы с языками программирования, в современных цифровых средах в условиях обеспечения информационной безопасности личности обучающегося.

Основные задачи учебного предмета «Информатика» – сформировать у обучающихся:

- базовые знания об информационном моделировании, в том числе о математическом моделировании;
- знание основных алгоритмических структур и умение применять эти знания для построения алгоритмов решения задач по их математическим моделям;

- умения и навыки составления простых программ по построенному алгоритму на одном из языков программирования высокого уровня; умения и навыки эффективного использования основных типов прикладных программ (приложений) общего назначения и информационных систем для решения с их помощью практических задач;
- умение грамотно интерпретировать результаты решения практических задач с помощью информационных технологий, применять полученные результаты в практической деятельности.

Особенности классов

Преподавание предмета «Программирование на Пайтон» осуществляется:

на параллели 5 классов с целью изучения базовых конструкций языка программирования Пайтон;

в профильных 6БАС, 6ИМ, 6М, 6ОМ классах с целью продолжения изучения курса 5 класса и подготовки к поступлению в специализированные 7 классы;

в специализированном 7ПИ (прошли предварительный отбор) и профильном 7ИМ классах с целью углубления в предмет и подготовке к профильным олимпиадам, в связи с этим будут использованы технологии деятельностного подхода с учетом использования задач повышенного и высокого уровня сложности;

в профильных 8БАС, 8М, 8ИТ классах с целью расширения содержательной линии «Алгоритмизация и программирование»;

в специализированном 8ПИ классе с целью углубления в предмет и подготовки к профильным олимпиадам в связи с этим будут использованы технологии деятельностного подхода с учетом использования задач повышенного и высокого уровня сложности;

в специализированном 9ПИ, профильных 9П, 9Т и 9ИТ классах с целью углубления в предмет и подготовки к профильным олимпиадам, в связи с этим будут использованы технологии деятельностного подхода с учетом использования задач повышенного и высокого уровня сложности.

Место предмета в учебном плане лица

Программа реализуется на углубленном уровне и относится к образовательной области Математика и Информатика. Предмет реализуется за счет части, формируемой участниками образовательных отношений

Учебный год	Количество часов									
	5	6БАС	6ИМ, 6М	6ОМ	7ПИ	7ИМ	8БАС, 8М, 8ИТ	8ПИ	9Т, 9ИТ	9ПИ, 9П
2025/2026	0,52	0,48	0,52	1	0,7	0,48	1	1,82	0,5	1,7
2026/2027	-	-	0,71	0,48	1,82	1	1	1,7	-	-
2027/2028	-	-	1,82	1	1,7	1	-	-	-	-
2028/2029	-	-	1,7	1	-	-	-	-	-	-

К тематическому планированию применяется модульный принцип построения образовательной программы, что позволяет выстраивать индивидуальную образовательную парадигму и обеспечивать саморазвитие при индивидуальном темпе работы с учебным материалом, контроль и самоконтроль знаний.

Используемые образовательные технологии, в том числе дистанционные

Обучение программированию может осуществляться с использованием дистанционных образовательных технологий (далее ДОТ), которое предполагает как самостоятельное прохождение учебного материала учеником, так и с помощью сопровождения учителя. При применении ДОТ используются платформы: лицейская платформа дистанционного обучения Moodle, ФГИС «Моя школа», ГИС «Электронная школа» Новосибирской области, Сферум.

При реализации рабочей программы могут быть использованы материалы для подготовки к профилям олимпиады КД НТИ и стандартов Всероссийского чемпионатного движения по профессиональному мастерству «Профессионалы».

Информация о промежуточной аттестации

Промежуточная аттестация осуществляется по окончании учебного модуля с целью проверки степени и качества усвоения материала по результатам изучения тематических модулей и проводится в форме письменных контрольных работ и практических заданий на компьютере.

Текущий контроль осуществляется с целью проверки степени и качества усвоения материала в ходе его изучения в следующих формах: тестовых и практических работ.

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением об осуществлении текущего контроля успеваемости и промежуточной аттестации обучающихся, их формах, периодичности и порядке проведения муниципального автономного общеобразовательного учреждения города Новосибирска «Лицей № 22 «Надежда Сибири» (протокол педагогического совета №1 от 29.08.2023).

Итоговая аттестация проводится в соответствии с законодательством РФ.

Промежуточная аттестация по Программированию на Пайтон в 5 классах

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Введение в программирование	17	16	Тест. Практическая работа

Промежуточная аттестация по Программированию на Пайтон в 6 БАС классе

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Циклы и сложные условия	16	15	Практическая работа

Промежуточная аттестация по Программированию на Пайтон в 6 ИМ, 6М классах

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Циклы и сложные условия	17	16	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 6ОМ классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Математический модуль, сложные условия	10	10	Тест. Практическая работа
МР № 2	Построение графиков функций	7	17	Тест. Практическая работа
МР № 3	Циклы	16	32	Тест. Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 7ПИ классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Строки и списки	13	13	Практическая работа
МР № 2	Множества	10	22	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 7ИМ классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Коллекции	16	15	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 8БАС классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Коллекции	10	10	Практическая работа
МР № 2	Функции	12	22	Практическая работа
МР № 3	Рекурсия	11	32	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 8М, 8ИТ классах**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
------------------------	------------------------	--------------------------------------	---------------------------	-----------------

МР № 1	Базовые конструкции языка	10	10	Практическая работа
МР № 2	Сложные условия и циклы	10	20	Практическая работа
МР № 3	Коллекции	13	32	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 8П, 8ПИ классах**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Базовые конструкции языка	10	10	Практическая работа
МР № 2	Обработка потока данных	10	20	Практическая работа
МР № 3	Строки	8	28	Практическая работа
МР № 4	Структуры	7	35	Практическая работа
МР № 5	Списки	10	45	Практическая работа
МР № 6	Вложенные списки	15	58	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 9ПИ классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Функции	12	12	Практическая работа
МР № 2	Функции с параметрами	10	22	Практическая работа
МР № 3	Коллекции	8	30	Практическая работа
МР № 4	Файлы	10	40	Практическая работа
МР № 5	ООП	11	50	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 9П классе**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Функции	12	12	Практическая работа
МР № 2	Функции с параметрами	10	22	Практическая работа

МР № 3	Коллекции	8	30	Практическая работа
МР № 4	Файлы	10	40	Практическая работа
МР № 5	Способы обработки потока данных	11	50	Практическая работа

**Промежуточная аттестация
по Программированию на Пайтон в 9Т, 9ИТ классах**

№ модульной	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Способы обработки потока данных	16	15	Практическая работа

1. Содержание учебного предмета

5 класс

Модуль 1 «Введение в программирование»

О языке программирования Python, блоки кода, отступы, комментарии, стандартные типы данных, идентификаторы, работа с регистром, присваивание значений переменным, множественное присваивание, проверка типа данных, преобразование различных типов данных, ввод данных, вывод данных. Арифметические операции в Python, операторы присваивания в Python, целочисленное деление Алгоритм ветвления (Управляющие конструкции), тип данных bool, операторы сравнения в Python, логические операторы в Python, , проверка выполнимости условия, оператор if и его синтаксис, оператор else и его синтаксис, оператор elif и его синтаксис, вложенные конструкции, трехместное выражение if/else и его синтаксис.

6 БАС класс

Модуль 1 «Циклы и сложные условия»

Логические операции (и, или, не), логические отношения, операции в скобках, логические переменные (true, false), сложные условия в условном операторе. Цикл for (конструкция и синтаксис), функция range, настройка функции print, счётчик. Цикл While, Составной оператор присваивания, подсчёт количества элементов, удовлетворяющих условию, сигнал остановки, поиск максимума и минимума, обработка потока данных.

6 ИМ, 6М классы

Модуль 1 «Циклы и сложные условия»

Логические операции (и, или, не), логические отношения, операции в скобках, логические переменные (true, false), сложные условия в условном операторе. Цикл for (конструкция и синтаксис), функция range, настройка функции print, счётчик. Цикл While, Составной оператор присваивания, подсчёт количества элементов, удовлетворяющих условию, сигнал остановки, поиск максимума и минимума, обработка потока данных.

6ОМ класс

Модуль 1 «Математический модуль, сложные условия»

Логические операции (и, или, не), логические отношения, операции в скобках, логические переменные (true, false), сложные условия в условном операторе, модуль math,

математические выражения, координатная плоскость, проверка координат, удовлетворяющим фигуре на координатной плоскости.

Модуль 2 «Построение графиков функций»

Построение графиков функций при помощи графического модуля turtle и математического модуля math

Модуль 3 «Циклы»

Цикл While, Составной оператор присваивания, подсчёт количества элементов, удовлетворяющих условию, сигнал остановки, поиск максимума и минимума, Цикл for (конструкция и синтаксис), функция range, настройка функции print, счётчик.

7ПИ класс

Модуль 1 «Множества и строки»

Теория множеств, операции над множествами, операции над двумя множествами, сравнение множеств, сокращенные операторы, тип данных «множество», свойства множеств, методы и функции работы с множествами, символьные (строковые) переменные, посимвольная обработка строк, подсчёт частоты появления символа в строке, встроенные функции для обработки строк.

Модуль 2 «Списки»

Индексация, срезы, методы и функции строк, списочные выражения, обращение к элементу списка.

7ИМ класс

Модуль 1 «Коллекции»

Теория множеств, операции над множествами, операции над двумя множествами, сравнение множеств, сокращенные операторы, тип данных «множество», свойства множеств, методы и функции работы с множествами, символьные (строковые) переменные, посимвольная обработка строк, подсчёт частоты появления символа в строке, встроенные функции для обработки строк. Индексация, срезы, методы и функции строк, списочные выражения, обращение к элементу списка.

8БАС классы

Модуль 1 «Коллекции»

Повторение. Теория множеств, операции над множествами, операции над двумя множествами, сравнение множеств, сокращенные операторы, тип данных «множество», свойства множеств, методы и функции работы с множествами, символьные (строковые) переменные, посимвольная обработка строк, подсчёт частоты появления символа в строке, встроенные функции для обработки строк. Индексация, срезы, методы и функции строк, списочные выражения, обращение к элементу списка.

Модуль 2 «Функции»

Функции как способ группировать команды и именовать участки кода, определение простейших функций, начальные знания о локальных переменных, аргументы функций, связь между математическими функциями и функциями в Пайтон, возвращаемые значения, возвращение значений встроенными функциями, аннотация типов, рекурсия, множественные точки возврата из функции, возврат из глубины функции, отладка, что можно возвращать из функции, возврат нескольких значений, функции строк.

Модуль 3 «Рекурсия»

Понятие рекурсии и как она работает. Основные принципы написания программ с рекурсией. Числовые задачи на рекурсию. Рекурсия по коллекциям

8М, 8ИТ классы

Модуль 1 «Базовые конструкции языка»

О языке программирования Python, блоки кода, отступы, комментарии, стандартные типы данных, идентификаторы, работа с регистром, присваивание значений переменным, множественное присваивание, проверка типа данных, преобразование различных типов данных, ввод данных, вывод данных. Арифметические операции в Python, операторы присваивания в Python, целочисленное деление Алгоритм ветвления (Управляющие конструкции), тип данных bool, операторы сравнения в Python, логические операторы в Python, , проверка выполнимости условия, оператор if и его синтаксис, оператор else и его синтаксис, оператор elif и его синтаксис, вложенные конструкции, трехместное выражение if/else и его синтаксис.

Модуль 2 «Сложные условия и циклы»

Логические операции (и, или, не), логические отношения, операции в скобках, логические переменные (true, false), сложные условия в условном операторе. Цикл for (конструкция и синтаксис), функция range, настройка функции print, счётчик. Цикл While, Составной оператор присваивания, подсчёт количества элементов, удовлетворяющих условию, сигнал остановки, поиск максимума и минимума, обработка потока данных.

Модуль 3 «Коллекции»

Теория множеств, операции над множествами, операции над двумя множествами, сравнение множеств, сокращенные операторы, тип данных «множество», свойства множеств, методы и функции работы с множествами, символьные (строковые) переменные, посимвольная обработка строк, подсчёт частоты появления символа в строке, встроенные функции для обработки строк. Индексация, срезы, методы и функции строк, списочные выражения, обращение к элементу списка.

8П и 8ПИ класс

Модуль 1 «Базовые конструкции языка»

Условный оператор, циклический алгоритм, повторение.

Модуль 2 «Обработка потока данных»

Вычисление количества, суммы, среднего арифметического, минимального и максимального значений элементов последовательности, удовлетворяющих заданному условию.

Модуль 3 «Строки»

Строки, Индексация, Срезы, Системы счисления с основаниями, являющимися степенью числа 2.

Модуль 4 «Структуры»

Элементы теории множеств, Множества в Python, Представление логических операций через диаграммы Эйлера-Венна

Модуль 5 «Списки»

Списки, Кортежи, Списочные выражения, Методы split() и join(), Другие методы списков и строк, Вложенные списки. Арифметические операции в различных системах счисления

Модуль 6 «Вложенные списки»

Вложенные списки. Двумерные вложенные списки (матрицы). Создание двумерного списка. Перебор элементов двумерного списка. Вывод списка на экран. Матрицы

9ПИ класс

Модуль 1 «Функции»

Функции как способ группировать команды и именовать участки кода, определение простейших функций, начальные знания о локальных переменных, аргументы функций, связь между математическими функциями и функциями в Пайтон, возвращаемые значения, возвращение значений встроенными функциями, аннотация типов, рекурсия, множественные точки возврата из функции, возврат из глубины функции, отладка, что можно возвращать из функции, возврат нескольких значений, функции строк.

Модуль 2 «Функции с параметрами»

Области видимости переменных, переопределение функции и декораторы, передача параметров, функции с переменным числом аргументов.

Модуль 3 «Коллекции»

Итерируемые объекты: итераторы и коллекции, встроенные итераторы. Комбинирование итераторов, совместные действия над коллекциями, функция zip, модуль itertools, свертка итератора, функция reduce, группировка элементов, функция groupby

Модуль 4 «Файлы»

Файловая система, файлы в Пайтон, хранение данных, работа с csv-документами, работа с файлами json.

Модуль 5 «ООП»

Введение в ООП, Наследование, Методы проектирования программ

9П класс

Модуль 1 «Функции»

Функции как способ группировать команды и именовать участки кода, определение простейших функций, начальные знания о локальных переменных, аргументы функций, связь между математическими функциями и функциями в Пайтон, возвращаемые значения, возвращение значений встроенными функциями, аннотация типов, рекурсия, множественные точки возврата из функции, возврат из глубины функции, отладка, что можно возвращать из функции, возврат нескольких значений, функции строк.

Модуль 2 «Функции с параметрами»

Области видимости переменных, переопределение функции и декораторы, передача параметров, функции с переменным числом аргументов.

Модуль 3 «Коллекции»

Итерируемые объекты: итераторы и коллекции, встроенные итераторы. Комбинирование итераторов, совместные действия над коллекциями, функция zip, модуль itertools, свертка итератора, функция reduce, группировка элементов, функция groupby

Модуль 4 «Файлы»

Файловая система, файлы в Пайтон, хранение данных, работа с csv-документами, работа с файлами json.

Модуль 5 «Способы обработки потока данных»

Обработка потока циклически полученных данных и данных, полученных списком. Решение типовых задач. Ручной перебор данных и поиск параметра.

9Т, 9ИТ классы

Модуль 1 «Способы обработки потока данных»

Обработка потока циклически полученных данных и данных, полученных списком и строкой. Решение типовых задач. Ручной перебор данных и поиск параметра.

2. Планируемые образовательные результаты освоения предмета Программирование на Пайтон

ЛИЧНОСТНЫЕ РЕЗУЛЬТАТЫ

В результате изучения программирования на уровне основного общего образования у обучающегося будут сформированы следующие личностные результаты в части:

1) патриотического воспитания:

Ценностное отношение к отечественному культурному, историческому и научному наследию, понимание значения информатики как науки в жизни современного общества, владение достоверной информацией о передовых мировых и отечественных достижениях в области информатики и информационных технологий, заинтересованность в научных знаниях о цифровой трансформации современного общества

2) духовно-нравственного воспитания:

Ориентация на моральные ценности и нормы в ситуациях нравственного выбора, готовность оценивать своё поведение и поступки, а также поведение и поступки других людей с позиции нравственных и правовых норм с учётом осознания последствий поступков, активное неприятие асоциальных поступков, в том числе в Интернете;

3) гражданского воспитания:

Представление о социальных нормах и правилах межличностных отношений в коллективе, в том числе в социальных сообществах, соблюдение правил безопасности, в том числе навыков безопасного поведения в Интернет-среде, готовность к разнообразной совместной деятельности при выполнении учебных, познавательных задач, создании учебных проектов, стремление к взаимопониманию и взаимопомощи в процессе этой учебной деятельности, готовность оценивать своё поведение и поступки своих товарищей с позиции нравственных и правовых норм с учётом осознания последствий поступков;

4) ценности научного познания:

Сформированность мировоззренческих представлений об информации, информационных процессах и информационных технологиях, соответствующих современному уровню развития науки и общественной практики и составляющих базовую основу для понимания сущности научной картины мира; интерес к обучению и познанию, любознательность, готовность и способность к самообразованию, осознанному выбору направленности и уровня обучения в дальнейшем; овладение основными навыками исследовательской деятельности, установка на осмысление опыта, наблюдений, поступков и стремление совершенствовать пути достижения индивидуального и коллективного благополучия; сформированность информационной культуры, в том числе навыков самостоятельной работы с учебными текстами, справочной литературой, разнообразными средствами информационных технологий, а также умения самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

5) формирования культуры здоровья:

Осознание ценности жизни, ответственное отношение к своему здоровью, установка на здоровый образ жизни, в том числе и за счёт освоения и соблюдения требований безопасной эксплуатации средств информационных и коммуникационных технологий;

6) трудового воспитания:

Интерес к практическому изучению профессий и труда в сферах профессиональной деятельности, связанных с информатикой, программированием и информационными технологиями, основанными на достижениях науки информатики и научно-технического прогресса; осознанный выбор и построение индивидуальной траектории образования и жизненных планов с учётом личных и общественных интересов и потребностей;

7) экологического воспитания:

Осознание глобального характера экологических проблем и путей их решения, в том числе с учётом возможностей информационных и коммуникационных технологий;

8) адаптации обучающегося к изменяющимся условиям социальной среды:

Освоение обучающимися социального опыта, основных социальных ролей, соответствующих деятельности возраста, норм и правил общественного поведения, форм социальной жизни в группах и сообществах, в том числе существующих в виртуальном пространстве.

МЕТАПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

Метапредметные результаты освоения образовательной программы по информатике отражают овладение универсальными учебными действиями – познавательными, коммуникативными, регулятивными.

Познавательные универсальные учебные действия

Базовые логические действия:

Умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логические рассуждения, проводить умозаключения (индуктивные, дедуктивные и по аналогии) и выводы; умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач; самостоятельно выбирать способ решения учебной задачи (сравнивать несколько вариантов решения, выбирать наиболее подходящий с учётом самостоятельно выделенных критериев).

Базовые исследовательские действия:

Формулировать вопросы, фиксирующие разрыв между реальным и желательным состоянием ситуации, объекта, и самостоятельно устанавливать искомое и данное; оценивать на применимость и достоверность информацию, полученную в ходе исследования; прогнозировать возможное дальнейшее развитие процессов, событий и их последствия в аналогичных или сходных ситуациях, а также выдвигать предположения об их развитии в новых условиях и контекстах.

Работа с информацией:

Выявлять дефицит информации, данных, необходимых для решения поставленной задачи; применять различные методы и инструменты при поиске и отборе информации из источников с учётом предложенной учебной задачи и заданных критериев; выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления; самостоятельно выбирать оптимальную форму представления информации и иллюстрировать решаемые задачи несложными схемами, диаграммами, иными графическими объектами и их комбинациями; оценивать достоверность информации по критериям, предложенным учителем или сформулированным самостоятельно; эффективно запоминать и систематизировать информацию.

Коммуникативные универсальные учебные действия

Общение:

Сопоставлять свои суждения с суждениями других участников диалога, обнаруживать различие и сходство позиций; публично представлять результаты выполненного опыта (эксперимента, исследования, проекта); самостоятельно выбирать формат выступления с учётом задач презентации и особенностей аудитории и в соответствии с ним составлять устные и письменные тексты с использованием иллюстративных материалов.

Совместная деятельность (сотрудничество):

Понимать и использовать преимущества командной и индивидуальной работы при решении конкретной проблемы, в том числе при создании информационного продукта; принимать цель совместной информационной деятельности по сбору, обработке, передаче, формализации информации, коллективно строить действия по её достижению: распределять роли, договариваться, обсуждать процесс и результат совместной работы; выполнять свою часть работы с информацией или информационным продуктом, достигая качественного результата по своему направлению и координируя свои действия с другими членами команды; оценивать качество своего вклада в общий информационный продукт по критериям, самостоятельно сформулированным участниками взаимодействия; сравнивать результаты с исходной задачей и вклад каждого члена команды в достижение результатов, разделять сферу ответственности и проявлять готовность к предоставлению отчёта перед группой.

Регулятивные универсальные учебные действия

Самоорганизация:

Выявлять в жизненных и учебных ситуациях проблемы, требующие решения; ориентироваться в различных подходах к принятию решений (индивидуальное принятие решений, принятие решений в группе); самостоятельно составлять алгоритм решения задачи (или его часть), выбирать способ решения учебной задачи с учётом имеющихся ресурсов и собственных возможностей, аргументировать выбор варианта решения задачи; составлять план действий (план реализации намеченного алгоритма решения), корректировать предложенный алгоритм с учётом получения новых знаний об изучаемом объекте; проводить выбор в условиях противоречивой информации и брать ответственность за решение.

Самоконтроль (рефлексия):

Владеть способами самоконтроля, самомотивации и рефлексии; давать оценку ситуации и предлагать план её изменения; учитывать контекст и предвидеть трудности, которые могут возникнуть при решении учебной задачи, адаптировать решение к меняющимся обстоятельствам; объяснять причины достижения (недостижения) результатов информационной деятельности, давать оценку приобретённому опыту, уметь находить позитивное в произошедшей ситуации; вносить коррективы в деятельность на основе новых обстоятельств, изменившихся ситуаций, установленных ошибок, возникших трудностей; оценивать соответствие результата цели и условиям.

Эмоциональный интеллект:

Ставить себя на место другого человека, понимать мотивы и намерения другого.

Принятие себя и других:

Осознавать невозможность контролировать всё вокруг даже в условиях открытого доступа к любым объёмам информации.

ПРЕДМЕТНЫЕ РЕЗУЛЬТАТЫ

К концу обучения в 5 классах у обучающегося будут сформированы следующие умения:

- понимать информационную и алгоритмическую культуры;
- оперировать понятиями: переменная, тип данных, операция присваивания, арифметические и логические операции, включая операции целочисленного деления и остатка от деления;
- использовать константы и переменные различных типов (числовых – целых и вещественных, логических, символьных), а также содержащие их выражения, использовать оператор присваивания;
- иметь представление о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- иметь представление об основных изучаемых понятиях: исполнитель, алгоритм - и их свойствах;
- применять умение составить и записать алгоритм для конкретного исполнителя;
- работать с одним из языков программирования и основными алгоритмическими структурами – линейной и условной
- применять знания об алгоритмических конструкциях, логических значениях и операциях;
- записывать логические выражения на изучаемом языке программирования;
- анализировать предложенные алгоритмы, в том числе определять, какие результаты возможны при заданном множестве исходных значений, определять возможные входные данные, приводящие к определённому результату;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки числовых данных с использованием ветвлений (нахождение минимума и максимума из двух, трёх и четырёх чисел);

- применять алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе;

К концу обучения в 6БАС классе у обучающегося будут сформированы следующие умения:

- применять знания об основных изучаемых понятиях: исполнитель, алгоритм - и их свойствах в решении задач разного уровня сложности;
- составлять и записывать алгоритм для конкретного исполнителя с применением базовых конструкций языка;
- работать с одним из языков программирования и основной алгоритмической структурой – циклической; работать со сложными условиями.
- применять навык алгоритмического мышления, необходимый для решения нестандартных задач;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки числовых данных с использованием циклов с переменной, циклов с условиями (алгоритмы нахождения наибольшего общего делителя двух натуральных чисел, проверки натурального числа на простоту, разложения натурального числа на простые сомножители, выделения цифр из натурального числа);
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки потока данных (вычисление количества, суммы, среднего арифметического, минимального и максимального значений элементов числовой последовательности, удовлетворяющих заданному условию);
- записывать логические выражения на изучаемом языке программирования;
- решать задачи углубленного уровня с применением полученных знаний.

К концу обучения в 6М, 6ИМ, 6ОМ классах у обучающегося будут сформированы следующие умения:

- применять знания об основных изучаемых понятиях: исполнитель, алгоритм - и их свойствах в решении задач разного уровня сложности;
- составлять и записывать алгоритм для конкретного исполнителя с применением базовых конструкций языка;
- работать с одним из языков программирования и основной алгоритмической структурой – циклической; работать со сложными условиями.
- применять навык алгоритмического мышления, необходимый для решения нестандартных задач;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки числовых данных с использованием циклов с переменной, циклов с условиями (алгоритмы нахождения наибольшего общего делителя двух натуральных чисел, проверки натурального числа на простоту, разложения натурального числа на простые сомножители, выделения цифр из натурального числа);
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки потока данных (вычисление количества, суммы, среднего арифметического, минимального и максимального значений элементов числовой последовательности, удовлетворяющих заданному условию);
- записывать логические выражения на изучаемом языке программирования;
- решать задачи углубленного уровня с применением полученных знаний;
- применять математический модуль программирования;

- работать с координатной плоскостью в программировании;
- строить графики функций при помощи графического и математического модулей.

К концу обучения в 7ПИ классе у обучающегося будут сформированы следующие умения:

- составлять и записывать алгоритм для конкретного исполнителя с применением освоенных базовых конструкций языка;
- работать с одним из языков программирования и применять полученные знания о булевых переменных, работать с вложенными циклами.
- применять навык алгоритмического мышления, необходимый для решения нестандартных задач;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки символьных данных (посимвольная обработка строк, подсчёт частоты появления символа в строке, использование встроенных функций для обработки строк);
- решать задачи углубленного уровня с применением полученных знаний;
- работать с типами данных строки, списки и множества.

К концу обучения в 7ИМ классе у обучающегося будут сформированы следующие умения:

- составлять и записывать алгоритм для конкретного исполнителя с применением освоенных базовых конструкций языка;
- работать с одним из языков программирования и применять полученные знания о булевых переменных, работать с вложенными циклами.
- применять навык алгоритмического мышления, необходимый для решения нестандартных задач;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки символьных данных (посимвольная обработка строк, подсчёт частоты появления символа в строке, использование встроенных функций для обработки строк);
- решать задачи углубленного уровня с применением полученных знаний;
- работать с типами данных строки, списки и множества.

К концу обучения в 8БАС классах у обучающегося будут сформированы следующие умения:

- понимать информационную и алгоритмическую культуры;
- оперировать понятиями: переменная, тип данных, операция присваивания, арифметические и логические операции, включая операции целочисленного деления и остатка от деления;
- использовать константы и переменные различных типов (числовых – целых и вещественных, логических, символьных), а также содержащие их выражения, использовать оператор присваивания;
- иметь представление о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- иметь представление об основных изучаемых понятиях: исполнитель, алгоритм - и их свойствах;
- применять умение составить и записать алгоритм для конкретного исполнителя;
- работать с одним из языков программирования и основными алгоритмическими структурами – линейной и условной

- применять знания об алгоритмических конструкциях, логических значениях и операциях;
- записывать логические выражения на изучаемом языке программирования;
- анализировать предложенные алгоритмы, в том числе определять, какие результаты возможны при заданном множестве исходных значений, определять возможные входные данные, приводящие к определённому результату;
- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки числовых данных с использованием ветвлений (нахождение минимума и максимума из двух, трёх и четырёх чисел);
- применять алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе;

К концу обучения в 8М, 8ИТ классах у обучающегося будут сформированы следующие умения:

- создавать и отлаживать программы, реализующие типовые алгоритмы обработки одномерных числовых массивов, на одном из языков программирования (Python): заполнение числового массива случайными числами, в соответствии с формулой или путём ввода чисел, линейный поиск заданного значения в массиве, подсчёт элементов массива, удовлетворяющих заданному условию, нахождение суммы, минимального и максимального значений элементов массива;
- работать с одним из языков программирования и применять полученные знания о множествах и строках, научиться применять функции и методы строк и списков.
- решать задачи углубленного уровня с применением полученных знаний.

К концу обучения в 8П, 8ПИ классах у обучающегося будут сформированы следующие умения:

- создавать и отлаживать программы, реализующие типовые алгоритмы обработки одномерных числовых массивов, на одном из языков программирования (Python): заполнение числового массива случайными числами, в соответствии с формулой или путём ввода чисел, линейный поиск заданного значения в массиве, подсчёт элементов массива, удовлетворяющих заданному условию, нахождение суммы, минимального и максимального значений элементов массива;
- работать с одним из языков программирования и применять полученные знания о множествах и строках, научиться применять функции и методы строк и списков.
- решать задачи углубленного уровня с применением полученных знаний.

К концу обучения в 9ПИ, 9П классах у обучающегося будут сформированы следующие умения:

- составлять и записывать алгоритм для конкретного исполнителя с применением освоенных конструкций языка (функции, библиотеки);
- работать с языком программирования Python и применять полученные знания об области видимых переменных, передаче параметров, функциях, функциональной обработке коллекций.
- работать с библиотеками графики, электронными таблицами, библиотеками графики + звука
- применению навыка алгоритмического мышления, необходимого для решения нестандартных задач;
- решать задачи углубленного уровня с применением полученных знаний.

К концу обучения в 9Т, 9ИТ классах у обучающегося будут сформированы следующие умения:

- создавать и отлаживать программы на современном языке программирования общего назначения (Python), реализующие алгоритмы обработки символьных данных (посимвольная обработка списков и строк, подсчёт частоты появления символа в строке и списке, использование встроенных функций для обработки строк и списков);

ПРОВЕРЯЕМЫЕ ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В 5 и 6 классе не проверяются предметные результаты освоения основной образовательной программы основного общего образования по информатике, так как этот курс введен в качестве пропедевтики и углубленного изучения информатики и ИКТ.

7 класс

Код проверяемого результата	Проверяемые предметные результаты освоения основной образовательной программы основного общего образования
1.9	Использовать современные сервисы интернет-коммуникаций
1.10	Соблюдать требования безопасной эксплуатации технических средств информационных и коммуникационных технологий, соблюдать сетевой этикет, базовые нормы информационной этики и права при работе с приложениями на любых устройствах и в Интернете, выбирать безопасные стратегии поведения в сети

8 класс

Код проверяемого результата	Проверяемые предметные результаты освоения основной образовательной программы основного общего образования
2.1	Раскрывать смысл понятий «исполнитель», «алгоритм», «программа», понимая разницу между употреблением этих терминов в быденной речи и в информатике
2.2	Описывать алгоритм решения задачи различными способами, в том числе в виде блок-схемы
2.3	Составлять, выполнять вручную и на компьютере несложные алгоритмы с использованием ветвлений и циклов для управления исполнителями
2.4	Использовать константы и переменные различных типов (числовых, логических, символьных), а также содержащие их выражения, использовать оператор присваивания
2.5	Использовать при разработке программ логические значения, операции и выражения с ними
2.6	Анализировать предложенные алгоритмы, в том числе определять, какие результаты возможны при заданном множестве исходных значений
2.7	Создавать и отлаживать программы на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык), реализующие несложные алгоритмы обработки числовых данных с использованием циклов и ветвлений, в том числе реализующие проверку делимости одного целого числа на другое,

	проверку натурального числа на простоту, выделения цифр из натурального числа
--	---

9 класс

Код проверяемого результата	Проверяемые предметные результаты освоения основной образовательной программы основного общего образования
3.1	Разбивать задачи на подзадачи, составлять, выполнять вручную и на компьютере несложные алгоритмы с использованием ветвлений, циклов и вспомогательных алгоритмов для управления исполнителями
3.2	Составлять и отлаживать программы, реализующие типовые алгоритмы обработки числовых последовательностей или одномерных числовых массивов (поиск максимумов, минимумов, суммы или количества элементов с заданными свойствами) на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык)

ПРОВЕРЯЕМЫЕ НА ОГЭ ПО ИНФОРМАТИКЕ ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ОСНОВНОГО ОБЩЕГО ОБРАЗОВАНИЯ

Код проверяемого требования	Проверяемые требования к предметным результатам базового уровня освоения основной образовательной программы основного общего образования на основе ФГОС
1	Знать (понимать)
1.1	Владение основными понятиями: информация, передача, хранение и обработка информации, алгоритм, модель, цифровой продукт и их использование для решения учебных и практических задач
2	Уметь
2.5	Умение составлять, выполнять вручную и на компьютере несложные алгоритмы для управления исполнителями (Черепашка, Чертёжник и другие); создавать и отлаживать программы на одном из языков программирования (Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык), реализующие несложные алгоритмы обработки числовых данных с использованием циклов и ветвлений; умение разбивать задачи на подзадачи, использовать константы, переменные и выражения различных типов (числовых, логических, символьных); анализировать предложенный алгоритм, определять, какие результаты возможны при заданном множестве исходных значений
2.6	Умение записать на изучаемом языке программирования алгоритмы проверки делимости одного целого числа на другое, проверки натурального числа на простоту, выделения цифр из натурального числа, поиск максимумов, минимумов, суммы числовой последовательности

3. Тематическое планирование

Предмет: Программирование на Пайтон

Класс: 5

№ п/ п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всег о	Контрольн ые работы	Практичес кие работы	
Модуль №1. Введение в программирование – 17 часов					

1	Базовые конструкции языка	8	0	2	https://informatics.msk.ru/course/view.php?id=156 Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
2	Алгоритм ветвления	8	0	2	
3	Модульная работа «Введение в программирование»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 6БАС

№ п/ п	Наименован ие разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всег о	Контрольн ые работы	Практическ ие работы	
Модуль №1. Введение в программирование – 16 часов					
1	Сложные условия	5	0	2	https://informatics.msk.ru/course/view.php?id=156 Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
2	Циклы	10	0	7	
3	Модульная работа «Циклы и сложные условия»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 6ИМ, 6М

№ п/ п	Наименован ие разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всег о	Контрольн ые работы	Практическ ие работы	
Модуль №1. Введение в программирование – 17 часов					
1	Сложные условия	5	0	2	https://informatics.msk.ru/course/view.php?id=156 Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
2	Циклы	11	0	7	
3	Модульная работа «Циклы и сложные условия»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 6ОМ

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Математический модуль, сложные условия – 10 часов					

1.1	Математический модуль	2	0	1	https://stepik.org/course/85388 https://pythontutor.ru https://informatics.msk.ru/course/view.php?id=156
1.2	Сложные условия	7	0	2	https://stepik.org/course/85388 https://pythontutor.ru https://informatics.msk.ru/course/view.php?id=156
1.3	Модульная работа «Математический модуль, сложные условия»	1	1	0	https://stepik.org/course/85388 https://pythontutor.ru https://informatics.msk.ru/course/view.php?id=156
Модуль №2. Построение графиков функций – 7 часов					
2.1	Оси координат, точка на плоскости	2	0	1	https://informatics.msk.ru/course/view.php?id=156
2.2	Построение графиков функций	4	0	2	https://informatics.msk.ru/course/view.php?id=156
2.3	Модульная работа «Построение графиков функций»	1	1	0	https://informatics.msk.ru/course/view.php?id=156
Модуль №3. Циклы – 16 часов					
3.1	Цикл с условием	7	0	2	https://informatics.msk.ru/course/view.php?id=156 Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
3.2	Цикл со счетчиком	8	0	3	
3.3	Модульная работа «Циклы»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 7ПИ

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Строки и списки – 13 часов					
1.1	Строки	4	0	1	https://lyceum.yandex.ru/
1.2	Списки	8	0	2	
1.3	Модульная работа «Строки и списки»	1	1	0	

Модуль №2. Множества – 10 часов					
2.1	Операции над множествами	9	0	2	https://lyceum.yandex.ru/
2.2	Модульная работа «Строки и списки»	1	0	0	

Предмет: Программирование на Пайтон

Класс: 7ИМ

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Коллекции – 16 часов					
1.1	Строки	5	0	1	https://informatics.msk.ru/course/view.php?id=156
1.2	Списки	5	0	1	
1.3	Множества	5	0	1	
1.3	Модульная работа «Коллекции»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 8БАС

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Коллекции – 11 часов					
1.1	Строки	5	0	1	https://lyceum.yandex.ru/ https://informatics.msk.ru/course/view.php?id=156
1.3	Множества	5	0	1	
1.3	Модульная работа «Коллекции»	1	1	0	
Модуль №2. Функции – 11 часов					
2.1	Функции. Введение	5	0	0	https://lyceum.yandex.ru/ https://informatics.msk.ru/course/view.php?id=156
2.2	Возвращение значений	5	0	1	
2.3	Модульная работа «Функции»	1	0	0	
Модуль №3. Рекурсия – 11 часов					
3.1	Понятие рекурсии	2	0	0	https://lyceum.yandex.ru/

3.2	Числовые задачи на рекурсию	8	0	1	https://informatics.msk.ru/course/view.php?id=156
3.3	Модульная работа «Рекурсия»	1	0	0	

Предмет: Программирование на Пайтон

Класс: 8М, 8ИТ

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Базовые конструкции языка – 10 часов					
1.1	Повторение · Операторы ввода и вывода	3	0	1	https://informatics.msk.ru/course/view.php?id=156
1.3	Арифметические операции	3	0	1	
1.3	Математический модуль и модуль случайных чисел	3	0	1	
1.4	Модульная работа «Базовые конструкции и языка»	1	1	0	
Модуль №2. Сложные условия и циклы – 10 часов					
2.1	Условный оператор	2	0	1	https://informatics.msk.ru/course/view.php?id=156
2.2	Цикл с условием	3	0	1	
2.3	Цикл со счетчиком	3	0	1	
2.4	Модульная работа «Сложные условия и циклы»	1	1	0	
Модуль №3. Коллекции – 13 часов					
3.1	Строки	3	0	1	https://informatics.msk.ru/course/view.php?id=156
3.2	Списки	4	0	1	
3.3	Множества	5	0	1	
3.4	Модульная работа	1	1	0	

	«Коллекции »				
--	-----------------	--	--	--	--

Предмет: Программирование на Пайтон

Класс: 8ПИ

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Базовые конструкции языка – 10 часов					
1.1	Знакомство с LMS. Синтаксис языка	1			https://lyceum.yandex.ru/
1.2	Типы данных. Числа в Python, математическая логика	7		1	
1.3	Модульная работа № 1 «Базовые конструкции и языка»	2	1		
Модуль 2. Обработка потока данных – 10 часов					
2.1	Обработка потока данных	9		2	https://lyceum.yandex.ru/
2.2	Модульная работа № 2 «Базовые конструкции и языка»	1	1		
Модуль 3. Строки – 8 часов					
3.1	Символьный тип данных	4		1	https://lyceum.yandex.ru/
3.2	Функции строк	3		1	
3.3	Модульная работа № 3 «Обработка символьных данных»	1	1		
Модуль 4. Структуры – 7 часов					
4.1	Тип данных	1			https://lyceum.yandex.ru/

	«Множеств о»				
4.2	Операции над множества ми	5		1	
4.3	Модульная работа № 4 «Множеств а»	1	1		
Модуль 5. Списки – 10 часов					
5.1	Списки	8		1	https://lyceum.yandex.ru/
5.2	Вложенные списки	1			
5.3	Модульная работа № 5 «Списки»	1	1		
Модуль 6. Словари – 15 часов					
6.1	Словари	7		2	https://lyceum.yandex.ru/
6.2	Графическ ий модуль Tkinter	7		2	
6.3	Модульная работа № 6 «Словари»	1	1		

Предмет: Программирование на Пайтон

Класс: 9ИТ, 9Т

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Способы обработки потока данных					
1.1	Циклы	5	0	1	https://informatics.msk.ru/course/view.php?id=156
1.2	Строки	4	0	1	
1.3	Списки	5	0	1	
1.4	Модульная работа «Способы обработки потока данных»	1	1	0	

Предмет: Программирование на Пайтон

Класс: 9ПИ

№ п/ п	Наименова ние разделов и тем программ ы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Все го	Контрольн ые работы	Практичес кие работы	

Модуль 1. Функции – 12 часов					
1.1	Функции введение	10		2	https://lyceum.yandex.ru/
1.2	Функции с переменным числом аргументов	1			
1.3	Модульная работа № 1 «Функции»	1	1		
Модуль 2. Функции с параметрами – 10 часов					
2.1	Функции с параметрами	8		4	https://lyceum.yandex.ru/
2.2	Модульная работа «Функции с параметрами»	2	1		
Модуль 3. Коллекции – 8 часов					
3.1	Обработка коллекций	3			https://lyceum.yandex.ru/
3.2	Итераторы и коллекции	3		1	
3.3	Модульная работа № 3 «Коллекции»	2	1		
Модуль 4. Файлы – 10 часов					
4.1	Файловая система	8		1	https://lyceum.yandex.ru/
4.2	Модульная работа № 4 «Файлы»	2	1		
Модуль 5. Объектно-ориентированное программирование – 11 часов					
5.1	Введение в ООП	4		2	https://lyceum.yandex.ru/
5.2	Наследование	3		1	
5.3	Методы проектирования программ	2		1	

5.4	Модульная работа № 4 «ООП»	2	1		
-----	----------------------------	---	---	--	--

Предмет: Программирование на Пайтон

Класс: 9П

№ п/ п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Функции – 12 часов					
1.1	Функции введение	10		2	https://lyceum.yandex.ru/
1.2	Функции с переменным числом аргументов	1			
1.3	Модульная работа № 1 «Функции»	1	1		
Модуль 2. Функции с параметрами – 10 часов					
2.1	Функции с параметрами	8		4	https://lyceum.yandex.ru/
2.2	Модульная работа «Функции с параметрами»	2	1		
Модуль 3. Коллекции – 8 часов					
3.1	Обработка коллекций	3			https://lyceum.yandex.ru/
3.2	Итераторы и коллекции	3		1	
3.3	Модульная работа № 3 «Коллекции»	2	1		
Модуль 4. Файлы – 10 часов					
4.1	Файловая система	8		1	https://lyceum.yandex.ru/
4.2	Модульная работа № 4 «Файлы»	2	1		

Модуль 5. Способы обработки потока данных – 11 часов					
5.1	Повторение Строки	3		1	https://education.yandex.ru/uchebnik/main
5.2	Повторение списки	3		1	
5.3	Повторение обработки потока данных	4		1	
5.4	Модульная работа «Способы обработки потока данных»	1	1		

ПРИЛОЖЕНИЯ К ПРОГРАММЕ

Поурочное планирование 5 класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Введение в программирование – 17 часов					
1	Вводное занятие. Техника безопасности	1			Библиотека ЦОК https://academy- content.apkpro.ru/ru/lesson/3f141523- -406f-426b-80b8- 3a4c0b07c0da?backUrl=%2Fru%2Fc atalog%2F05%2F77%3Fclass%3D77 %26term%3D%25D1%2582%25D0 %25B5%25D1%2585%25D0%25B D%25D0%25B8%25D0%25BA%25 D0%25B0%2520%25D0%25B1%25 D0%25B5%25D0%25B7%25D0%2 5BE%25D0%25BF%25D0%25B0% 25D1%2581%25D0%25BD%25D0 %25BE%25D1%2581%25D1%2582 %25D0%25B8
2	Знакомство со средой, синтаксис и типы данных	1			Библиотека ЦОК https://academy- content.apkpro.ru/ru/lesson/c07f83e9 -5223-4e03-955a-7d2f6ee1f619

3	Оператор вывода	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
4	Оператор ввода	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
5	Практическая работа «Ввод и вывод данных»	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
6	Простые арифметические операции	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/c07f83e9-5223-4e03-955a-7d2f6ee1f619
7	Деление нацело и деление с остатком	1			
8	Практическая работа «калькулятор»	1		1	
9	Величины логического типа. Вычисление логических выражений.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
10	Условный оператор. Неполная форма ветвления.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
11	Условный оператор. Полная форма ветвления.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
12	Целочисленная арифметика и условный оператор.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
13	Использование сложных условий.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4

14	Практическая работа: «Решение задач с использованием условного оператора»	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
15	Оператор выбора.	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
16	Практическая работа: Использование оператора выбора для решения задач.	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
17	Модульная работа «Введение в программирование»	1	1		

6 БАС класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Циклы и сложные условия – 16 часов					
1	Базовые логические операции (and, or, not).	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
2	Таблицы истинности. Решение задач	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
3	Комбинирование условий в программах	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
4	Практикум по логическим	1		1	Библиотека ЦОК

	операциям и условиям				https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
5	Вложенные сложные условия				
6	Цикл for: основы работы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
7	Решение задач с циклом for	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
8	Обработка последовательностей данных	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
9	Цикл while: основы работы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
10	Решение задач с циклом while	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
11	Обработка последовательностей данных	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
12	Практическая работа на решение задач с использованием подходящего цикла	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
13	Поиск максимума и минимума в последовательности	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1

14	Вложенные циклы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
15	Практическая работа «Вложенные циклы»	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
16	Модульная работа «Циклы и сложные условия»	1	1		

6 ИМ, 6М класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Циклы и сложные условия – 17 часов					
1	Базовые логические операции (and, or, not).	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
2	Таблицы истинности. Решение задач	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
3	Комбинирование условий в программах	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
4	Практикум по логическим операциям и условиям	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
5	Вложенные сложные условия				
6	Цикл for: основы работы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1

7	Решение задач с циклом for	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
8	Обработка последовательностей данных	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
9	Цикл while: основы работы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
10	Решение задач с циклом while	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
11	Обработка последовательностей данных	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
12	Практическая работа на решение задач с использованием подходящего цикла	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
13	Поиск максимума и минимума в последовательности	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
14	Вложенные циклы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
15	Практическая работа «Вложенные циклы»	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
16	Комбинирование вложенных циклов и	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1

	сложных. условий				
17	Модульная работа «Циклы и сложные условия»	1	1		

60М класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Математический модуль, сложные условия – 10 часов					
1	Математический модуль	1			https://informatics.msk.ru/course/view.php?id=156
2	Практическая работа «написание математических выражений на языке программирования»	1		1	https://informatics.msk.ru/course/view.php?id=156
3	Базовые логические операции (and, or, not).	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
4	Таблицы истинности. Решение задач	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
5	Комбинирование условий в программах	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
6	Практикум по логическим операциям и условиям	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
7	Вложенные сложные условия	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4

8	Практическая работа «Квест»	2		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/8faeceb4-f75b-4df9-b60e-9a663512d6b4
9	Модульная работа «Математический модуль, сложные условия»	1	1		
Модуль №2. Построение графиков функций – 7 часов					
1	Оси координат, точка на плоскости	1			https://informatics.msk.ru/course/view.php?id=156
2	Практическая работа «Оси координат, точка на плоскости»	1		1	https://informatics.msk.ru/course/view.php?id=156
3	Построение графиков функций на языке программирования	1			https://informatics.msk.ru/course/view.php?id=156
4	Практическая работа «Построение графиков функций на языке программирования»	1		1	https://informatics.msk.ru/course/view.php?id=156
5	Проверка нахождения точки на плоскости	1			https://informatics.msk.ru/course/view.php?id=156
6	Практическая работа «Точка на плоскости»	1		1	https://informatics.msk.ru/course/view.php?id=156
7	Модульная работа «Построение	1	1		

	графиков функций»				
Модуль №3. Циклы – 16 часов					
1	Введение в циклы. Виды циклов в Python	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
2	Цикл for: синтаксис и базовое использование	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
3	Функция range() и её применение в цикле for	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
4	Итерация по строкам в цикле for	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
5	Вложенные циклы for	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
6	Практикум: задачи на цикл for	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
7	Цикл while: синтаксис и принцип работы	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
8	Управление циклом while: break, continue	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
9	Бесконечные циклы и способы их избежания	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1

10	Практикум: задачи на цикл while	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
11	Сравнение циклов for и while	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
12	Обработка данных в циклах (сумма, максимум, минимум)	1			Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
13	Решение задач на комбинацию условий и циклов	1		1	Библиотека ЦОК https://academy-content.apkpro.ru/ru/lesson/75d63645-6f43-45af-bbd1-7ed55cd4a8a1
14	Мини- проект: "Генератор паролей" или "Угадай число"	2		2	
15	Модульная работа «Циклы»	1	1		

7ПИ класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Строки и списки – 13 часов					
1	Основы работы со строками в Python, индексация и срезы	1			https://informatics.msk.ru/course/view.php?id=156
2	Основные методы строк	1			https://informatics.msk.ru/course/view.php?id=156

3	Форматирование строк	1			https://informatics.msk.ru/course/view.php?id=156
4	Решение задач на обработку строк	1		1	https://informatics.msk.ru/course/view.php?id=156
5	Введение в списки, индексация и срезы списков	1			https://informatics.msk.ru/course/view.php?id=156
6	Основные методы списков	1			https://informatics.msk.ru/course/view.php?id=156
7	Генерация списков	1			https://informatics.msk.ru/course/view.php?id=156
8	Вложенные списки и многомерные структуры	1			https://informatics.msk.ru/course/view.php?id=156
9	Сортировка и поиск в списках	1			https://informatics.msk.ru/course/view.php?id=156
10	Практическая работа нахождение элемента списка	1		1	https://informatics.msk.ru/course/view.php?id=156
11	Перевод строки в список и обратно	1			https://informatics.msk.ru/course/view.php?id=156
12	Решение задач на работу со списками	1		1	https://informatics.msk.ru/course/view.php?id=156
13	Модульная работа «Строки и списки»	1	1		
Модуль №2. Множества – 10 часов					
1	Основы работы с	1			https://informatics.msk.ru/course/view.php?id=156

	множествами (set)				
2	Операции с множествами	1			https://informatics.msk.ru/course/view.php?id=156
3	Методы множеств	1			https://informatics.msk.ru/course/view.php?id=156
4	Сравнение множеств	1			https://informatics.msk.ru/course/view.php?id=156
5	Генераторы множеств	1			https://informatics.msk.ru/course/view.php?id=156
6	Работа с frozenset	1			https://informatics.msk.ru/course/view.php?id=156
7	Множества и другие структуры данных	1			https://informatics.msk.ru/course/view.php?id=156
8	Практикум по операциям с множествами	1		1	https://informatics.msk.ru/course/view.php?id=156
9	Решение задач с использованием множеств	1		1	https://informatics.msk.ru/course/view.php?id=156
10	Модульная работа «Множества»	1	1		

7ИМ класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Коллекции – 16 часов					
1	Основы работы со строками	1			https://informatics.msk.ru/course/view.php?id=156
2	Индексация и срезы строк	1			https://informatics.msk.ru/course/view.php?id=156
3	Основные строковые методы	1			https://informatics.msk.ru/course/view.php?id=156

4	Форматирование строк (f-строки)	1			https://informatics.msk.ru/course/view.php?id=156
5	Решение задач на строки	1		1	https://informatics.msk.ru/course/view.php?id=156
6	Введение в списки	1			https://informatics.msk.ru/course/view.php?id=156
7	Операции со списками	1			https://informatics.msk.ru/course/view.php?id=156
8	Методы списков	1			https://informatics.msk.ru/course/view.php?id=156
9	Генераторы списков	1			https://informatics.msk.ru/course/view.php?id=156
10	Обработка списка	1		1	https://informatics.msk.ru/course/view.php?id=156
11	Основы множеств (set)	1			https://informatics.msk.ru/course/view.php?id=156
12	Операции с множествами	1			https://informatics.msk.ru/course/view.php?id=156
13	Методы множеств	1			https://informatics.msk.ru/course/view.php?id=156
14	Практикум по множествам	1		1	https://informatics.msk.ru/course/view.php?id=156
15	Решение задач на обработку коллекций	1			https://informatics.msk.ru/course/view.php?id=156
16	Модульная работа «Коллекции»	1	1		

8ИТ, 8М классы

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Базовые конструкции языка – 10 часов					
1	Повторение синтаксиса языка и	1			https://informatics.msk.ru/course/view.php?id=156

	работы с переменной				
2	Оператор ввода и оператор вывода	1			https://informatics.msk.ru/course/view.php?id=156
3	Типы данных	1			
4	Арифметические операции	1			https://informatics.msk.ru/course/view.php?id=156
5	Деление нацело и деление с остатком	1			https://informatics.msk.ru/course/view.php?id=156
6	Математический модуль	1			https://informatics.msk.ru/course/view.php?id=156
7	Модуль случайных чисел	1			https://informatics.msk.ru/course/view.php?id=156
8	Практическая работа «Калькулятор»	1		1	https://informatics.msk.ru/course/view.php?id=156
9	Решение задач	1			https://informatics.msk.ru/course/view.php?id=156
10	Модульная работа «Базовые конструкции языка»	1	1		
Модуль №2. Сложные условия и циклы – 10 часов					
1	Базовые логические операции (and, or, not), таблицы истинности	1			https://informatics.msk.ru/course/view.php?id=156
2	Практикум по логическим операциям и условиям	1		1	https://informatics.msk.ru/course/view.php?id=156

3	Цикл for: основы работы	1			https://informatics.msk.ru/course/view.php?id=156
4	Решение задач с циклом for	1			https://informatics.msk.ru/course/view.php?id=156
5	Обработка последователь ностей данных	1			https://informatics.msk.ru/course/view.php?id=156
6	Цикл while: основы работы	1			https://informatics.msk.ru/course/view.php?id=156
7	Решение задач с циклом while	1		1	https://informatics.msk.ru/course/view.php?id=156
8	Поиск максимума и минимума в последователь ности	1			https://informatics.msk.ru/course/view.php?id=156
9	Вложенные циклы	1			https://informatics.msk.ru/course/view.php?id=156
10	Модульная работа «Циклы и сложные условия»	1	1		
Модуль №3. Коллекции – 13 часов					
1	Основы работы со строками, Индексация и срезы строк	1			https://informatics.msk.ru/course/view.php?id=156
2	Основные строковые методы	1			https://informatics.msk.ru/course/view.php?id=156
3	Решение задач на строки	1			https://informatics.msk.ru/course/view.php?id=156
4	Введение в списки	1			https://informatics.msk.ru/course/view.php?id=156

5	Операции со списками	1			https://informatics.msk.ru/course/view.php?id=156
6	Методы списков и их генерация	1			https://informatics.msk.ru/course/view.php?id=156
7	Обработка списка	1		1	https://informatics.msk.ru/course/view.php?id=156
8	Основы множеств (set)	1			https://informatics.msk.ru/course/view.php?id=156
9	Операции с множествами	1			https://informatics.msk.ru/course/view.php?id=156
10	Методы множеств	1			https://informatics.msk.ru/course/view.php?id=156
11	Практикум по множествам	1		1	https://informatics.msk.ru/course/view.php?id=156
12	Решение задач на обработку коллекций	1			https://informatics.msk.ru/course/view.php?id=156
13	Модульная работа «Коллекции»	1			https://informatics.msk.ru/course/view.php?id=156

8ПИ класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольны е работы	Практические работы	
Модуль 1. Базовые конструкции языка – 10 часов					
1	Знакомство с LMS. Синтаксис языка	1			https://lyceum.yandex.ru/
2	Основные типы данных в Python	1			
3	Целые и вещественн ые числа	1			
4	Математич еские операции в Python	1			
5	Битовые операции и	1			

	системы счисления				
6	Логические операторы и выражения	1			
7	Сравнение чисел и логические выражения	1			
8	Практикум по числам и логике	1		1	
9	Модульная работа № 1 «Базовые конструкции и языка»	2	1		
Модуль 2. Обработка потока данных – 10 часов					
1	Введение в циклы: for и while	1			https://lyceum.yandex.ru/
2	Цикл for: синтаксис и базовые примеры	1			
3	Функция range() и итерация по последовательностям	1			
4	Вложенные циклы и их применение	1			
5	Цикл while: основы и условия выхода	1			
6	Управление циклами: break, continue, else	1			
7	Практикум : задачи на циклы (перебор, сумма, поиск)	1		1	

8	Генераторы списков (list comprehension)	1			
9	Решение задач с комбинацией условий и циклов	1		1	
10	Модульная работа № 2 «Обработка потока данных»	1	1		
Модуль 3. Строки – 8 часов					
1	Основы работы со строками. Создание и индексация	1			https://lyceum.yandex.ru/
2	Срезы строк и базовые операции	1			
3	Основные строковые методы (часть 1)	1			
4	Основные строковые методы (часть 2)	1			
5	Форматирование строк (f-строки, format())	1			
6	Решение задач на обработку строк	1		1	
7	Регулярные выражения (базовые понятия)	1			
8	Модульная работа № 3 «Обработка	1	1		

	а символьны х данных»				
Модуль 4. Множества – 7 часов					
1	Введение в множества (set). Создание и основные свойства	1			https://lyceum.yandex.ru/
2	Операции с множествами (объединение, пересечение, разность)	1			
3	Методы множеств (добавление, удаление, проверка элементов)	1			
4	Генераторы множеств (set comprehension)	1			
5	Неизменяемые множества (frozenset)	1			
6	Решение задач над множествами	1		1	
7	Модульная работа № 4 «Множества»	1	1		
Модуль 5. Списки – 10 часов					
1	Введение в списки (list). Создание и базовые операции	1			https://lyceum.yandex.ru/
2	Индексация и срезы списков	1			

3	Основные методы списков (добавление и удаление элементов)	1			
4	Методы списков (поиск и сортировка)	1			
5	Генераторы списков (list comprehension)	1			
6	Копирование списков. Поверхностное и глубокое копирование	1			
7	Вложенные списки и многомерные структуры	1			
8	Преобразование между списками и другими типами данных	1			
9	Практикум : решение задач со списками	1		1	
10	Модульная работа № 5 «Списки»	1	1		
Модуль 6. Словари – 15 часов					
1	Введение в словари (dict). Создание и базовые операции	1			
2	Основные методы	1			

	работы со словарями				
3	Перебор элементов словаря (keys, values, items)	1			
4	Генераторы словарей (dict comprehension)	1			
5	Вложенные словари и сложные структуры данных	1			
6	Практикум : решение задач со словарями (часть 1)	1		1	
7	Практикум : решение задач со словарями (часть 2)	1		1	
8	Введение в Tkinter. Основы GUI-программирования	1			
9	Основные виджеты Tkinter (Label, Button, Entry)	1			
10	Управление компоновкой (pack, grid, place)	1			
11	Работа с событиями и обработчиками в Tkinter	1			

12	Продвинутые виджеты Tkinter (Listbox, Combobox, Checkbutton)	1			
13	Разработка приложения-словаря на Tkinter (часть 1)	1		1	
14	Разработка приложения-словаря на Tkinter (часть 2)	1		1	
15	Модульная работа № 6 «Словари»	1	1		

8БАС

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Коллекции – 11 часов					
1	Основы работы со строками. Создание и индексация	1			https://informatics.msk.ru/course/view.php?id=156 https://lyceum.yandex.ru/
2	Срезы строк и базовые операции	1			
3	Основные строковые методы	1			
4	Форматирование строк (f-строки, format())	1			
5	Решение задач на обработку строк	1		1	
6	Введение в множества (set). Создание и основные свойства	5	0	1	

7	Операции с множествами и методы	1			
8	Генераторы множеств (set comprehension)	1			
9	Неизменяемые множества (frozenset)	1			
10	Практикум: решение задач с множествами	1		1	
11	Модульная работа «Коллекции»	1	1	0	
Модуль №2. Функции – 11 часов					
1	Введение в функции. Создание и вызов функций	1			https://informatics.msk.ru/course/view.php?id=156 https://lyceum.yandex.ru/
2	Параметры и аргументы функций	1			
3	Возврат значений из функций (return)	1			
4	Области видимости переменных (global, local)	1			
5	Функции с произвольным числом аргументов (*args, **kwargs)	1			
6	Рекурсивные функции	1			
7	Анонимные функции (lambda)	1			
8	Декораторы функций	1			
9	Документирование функций (docstrings)	1			

10	Практикум: решение задач с функциями	1		1	
11	Модульная работа «Функции»	1	1		
Модуль №3. Рекурсия – 11 часов					
1	Что такое рекурсия? Простые примеры	1			https://informatics.msk.ru/course/view.php?id=156 https://lyceum.yandex.ru/
2	Как работает рекурсия в Python	1			
3	Факториал и числа Фибоначчи	1			
4	Сумма чисел и списков рекурсией	1			
5	Глубина рекурсии и стек вызовов	1			
6	Когда использовать рекурсию?	1			
7	Рекурсия для работы со списками	1			
8	Простые рекурсивные алгоритмы	1			
9	Ошибки в рекурсии и как их избежать	1			
10	Практика: задачи на рекурсию	1		1	
11	Модульная работа «Рекурсия»	1	1		

9ИТ, 9Т классы

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль №1. Способы обработки потока данных					

1	Введение в циклы: for и while	1			https://informatics.msk.ru/course/view.php?id=156
2	Цикл for: синтаксис и базовые примеры	1			https://informatics.msk.ru/course/view.php?id=156
3	Вложенные циклы и их применение	1			https://informatics.msk.ru/course/view.php?id=156
4	Цикл while: основы и условия выхода	1			https://informatics.msk.ru/course/view.php?id=156
5	Практикум: задачи на циклы (перебор, сумма, поиск)	1		1	https://informatics.msk.ru/course/view.php?id=156
6	Основы работы со строками. Создание и индексация	1			https://informatics.msk.ru/course/view.php?id=156
7	Срезы строк и базовые операции	1			https://informatics.msk.ru/course/view.php?id=156
8	Основные строковые методы	1			https://informatics.msk.ru/course/view.php?id=156
9	Решение задач на обработку строк	1		1	https://informatics.msk.ru/course/view.php?id=156
10	Введение в списки (list). Создание и базовые операции	1			https://informatics.msk.ru/course/view.php?id=156
11	Основные методы списков (добавление и удаление элементов)	1			https://informatics.msk.ru/course/view.php?id=156
12	Методы списков (поиск и сортировка)	1			https://informatics.msk.ru/course/view.php?id=156

13	Генераторы списков (list comprehension)	1			https://informatics.msk.ru/course/view.php?id=156
14	Практикум: решение задач со списками	1		1	https://informatics.msk.ru/course/view.php?id=156
15	Решение задач на обработку потока данных	1			https://informatics.msk.ru/course/view.php?id=156
16	Модульная работа «Способы обработки потока данных»	1	1		

9ПИ класс

11 класс					
№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Функции – 11 часов					
1	Введение в функции. Создание и вызов функций				https://lyceum.yandex.ru/
2	Параметры и аргументы функций				
3	Возврат значений из функций (return)				
4	Области видимости переменных (global, local)				
5	Функции с произвольным числом аргументов (*args, **kwargs)				
6	Рекурсивные функции			1	

7	Анонимные функции (lambda)				
8	Декораторы функций				
9	Документирование функций (docstrings)				
10	Практикум: решение задач с функциями			1	
11	Модульная работа «Функции»				

Модуль 2. Функции с параметрами – 10 часов

1	Распаковка и упаковка значений	1			https://lyceum.yandex.ru/
2	Решение задач	1		1	
3	Аргументы по умолчанию	1			
4	Решение задач	1		1	
5	Именованные аргументы	1			
6	Решение задач	1		1	
7	Инструкция pass	1			
8	Решение задач	1		1	
9	Согласованность аргументов	1			
10	Модульная работа «Функции с параметром»	1	1		

Модуль 3. Коллекции – 8 часов

1	Итерируемые объекты	1			https://lyceum.yandex.ru/
2	Почему filter и map возвращают не список	1			
3	Функции max/min/sort ed и использован ие ключа сортировки	1			
4	Проверка коллекций: all, any Потоковый ввод stdin	1			
5	Модуль itertools	1			
6	Функция groupby	1			
7	Решение задач	1		1	
8	Модульная работа «Коллекции»	1	1		
Модуль 4. Файлы – 10 часов					
1	Повторение	1			https://lyceum.yandex.ru/
2	Файловая система в Пайтон	1			
3	Файловая система в Пайтон	1			
4	Файловая система в Пайтон	1			
5	Хранение данных	1			
6	Хранение данных	1			

7	Работа с csv-документами	1			
8	Работа с csv-документами	1			
9	Практическая работа	1		1	
10	Модульная работа «Файлы»	1	1		
Модуль 5. Объектно-ориентированное программирование – 11 часов					
1	Введение (Основные понятия Создание классов Методы классов)	1		1	https://lyceum.yandex.ru/
2	Инициализация экземпляров класса	1			
3	Соглашения об именовании, вызов методов и обращение к атрибутам	1			
4	Иерархия классов Механизм наследования Наследование на примере геометрических фигур.	1		1	
5	Расширение методов Использование методов наследников	1			

	в базовом классе				
6	Переопределение методов Множественное наследование	1			
7	Подготовительный этап Движение фигуры по пустой доске. Создание и использование своих модулей	1		1	
8	Ходы на доске с другими фигурами и взятие фигур.	1			
9	Варианты проектирования	1			
10	Реализация взятия фигур. Проверка того, что фигура не проходит сквозь другие	1		1	
11	Модульная работа «ООП»	1	1		

9П класс

№ п/п	Тема урока	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Функции – 11 часов					

1	Введение в функции. Создание и вызов функций				https://lyceum.yandex.ru/
2	Параметры и аргументы функций				
3	Возврат значений из функций (return)				
4	Области видимости переменных (global, local)				
5	Функции с произвольным числом аргументов (*args, **kwargs)				
6	Рекурсивные функции			1	
7	Анонимные функции (lambda)				
8	Декораторы функций				
9	Документирование функций (docstrings)				
10	Практикум: решение задач с функциями			1	
11	Модульная работа «Функции»				
Модуль 2. Функции с параметрами – 10 часов					
1	Распаковка и упаковка значений	1			https://lyceum.yandex.ru/
2	Решение задач	1		1	

3	Аргументы по умолчанию	1			
4	Решение задач	1		1	
5	Именованные аргументы	1			
6	Решение задач	1		1	
7	Инструкция pass	1			
8	Решение задач	1		1	
9	Согласованность аргументов	1			
10	Модульная работа «Функции с параметром»	1	1		
Модуль 3. Коллекции – 8 часов					
1	Итерируемые объекты	1			https://lyceum.yandex.ru/
2	Почему filter и map возвращают не список	1			
3	Функции max/min/sorted и использование ключа сортировки	1			
4	Проверка коллекций: all, any Потоковый ввод stdin	1			
5	Модуль itertools	1			
6	Функция groupby	1			

7	Решение задач	1		1	
8	Модульная работа «Коллекции»	1	1		
Модуль 4. Файлы – 10 часов					
1	Повторение	1			https://lyceum.yandex.ru/
2	Файловая система в Пайтон	1			
3	Файловая система в Пайтон	1			
4	Файловая система в Пайтон	1			
5	Хранение данных	1			
6	Хранение данных	1			
7	Работа с csv-документами	1			
8	Работа с csv-документами	1			
9	Практическая работа	1		1	
10	Модульная работа «Файлы»	1	1		
Модуль 5. Способы обработки потока данных – 11 часов					
1	Повторение «тип данных строка»	1			https://education.yandex.ru/uchebnik/main
2	Обработка строки	1			
3	Решение задач	1		1	
4	Повторение «тип данных список»	1			

5	Обработка списка	1			
6	Решение задач	1		1	
7	Обработка потока данных	1			
8	Обработка потока данных	1			
9	Поиск искомого значения	1			
10	Решение задач	1		1	
11	Модульная работа «Способы обработки потока данных»	1	1		

4. Приложения к программе

КИМы

5 классы

Модуль 1. Введение в программирование

Часть 1. Базовые конструкции языка

1. Оператор ввода _____

2. Выберите верные названия переменной

☐ a

☐ 1a

☐ 1_a

☐ a_1

3. Оператор вывода _____

4. Что _____
будет выведено на экран, если в списке вывода записано арифметическое выражение?

5. Выберите верный список арифметических операций в Python

a. +, -, *, /, =

b. +, -, *, /

c. +, -, *, /, ==

6. Запишите оператор присваивания _____

7. В какой строке программы ошибка

a = int(input())

b = int(input())

```
print a + b
```

8. Укажите целочисленный тип данных _____

9. Укажите тип данных, задающих дробное число _____

10. Задано условие для решения задачи: Даны стороны прямоугольника. Найти его периметр.

Из приведённых вариантов выберите вариант решения, который будет верно находить периметр прямоугольника.

```
a) a = int(input('Введите первую сторону прямоугольника'))
b = int(input('Введите вторую сторону прямоугольника'))
p = a + b * 2
print('Периметр равен', p)
```

```
б) a = int(input('Введите первую сторону прямоугольника'))
b = int(input('Введите вторую сторону прямоугольника'))
p = (a + b) * 2
print('Периметр равен', p)
```

```
в) a = int(input('Введите первую сторону прямоугольника'))
b = int(input('Введите вторую сторону прямоугольника'))
p = (a + b) * 4
print('Периметр равен', p)
```

11. Найдите и исправьте ошибку в программном алгоритме.

Условие задачи: Возраст Тани — X лет, а возраст Мити — Y лет. Найти их средний возраст.

```
x = int(input('Введите возраст Тани'))
y = int(input('Введите возраст Мити'))
sr = (x + y) / 2
print('Средний возраст равен', sr)
```

12. Выберите арифметический оператор, который находит остаток от деления одной величины от другой.

а) /

б) %

в) //

г) *

13. «Дано двузначное число, выведите количество десятков в нём».

Выберите правильный вариант решения данной задачи.

```
1) a = int(input()) 2) a = int(input()) 3) a = int(input()) 4) a = int(input())
a1 = a//10 a1 = a%10 a1 = a%100 a1 = a//100
```

14. Дано трехзначное число.

Запишите строку программы, которая позволит определить число сотен в нём.

Часть 2. Условный оператор

1. Укажите полная или неполная форма ветвления представлена в задании

```
if a > b:
```

```
    M = a
```

```
    N = b
```

Ответ: _____

2. Напишите, что будет выведено в результате выполнения программы

(в ответ укажите число, которое будет выведено в результате выполнения программы)

```
a = 5
```

```
b = 10
```

```
if a > b:
```

```
    print(a)
```

```
else:
```

print(b)

Ответ: _____

3. Из предложенных вариантов выберите, какой это знак отношения: \leq

- больше
- меньше
- больше или равно
- меньше или равно

4. Напишите, как называется логический тип данных и какие значения могут принимать переменные данного типа.

Ответ: _____

5. Вычислите значение логического выражения при следующих значениях логических величин А, В: А = Истина, В = Ложь:

А и В Ответ: _____

6. Вычислите значение логического выражения при следующих значениях логических величин А, В и С: А = Ложь, В = Ложь, С = Истина:

(не А или не В) и (А или В)

Ответ: _____

7. Напишите программу для следующего условия:

Дано натуральное число. Определить:

- а) является ли оно нечетным;
- б) оканчивается ли оно цифрой 7

8. Напишите программу для следующего условия:

Дано двузначное число. Определить: какая из его цифр больше: первая или вторая

6БАС, 6М, 6ИМ

Часть 1. Сложные условия

1. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
b = 5
if not (a < 5):
    a = 5
```

2. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
b = 5
if a > 5 and a < b:
    a -= 5
```

3. Какую логическую операцию нужно добавить в программу вместо многоточия, чтобы значение переменной «а» после выполнения фрагмента программы стало равно 3?

```
a = 10
b = 5
if a < 1 ... a > b:
    a -= 7
else:
    a += 7
```

4. Определите значение переменной «а» после выполнения фрагмента программы:

```
a = 10
b = 5
if a > 1 and a < b:
    a -= 5
if a > 1 and a == b:
    a -= 5
```

5. Какое число нужно написать вместо многоточия, чтобы цикл выполнился ровно 2 раза?

```
i = 3
while i >= ...:
    print ( "Привет!" )
    i -= 1
```

6. Какое число нужно написать вместо многоточия, чтобы цикл выполнился ровно 3 раза?

```
i = ...
while i <= 7:
    print ( "Привет!" )
    i += 1
```

7. Чему будет равно значение переменной «a» после выполнения этого цикла?

```
i = 2
a = 12
while i < 5:
    a += i
    i += 1
```

8. Чему будет равно значение целой переменной «a» после выполнения этого фрагмента программы?

```
a = 2
for i in range(3):
    a += i
```

9. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
print ( 4, end="" )
for i in range(1,2,-1):
    print ( 2*i, end="" )
```

10. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
print ( 1, end="" )
for i in range(3,6):
    print ( i, end="" )
```

Часть 2. Циклы

Задача 1: Вывод чисел от 1 до 10

Напишите программу, которая выводит все числа от 1 до 10 с помощью цикла for.

Пример вывода:

1 2 3 4 5 6 7 8 9 10

Задача 2: Сумма чисел

Напишите программу, которая вычисляет сумму всех чисел от 1 до 100 с помощью цикла for.

Пример вывода:

Сумма чисел от 1 до 100: 5050

Задача 3: Таблица умножения

Напишите программу, которая выводит таблицу умножения на 5 (от 1 до 10) с помощью цикла for.

Пример вывода:

```
5 × 1 = 5
5 × 2 = 10
...
5 × 10 = 50
```

Задача 4: Обратный отсчёт

Напишите программу, которая выводит числа от 10 до 1 в обратном порядке с помощью цикла while.

Пример вывода:

10 9 8 7 6 5 4 3 2 1

Задача 5: Ввод чисел до нуля

Напишите программу, которая запрашивает у пользователя числа, пока он не введёт 0, а затем выводит сумму всех введённых чисел.

Пример работы:

Введите число: 5

Введите число: 3

Введите число: 0

Сумма: 8

Задача 6: Поиск делителя

Напишите программу, которая находит первый делитель числа 91, отличный от 1 и самого числа, с помощью цикла while.

Пример вывода:

Первый делитель числа 91 (кроме 1 и 91): 7

60М

Модуль 1 «Математический модуль, сложные условия»

1. Записать по правилам изучаемого языка программирования следующие выражения:

а) $\sqrt{x_1^2 + x_2^2}$;

ж) $2\pi R$;

н) $\frac{ad + bc}{ad}$;

б) $x_1x_2 + x_1x_3 + x_2x_3$;

з) $b^2 - 4ac$;

о) $\sqrt{1 - \sin^2 x}$;

в) $v_0t + \frac{at^2}{2}$;

и) $\gamma \frac{m_1m_2}{r^2}$;

п) $\frac{1}{\sqrt{ax^2 + bx + c}}$;

г) $\frac{mv^2}{2} + mgh$;

к) I^2R ;

р) $\frac{\sqrt{x+1} + \sqrt{x-1}}{2\sqrt{x}}$;

д) $\frac{1}{R_1} + \frac{1}{R_2}$;

л) $ab \sin c$;

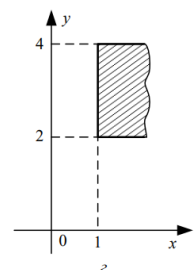
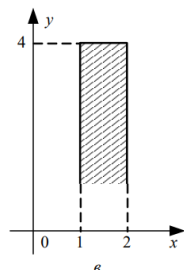
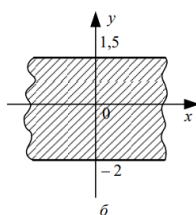
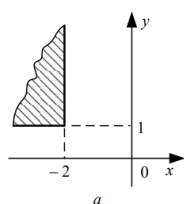
с) $|x| + |x+1|$;

е) $mg \cos \alpha$;

м) $\sqrt{a^2 + b^2 - 2ab \cos c}$;

т) $|1 - |x||$.

2. Записать условие, которое является истинным, когда точка с координатами (x, y) попадает в заштрихованные участки плоскости, включая их границы



Модуль 2 «Построение графиков функций»

Построить график линейной функции и квадратичной функции средствами программирования с применением графического и математического модулей.

Модуль 3 «Циклы»

4. Какое число нужно написать вместо многоточия, чтобы цикл выполнился ровно 2 раза?

```
i = 3
while i >= ...:
    print ( "Привет!" )
    i -= 1
```

5. Какое число нужно написать вместо многоточия, чтобы цикл выполнился ровно 3 раза?

```
i = ...
while i <= 7:
    print ( "Привет!" )
    i += 1
```

6. Чему будет равно значение переменной «a» после выполнения этого цикла?

```
i = 2
a = 12
while i < 5:
    a += i
    i += 1
```

7. Чему будет равно значение целой переменной «a» после выполнения этого фрагмента программы?

```
a = 2
for i in range(3):
    a += i
```

8. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
print ( 4, end="" )
for i in range(1,2,-1):
    print ( 2*i, end="" )
```

9. Какое число будет выведено на экран после выполнения этого фрагмента программы?

```
print ( 1, end="" )
for i in range(3,6):
    print ( i, end="" )
```

7ПН

Модуль 1 «Строки и списки»

1. Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод count.

2. Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.

При решении этой задачи не стоит пользоваться циклами и инструкцией if.

3. Дана строка. Замените в этой строке все цифры 1 на слово one.

4. Дана строка. Удалите из этой строки все символы @.

5. Выведите все четные элементы списка. При этом используйте цикл for, перебирающий элементы списка, а не их индексы!

6. Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку у них недостаточно соседей.

7. Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.

8. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

9. Переставьте соседние элементы списка (A[0] с A[1], A[2] с A[3] и т. д.). Если элементов нечетное число, то последний элемент остается на своем месте.

Модуль 2 «Множества»

1. Дан список чисел. Определите, сколько в нем встречается различных чисел.
2. Даны два списка чисел. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
3. Даны два списка чисел. Найдите все числа, которые входят как в первый, так и во второй список и выведите их в порядке возрастания.
4. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
5. Дан текст: в первой строке записано число строк, далее идут сами строки. Определите, сколько различных слов содержится в этом тексте.
Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.
6. Каждый из некоторого множества школьников некоторой школы знает некоторое количество языков. Нужно определить сколько языков знают все школьники, и сколько языков знает хотя бы один из школьников.
В первой строке задано количество школьников. Для каждого из школьников сперва записано количество языков, которое он знает, а затем - названия языков, по одному в строке.
В первой строке выведите количество языков, которые знают все школьники. Начиная со второй строки - список таких языков. Затем - количество языков, которые знает хотя бы один школьник, на следующих строках - список таких языков. Языки нужно выводить в лексикографическом порядке, по одному на строке.

7ИМ

Модуль 1 «Коллекции»

1. Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод count.
2. Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.
При решении этой задачи не стоит пользоваться циклами и инструкцией if.
3. Дана строка. Замените в этой строке все цифры 1 на слово one.
4. Дана строка. Удалите из этой строки все символы @.
5. Выведите все четные элементы списка. При этом используйте цикл for, перебирающий элементы списка, а не их индексы!
6. Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку у них недостаточно соседей.
7. Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.
8. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.
9. Переставьте соседние элементы списка ($A[0]$ с $A[1]$, $A[2]$ с $A[3]$ и т. д.). Если элементов нечетное число, то последний элемент остается на своем месте.
10. Дан список чисел. Определите, сколько в нем встречается различных чисел.
11. Даны два списка чисел. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
12. Даны два списка чисел. Найдите все числа, которые входят как в первый, так и во второй список и выведите их в порядке возрастания.
13. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.

14. Дан текст: в первой строке записано число строк, далее идут сами строки. Определите, сколько различных слов содержится в этом тексте.

Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.

15. Каждый из некоторого множества школьников некоторой школы знает некоторое количество языков. Нужно определить сколько языков знают все школьники, и сколько языков знает хотя бы один из школьников.

В первой строке задано количество школьников. Для каждого из школьников сперва записано количество языков, которое он знает, а затем - названия языков, по одному в строке.

В первой строке выведите количество языков, которые знают все школьники. Начиная со второй строки - список таких языков. Затем - количество языков, которые знает хотя бы один школьник, на следующих строках - список таких языков. Языки нужно выводить в лексикографическом порядке, по одному на строке.

8БАС

Модуль 1 «Коллекции»

Задачи на строки

Задача 1: Подсчёт гласных букв

Напишите программу, которая принимает строку и выводит количество гласных букв (а, е, ё, и, о, у, ы, э, ю, я) в ней.

Пример:

Введите строку: Привет, мир!

Количество гласных: 3

Задача 2: Замена символов

Напишите программу, которая заменяет все буквы "а" на "о" в заданной строке и выводит результат.

Пример:

Введите строку: банан

Результат: бонон

Задача 3: Переворот строки

Напишите программу, которая принимает строку и выводит её в обратном порядке.

Пример:

Введите строку: Python

Результат: nohtyP

Задачи на множества

Задача 4: Общие элементы

Даны два множества:

$A = \{1, 2, 3, 4, 5\}$

$B = \{4, 5, 6, 7, 8\}$

Напишите программу, которая выводит общие элементы этих множеств.

Пример вывода:

Общие элементы: {4, 5}

Задача 5: Уникальные символы в строке

Напишите программу, которая принимает строку и выводит множество уникальных символов в ней (без повторов).

Пример:

Введите строку: программирование

Уникальные символы: {'п', 'р', 'о', 'г', 'а', 'м', 'и', 'н', 'е', 'в'}

Задача 6: Объединение и разность множеств

Даны два множества:

$X = \{10, 20, 30, 40\}$

$Y = \{30, 40, 50, 60\}$

Напишите программу, которая выводит:

Объединение множеств

Разность $X - Y$

Пример вывода:

Объединение: {10, 20, 30, 40, 50, 60}

Разность $X - Y$: {10, 20}

Модуль 2 «Функции»

1. Простая функция с приветствием

Напишите функцию `greet(name)`, которая принимает имя и выводит приветствие в формате: "Привет, [имя]! Как дела?"

Пример вызова:

```
greet("Анна")
```

Вывод:

Привет, Анна! Как дела?

2. Функция вычисления суммы трёх чисел

Напишите функцию `sum_three(a, b, c)`, которая принимает три числа и возвращает их сумму.

Пример вызова:

```
result = sum_three(5, 10, 15)
```

```
print(result)
```

Вывод:

30

3. Функция проверки чётности числа

Напишите функцию `is_even(num)`, которая принимает число и возвращает `True`, если оно чётное, и `False`, если нечётное.

Пример вызова:

```
print(is_even(4)) # True
```

```
print(is_even(7)) # False
```

4. Функция нахождения максимального числа

Напишите функцию `max_number(a, b)`, которая принимает два числа и возвращает наибольшее из них.

Пример вызова:

```
print(max_number(8, 12)) # 12
```

5. Функция подсчёта длины строки

Напишите функцию `str_length(text)`, которая принимает строку и возвращает её длину (без использования `len()`).

Пример вызова:

```
print(str_length("Python")) # 6
```

6. Функция проверки палиндрома

Напишите функцию `is_palindrome(word)`, которая проверяет, является ли слово палиндромом (читается одинаково слева направо и справа налево, например, "топот"). Функция должна возвращать `True` или `False`.

Пример вызова:

```
print(is_palindrome("топот")) # True
print(is_palindrome("привет")) # False
```

Модуль 3 «Рекурсия»

1. Рекурсивный вывод чисел от 1 до N

Напишите рекурсивную функцию `print_numbers(n)`, которая выводит числа от 1 до n.

Пример:

```
print_numbers(5)
```

Вывод:

```
1
2
3
4
5
```

Подсказка:

Функция должна сначала вызвать себя для $n-1$, а потом вывести n .

2. Рекурсивная сумма чисел от 1 до N

Напишите рекурсивную функцию `sum_numbers(n)`, которая возвращает сумму чисел от 1 до n.

Пример:

```
print(sum_numbers(3)) # 1 + 2 + 3 = 6
```

Подсказка:

Базовый случай: если $n == 1$, вернуть 1.

Иначе: вернуть $n + \text{sum_numbers}(n - 1)$.

3. Рекурсивный факториал

Напишите рекурсивную функцию `factorial(n)`, которая вычисляет факториал числа n (произведение всех чисел от 1 до n).

Пример:

```
print(factorial(5)) # 1 * 2 * 3 * 4 * 5 = 120
```

Подсказка:

Базовый случай: `factorial(1) = 1`.

Рекурсивный случай: `factorial(n) = n * factorial(n - 1)`.

4. Рекурсивная проверка чётности числа

Напишите рекурсивную функцию `is_even(n)`, которая возвращает `True`, если число чётное, и `False`, если нечётное.

Пример:

```
print(is_even(4)) # True
print(is_even(7)) # False
```

Подсказка:

Если $n == 0$, вернуть `True`.

Если $n == 1$, вернуть `False`.

Иначе: вернуть `is_even(n - 2)`.

5. Рекурсивный вывод цифр числа

Напишите рекурсивную функцию `print_digits(n)`, которая выводит цифры числа по одной (сначала последнюю, потом предпоследнюю и т. д.).

Пример:

```
print_digits(123)
```

Вывод:

3
2
1

Подсказка:

Сначала вывести $n \% 10$ (последняя цифра).

Затем вызвать `print_digits(n // 10)` (число без последней цифры).

Базовый случай: если $n < 10$, просто вывести n .

6. Рекурсивный подсчёт количества цифр в числе

Напишите рекурсивную функцию `count_digits(n)`, которая возвращает количество цифр в числе.

Пример:

```
print(count_digits(1234)) # 4
```

Подсказка:

Базовый случай: если $n < 10$, вернуть 1.

Иначе: вернуть $1 + \text{count_digits}(n // 10)$.

8М, 8ИТ классы

Модуль 1. Базовые конструкции языка

1. Оператор ввода _____

2. Выберите верные названия переменной

☐ a

☐ 1a

☐ 1_a

☐ a_1

3. Оператор вывода _____

4. Что _____

будет выведено на экран, если в списке вывода записано арифметическое выражение?

5. Выберите верный список арифметических операций в Python

a. +, -, *, /, =

b. +, -, *, /

c. +, -, *, /, ==

6. Запишите оператор присваивания _____

7. В какой строке программы ошибка

```
a = int(input())
```

```
b = int(input())
```

```
print a + b
```

8. Укажите целочисленный тип данных _____

9. Укажите тип данных, задающих дробное число _____

10. Задано условие для решения задачи: Даны стороны прямоугольника. Найти его периметр.

Из приведённых вариантов выберите вариант решения, который будет верно находить периметр прямоугольника.

a) `a = int(input('Введите первую сторону прямоугольника'))`

`b = int(input('Введите вторую сторону прямоугольника'))`

`p = a + b * 2`

`print('Периметр равен', p)`

б) `a = int(input('Введите первую сторону прямоугольника'))`

`b = int(input('Введите вторую сторону прямоугольника'))`

`p = (a + b) * 2`

```
print('Периметр равен', p)
```

```
в) a = int(input('Введите первую сторону прямоугольника'))
```

```
b = int(input('Введите вторую сторону прямоугольника'))
```

```
p = (a + b) * 4
```

```
print('Периметр равен', p)
```

11. Найдите и исправьте ошибку в программном алгоритме.

Условие задачи: Возраст Тани — X лет, а возраст Мити — Y лет. Найти их

средний возраст.

```
x = int(input('Введите возраст Тани'))
```

```
y = int(input('Введите возраст Мити'))
```

```
sr = (x + y) / 2
```

```
print(Средний возраст равен, sr)
```

12. Выберите арифметический оператор, который находит остаток от деления одной величины от другой.

а) /

б) %

в) //

г) *

13. «Дано двузначное число, выведите количество десятков в нём».

Выберите правильный вариант решения данной задачи.

1) a = int(input()) 2) a = int(input()) 3) a = int(input()) 4) a = int(input())

a1 = a//10 a1 = a%10 a1 = a%100 a1 = a//100

Модуль 2. Сложные условия и циклы

1. Чётные и нечётные числа в диапазоне

Напишите программу, которая выводит все чётные числа от 1 до N и отдельно все нечётные числа от 1 до N.

Ввод:

N = 10

Вывод:

Чётные: 2 4 6 8 10

Нечётные: 1 3 5 7 9

2. Поиск делителей числа

Напишите программу, которая находит все делители числа N (кроме 1 и самого числа).

Ввод:

N = 12

Вывод:

Делители: 2, 3, 4, 6

3. Проверка на простое число

Напишите программу, которая проверяет, является ли число N простым (делится только на 1 и на себя).

Ввод:

N = 7

Вывод:

7 — простое число

4. Подсчёт суммы цифр числа

Напишите программу, которая считает сумму цифр числа N.

Ввод:

N = 1234

Вывод:

Сумма цифр: 10

5. Таблица умножения с условием

Напишите программу, которая выводит таблицу умножения для числа K, но пропускает умножение на 5.

Ввод:

K = 3

Вывод:

$3 \times 1 = 3$

$3 \times 2 = 6$

$3 \times 3 = 9$

$3 \times 4 = 12$

$3 \times 6 = 18$

...

$3 \times 10 = 30$

6. Поиск минимальной цифры в числе

Напишите программу, которая находит наименьшую цифру в числе N.

Ввод:

N = 5724

Вывод:

Минимальная цифра: 2

7. Проверка числа на "палиндромность"

Напишите программу, которая проверяет, является ли число N палиндромом (читается одинаково слева направо и справа налево, например, 121).

Ввод:

N = 12321

Вывод:

12321 — палиндром

8. Вывод чисел Фибоначчи до N

Напишите программу, которая выводит все числа Фибоначчи (каждое следующее — сумма двух предыдущих), не превышающие N.

Ввод:

N = 20

Вывод:

1, 1, 2, 3, 5, 8, 13

Модуль 3 «Коллекции»

1. Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод count.

2. Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.

При решении этой задачи не стоит пользоваться циклами и инструкцией if.

3. Дана строка. Замените в этой строке все цифры 1 на слово one.

4. Дана строка. Удалите из этой строки все символы @.

5. Выведите все четные элементы списка. При этом используйте цикл for, перебирающий элементы списка, а не их индексы!

6. Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку у них недостаточно соседей.
7. Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.
8. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.
9. Переставьте соседние элементы списка ($A[0]$ с $A[1]$, $A[2]$ с $A[3]$ и т. д.). Если элементов нечетное число, то последний элемент остается на своем месте.
10. Дан список чисел. Определите, сколько в нем встречается различных чисел.
11. Даны два списка чисел. Посчитайте, сколько чисел содержится одновременно как в первом списке, так и во втором.
12. Даны два списка чисел. Найдите все числа, которые входят как в первый, так и во второй список и выведите их в порядке возрастания.
13. Во входной строке записана последовательность чисел через пробел. Для каждого числа выведите слово YES (в отдельной строке), если это число ранее встречалось в последовательности или NO, если не встречалось.
14. Дан текст: в первой строке записано число строк, далее идут сами строки. Определите, сколько различных слов содержится в этом тексте.
Словом считается последовательность непробельных символов идущих подряд, слова разделены одним или большим числом пробелов или символами конца строки.
15. Каждый из некоторого множества школьников некоторой школы знает некоторое количество языков. Нужно определить сколько языков знают все школьники, и сколько языков знает хотя бы один из школьников.
В первой строке задано количество школьников. Для каждого из школьников сперва записано количество языков, которое он знает, а затем - названия языков, по одному в строке.
В первой строке выведите количество языков, которые знают все школьники. Начиная со второй строки - список таких языков. Затем - количество языков, которые знает хотя бы один школьник, на следующих строках - список таких языков. Языки нужно выводить в лексикографическом порядке, по одному на строке.

8ПМ класс

Модуль 1 «Базовые конструкции языка»

1. К удалому флейтисту из Конго

Раз в сапог заползла анаконда.

Но настолько отвратно

Он играл, что обратно

Через час уползла анаконда.

Ну, вы, конечно, подумали, что это Anaconda! Или Python! А вот и нет. Простая змея. Большая. И не одна, а втроем. И пока ползли, договаривались, как будут делить добычу. Самая длинная змея сказала, что очень голодна, не ела уже полгода, поэтому ей нужно в 4 раза больше, чем самой толстой змее. Самая толстая сказала, что не ела два месяца, поэтому ей нужно в 2 раза больше, чем оставшемуся Питончику. А что ему оставалось делать? Маленького питона обидеть может каждый! Он согласился.

Напишите программу, которая выводит в строку через пробел, сколько добычи (что бы это ни значило) достанется каждой змее, начиная с самой голодной. Что-то может и остаться, но как можно меньше...

Формат ввода

Целое число – количество добычи.

Формат вывода

Четыре числа: сколько добычи достанется Самой длинной змее, Самой толстой змее, Питончику и сколько останется. Числа записаны в одну строку через пробел.

Пример 1

Ввод	Вывод
60	40 10 5 5

Пример 2

Ввод	Вывод
33	24 6 3 0

2. Похвастаться-то они похвастались своим обжорством, да вот смогли ли съесть, сколько захапали? Вам известен аппетит каждой змеи, прошедшее время и количество реально съеденной добычи. Определите, кто из змей мог все это съесть? (Ели по одной, сильно жадные.)

Формат ввода

Аппетит (количество добычи, съедаемой **в минуту**) Самой длинной змеи (действительное число).
Аппетит Самой толстой змеи (действительное число).
Аппетит Питончика (действительное число).
Количество съеденной добычи (целое число).
Время **в часах** (целое число).

Формат вывода

В одной строке через пробел вывести всех змей, которые ЭТО могли съесть. Если никто не успеет за такое время, вывести: **Не они**.

Пример 1

Ввод	Вывод
15.5	Самая длинная змея Самая толстая змея Питончик
18.3	
12.2	
600	
1	

Пример 2

Ввод	Вывод
5.5	Самая толстая змея
8.3	
2.2	
990	
2	

Пример 3

Ввод	Вывод
3.128	Не они
1.535	
2.114	
940	

Примечания

Если подозреваемых несколько, выводить в порядке: **Самая длинная змея, Самая толстая змея, Питончик.**

3. Теперь про флейтиста. Он совсем не умеет играть на флейте, или ему слон на ухо наступил. В данном случае, это неплохо, иначе стать бы ему ДОБЫЧЕЙ! Вам нужно проверить, может ли введенная строка быть СЫГРАНА фальшивящим флейтистом. Определять будем так:

1. строка должна быть не длиннее 7 букв;
2. там не должно быть слова **нота**;
3. если там есть сочетание **до** или **ля**, то количество букв должно быть четным, если **соль** – то нечетным (если есть оба сочетания – смотреть по первому);
4. в словаре строка не должна идти после слова **флейта**.

Если все условия выполнены, нужно вывести **МОЖЕТ**, иначе **НЕ МОЖЕТ**.

Формат ввода

Слово в нижнем регистре.

Формат вывода

МОЖЕТ или **НЕ МОЖЕТ**.

Пример 1

Ввод Вывод
худо НЕ МОЖЕТ

Пример 2

Ввод Вывод
гнездо МОЖЕТ

Модуль 2 «Обработка потока данных»

1. Напишите программу, которая в последовательности натуральных чисел определяет количество трехзначных чисел, кратных 4. Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность).

Программа должна вывести одно число: количество трехзначных чисел, кратных 4.

Пример работы программы:

Входные данные	Выходные данные
120 9 365 4 0	1

2. Напишите программу, которая в последовательности натуральных чисел определяет сумму всех чисел, кратных 6 и оканчивающихся на 6. Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 – признак окончания ввода, не входит в последовательность).
 Программа должна вывести одно число: сумму всех чисел, кратных 6 и оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
36 12 16 66 11 0	102

3. Напишите программу для решения следующей задачи. Ученики 4 класса вели дневники наблюдения за погодой и ежедневно записывали дневную температуру. Найдите среднюю температуру за время наблюдения. Если количество дней, когда температура поднималась выше нуля градусов, не менее 5, выведите YES, иначе выведите NO. Программа получает на вход количество дней, в течение которых проводилось наблюдение N ($1 \leq N \leq 31$), затем для каждого дня вводится температура.

Пример работы программы:

Входные данные	Выходные данные
4 -4 12 -2 8	3.5 NO

Модуль 3 «Строки»

1. Вводятся слова. Выведите только те из них, в которых нет повторяющихся букв. Ввод заканчивается пустой строкой.

Пример

Ввод	Вывод
whirling	round
twirling	and
round	round
and	
round	

2. Словом в данной задаче считается последовательность букв, ограниченная пробелами или началом или концом строки.
 Выведите все слова из строки в столбик.

Пример 1

Ввод	Вывод
Hello, world!	Hello, world!

Пример 2

Ввод	Вывод
My heart in the Highland	My heart in the Highland

3. Зашифруйте введенную строку таким образом:

разбить строку на две части пополам;

записать новую строку так, чтобы символы из первой части чередовались символами из второй части.

Если длина строки нечетная, то в первой части на 1 символ больше.

Пример 1

Ввод	Вывод
python	phyotn

Пример 2

Ввод	Вывод
world	wlodr

Модуль 4 «Структуры»

1. Ученые часто славятся своей рассеянностью. И немудрено – уже столько всего открыто, разве упомнишь! И вот, когда обнаруживают новый вид каких-нибудь гусениц, нужно придумать им название, но не повториться.

Напишите программу, которая проверяет, не было ли название использовано раньше.

Формат ввода

Вводится количество существующих названий, затем сами названия гусениц.

Формат вывода

Для каждого предложенного названия вывести, использовалось ли оно раньше, то есть **ДА** или **НЕТ**.

Пример

Ввод	Вывод
6	НЕТ
Lonomia_Obliqua	НЕТ
Limacodidae_Euclidae	НЕТ
Acharia_Stimulea	НЕТ
Coscinocera_Hercules	ДА
Limacodidae_Euclidae	ДА

Ввод

Вывод

Limacodidae_Euclidae

2. Для красивого обозначения номеров домов используются трафареты. Определите, трафареты скольких цифр вам нужно купить, чтобы можно было написать все номера.

Формат ввода

Вводится количество чисел, затем сами числа.

Формат вывода

Выведите количество различных использованных цифр.

Пример

Ввод	Вывод
3	4
100	
500	
20	

3. Для получения полноценного иммунитета нужно сделать две дозы прививки. Есть две базы данных прививок, сделанных с разницей в 21 день. Узнайте, кто сделал обе дозы.

Формат ввода

Два раза вводятся число – количество данных – и сами данные в виде многозначных чисел.

Формат вывода

Выведите без повторов числа, которые есть в обеих базах данных. Порядок вывода не важен, каждое значение с новой строки.

Пример

Ввод	Вывод
3	123
123	301
231	
301	
4	
301	
200	
222	
123	

4. Есть три множества многоножек: многоножки, у которых больше 40 ножек; многоножки с четным числом ножек и многоножки с числом ножек, кратным трем.

Определите сколько из введенных многоножек принадлежат ровно двум множествам.

Формат ввода

Вводится число – количества многоножек, затем сами ножки.

Формат вывода

Выведите числа (ножки у многоножек), которые находятся ровно в двух множествах. Выводить в одну строку через пробел в любом порядке.

Пример

Ввод	Вывод
7	12 86
86	
20	
71	
66	
12	
13	
72	

Модуль 5 «Списки»

1. Нужно найти место слову в упорядоченном по алфавиту списке.

Формат ввода

Вводится количество слов, затем слова. Затем вводится слово для вставки.

Формат вывода

Определите, под каким индексом нужно вставить в список слово, чтобы алфавитный порядок не нарушался. Если такое слово уже есть в строке, то вставлять нужно после него.

Пример 1

Ввод	Вывод
5	4
На	
златом	
крыльце	
сидели	
царь	
царевич	

Пример 2

Ввод	Вывод
4	4
Как-то	
крошка-крокодил	
от	
родителей	
уплыл	

2. Нужно раздать подарки всем детям, но строго по одному, а то кому-то не достанется.

Формат ввода

Вводится количество детей, затем их имена. Потом вводится количество запросов, затем имена детей, подходящих за подарками.

Формат вывода

Если этот ребенок еще подарок не получал, то вывести:

Вот твой подарок, <имя>!

Если он уже брал подарок, вывести:

<Имя>, всем по одному подарку!

Пример

Ввод	Вывод
4	Вот твой подарок, Петечка!
Ванечка	Вот твой подарок, Машенька!
Машенька	Петечка, всем по одному подарку!
Петечка	
Толечка	
3	
Петечка	
Машенька	
Петечка	

3. Для украшения улиц купили флаги разных цветов. Нужно составить гирлянду из некоторого количества флагов, в которой цвета идут подряд, а затем повторяются в том же порядке, пока не наберется нужное количество флагов.

Формат ввода

Вводится количество различных цветов флагов, затем сами цвета. Затем вводится количество флагов в гирлянде.

Формат вывода

Вывести цвета флагов, как они должны идти в гирлянде.

Пример

Ввод	Вывод
3	red
red	green
green	blue
blue	red
7	green
	blue
	red

Модуль 6 «Словари»

1. В растровом изображении каждый пиксель имеет какой-либо цвет. В цветовой модели RGB цвета задаются кортежем из интенсивностей трех компонент: красной, зеленой и синей.

Напишите программу, которая подсчитает количество различных цветов в изображении и количество пикселей каждого цвета.

Формат ввода

Вводится количество строк в развертке изображения. Затем вводятся строки одинаковой длины, в которых пиксели разделены символом табуляции, а компоненты цвета в пикселе – пробелом.

Формат вывода

Выведите кортеж компонентов цвета, который встречается чаще всего. Если таких несколько, выведите их все в любом порядке, каждый кортеж с новой строки.

Пример

Ввод	Вывод
5	0 255 0
255 0 255 0 255 0 0 255 255 255 0 255	0 255 255
255 0 255 0 255 0 0 255 0 255 255 0	255 0 255
255 0 0 0 255 255 0 255 255 255 0 255	
0 0 255 0 0 0 255 255 255 0 255 255	
255 0 0 0 0 255 255 255 0 0 255 0	

2. Рыба ищет, где глубже, а человек, где лучше.

Но задача именно про рыбу.

Формат ввода

Вводятся названия рыб и глубины, на которых их обнаружили, пока не будет введена пустая строка.

Формат вывода

Составьте и выведите словарь, в котором по ключу – названию рыбы – находится значение – кортеж из минимальной и максимальной глубин, на которых ее встретили. Если ее встречали один раз, то оба значения одинаковы.

Пример

Ввод	Вывод
carp 8	{'carp': (8, 8), 'asp': (3, 14), 'flatfish': (25, 150), 'hake': (15, 28)}
asp 3	
flatfish 150	
hake 15	
flatfish 98	
flatfish 25	
asp 14	
hake 28	

3. Для приготовления одной порции самых популярных молочных смузи нужно:

для смузи "Сладкоежка": банан, молоко, мороженое;

для "Клубнично-банановый": клубника, молоко, банан, мороженое;

для "Клубничный": клубника, молоко, мороженое;

для "Хушаф": финики, молоко;

для "Банановый": банан, молоко, финики.

Определите, какие из заказов покупателей могут быть выполнены.

Формат ввода

Сначала вводится 5 строк – количество имеющихся в наличии продуктов (через двоеточие и пробел): *молоко, мороженое, клубника, банан, финики*.

Затем вводится число – количество запросов, затем сами запросы – названия заказываемых коктейлей.

Формат вывода

Если заказ может быть выполнен, печатается строка:

Пожалуйста, ваш <название>. Приятного аппетита!

Если для выполнения заказа не хватает продуктов, выводится строка:

Извините, не можем выполнить заказ.

Считается, что порции продуктов в каждом заказе одинаковы и равны условной единице.

Пример

Ввод

молоко: 5
мороженое: 3
клубника: 1
банан: 0
финики: 1
5
Хушаф
Хушаф
Сладкоежка
Клубничный
Клубнично-банановый

Вывод

Пожалуйста, ваш Хушаф. Приятного аппетита!
Извините, не можем выполнить заказ.
Извините, не можем выполнить заказ.
Пожалуйста, ваш Клубничный. Приятного аппетита!
Извините, не можем выполнить заказ.

9ИТ, 9Т классы

Модуль 1 «Способы обработки потока данных»

Часть 1 «Циклы»

1. Напишите программу для решения следующей задачи. Камера наблюдения регистрирует в автоматическом режиме скорость проезжающих мимо нее автомобилей, округляя значения скорости до целых чисел. Необходимо определить максимальную зарегистрированную скорость автомобиля. Если скорость хотя бы одного автомобиля была меньше 30 км/ч, выведите «YES», иначе выведите «NO».

Программа получает на вход число проехавших автомобилей N ($1 < N < 30$), затем указываются их скорости. Значение скорости не может быть меньше 1 и больше 300. Программа должна сначала вывести максимальную скорость, затем YES или NO.

Пример работы программы:

Входные данные	Выходные данные
----------------	-----------------

4	74 NO
74	
69	
63	
66	

2. Напишите программу, которая в последовательности натуральных чисел определяет наименьшее число, кратное 3. Программа получает на вход целые числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не является членом последовательности). Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Гарантируется, что в последовательности есть хотя бы одно число, кратное 3.

Программа должна вывести одно число: наименьшее число, кратное 3.

Пример работы программы:

Входные данные	Выходные данные
45 10 9 0	9

3. Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое двузначных чисел или сообщает, что таких чисел нет (выводит «NO»).

Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность). Количество чисел не превышает 100. Введенные числа не превышают 300. Программа должна вывести среднее арифметическое двузначных чисел или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых

Пример работы программы:

Входные данные	Выходные данные
10 120 49 0	29.5
111 1 0	NO

4. Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое чисел, кратных 8, или сообщает, что таких чисел нет (выводит «NO»). Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введенные числа не превышают 300. Программа должна

вывести среднее арифметическое чисел, кратных 8, или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых.

Пример работы программы:

Входные данные	Выходные данные
8 122 64 16 0	29.3
111 1 0	NO

5. Напишите программу для решения следующей задачи. Камера наблюдения регистрирует в автоматическом режиме скорость проезжающих мимо нее автомобилей, округляя значения скорости до целых чисел. Необходимо определить среднюю зарегистрированную скорость всех автомобилей. Если не менее двух автомобилей двигались со скоростью не больше 40 км/ч, выведите «YES», иначе выведите «NO».

Программа получает на вход число проехавших автомобилей N ($1 \leq N \leq 30$), затем указываются их скорости. Значение скорости не может быть меньше 1 и больше 300. Программа должна сначала вывести среднюю скорость, с точностью до одного знака после запятой, затем «YES» или «NO».

Пример работы программы:

Входные данные	Выходные данные
4 74 69 63 96	75.5 NO

6. Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число,

оканчивающееся на 4. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число — сумму чисел, оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3	38
14	
25	
24	

Часть 2 «Строки»

1. Подсчет гласных букв

Напишите программу, которая подсчитывает количество гласных букв (а, е, ё, и, о, у, ы, э, ю, я) в строке. Учитывайте как строчные, так и заглавные буквы.

Ввод:

```
s = "Программирование - это интересно!"
```

Вывод:

```
11
```

2. Замена символов

Напишите программу, которая заменяет все вхождения буквы "а" на "о" в строке. Учитывайте как строчные, так и заглавные буквы.

Ввод:

```
s = "банан"
```

Вывод:

```
бонон
```

3. Переворот строки

Напишите программу, которая переворачивает строку.

Ввод:

```
s = "Python"
```

Вывод:

```
nohtyP
```

4. Подсчет слов в строке

Напишите программу, которая подсчитывает количество слов в строке (слова разделены пробелами).

Ввод:

```
s = "Hello world and good morning"
```

Вывод:

```
5
```

5. Удаление цифр из строки

Напишите программу, которая удаляет все цифры из строки.

Ввод:

```
python
```

```
s = "a1b2c3d4e5"
```

Вывод:

```
abcde
```

6. Проверка на палиндром

Напишите программу, которая проверяет, является ли строка палиндромом (читается одинаково слева направо и справа налево, игнорируя пробелы и регистр).

Ввод:

```
python
```

```
s = "А роза упала на лапу Азора"
```

Вывод:

```
Да
```

7. Поиск самого длинного слова

Напишите программу, которая находит самое длинное слово в строке.

Ввод:

```
python
```

```
s = "Python является одним из самых популярных языков программирования"
```

Вывод:

```
программирования
```

8. Шифр Цезаря (сдвиг букв)

Напишите программу, которая шифрует строку с помощью шифра Цезаря (сдвиг на 3 буквы вперед по алфавиту). Сдвигайте только буквы, остальные символы оставляйте без изменений.

Ввод:

```
s = "abcxyz"
```

Вывод:

```
defabc
```

Часть 3 «Списки»

1. Сумма элементов списка

Напишите программу, которая вычисляет сумму всех элементов в списке целых чисел.

Пример ввода:

```
numbers = [1, 2, 3, 4, 5]
```

Пример вывода:

```
15
```

2. Поиск максимального элемента

Напишите программу, которая находит максимальный элемент в списке чисел.

Пример ввода:

```
numbers = [12, 45, 23, 67, 34]
```

Пример вывода:

```
67
```

3. Подсчет четных чисел

Напишите программу, которая подсчитывает количество четных чисел в списке.

Пример ввода:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8]
```

Пример вывода:

```
4
```

4. Обратный порядок списка

Напишите программу, которая выводит элементы списка в обратном порядке.

Пример ввода:

```
items = ['яблоко', 'банан', 'апельсин']
```

Пример вывода:

```
['апельсин', 'банан', 'яблоко']
```

5. Удаление дубликатов

Напишите программу, которая удаляет повторяющиеся элементы из списка, сохраняя исходный порядок.

Пример ввода:

```
items = [1, 2, 2, 3, 4, 4, 5]
```

Пример вывода:

```
[1, 2, 3, 4, 5]
```

6. Объединение списков

Напишите программу, которая объединяет два списка в один, чередуя их элементы.

Пример ввода:

```
list1 = [1, 3, 5]
```

```
list2 = [2, 4, 6]
```

Пример вывода:

```
[1, 2, 3, 4, 5, 6]
```

7. Поиск индекса элемента

Напишите программу, которая находит индекс первого вхождения заданного элемента в списке.

Если элемента нет, выведите -1.

Пример ввода:

```
items = ['яблоко', 'банан', 'апельсин']
```

```
target = 'банан'
```

Пример вывода:

```
1
```

8. Сортировка списка строк

Напишите программу, которая сортирует список строк по длине (от самой короткой до самой длинной).

Пример ввода:

```
words = ['Python', 'список', 'строка', 'алгоритм']
```

Пример вывода:

```
['список', 'строка', 'Python', 'алгоритм']
```

9П, 9ПИ классы

Модуль 1 «Функции»

1. Торговец поставил сундук у ног калифа и открыл крышку. Великий аллах! Чего только тут не было! И ожерелья из жемчуга, и самоцветные перстни, и оружие в серебряной оправе, и золотые кубки, и роговые гребни.

Напишите программу, которая среди всех товаров выберет самые диковинные.

Формат ввода

Вводятся строки из положительных целых чисел, записанных через точку с запятой и пробел.

Формат вывода

Нужно найти строку с наибольшим количеством двузначных чисел. Если таких несколько, выбрать последнюю встреченную. Вывести из этой строки все двузначные числа в исходном порядке через пробел.

Пример 1

Ввод

```
12; 5; 43; 9; 29; 9; 3; 6
24; 44; 25; 49; 49; 20; 38; 2; 9; 3
30; 8; 9; 5; 45; 30; 2; 49; 35; 46; 25; 8
```

Вывод

```
30 45 30 49 35 46 25
```

Пример 2

Ввод

```
24; 45; 38; 49; 28; 14; 39; 10; 0; 29
34; 42; 32; 27; 44; 358
309; 28; 41; 43; 19; 29; 49
12; 19; 33; 39; 47; 36; 12; 6; 27; 0; 8; 191
```

Вывод

```
24 45 38 49 28 14 39 10 29
```

2. – Не знаешь ли ты, визирь, кто бы мог прочесть эти письма?

– Всемиловейший господин и повелитель, – ответил визирь, – у Большой мечети живёт человек, по имени Селим, по прозвищу «Учёный». Он может прочесть всякую книгу. Прикажи позвать его, может быть, он проникнет в тайну этих загадочных знаков.

Напишите функцию **scientist()**, которая будет расшифровывать письма очень странным образом.

Функция принимает строку в качестве позиционного аргумента и именованный параметр *to_reverse* со значением по умолчанию **False**.

А в глобальной переменной **writing** находится список слов (все буквы в нижнем регистре) для перевода. Функция должна сравнить каждое слово в списке со строкой-аргументом, и, если слово идет позже по алфавиту, чем аргумент, то взять из него каждый второй символ (начиная с нулевого), иначе каждый третий. Если значение *to_reverse* **True**, то полученную строку перевернуть в обратном порядке. Строку после всех преобразований поместить на прежнее место в список.

Пример 1

Ввод

```
writing = ['yqrweetrstyym', 'fuiopo', 'ezxscvebbhastd',
'slukojihrgeftdsgydma', 'spnogiussy']
scientist('penetrate', to_reverse=True)
print(writing)
```

Вывод

```
['mystery', 'of', 'these',
'mysterious', 'signs']
```

Пример 2

Ввод

```
writing = ['lqweeratyrgndseasdx', 'mzxafan', 'sweetyluiiopm']
scientist('wizard')
print(writing)
```

Вывод

```
['learned', 'man', 'selim']
```


3. Арауко — столица Араукании, государства, имеющего в длину сто пятьдесят лье, а в ширину тридцать. Населяют Арауканию молуче, эти первородные сыны чилийской расы – гордое и сильное племя, единственное из американских племен, которое никогда не подпадало под иноземное владычество. Если город Арауко был когда-то подчинен испанцам, то население Араукании оставалось всегда независимым.

Напишите функцию **independence()**, которая определит подчиненность и независимость.

Функция принимает произвольное количество позиционных аргументов-кортежей натуральных чисел и произвольное количество именованных аргументов, среди которых могут быть такие:

- `first_larger` – первое число в кортеже больше указанного значения;
- `first_not_ends` – первое число не оканчивается на указанную цифру;
- `second_not_mult` – второе число в кортеже не кратно указанному числу;
- `second_digits` – второе число должно иметь указанное количество разрядов.

Функция возвращает наибольшее среди подходящих первых значений кортежей и наименьшее среди вторых подходящих.

Пример 1

Ввод	Вывод
<pre>data = [(12, 36), (20, 7), (4, 61), (13, 11), (54, 19), (22, 14)] conditions = {} print(*independence(*data, **conditions))</pre>	54 7

Пример 2

Ввод	Вывод
<pre>data = [(12, 36), (20, 7), (4, 61), (13, 11), (54, 19), (22, 14)] conditions = {'first_larger': 11, 'first_not_ends': 4, 'second_not_mult': 3, 'second_digits': 2} print(*independence(*data, **conditions))</pre>	22 11

Модуль 2 «Функции с параметрами»

1. Боба очень любит попугаев, но у него аллергия на птиц. Поэтому он захотел создать себе электронного попугая.

Напишите функцию **parrot(phrase)**, которая будет симулировать речь попугая. Если в переменной `phrase` находится предложение, которое попугай слышал хотя бы один раз, то функция печатает эту фразу.

Формат ввода

Строка

Пример 1

Ввод	Вывод
<pre>parrot("Привет!")</pre>	Привет!
<pre>parrot("Привет!")</pre>	
<pre>parrot("Как дела?")</pre>	

Пример 2

Ввод	Вывод
<code>parrot("Привет")</code>	Привет
<code>parrot("Как тебя зовут?")</code>	
<code>parrot("Привет")</code>	

2. Напишите пример, который продемонстрирует отличия между методом `sort()` списка и функцией `sorted(arr)`.

На основе вашего кода должно быть возможно показать, какой из вариантов меняет список, а какой – создает новый, отсортированный список.

Примечания

В качестве решения приложите файл с кодом, в котором обязательно должны присутствовать комментарии, поясняющие работу программы.

Данная задача проверяется преподавателем.

3. Приведите пример, когда

`value = value + addition`

и

`value += addition`

дают одинаковый результат и когда разный.

В чем будут отличия, и как их пронаблюдать?
Напишите код, который их продемонстрирует.

Примечания

В качестве решения приложите файл с кодом, в котором обязательно должны присутствовать комментарии, поясняющие работу программы.

Данная задача проверяется преподавателем.

Модуль 3 «Коллекции»

1. Нужно найти место слову в упорядоченном по алфавиту списке.

Формат ввода

Вводится количество слов, затем слова. Затем вводится слово для вставки.

Формат вывода

Определите, под каким индексом нужно вставить в список слово, чтобы алфавитный порядок не нарушался. Если такое слово уже есть в строке, то вставлять нужно после него.

Пример 1

Ввод	Вывод
5	4
На	

Ввод	Вывод
златом	
крыльце	
сидели	
царь	
царевич	

Пример 2

Ввод	Вывод
4	4
Как-то	
крошка-крокодил	
от	
родителей	
уплыл	

2. Нужно раздать подарки всем детям, но строго по одному, а то кому-то не достанется.

Формат ввода

Вводится количество детей, затем их имена. Потом вводится количество запросов, затем имена детей, подходящих за подарками.

Формат вывода

Если этот ребенок еще подарок не получал, то вывести:

Вот твой подарок, <имя>!

Если он уже брал подарок, вывести:

<Имя>, всем по одному подарку!

Пример

Ввод	Вывод
4	Вот твой подарок, Петечка!
Ванечка	Вот твой подарок, Машенька!
Машенька	Петечка, всем по одному подарку!
Петечка	
Толечка	
3	
Петечка	
Машенька	
Петечка	

3. В растровом изображении каждый пиксель имеет какой-либо цвет. В цветовой модели RGB цвета задаются кортежем из интенсивностей трех компонент: красной, зеленой и синей.

Напишите программу, которая подсчитает количество различных цветов в изображении и количество пикселей каждого цвета.

Формат ввода

Вводится количество строк в развертке изображения. Затем вводятся строки одинаковой длины, в которых пиксели разделены символом табуляции, а компоненты цвета в пикселе – пробелом.

Формат вывода

Выведите кортеж компонентов цвета, который встречается чаще всего. Если таких несколько, выведите их все в любом порядке, каждый кортеж с новой строки.

Пример

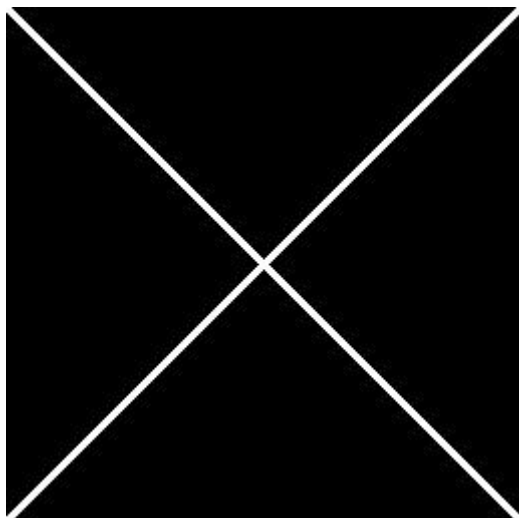
Ввод	Вывод
5	0 255 0
255 0 255 0 255 0 0 255 255 255 0 255	0 255 255
255 0 255 0 255 0 0 255 0 255 255 0	255 0 255
255 0 0 0 255 255 0 255 255 255 0 255	
0 0 255 0 0 0 255 255 255 0 255 255	
255 0 0 0 0 255 255 255 0 0 255 0	

Модуль 4 «Файлы»

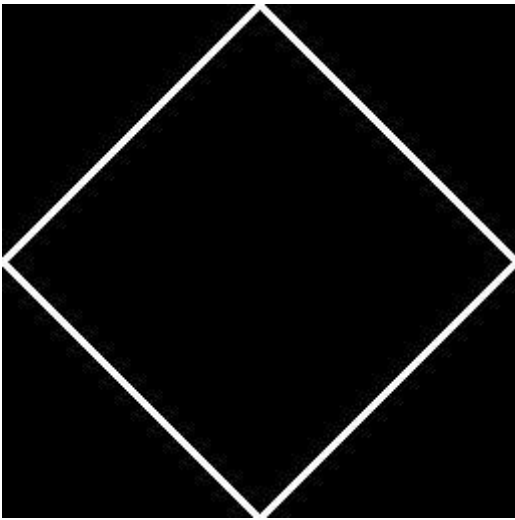
1. Напишите функцию `twist_image(input_file_name, output_file_name)`, которая будет менять местами левую и правую половины изображения.

Параметр `input_file_name` задаёт имя исходного файла, а `output_file_name` — имя файла, куда следует сохранить результат.

При передаче данного изображения:



должно получиться:



Формат ввода

Функция не должна ничего считывать.

Формат вывода

Функция не должна ничего выводить.

Примечания

Все имена файлов имеют расширение `.jpg`.

2. Мы очень любим мультфильм «Чип и Дейл». А вы?

Давайте попробуем написать функцию **chip_and_dale(number)**, которая сможет монозвуковой файл с любой речью переделать так, чтобы казалось, что слова произносит герой любимого нами мультфильма **Чип**.

Для этого надо ускорить воспроизведение.

В функцию передаётся натуральное число **i** из диапазона [2..5].

Необходимо прочитать файл **in.wav**, оставить каждый **i-ый** фрейм и сохранить результат в файле **out.wav**

В качестве образца по работе со звуковым файлом возьмите пример из урока.

Формат ввода

chip_and_dale(3)

Формат вывода

Звуковой файл **out.wav**

3. Напишите функцию **pitch_and_toss()**, которая будет делить файл **in.wav** на четыре части, менять местами первую часть с третьей, а вторую с четвертой и сохранять полученный результат в файле **out.wav**.

Обратите внимание, что в общем случае файл может содержать количество фреймов, не кратное 4. Поэтому первые три части можно сделать одинаковыми с помощью целочисленного деления, а в четвертую записать все оставшиеся данные.

Формат ввода

Звуковой файл **in.wav** в формате «моно», лежащий в папке с программой.

Формат вывода

Звуковой файл **out.wav**, лежащий в папке с программой.

Для 9ПИ

Модуль 5 «Объектно-ориентированное программирование»

1. Ранней весной Ларт отправлялся в один из своих вояжей – как всегда неожиданно и как всегда спешно.

Выехал он на рассвете, снабдив меня множеством инструкций. Я должен был исполнить несколько мелких поручений в поселке, привести в порядок дом, собрать дорожный сундучок и встретиться с Лартом в порту, чтобы вечером поднять паруса.

Напишите программу для сортировки предстоящих дел по важности: сначала пусть идут дела с наибольшим количеством слов, если количество одно и то же, то порядок менять не нужно. Ну и при выводе нужно сократить каждое дело до первых трех слов – ведь это только напоминание.

Формат ввода

Вводятся строки слов, записанных через пробел.

Формат вывода

Вывести эти же строки, сокращенные до первых трех слов в описанном порядке.

Пример 1

Ввод

```
carry out a few small tasks
put the house in order
collect a travel chest
knock the dust out of a traveling cloak
meet at the port
raise sails
```

Вывод

```
knock the dust
carry out a
put the house
collect a travel
meet at the
raise sails
```

Пример 2

Ввод

```
clean the chimney
fix the chains on the spell book
turn the skeleton to the wall so that it does not grin
feed the raven
fix the spider web for the house spider
```

Вывод

```
turn the skeleton
fix the spider
fix the chains
clean the chimney
feed the raven
```

2. Прямо передо мной располагалось так называемое лохматое пятно – в этом месте постоянно отрастала шерсть на ворсистом ковре, и в мои обязанности входило регулярно ее подстригать, уравнивая с остальной ковровой поверхностью. Остриженную шерсть я собирал в полотняный мешочек, надеясь со временем связать себе шарф.

Напишите функцию **cutting_spot()**, подстригающую строки строго по правилу:

- если длина строки кратна 5, то её нужно записать с прописной буквы, остальные строчные;
- если кратна 3, то все буквы строчные;
- если длина кратна 2, то регистр меняется на противоположный – большие буквы становятся маленькими, маленькие большими.

Функция принимает произвольное число аргументов-строк, а возвращает словарь: ключи – числа 2, 3 и 5, а значения – списки преобразованных по правилам строк в алфавитном порядке.

Примечание

Ключи словаря могут быть как числового, так и строкового типа, на результат проверки это не повлияет.

Пример 1

Ввод

```
data = [  
    'Carpet', 'pILe', 'surface',  
    'wOOl', 'graSs', 'Shell', 'armoR'  
]  
print(cutting_spot(*data))
```

Вывод

```
{  
    "2": [  
        "PiLe",  
        "WooL",  
        "cARPET"  
    ],  
    "3": [  
        "carpet"  
    ],  
    "5": [  
        "Armor",  
        "Grass",  
        "Shell"  
    ]  
}
```

Пример 2

Ввод

```
data = ['scarf', 'Hat', 'sweAter', 'Mittens']  
print(cutting_spot(*data))
```

Вывод

```
{  
    "2": [],  
    "3": [  
        "hat"  
    ],  
    "5": [  
        "Scarf"  
    ]  
}
```

3. А мрачное чудовище в углу называлось вешалкой. Трудно сказать, на что она больше была похожа – на больное дерево или скелет уродливого животного. И получилось как-то незаметно, что в доме, где полно чудес и диковин, эта вешалка оказалась едва ли не самой странной странностью – Ларт явно выделял ее среди всех предметов.

Напишите класс **Hanger** (Вешалка), экземпляр которого при инициализации принимает аргументы:

- имя заколдованного;
- время пребывания в заколдованном виде;

- именованный аргумент *oddness* (странность) со значением по умолчанию 0.

Класс обеспечивает выполнение методов (*hg* – экземпляр класса):

- time_inc()* – увеличить время на значение аргумента – одновременно странность увеличивается на целую часть от деления аргумента на 2;
- экземпляры можно складывать: *hg_new = hg + hg1* – получается новый экземпляр с именем, составленным из первого слова из имени первого и добавленным через пробел последнего слова из имени второго; время у новой Вешалки – 0, странность – среднее, округленное до целого вниз; исходные экземпляры не меняются;
- для экземпляра определена операция взятия остатка от деления на число: *hg %= n* – в результате у него меняется странность, становится равной остатку от деления исходного значения на это число;
- экземпляры можно сравнивать: сначала по странности, затем по времени, затем по имени по алфавиту;
- экземпляр можно вызвать без аргументов – возвращается произведение времени на странность;
- для вывода возвращается строка вида:

```
Hanger by name <name> (<time>, <oddness>)
Вешалка по имени <имя> (<время>, <странность>)
```

Пример 1

Ввод

```
hg = Hanger('Rual', 3)
print(hg)
hg.time_inc(8)
hg %= 8
print(hg)
print(hg())
```

Вывод

```
Hanger by name Rual (3, 0)
Hanger by name Rual (11, 4)
44
```

Пример 2

Ввод

```
hg = Hanger('Rual Illmarrannen', 3)
hg1 = Hanger('Lart Marran', 1, 5)
hg2 = hg + hg1
print(hg > hg1, hg <= hg2, hg != hg1, sep='\n')
print(hg, hg1, hg2, sep='\n')
hg2.time_inc(3)
id_hg1 = id(hg1)
hg1 %= 4
print(hg < hg2, hg == hg2, hg1 >= hg2, sep='\n')
print(hg, hg1, hg2, sep='\n')
print(id_hg1 == id(hg1))
```

Вывод

```
False
True
True
Hanger by name Rual Illmarrannen (3, 0)
Hanger by name Lart Marran (1, 5)
Hanger by name Rual Marran (0, 2)
True
False
False
Hanger by name Rual Illmarrannen (3, 0)
Hanger by name Lart Marran (1, 1)
Hanger by name Rual Marran (3, 3)
True
```

Для 9П

Модуль 5 «Способы обработки потока данных»

1. Напишите программу для решения следующей задачи. Камера наблюдения регистрирует в автоматическом режиме скорость проезжающих мимо нее автомобилей, округляя значения скорости до целых чисел. Необходимо определить максимальную зарегистрированную скорость автомобиля. Если скорость хотя бы одного автомобиля была меньше 30 км/ч, выведите «YES», иначе выведите «NO».

Программа получает на вход число проехавших автомобилей N ($1 < N < 30$), затем указываются их скорости. Значение скорости не может быть меньше 1 и больше 300. Программа должна сначала вывести максимальную скорость, затем YES или NO.

Пример работы программы:

Входные данные	Выходные данные
4 74 69 63 66	74 NO

2. Напишите программу, которая в последовательности натуральных чисел определяет наименьшее число, кратное 3. Программа получает на вход целые числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не является членом последовательности). Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Гарантируется, что в последовательности есть хотя бы одно число, кратное 3.

Программа должна вывести одно число: наименьшее число, кратное 3.

Пример работы программы:

Входные данные	Выходные данные
45 10 9 0	9

3. Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое двузначных чисел или сообщает, что таких чисел нет (выводит «NO»). Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность). Количество чисел не превышает 100. Введенные числа не превышают 300. Программа должна вывести среднее арифметическое двузначных чисел или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых

Пример работы программы:

Входные данные	Выходные данные
10 120 49 0	29.5
111 1 0	NO

4. Напишите программу, которая в последовательности натуральных чисел находит среднее арифметическое чисел, кратных 8, или сообщает, что таких чисел нет (выводит «NO»). Программа получает на вход натуральные числа, количество введенных чисел неизвестно, последовательность чисел заканчивается числом 0 (0 — признак окончания ввода, не входит в последовательность).

Количество чисел не превышает 100. Введенные числа не превышают 300. Программа должна вывести среднее арифметическое чисел, кратных 8, или вывести «NO», если таких чисел нет. Значение выводить с точностью до десятых.

Пример работы программы:

Входные данные	Выходные данные
8 122 64 16 0	29.3
111 1 0	NO

5. Напишите программу для решения следующей задачи. Камера наблюдения регистрирует в автоматическом режиме скорость проезжающих мимо нее автомобилей, округляя значения скорости до целых чисел. Необходимо определить среднюю зарегистрированную скорость всех автомобилей. Если не менее двух автомобилей двигались со скоростью не больше 40 км/ч, выведите «YES», иначе выведите «NO».

Программа получает на вход число проехавших автомобилей N ($1 \leq N \leq 30$), затем указываются их скорости. Значение скорости не может быть меньше 1 и больше 300. Программа должна сначала вывести среднюю скорость, с точностью до одного знака после запятой, затем «YES» или «NO».

Пример работы программы:

Входные данные	Выходные данные
4 74 69 63 96	75.5 NO

6. Напишите программу, которая в последовательности натуральных чисел определяет сумму чисел, оканчивающихся на 4. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 4. Количество чисел не превышает 1000. Введенные числа не превышают 30 000. Программа должна вывести одно число — сумму чисел, оканчивающихся на 4.

Пример работы программы:

Входные данные	Выходные данные
3 14 25 24	38