

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного автономного образовательного учреждения
высшего образования
«Национальный исследовательский технологический университет «МИСИС»
СТИ НИТУ «МИСИС»

Рабочая программа утверждена
решением Ученого совета
СТИ НИТУ «МИСИС»
от «23» июня 2026 г.
протокол № 38

Рабочая программа дисциплины

Информатика

Закреплена за кафедрой	<u>Кафедра высшей математики и информатики</u>
Направление подготовки	15.03.04 – Автоматизация технологических процессов и производств
Профиль	Автоматизация технологических процессов и производств (горно-металлургическое производство)
Квалификация	<u>Бакалавр</u>
Форма обучения	<u>Очная</u>
Общая трудоемкость	5 ЗЕТ

Часов по учебному плану	<u>180</u>
в том числе:	
аудиторные занятия	<u>85</u>
самостоятельная работа	<u>41</u>
часов на контроль	<u>54</u>

Формы контроля в семестрах:
экзамен 1

Распределение часов дисциплины по семестрам

Семестр	1		Итого	
Вид занятий	УП	РП	УП	РП
Лекции	17	17	17	17
Лабораторные	68	68	68	68
Контактная работа	85	85	85	85
Сам. работа	41	41	41	41
Часы на контроль	54	54	54	54
Итого:	180	180	180	180

Год набора 2026 г.

Программу составил:

Доцент, кандидат педагогических наук

Некрасова Наталья Ивановна

Должность, уч. ст., уч. зв. ФИО полностью



подпись

Рабочая программа дисциплины

Информатика

наименование

Разработана в соответствии с ОС ВО НИТУ «МИСИС»:

Самостоятельно устанавливаемый образовательный стандарт высшего образования - бакалавриат

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС» по направлению подготовки 15.03.04 – Автоматизация технологических процессов и производств (приказ от 25.11.2021 г. № 465 о.в.)

Составлена на основании учебного плана 2026 года набора:

15.03.04 – Автоматизация технологических процессов и производств,

Профиль: Автоматизация технологических процессов и производств (горно-металлургическое производство),
утвержденного Ученым советом СТИ НИТУ «МИСИС» «23» июня 2026 г., протокол № 38.

Рабочая программа одобрена на заседании кафедры

Высшей математики и информатики

наименование кафедры

Протокол от «19» июня 2026 г. № 9.

Зав. кафедрой ВМиИ

аббревиатура наименования кафедры

«19» июня 2026 г.



подпись

Е.Г. Кабулова

И.О. Фамилия

Руководитель ОПОП ВО

И.о. зав. кафедрой АИСУ, кандидат

технических наук, доцент

должность, уч. ст., уч. зв.

«22» июня 2026 г.



подпись

Д.А. Полещенко

И.О. Фамилия

1. ЦЕЛИ ОСВОЕНИЯ

Цели освоения дисциплины:

1. **Формирование системы знаний** о вкладе информатики в формирование целостной научной картины мира, роли информационных и коммуникационных технологий в современном информационном обществе, а также об основных информационных процессах, технических и программных средствах их реализации.
2. **Развитие логического и алгоритмического мышления**, способности к формализации, критическому анализу и синтезу информации, применению системного подхода при решении профессиональных задач в области информационных систем и технологий.
3. **Приобретение практического опыта** использования информационных ресурсов, современных инструментальных средств (электронные таблицы, СУБД, среды программирования) и коммуникационных технологий в учебной, научно-исследовательской и будущей профессиональной деятельности.
4. **Воспитание ответственного отношения** к соблюдению этических и правовых норм информационной деятельности, правил сетевого этикета, требований информационной безопасности

Задачи дисциплины:

1. **Теоретическая подготовка:**
 - 1.1 изучить базовые понятия информатики, архитектуру ЭВМ, классификацию программного обеспечения, основы функционирования локальных и глобальных сетей, принципы информационной безопасности;
 - 1.2 освоить основные положения реляционной модели данных, нормализацию, язык SQL;
 - 1.3 ознакомиться с этическими и правовыми аспектами работы с информацией в цифровой среде.
2. **Формирование практических умений и навыков:**
 - 2.1 квалифицированно использовать прикладные программы общего назначения (электронные таблицы, СУБД) для реализации базовых информационных процессов – сбора, обработки, анализа, визуализации и хранения структурированных и неструктурированных данных;
 - 2.2 применять современные сетевые и облачные технологии для обмена информацией, совместной работы и публичного представления результатов.
3. **Алгоритмическая и программная подготовка:**
 - 3.1 освоить основы алгоритмизации (структурное, модульное, нисходящее проектирование);
 - 3.2 сформировать умения и навыки практического применения алгоритмических конструкций при составлении программ на языке Python;
 - 3.3 научиться разрабатывать, проводить отладку, тестирование и документирование программ, предназначенных для практического применения в области информационных систем и технологий.
4. **Подготовка к профессиональной деятельности:**
 - 4.1 заложить базу для успешного освоения последующих дисциплин (программирование, базы данных, компьютерные сети, информационная безопасность);
 - 4.2 выработать навыки командной работы, эффективной коммуникации и ответственного отношения к своей роли в коллективных IT-проектах.

2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОП:	Б1.Б.06
2.1 Требования к предварительной подготовке обучающегося:	
2.1.1	Знания, умения и навыки, полученные при изучении образовательной программы среднего общего образования или среднего профессионального образования
2.2 Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее:	
2.2.1	Современные информационные технологии
2.2.2	Инженерная и компьютерная графика
2.2.3	Информатизация общества
2.2.4	Алгоритмизация и программирование задач управления
2.2.5	Защита информации
2.2.6	Основы программирования
2.2.7	Основы программирования на Python
2.2.8	Разработка баз данных
2.2.9	Вычислительные сети и системы
2.2.10	Инструментальные средства моделирования и проектирования
2.2.11	Моделирование процессов и систем
2.2.12	Основы кибернетики

3. ИНДИКАТОРЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ, СОВМЕЩЕННЫЕ С РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	
УК-1: Способен осуществлять поиск, критический анализ и синтез информации, умение анализировать процессы и системы с использованием соответствующих аналитических, вычислительных и экспериментальных методов, применять системный подход для решения поставленных задач	
Знать:	УК-1-31 – основные методы и технологии поиска, сбора, верификации, критической оценки и синтеза информации в электронной информационно-образовательной среде; УК-1-32 – перечень и возможности аналитических, вычислительных и экспериментальных методов, применяемых для обработки структурированных данных с использованием электронных таблиц (включая сводные таблицы, математическое моделирование), баз данных (языки QBE, SQL) и языков программирования (Python с библиотеками для анализа данных); УК-1-33 – типовые программные средства (электронные таблицы, СУБД, средства визуализации, языки программирования высокого уровня) для реализации информационного поиска, анализа и синтеза.
Уметь:	УК-1-У1 – осуществлять целенаправленный поиск информации в глобальных и локальных сетях, специализированных базах данных, реферативных системах и электронных библиотеках с применением современных информационно-коммуникационных технологий; УК-1-У2 – синтезировать новую информацию путём обобщения, структурирования и интеграции данных из разнородных источников для решения поставленной задачи; УК-1-У3 – применять аналитические и вычислительные методы (включая алгоритмизацию запросов) для анализа информационных процессов и систем.
Владеть:	УК-1-В1 – навыками поиска, критического анализа, систематизации и синтеза информации с использованием современных цифровых инструментов (офисные пакеты, средства поиска, системы управления базами данных, онлайн-сервисы); УК-1-В2 – навыками применения аналитических, вычислительных и экспериментальных методов для обработки данных и получения обоснованных выводов (построение таблиц, графиков, диаграмм, расчёт статистических показателей, разработка простых программных моделей, SQL-запросов); УК-1-В3 – навыками работы с прикладным программным обеспечением для автоматизации аналитической деятельности и поддержки принятия решений в профессиональной области.
УК-3: Способен эффективно обмениваться информацией, идеями, проблемами и решениями с инженерным сообществом и обществом в целом, осуществлять социальное взаимодействие и реализовывать свою роль в команде	
Знать:	УК-3-31 – основные принципы, протоколы и технологии передачи информации в глобальных и локальных компьютерных сетях, обеспечивающие эффективный обмен данными; УК-3-32 – архитектуру персональных компьютеров, структуру программного обеспечения, а также современные программные и аппаратные средства для коллективной работы (облачные сервисы, системы контроля версий, офисные пакеты совместного доступа) и их функциональные возможности; УК-3-33 – принципы организации командной работы и этики сетевого общения при взаимодействии в цифровой среде.
Уметь:	УК-3-У1 – использовать сетевое программное обеспечение для оперативного обмена технической информацией, файлами и данными; УК-3-У2 – формулировать и представлять технические проблемы и найденные решения в электронных документах с использованием современных прикладных программных средств; УК-3-У3 – работать в команде над общим IT-проектом, осуществляя обмен информацией, знаниями и опытом с членами команды; оценивать идеи других членов команды для достижения поставленной цели, нести ответственность за свой участок работы.
Владеть:	УК-3-В1 – навыками публичного представления результатов проектной деятельности и технической информации с использованием мультимедийных технологий; УК-3-В2 – инструментами облачных технологий для совместного создания и редактирования текстовых документов и табличных данных; УК-3-В3 – навыками работы в распределенной команде через цифровую среду, понимая свою роль и взаимодействуя с другими участниками для достижения общего результата.
ОПК-6: Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий	
Знать:	ОПК-6-31 – основные этапы и методики проектирования алгоритмов (структурное, модульное, нисходящее проектирование); ОПК-6-32 – актуальные инструментальные средства для разработки программ (интегрированные среды разработки, компиляторы/интерпретаторы, отладчики, среды для работы с SQL, СУБД); ОПК-6-33 – базовые конструкции языка Python: типы данных, условные операторы, циклы, функции (в том числе рекурсивные), коллекции (списки, словари, множества, кортежи), ввод-вывод с использованием стандартных средств и файловых операций; ОПК-6-34 – принципы разработки программ, предназначенных для практического применения (понятие надёжности, тестирования, документирования, пользовательского интерфейса).
Уметь:	ОПК-6-У1 – выбирать методику проектирования (например, нисходящее проектирование с пошаговой детализацией) в зависимости от сложности и типа решаемой задачи; ОПК-6-У2 – разрабатывать алгоритмы решения типовых задач обработки данных (вычисление сумм, поиск экстремумов, сортировка, работа с массивами и строками, простые рекурсивные алгоритмы; создание SQL-запросов на выборку, вставку, обновление, удаление); ОПК-6-У3 – реализовывать разработанные алгоритмы в виде программ на языке программирования Python; ОПК-6-У4 – создавать программы, предназначенные для практического применения: с понятным интерфейсом, обработкой ошибок ввода, комментариями и документацией.

Владеть:	ОПК-6-B1 – навыками применения методик проектирования алгоритмов для формализации и структурирования решения задачи; ОПК-6-B2 – навыками разработки и отладки программ на языке программирования Python для решения вычислительных, логических и информационно-поисковых задач; ОПК-6-B3 – навыками тестирования и верификации программ (разработка тестовых примеров, анализ результатов); ОПК-6-B4 – навыками документирования программного кода и создания инструкций пользователя для разработанных программ.
----------	--

4. СТРУКТУРА И СОДЕРЖАНИЕ						
Код занятия	Наименование разделов и тем /вид занятия/	Семестр	Часов	Компетенции	Литература и электронные ресурсы	Примечание
	Раздел 1. Технические и программные средства реализации информационных процессов.					
1.1	Информация и информатика. Актуальные философские и научно-методологические проблемы развития информатики. Проблема полисемии термина «информация». Информационные процессы в природе, технике и обществе. Свойства информации (атрибутивные, прагматические, динамические). Подходы к измерению информации (содержательный, алфавитный, вероятностный). Единицы измерения информации. Особенности кодирования информации различной природы. Системы счисления. /Лек/	1	1	УК-1-31, УК-1-32, УК-1-33	Л1.2, Л1.8, Л2.4, Э2, Э3, Э4	
1.2	Освоение материала по теме «Информация и информатика». /Ср/	1	2	УК-1-31, УК-1-32, УК-1-33, УК-1-У1, УК-1-У2, УК-1-В1	Л1.2, Л1.8, Л2.4, Э2, Э3, Э4	
1.3	Технические средства реализации информационных процессов. Предшествующие этапы и современные тенденции развития вычислительной техники и компьютерных технологий. Этические проблемы цифровых технологий. Принципы архитектуры Дж. фон Неймана. Состав компонентов ЭВМ Дж. фон Неймана. Магистрально-модульная архитектура ЭВМ. Процессор. Основные параметры процессоров. Запоминающие устройства: классификация, принцип работы, основные характеристики. Устройства ввода/вывода данных, их разновидности и основные характеристики. /Лек/	1	1	УК-3-32, УК-3-33	Л1.2, Л1.8, Л2.2, Э2, Э3, Э4	
1.4	Программные средства реализации информационных процессов. Программное обеспечение: базовый, системный, служебный, прикладной уровни. Операционная система. Классификация прикладных программных средств. Классификация служебных программных средств. Вредоносные программы. Антивирусные программы. Классификация программного обеспечения по условиям распространения и использования. Средства разработки программ. /Лек/	1	1	УК-1-33, УК-3-32, УК-3-33, ОПК-6-32	Л1.2, Л1.8, Л2.1, Э2, Э3, Э4	

1.5	Освоение материала по теме «Технические и программные средства реализации информационных процессов». /Ср/	1	2	УК-1-33, УК-3-32, УК-3-33, ОПК-6-32, УК-1-У3, УК-3-У1, УК-3-У3	Л1.2, Л1.8, Л2.1, Л2.2, Л2.4, Э2, Э3, Э4	
1.6	Подготовка к лабораторной работе «Информационные технологии анализа структурированных данных». /Ср/	1	1	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2, УК-3-В1	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
1.7	Информационные технологии анализа структурированных данных /Лаб/	1	6	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2 УК-3-В2	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
1.8	Подготовка к лабораторной работе «Информационные технологии визуализации структурированных данных». /Ср/	1	1	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2, УК-3-В1	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
1.9	Информационные технологии визуализации структурированных данных. Сводные таблицы. /Лаб/	1	4	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2 УК-3-В2	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
1.10	Подготовка к лабораторной работе «Математическое моделирование средствами электронных таблиц». /Ср/	1	1	УК-1-32 УК-1-33 УК-1-У2, УК-1-У3, УК-1-В2	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
1.11	Математическое моделирование средствами электронных таблиц. /Лаб/	1	4	УК-1-32 УК-1-33, УК-1-У3, УК-1-В2, УК-1-В3, УК-3-У2	Л1.5, Л2.3, Л2.11, Л3.1, Э3, Э13, Э14, Э15	
	Раздел 2. Основы работы с базами данных.					
2.1	Основы работы с базами данных. Информационные системы и базы данных. Типы СУБД и архитектура клиент-сервис. Модели данных. Реляционная модель данных. Проектирование баз данных. Целостность БД (языковая, структурная, ссылочная). Разработка схемы данных. Типы данных. Свойства полей базы данных. Объекты базы данных. Основные операции с данными. Язык запросов QBE. Язык структурированных запросов SQL. /Лек/	1	4	УК-1-32, УК-1-33	Л1.1, Л1.2, Л1.8, Л2.4, Л2.5, Э2, Э3, Э8, Э10	
2.2	Освоение материала по теме «Основы работы с базами данных». /Ср/	1	5	УК-1-32, УК-1-33, УК-1-У3, УК-1-В2	Л1.1, Л1.2, Л1.8, Л2.4, Л2.5, Э2, Э3, Э8, Э10	

2.3	Подготовка к лабораторной работе «Работа с базой данных. Язык QBE». /Ср/	1	1	УК-1-32 УК-1-33 УК-1-У3, УК-3-У1 УК-1-В1, УК-1-В3, ОПК-6-32	Л1.1, Л1.2, Л3.1, Л2.4, Э2, Э3, Э8, Э10	
2.4	Работа с базой данных. Язык QBE. /Лаб/	1	8	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2 УК-3-В1, ОПК-6-32	Л1.1, Л3.1, Э3, Э8, Э10	
2.5	Подготовка к лабораторной работе «Работа с базой данных. Язык SQL». /Ср/	1	1	УК-1-32 УК-1-33 УК-1-У3, УК-1-В1, УК-1-В2 УК-1-В3, ОПК-6-32, ОПК-6-У2	Л1.1, Л3.1, Л2.5, Э3, Э8, Э10	
2.6	Работа с базой данных. Язык SQL. /Лаб/	1	6	УК-1-32, УК-1-33, УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, ОПК-6-32, ОПК-6-У2, ОПК-6-В3, УК-3-В1, УК-3-У3	Л1.1, Л3.1, Л2.5, Э3, Э8, Э10	
	Раздел 3. Основы алгоритмизации и программирования.					
3.1	Основы алгоритмизации и программирования. История создания языка Python. Актуальные области применения и перспективы. Инструкции и структура программы. PEP 8 - руководство по написанию кода на Python. Базовые типы данных. Динамическая типизация. Операторы языка Python: арифметические, присваивания, сравнения, логические, операторы членства и тождественности. Приоритет их выполнения. Модуль Math. Особенности ввода/вывода данных. Инструкция if-elif-else, трехместное выражение if/else. Циклы for и while, инструкции break и continue, ветвь else в циклах. Функции range() и enumerate(). Вложенные циклы. Типы-коллекции в языке Python (индексирование, срезы, функции вычисления длины, максимального и минимального значений, сортировки, смены порядка следования элементов и др.). Списки (list), кортежи (tuple), множества (set), их функции и методы. Строки. Функции и методы строк. Словари (dict) и работа с ними. Модульное программирование. Определение функций. Функции с переменным числом аргументов. Локальные и глобальные переменные. Функции высшего порядка: map, filter, reduce и лямбда-выражения. Рекурсия. Использование стека для организации рекурсивных вызовов. Работа с файлами. /Лек/	1	8	ОПК-6-31, ОПК-6-32, ОПК-6-33, ОПК-6-34	Л1.3, Л1.4, Л1.6, Л1.8, Л2.6, Л2.7, Л2.8, Л2.12, Л3.2, Э5, Э6, Э7, Э9, Э11	

3.2	Освоение материала по теме «Основы алгоритмизации и программирования». /Ср/	1	8	ОПК-6-31, ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У1, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л1.8, Л2.6, Л2.7, Л2.8, Л2.12, Л3.2, Э5, Э6, Э7, Э9, Э11	
3.3	Подготовка к лабораторной работе «Программирование разветвляющихся вычислительных процессов». /Ср/	1	1	ОПК-6-33, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	
3.4	Программирование разветвляющихся вычислительных процессов. /Лаб/	1	6	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У3, ОПК-6-В2, ОПК-6-В3	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.5	Подготовка к лабораторной работе «Программирование циклических вычислительных процессов». /Ср/	1	1	ОПК-6-33, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	
3.6	Программирование циклических вычислительных процессов. /Лаб/	1	6	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У3, ОПК-6-В2, ОПК-6-В3	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.7	Подготовка к лабораторной работе «Одномерные коллекции Python: списки и генераторы». /Ср/	1	1	ОПК-6-33, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	
3.8	Одномерные коллекции Python: списки и генераторы. /Лаб/	1	6	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У1, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.9	Подготовка к лабораторной работе «Матричные вычисления: вложенные списки и библиотека NumPy». /Ср/		1	ОПК-6-33, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	
3.10	Матричные вычисления: вложенные списки и библиотека NumPy. /Лаб/		6	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У1, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.11	Подготовка к лабораторной работе «Модульное программирование: функции, lambda, рекурсия». /Ср/	1	1	ОПК-6-33, ОПК-6-У1, ОПК-6-У2, ОПК-6-В1	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	

3.12	Модульное программирование: функции, lambda, рекурсия. /Лаб/	1	8	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.13	Подготовка к лабораторной работе «Словарные структуры данных в Python». /Ср/	1	1	ОПК-6-33, ОПК-6-У1, ОПК-6-У2, ОПК-6-В1, УК-1-У3	Л1.3, Л1.4, Л1.6, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Э5, Э6, Э7, Э9, Э11	
3.14	Словарные структуры данных в Python. /Лаб/	1	8	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У2, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4, УК-3-В1, УК-3-У3	Л1.3, Л1.4, Л1.6, Л3.2, Э5, Э6, Э7, Э9	
3.15	Выполнение домашнего задания «Основы алгоритмизации и программирования» /Ср/	1	9	ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У2, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4, УК-3-У3, УК-3-В3	Л1.3, Л1.4, Л1.6, Л1.8, Л3.2, Л2.6, Л2.7, Л2.8, Л2.12, Л3.2, Э5, Э6, Э7, Э9, Э11	
	Раздел 4. Локальные и глобальные сети ЭВМ.					
4.1	Локальные и глобальные сети ЭВМ. Архитектура сети ЭВМ. Классификация сетей. Базовые топологии. Модель взаимодействия открытых систем ISO/OSI. Основы функционирования Интернета (стек протоколов TCP/IP). Сетевые и доменные адреса. Протоколы обмена информацией и службы Интернета. Унифицированный указатель ресурса (URL). Технологии проводного и беспроводного подключения к Интернету. Эволюция технологий ЛВС. Конвергенция сетей. Программно-определяемые сети (SDN). Защита от удаленного администрирования. Защита от компьютерных вирусов и их профилактика. Информационная безопасность в Интернете (конфиденциальность, целостность и доступность информации). Электронная цифровая подпись. /Лек/	1	2	УК-3-31, УК-3-32, УК-3-33, УК-1-В1	Л1.2, Л1.7, Л1.8, Л2.4, Э2, Э3, Э4, Э12	
4.2	Освоение материала по теме «Локальные и глобальные сети ЭВМ». /Ср/	1	4	УК-3-У1, УК-3-У2, УК-3-В3	Л1.2, Л1.7, Л1.8, Л2.4, Э2, Э3, Э4, Э12	
	Часы на контроль. / Контроль/	1	54	УК-1-31 УК-1-32 УК-1-33 УК-1-У1 УК-1-У2 УК-1-У3 УК-1-В1 УК-1-В2	Л1.1 – Л1.8, Л2.1 – Л2.12, Л3.1, Л3.2, Э1– Э15	

				УК-1-В3 УК-3-31 УК-3-32 УК-3-33 УК-3-У1 УК-3-У2 УК-3-У3 УК-3-В1 УК-3-В2 УК-3-В3 ОПК-6-31 ОПК-6-32 ОПК-6-33 ОПК-6-34 ОПК-6-У1 ОПК-6-У2 ОПК-6-У3 ОПК-6-У4 ОПК-6-В1 ОПК-6-В2 ОПК-6-В3 ОПК-6-В4		
--	--	--	--	--	--	--

5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

5.1. Вопросы для самостоятельной подготовки к экзамену

Контрольные вопросы, используемые при формировании экзаменационных билетов.

Экзаменационный билет состоит из 4 вопросов. Вопросы 1 и 2 – теоретические вопросы (по разделам курса 1-5) (УК-1-31; УК-1-32; УК-3-31; УК-3-32; УК-3-33; ОПК-6-31; ОПК-6-32; ОПК-6-33; ОПК-6-34):

1. Актуальные философские и научно-методологические проблемы развития информатики (УК-1-31, УК-1-У1).
2. Проблема полисемии термина «информация» (УК-1-31, УК-1-32).
3. Свойства информации (атрибутивные, прагматические, динамические) (УК-1-32).
4. Подходы к измерению информации (содержательный, алфавитный, вероятностный). Единицы измерения информации (УК-1-32).
5. Особенности кодирования информации различной природы (УК-1-32, УК-1-33).
6. Системы счисления (УК-1-32).
7. Этапы развития вычислительной техники. Поколения ЭВМ. Тенденции развития вычислительной техники и компьютерных технологий (УК-3-32, УК-1-У2).
8. Этические проблемы цифровых технологий (УК-3-33, УК-3-У2).
9. Принципы архитектуры Дж. фон Неймана. Состав компонентов ЭВМ Дж. фон Неймана (УК-3-32).
10. Магистрально-модульная архитектура ЭВМ (УК-3-32).
11. Процессор. Основные характеристики процессоров (УК-3-32).
12. Запоминающие устройства (внутренние, внешние) (УК-3-32).
13. Устройства ввода информации. Основные характеристики и виды (УК-3-32).
14. Устройства вывода информации. Основные характеристики и виды (УК-3-32).
15. Программное обеспечение: базовый, системный, служебный, прикладной уровни (УК-3-32).
16. Операционная система (УК-3-32).
17. Прикладные программные средства, их классификация (УК-1-33, УК-3-32, УК-1-В3, УК-3-В2).
18. Служебные программные средства (УК-1-33).
19. Классификация вредоносных программ. Антивирусные программы (УК-3-33, УК-1-33).
20. Классификация программного обеспечения по условиям распространения и использования (УК-3-32, УК-3-33).
21. Базовые типы данных языка Python. Динамическая типизация (ОПК-6-33, ОПК-6-32).
22. Операторы языка Python: арифметические, присваивания, сравнения, логические, операторы членства и тождественности. Приоритет их выполнения (ОПК-6-33).
23. Особенности ввода/вывода данных в языке Python (ОПК-6-33).
24. Инструкция if-elif-else, трехместное выражение if/else (ОПК-6-33).
25. Циклы for и while, инструкции break и continue, ветвь else в циклах. Функции range() и enumerate() (ОПК-6-33).
26. Определение функций. Функции с переменным числом аргументов. Локальные и глобальные переменные. Функции высшего порядка: map, filter, reduce и лямбда-выражения. Рекурсия (ОПК-6-33, ОПК-6-31).
27. Типы-коллекции в языке Python (индексирование, срезы, функции вычисления длины, максимального и минимального значений, сортировки, смены порядка следования элементов) (ОПК-6-33).
28. Строки. Функции и методы строк (ОПК-6-33).
29. Списки (list), кортежи (tuple), множества (set), их функции и методы (ОПК-6-33).
30. Словари (dict) и работа с ними (ОПК-6-33).
31. Работа с файлами (ОПК-6-33, ОПК-6-34).
32. Модели данных (УК-1-32, УК-1-33).

33. Реляционная модель данных (УК-1-33).
34. Межтабличные связи в реляционной БД. Первичные и внешние ключи (УК-1-33).
35. Целостность БД (языковая, структурная, ссылочная) (УК-1-33).
36. Нормализация отношений в реляционной БД (1НФ, 2 НФ, 3 НФ) (УК-1-32, УК-1-33).
37. Свойства полей таблиц базы данных (на примере СУБД Microsoft Access) (УК-1-33).
38. Типы данных (на примере СУБД Microsoft Access) (УК-1-33).
39. Объекты базы данных (на примере СУБД Microsoft Access) (УК-1-33).
40. Классификация сетей ЭВМ по степени удаленности компьютеров в сети (УК-3-31).
41. Классификация сетей ЭВМ по типу функционального взаимодействия (УК-3-31).
42. Базовые топологии компьютерной сети (УК-3-31).
43. Модель взаимодействия открытых систем ISO/OSI (УК-3-31).
44. Основы функционирования Интернета (стек протоколов TCP/IP) (УК-3-31, УК-3-У1).
45. Сетевые и доменные адреса (УК-3-31).
46. Протоколы обмена информацией и службы Интернета (УК-3-31).
47. Унифицированный указатель ресурса (URL) (УК-3-31).
48. Технологии проводного подключения к Интернету (УК-3-31, УК-3-32).
49. Технологии беспроводного подключения к Интернету (УК-3-31, УК-3-32).
50. Эволюция технологий ЛВС. Конвергенция сетей. Программно-определяемые сети (SDN) (УК-3-31, УК-3-32).
51. Защита от удаленного администрирования. Защита от компьютерных вирусов и их профилактика (УК-3-33).
52. Информационная безопасность в Интернете (конфиденциальность, целостность и доступность информации) (УК-3-33, УК-3-31, УК-1-В1).
53. Электронная цифровая подпись (УК-3-33).

Вопрос 3 – задача (раздел 2) (УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, ОПК-6-У2, ОПК-6-В3, УК-3-В1).

Примеры задачи.

Пример №1.

Имеются данные о каталоге книг библиотеки института: шифр книги; ФИО автора; год издания; номер шкафа; номер стеллажа. Используя операторы языка SQL создайте следующие запросы:

- a. запрос на создание таблицы в соответствии с имеющейся информацией;
- b. запрос на внесение записей в таблицу;
- c. запрос, определяющий среднее количество книг на каждом стеллаже указанного шкафа;
- d. запрос, удаляющий информацию о книгах указанного автора, изданных во второй половине XX века.

Пример №2.

Имеются данные об участниках конференции: код участника, ФИО участника, адрес, место работы, номер телефона, дата заявки, сумма оплаты за проживание. Используя операторы языка SQL создайте следующие запросы:

- a. запрос на создание таблицы в соответствии с имеющейся информацией;
- b. запрос на внесение записей в таблицу;
- c. запрос, определяющий ФИО и номер телефона участника, который внес наибольшую сумму оплаты за проживание;
- d. запрос, удаляющий информацию об участниках, которые подали заявки позже указанной даты

Вопрос 4 – задача (раздел 3) (ОПК-6-У2, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4, ОПК-6-У1, ОПК-6-В1, УК-3-У3, УК-3-В3).

Примеры задачи.

Пример №1.

a. Дан одномерный список, состоящий из N целых неотрицательных чисел. Количество элементов списка (N) и их значения вводятся пользователем. Требуется вставить на позицию с заданным пользователем индексом значение среднего арифметического положительных элементов этого массива, восьмеричная запись которых оканчивается цифрой 0. В случае отсутствия указанных элементов, на позицию, указанную пользователем, следует вставить значение, равное индексу максимального элемента данного списка.

b. Дан словарь *servers*, где ключом является *ID сервера* (например, "SRV-001"), а значением – другой словарь, содержащий информацию о сервере: *name* (имя сервера (строка)); *ip* (IP-адрес (строка)); *response_times* (список времен отклика в миллисекундах (целые или вещественные числа), полученных в результате последовательных ping-запросов). Например,

"SRV-001": {"name": "Web1", "ip": "192.168.1.1", "response_times": [25, 30, 28, 32]}

Требуется написать функцию *calculate_average_response_time(servers_dict)*, которая возвращает новый словарь, где ключ – ID сервера (из исходного словаря); значение – среднее арифметическое времени отклика для этого сервера (тип float), округлённое до двух знаков после запятой.

Пример №2.

a. Дан одномерный список, состоящий из N целых чисел (могут быть как положительные, так и отрицательные). Количество элементов списка N и их значения вводятся пользователем с клавиатуры. Требуется вставить на позицию с заданным пользователем индексом значение, равное среднему арифметическому элементов, которые являются двузначными числами-палиндромами (например, 11, 22, 33, ..., 99). Если такие элементы отсутствуют, то на указанную позицию следует вставить значение, равное индексу минимального элемента исходного списка.

б. Дан словарь *products*, где ключом является артикул товара (например, "A123"), а значением – другой словарь, содержащий информацию о товаре: *name* (название товара (строка)); *category* (категория товара (строка)); *monthly_sales* (список целых чисел, обозначающих количество проданных единиц товара за каждый месяц (первый элемент – продажи за январь, второй – за февраль и т.д.)). Например,

"A123": {"name": "Ноутбук", "category": "Электроника", "monthly_sales": [15, 22, 18, 30, 25]}

Требуется написать функцию ***find_max_sales(products_dict)***, которая возвращает новый словарь, где ключ – артикул товара (из исходного словаря); значение – максимальное количество продаж за один месяц (целое число) для этого товара.

Если список *monthly_sales* пуст, значение должно быть равно 0.

5.2. Перечень работ, выполняемых по дисциплине

Домашнее задание «Основы алгоритмизации и программирования» (ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4, УК-3-У3, УК-3-В3).

Домашнее задание выполняется в форме практической работы. Требуется составить алгоритм решения приведенных в варианте задач и реализовать его на языке программирования Python. Решение каждой задачи необходимо сопроводить пояснительной запиской.

Репродуктивный уровень.

Задача 1.

Даны две расчётные формулы для параметров *a* и *b*, зависящие от одной или нескольких переменных, значения которых вводятся пользователем. Формулы могут содержать арифметические операции, извлечение корня, логарифмирование, возведение в степень и другие математические функции.

Требуется написать программу, которая:

1. Запрашивает у пользователя значения переменных, необходимых для вычисления *a* и *b* (количество и тип переменных зависят от варианта).
2. Вычисляет значения *a* и *b* по заданным формулам, используя встроенные математические функции (например, `math.sqrt`, `math.log`, `math.pow` и т.п.).
3. Проверяет область допустимых значений для всех операций:
 - знаменатель дроби не должен быть равен нулю;
 - подкоренное выражение должно быть неотрицательным (для квадратного корня);
 - аргумент логарифма должен быть положительным, а основание – положительным и не равным 1;
 - при возведении в степень с нецелым показателем основание должно быть положительным;
 - другие ограничения, характерные для используемых функций.
4. Обрабатывает случаи, когда вычисление невозможно (нарушение ОДЗ): если хотя бы одно из выражений не может быть вычислено корректно, программа должна вывести сообщение об ошибке с указанием причины (например, «деление на ноль», «отрицательное подкоренное выражение» и т.п.) и завершить работу (или запросить новые данные, но по условию обычно достаточно вывести ошибку и остановиться).
5. Если оба значения вычислены успешно, программа должна сравнить их и вывести результат сравнения в виде:
 - $a > b$
 - $a < b$
 - $a = b$ (с учётом погрешности вычислений, если значения вещественные, рекомендуется сравнивать с точностью, например, $\text{abs}(a - b) < 1e-9$).
6. Выводит вычисленные значения *a* и *b* с точностью до 4–6 знаков после запятой (или по требованию варианта), а также результат сравнения.

Требования к реализации.

- Программа должна использовать условные операторы (`if-elif-else`) для проверки ОДЗ и сравнения результатов.
- Для математических вычислений используется модуль `math`.
- Ввод данных осуществляется с клавиатуры (числа с плавающей точкой).
- Программа должна корректно обрабатывать проверку ОДЗ.

Вариант 1

$$a = \frac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}, \quad b = x(\arctg z + e^{-(x+3)})$$

Вариант 2

$$a = \frac{3 + e^{y-1}}{1 + x^2|y - \text{tg } z|}, \quad b = 1 + |y - x| + \frac{(y - x)^2}{2} + \frac{|y - x|^3}{3}.$$

Задача 2.

Дана таблица соответствий, содержащая интервалы значений, конкретные значения или комбинации параметров, которые позволяют по одному входному числу (или строке) однозначно определить выходной результат – категорию, уровень, тип, рекомендацию, сообщение и т.п.

Требуется написать программу, которая:

1. запрашивает у пользователя входные данные в соответствии с вариантом (одно число, например: скорость соединения, объём памяти, частота процессора, время отклика, загрузка, пинг, размер переменной и т.д.);
2. обрабатывает введённые данные с помощью условных операторов (if-elif-else):
 - если значение попадает в заданный интервал или совпадает с конкретным значением из таблицы, программа выбирает соответствующую категорию, уровень или сообщение;
 - для интервалов проверка выполняется последовательно (например, if $x < a$, elif $x < b$ и т.д.);
 - для точных совпадений используются сравнения чисел (или строк, если предусмотрено вариантом);
3. выводит на экран требуемый результат: категорию, уровень, рекомендацию, тип, сообщение или дополнительную информацию;
4. обрабатывает особые случаи, если они предусмотрены вариантом:
 - пограничные значения (точное совпадение с границей интервала) – вывод специального сообщения («пограничное значение, относится к ...»);
 - ошибочные значения (отрицательные, нулевые, некорректные) – вывод сообщения об ошибке;
 - запредельные значения (слишком большие или слишком малые) – вывод предупреждения или специального сообщения.

Вариант 1

Вводится скорость соединения (в Мбит/с). Требуется определить категорию и рекомендацию по применению согласно таблице.

Таблица. Классификация скорости интернет-соединения

Скорость, Мбит/с	Категория	Применение
< 1	Очень низкая	Только текстовые сообщения
1 – 9	Низкая	Просмотр веб-страниц, почта
10 – 49	Средняя	Видеозвонки, потоковое видео (SD)
50 – 99	Высокая	Видео 4К, онлайн-игры
100 – 499	Очень высокая	Профессиональная работа, серверы
≥ 500	Гигабитная	Дата-центры, магистральные сети

Особые случаи:

- Скорость точно 1 Мбит/с – пограничное значение, относится к низкой.
- Скорость точно 50 Мбит/с – пограничное значение, относится к высокой.
- Скорость < 0 – ошибка.
- Скорость > 10000 – сообщение «запредельное значение».

Вариант 2

Вводится объём данных (в ГБ). Требуется определить рекомендуемый тип носителя и определить область его применения.

Таблица. Классификация типа хранилища данных

Объём, ГБ	Тип носителя	Применение
< 1	Оперативная память (кэш)	Для быстрого доступа
1 – 15	SSD-накопитель	Для операционной системы
16 – 255	HDD (ноутбучный)	Для хранения файлов
256 – 1023	HDD (настольный)	Для архива данных
1024 – 4095	Облачное хранилище	Для удалённого доступа
≥ 4096	Серверный массив	Для масштабируемых систем

Особые случаи:

- Объём точно 16 – «SSD-накопитель».
- Объём точно 1024 – «облачное хранилище».
- Объём < 0 – ошибка.
- Объём > 100000 – «промышленное хранилище».

Задача 3.

Вычислить и вывести на экран монитора в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от $X_{нач}$ до $X_{кон}$ с шагом dx и точностью ε (значения указанных параметров задаются пользователем при запуске программы, например, $\varepsilon = 0.00003$). Таблицу следует снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.

Примечание. При решении задачи следует получить рекуррентную формулу для вычисления очередного члена заданного ряда в виде $a_{n+1} = k \cdot a_n$, где $k = k \frac{a_{n+1}}{a_n}$

Вариант 1

$$\ln \frac{x+1}{x-1} = 2 \cdot \sum_{n=0}^{\infty} \frac{1}{(2 \cdot n + 1) \cdot x^{2 \cdot n + 1}} = 2 \cdot \left(\frac{1}{x} + \frac{1}{3 \cdot x^3} + \frac{1}{5 \cdot x^5} + \dots \right), \quad |x| > 1$$

Вариант 2

$$e^{-x} = \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots, \quad |x| < \infty$$

Задача 4.

Дана прямоугольная таблица (матрица) размера $M \times N$, где:

- M строк — объекты одного типа (например, студенты, сотрудники, товары, регионы, спортивные команды);
- N столбцов — объекты другого типа (например, предметы, месяцы, характеристики, показатели).
- Элемент таблицы $a[i][j]$ — это числовая характеристика (оценка, объём продаж, количество, балл), связанная с i -м объектом первого типа и j -м объектом второго типа.

Требуется написать программу, которая:

1. выводит таблицу на экран в удобочитаемом виде, указав названия строк и столбцов;
2. вычисляет сумму / произведение заданных элементов;
3. для каждого объекта первого типа (каждой строки) определяет максимальное значение в строке и указывает соответствующий ему объект второго типа (название столбца). Если максимумов несколько – указывает первый из них или все (по условию в варианте);
4. для каждого объекта второго типа (каждого столбца) определяет минимальное значение в столбце и указывает соответствующий ему объект первого типа (название строки). Если минимумов несколько – указывает первый из них или все (по условию в варианте).

Примечания. Все исходные данные (размеры, названия, значения) задаются в коде для каждого варианта. В выводе должны присутствовать поясняющие тексты.

Вариант 1

Матрица *Таблица успеваемости* содержит информацию об оценках 5 студентов по 5 предметам. Строки – студенты (1–5), столбцы – предметы: математика, физика, химия, биология, история.

Исходные данные:

[5, 4, 5, 3, 4]

[3, 5, 4, 4, 5]

[4, 4, 5, 5, 4]

[5, 5, 5, 4, 5]

[3, 4, 3, 5, 4]

Требуется:

1. Вывести таблицу успеваемости, указав для строк «Студент 1» ... «Студент 5» и для столбцов названия предметов.

Таблица успеваемости:

	математика	физика	химия	биология	история
Студент 1	5	4	5	3	4
Студент 2	3	5	4	4	5
Студент 3	4	4	5	5	4
Студент 4	5	5	5	4	5
Студент 5	3	4	3	5	4

2. Найти суммарный балл студентов, получивших оценки на главной диагонали (студент 1 – математика, студент 2 – физика, ..., студент 5 – история).
3. Найти произведение оценок на побочной диагонали (студент 1 – история, студент 2 – биология, ..., студент 5 – математика).
4. Для каждого студента определить лучший предмет (максимальная оценка в строке).
5. Для каждого предмета определить худшую оценку (минимальная оценка в столбце).

Вариант 2

Матрица *Температура воздуха* показывает среднедневную температуру (°C) в 5 городах за 5 дней. Строки – города: Москва, Санкт-Петербург, Новосибирск, Екатеринбург, Казань. Столбцы – дни (1–5).

Исходные данные:

[22, 24, 23, 21, 20]

[18, 19, 20, 19, 18]

[25, 26, 27, 28, 26]

[15, 16, 15, 14, 15]

[30, 31, 29, 30, 32]

Требуется:

1. Вывести таблицу температур, указав названия городов и номера дней.

Таблица температур (°C):

	День 1	День 2	День 3	День 4	День 5
Москва	22	24	23	21	20
СПб	18	19	20	19	18
Новосибирск	25	26	27	28	26
Екатеринбург	15	16	15	14	15
Казань	30	31	29	30	32

2. Найти сумму температур на главной диагонали (Москва – день 1, Санкт-Петербург – день 2, ..., Казань – день 5).
3. Найти произведение температур на побочной диагонали (Москва – день 5, Санкт-Петербург – день 4, ..., Казань – день 1).
4. Определить максимальную температуру для каждого города (за 5 дней).
5. Определить минимальную температуру в каждый день (среди всех городов).

Задача 5.

Дан словарь *data* фиксированного размера (10 записей), где каждый ключ – идентификатор некоторого объекта (товар, студент, валюта, сотрудник, фильм и т.п.), а значение – числовая или строковая характеристика этого объекта (количество, цена, курс, рейтинг, расход топлива и т.п.). Требуется написать программу, которая:

1. запрашивает у пользователя набор запросов. Формат ввода может быть одним из двух типов:
 - только ключи (например, названия товаров, фамилии студентов, коды валют): ввод через пробел в одной строке или построчно до пустой строки;
 - пары «ключ значение» (например, продукт и вес, модель и расстояние, город и длительность): ввод через пробел или построчно до пустой строки, где каждая строка содержит ключ и связанное с ним дополнительное число (вес, расстояние, длительность);
2. для каждого запроса (ключа или пары) выполняет:
 - поиск ключа в словаре *data*.
 - если ключ найден:
 - извлекает соответствующее значение из словаря;
 - если запрос содержал дополнительное число (пара), выполняет вычисление с использованием извлечённого значения (например, стоимость = цена × вес, расход = норма × расстояние);
 - выводит результат для данного ключа в удобочитаемом виде;
 - если ключ не найден:
 - выводит заранее определённое сообщение об ошибке (например, «не аттестован», «конвертация невозможна», «абонент не найден»);
 - такой запрос не учитывается при подсчёте итоговых агрегатов;
3. вычисляет итоговый агрегат (сумма, среднее арифметическое и т.п.) по всем успешно обработанным запросам (тем, где ключ был найден);
4. выводит итоговый агрегат с пояснительным текстом.

Дополнительные требования.

- Все исходные данные (словарь из 10 записей) задаются в коде.
- Программа должна корректно обрабатывать ввод:
 - Ввод через пробел в одной строке (например, "яблоко молоко хлеб").
 - Построчный ввод до пустой строки (пользователь вводит строки, пока не нажмёт *Enter* без ввода).
- Для вариантов с парами «ключ значение» – значения могут быть целыми или вещественными числами.

Вывод должен быть понятным: для каждого ключа – отдельная строка с результатом, в конце – итоговый агрегат

Вариант 1

Требуется написать программу, запрашивающую у пользователя названия нескольких товаров (ввод через пробел или построчно до пустой строки) и определяющую количество каждого из указанных товаров, а также их суммарное количество. Исходные данные хранятся в словаре *stock*, где ключи – названия товаров, значения – их количество на складе (целые числа). Словарь содержит информацию о 10 товарах.

Вариант 2

Требуется написать программу, запрашивающую у пользователя фамилии нескольких студентов (ввод через пробел или построчно до пустой строки) и определяющую оценку каждого из указанных студентов, а также средний балл только среди тех, кто есть в словаре. Исходные данные хранятся в словаре *grades*, где ключи – фамилии студентов, значения – их оценки (целые числа от 2 до 5). Словарь содержит информацию о 10 студентах. Для отсутствующих студентов выводить сообщение «не аттестован» и не учитывать при расчёте среднего.

Задача 6.

Дано несколько векторов (списков). Требуется обработать элементы векторов согласно варианту задания, реализовав набор функций:

- ввод вектора;
- вывод вектора;
- проверка элемента на заданное нетривиальное условие (простота, числа Фибоначчи, Армстронга и т.д.);
- вычисление для вектора некоторой агрегированной характеристики (сумма, количество, среднее, доля и т.п.);
- выбор одного из векторов по сложному критерию.

Каждый вариант должен содержать следующие функции (имена можно изменять):

```
def input_vector(name):          # ввод элементов вектора
def output_vector(vec):         # вывод вектора на экран
def condition(x):               # проверка условия
```

```
# вычисление заданной характеристики вектора
# (напр. сумма_подходящих)
def compute_characteristic(vec):
```

```
# критерий выбора
def choose_best(vectors, characteristics):
```

В основной программе:

- создать список векторов;
- для каждого вектора вычислить характеристику.
- среди всех векторов выбрать один по критерию.

Вывести выбранный вектор и его характеристику.

Вариант 1

Дано несколько векторов (списков) целых положительных чисел. Элемент вектора считается «особым», если он является простым числом (делится только на 1 и на себя). Для каждого вектора требуется вычислить сумму особых элементов. Среди всех векторов выбрать тот, у которого эта сумма максимальна. Если несколько векторов имеют одинаковую максимальную сумму, выбрать среди них тот, у которого количество особых элементов больше.

Вариант 2

Дано несколько векторов целых положительных чисел. Элемент считается совершенным числом, если он равен сумме всех своих собственных делителей (например, $6 = 1+2+3$). Для каждого вектора требуется вычислить плотность совершенных чисел = (количество совершенных) / (длина вектора). Среди всех векторов выбрать тот, у которого плотность максимальна. При равенстве плотности выбрать вектор с наибольшей суммой совершенных чисел.

Реконструктивный уровень.

Требуется реализовать программу для работы с коллекцией объектов, хранящейся в словаре. Структура данных:

- *Внешний словарь*: ключ — уникальный идентификатор объекта (ID, ISBN, артикул и т.п.), значение – *внутренний словарь*, содержащий атрибуты объекта (название, характеристики, числовые показатели).

Программа должна:

1. создать словарь и заполнить его данными (не менее 10 записей);
2. реализовать функцию ***print_all...()*** для вывода всей информации на экран;
3. реализовать функцию вычисления агрегированной характеристики (средний балл, средний рейтинг по жанрам, средняя цена, общий баланс и т.п.);
4. реализовать функцию поиска объекта с максимальным (или минимальным) значением некоторого показателя (лучший студент, самая популярная книга, самая интенсивная тренировка и т.д.).

Требования к реализации.

- Все функции должны быть вынесены в отдельный модуль *functions.py*.
- Во втором файле (*main.py*) должно быть реализовано консольное меню, позволяющее пользователю последовательно вызывать каждую из трёх функций (вывод всех, вычисление агрегата, поиск лучшего).
- Модуль *functions* должен импортироваться в *main.py*.

Вариант 1

1. Создайте словарь *students*, где ключом является *ID студента*, значением – другой словарь с информацией о студенте: *name, age, grades* (список оценок).
2. Добавьте данные как минимум для десяти студентов.
3. Напишите функцию ***print_all_students(students_dict)***, которая выводит информацию о всех студентах.

4. Напишите функцию `calculate_average_grades(students_dict)`, которая возвращает новый словарь, где ключ - ID студента, а значение – его средний балл.
5. Напишите функцию `find_top_student(students_dict)`, которая находит ID студента с самым высоким средним баллом.

Вариант 2

1. Создайте словарь `library`, где ключом является ISBN книги, а значением — другой словарь с информацией о книге: `title, author, year, genre, rating`.
2. Добавьте данные как минимум для десяти книг.
3. Напишите функцию `print_all_books(library_dict)`, которая выводит информацию обо всех книгах.
4. Напишите функцию `calculate_average_rating(library_dict)`, которая возвращает новый словарь, где ключ - жанр, а значение - средний рейтинг по жанрам.
5. Напишите функцию `find_top_book(library_dict)`, которая находит самую популярную книгу.

При изучении дисциплины предусмотрено **выполнение и защита лабораторных работ**. Каждая лабораторная работа предусматривает выполнение двух практических заданий (задания репродуктивного уровня и задания реконструктивного уровня) и ответы на контрольные вопросы.

Лабораторная работа «Информационные технологии анализа структурированных данных» (УК-1-32, УК-1-33, УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2, УК-3-В2)

Практическое задание

Задача 1. Репродуктивный уровень.

Требуется провести анализ заказов компьютерной техники на основании исходных данных, представленных в таблицах «Заказы» (список заказов на поставку компьютерной техники), «ПрайсЛист», «Клиенты». Необходимо реализовать следующие вычисления и преобразования:

Форматирование и подстановка данных

- Произвольное оформление заголовка таблицы.
- Автоматический подбор цены через функцию ВПР с обработкой #Н/Д.
- Условное форматирование ячеек с ценой 0 (заливка, цвет шрифта).

Расчётные столбцы

- Стоимость заказа (количество × цена)
- Скидка по прогрессивной шкале (0%/5%/10%)
- НДС 20%
- Сумма к оплате (стоимость – скидка + НДС)

Агрегатные формулы с массивами

- Общая выручка (без скидок).
- Наибольшее значение по условию варианта.
- Сумма, среднее, количество ячеек по условию варианта.

Условное форматирование, сортировка и фильтрация

- Ячейки суммы к оплате >1 000 000 – особый стиль.
- На новых листах: многоуровневая сортировка, выборка по условию, подведение итогов (суммы, средние по группам).

Работа с таблицей («умная таблица»)

- Преобразовать исходный список на листе «Заказы» в таблицу.
- Добавить столбец «Премия менеджера»: 3% от итоговой суммы, если итог >100 000, иначе 0.
- Вставить строку нового заказа.
- Вставить строку итогов: средний процент скидки, общая сумма к оплате.
- Те же итоги отфильтровать по одному городу (например, Белгород).

Текстовые функции, функции Дата и время

На листе «Клиенты»: объединить ФИО в строку (Фамилия Имя Отчество).

- Сформировать сокращённый вариант «Фамилия И.О.».
- Вычислить возраст клиентов (количество полных лет) на текущую дату.

Примеры вариантов

№	Постановка задачи
1	Вычислите: ▪ стоимость максимального заказа на поставку компьютеров в г. Орел; ▪ суммарное количество ноутбуков (шт.), проданных в г. Воронеж; ▪ среднюю стоимость заказов (руб.), оформленных менеджером Слесаревым С.Ю.; ▪ количество заказов на поставку компьютеров, отправленных в г. Белгород. На новых листах выполните сортировку данных на трех уровнях по возрастанию: а. по Наименованию, Дате, Сумме к оплате; б. по Городу, Дате, Количеству. На новых листах сделайте 2 выборки данных, используя следующие критерии: а. выбрать сведения о заказах на поставку офисных АТС и телефонов в г. Воронеж в январе 2024 г.; б. выбрать сведения о заказах на поставку устройства связи в города Белгород и Воронеж в первом полугодии 2024 г.

	На новом листе для каждой категории товаров подведите итоги продаж (по столбцу <i>Количество (шт.)</i>) по каждому городу.
2	Вычислите: - стоимость максимального заказа на поставку планшетов, оформленного менеджером Шацких А.С.; - суммарное количество принтеров (шт.), проданных в г. Курск; - среднюю стоимость заказов (руб.) клиентов из г. Орел; - количество заказов на поставку 3D-принтеров, отправленных в г. Орел. На новых листах выполните сортировку данных на трех уровнях по возрастанию: - по Категории, Наименованию, Количеству; - по Городу, Дате, Сумме к оплате. На новых листах сделайте 2 выборки данных, используя следующие критерии: - выбрать сведения о заказах на поставку печатных устройства в г. Белгород, оформленных менеджером Слесаревым С.Ю. осенью 2024 г.; - выбрать сведения о заказах на поставку компьютеров в г. Белгород, при оформлении которых была предоставлена скидка в размере от 100 000 до 150 000 руб.

Задача 2. Реконструктивный уровень.

Дана электронная книга, содержащая несколько связанных таблиц (листов) с информацией об объектах некоторой предметной области (клиенты, заказы, услуги/товары/книги/фильмы и т.п.). Каждая таблица имеет фиксированную структуру: столбцы с идентификаторами, наименованиями, количественными характеристиками. Требуется выполнить следующие действия, *не используя стандартные средства фильтрации и сортировки данных* (т.е. без автофильтра, расширенного фильтра, сортировки через меню). Допускается использование встроенных функций: ВПР, СУММЕСЛИ, СЧЁТЕСЛИ, СРЗНАЧЕСЛИ, СУММЕСЛИМН, СЧЁТЕСЛИМН, СРЗНАЧЕСЛИМН, а также формул с массивами, логических и текстовых функций, функций для работы с датами. Выборка данных по условиям может выполняться с помощью расширенного фильтра, формул массива или других неразрушающих методов.

1. На основании данных, содержащихся в таблицах, требуется вычислить следующие показатели (конкретные условия и агрегаты зависят от варианта):

- суммарное значение (сумма по условию) – например, сумма заказов, выручка, количество, стоимость.
- максимальное / минимальное значение по условию (самый старый, самый поздний, максимальный бюджет, максимальная партия и т.п.).
- среднее значение по условию (средняя стоимость, средняя продолжительность, средний размер скидки и пр.).
- относительная доля (процент) от общего количества или суммы по заданному условию.
- все вычисления должны быть реализованы формулами, которые автоматически пересчитываются при изменении исходных данных. Фильтрация и сортировка через стандартные диалоговые окна не применяются.

2. Создать новый лист. На нём для каждой группы объектов (например, для каждого типа услуг, для каждого автора, для каждого города, для каждой даты, для каждого района и т.п.) вычислить:

- суммарную стоимость / количество / объём по группе;
- дополнительные агрегаты (по варианту – например, отдельно для выполненных и невыполненных заказов, отдельно по категориям номеров, отдельно по поставщикам).

Результаты представить в виде таблицы с группировкой (без использования сводных таблиц, только промежуточными итогами).

3. На новых листах (отдельных для каждой выборки) с помощью расширенного фильтра (но без изменения исходных данных) получить выборки, удовлетворяющие заданным критериям (варьируются по варианту).

Примеры вариантов

№	Постановка задачи
1	1. В электронные таблицы занесли данные, отражающие работу рекламного агентства. Таблица «Клиенты» содержит данные о клиентах, таблица «Заказы» – о совершённых заказах, таблица «Услуги» – о видах услуг, таблица «Типы услуг» – о типах услуг. На основании данных, содержащихся в приведенных таблицах, ответьте на следующие вопросы (без использования Фильтра и Сортировки!): - На какую сумму было выполнено заказов на печать визиток клиентам из г. Владивосток. - Какова средняя стоимость заказов на изготовление наружной рекламы, выполненных в первом полугодии 2024 г. - Чему равна наибольшая стоимость заказа на разработку дизайна визиток, выполненного для клиентов из г. Дивногорск. - Сколько процентов от общего заказов составили заказы на разработку дизайна сайта. 2. На новом листе для каждого типа услуг подведите итоги, отражающие суммарную стоимость (руб.) выполненных заказов по каждому наименованию предоставляемых услуг. 3. На новых листах сделайте 2 выборки данных, используя следующие критерии: - выбрать сведения о заказах на печать блокнотов для клиентов из г. Владивосток и печать визиток для клиентов из г. Алексеевка; - выбрать сведения о заказах на изготовление билбордов, итоговая стоимость которых превысила 1 млн руб., и изготовление штендеров с итоговой стоимостью выше 500000 руб.
2	1. В электронные таблицы занесли данные, отражающие работу библиотеки. Таблица «Книги» содержит полную информацию о книгах, таблица «Читатели» – данные о читателях, таблица «Бронирование» – записи о забронированных книгах, таблица «Выдача книг» – информацию о выдаче книг и сроках их возвращения.

- На основании данных, содержащихся в этой таблице, ответьте на следующие вопросы (без использования Фильтра и Сортировки!):
- Определите общую стоимость книг, выданных читателям из г. Сочи в первом квартале 2024 г.
 - Определите год издания самой старой из забронированных читателям из г. Билибино книг Н.В. Гоголя.
 - Какова средняя стоимость книг, изданных в прошлом веке, забронированных читателями г. Чита.
 - Сколько процентов от общего количества наименований составляют книги А.П. Чехова, изданные в последнем десятилетии 19 века.
2. На новом листе определите количество выданных книг каждого автора. Для первых трех авторов в списке дополнительно определите количество выданных книг каждого года издания.
3. На новых листах сделайте 2 выборки данных, используя следующие критерии:
- выбрать сведения о книгах А.П. Чехова, изданных в 1899 г., и книгах Л.Н. Толстого, изданных в 1910 г.;
 - выбрать сведения о книгах, забронированных в июне читателями из г. Сочи и в июле читателями из г. Каспийск.

Контрольные вопросы

- Какие данные может содержать ячейка рабочего листа?
- Перечислите допустимые операнды в формуле в электронной таблице.
- Перечислите арифметические операции и операции отношения допустимые в формуле в электронной таблице.
- Какой вид адресации следует использовать, если при копировании формулы необходимо, чтобы ссылки, входящие в эту формулу, оставались неизменными?
- В ячейке B3 содержится формула
=A2+\$C\$4+\$B2
Каким образом изменятся операнды при копировании формулы в ячейку D5?
- Приведите пример формулы, содержащей абсолютную ссылку.
- Перечислите последовательность действий при построении формулы в электронной таблице.
- Перечислите категории функций табличного процессора.
- В какой последовательности располагаются при сортировке строк в списке числа, текст, даты, логические значения и пустые ячейки:
 - при сортировке по возрастанию?
 - при сортировке по убыванию?
- Что означает появление ##### при выполнении расчетов?
- В диапазоне B1:B5 по порядку указаны значения 1, 5, 3, 4, 2. Какой результат получится в ячейке C1, если в ней указать формулу:
 - =МАКС(B1:B5)
 - =МАКС(B1;B5)
- На что ссылается ячейка в документе MS Excel, если она содержит следующее значение [Выручка.xls]Январь!A5?
- Если ячейку, в которой указана формула, скопировать (Ctrl+C) и вставить в другую ячейку (Ctrl+V), то вставится формула или результат вычисления формулы?
- Если к диапазону, содержащему 4 столбца, применить команду Сортировка, по какому столбцу будет отсортирован диапазон?
- Какие аналитические функции и формулы используются для выявления статистических закономерностей в наборе данных? Перечислите и кратко охарактеризуйте функции для расчета среднего арифметического, медианы, моды, стандартного отклонения, а также функции для условного подсчета (СЧЁТЕСЛИ) и суммирования (СУММЕСЛИ).?
- Какой опцией специальной вставки можно горизонтальный диапазон данных вставить вертикально и наоборот?
- Что считает формула "=СУММ(A:A)"?
- Для чего используется функция ВПР? Опишите её синтаксис и аргументы. Как обработать ситуацию, когда искомое значение не найдено (ошибка #Н/Д)?
- Что такое условное форматирование и для каких целей оно применяется?

Лабораторная работа «Информационные технологии визуализации структурированных данных» (УК-1-32, УК-1-33, УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2, УК-3-В2)

Практическое задание

Задача 1. Репродуктивный уровень.

В электронную таблицу занесли информацию о заказах на поставку компьютерной техники. На основании исходных данных, представленных в таблице *Заказы*, требуется:

- построить сводную таблицу, отображающую квартальные итоги продаж по каждому наименованию товара каждой категории с возможностью выбора города (в качестве фильтра отчёта);
- сгенерировать аналогичные итоги для каждого города на отдельных листах активной рабочей книги;

3. построить сводную таблицу, отображающую итоги продаж каждого менеджера для каждого города, и настроить для неё интерактивные фильтры (срезы и/или временную шкалу), с помощью которых продемонстрировать показатели, указанные в варианте задания;
4. построить сводную диаграмму (объёмная гистограмма) под названием «Итоги продаж», отображающую итоги продаж указанных в варианте категорий товаров. При построении предусмотреть возможность выбора даты оформления заказа (фильтр отчёта);
5. отформатировать диаграмму, настроив произвольным образом:
 - формат области диаграммы (фон, границы);
 - формат области построения диаграммы (заливка, сетка);
 - формат стенок и основания (для объёмной диаграммы);
 - формат рядов данных (фигура, фронтальный зазор, боковой зазор, эффекты (тень, свечение), заливка, контур).
6. построить ещё одну сводную диаграмму на отдельном листе, демонстрирующую итоги продаж указанных в варианте категорий товаров по заданным городам в заданном периоде (например, в первом квартале 2026 г.).

Примеры вариантов

№	Критерии фильтрации
1	1. продажи моноблоков в январе и феврале 2021 г.; 2. продажи устройств связи (офисная МТС, телефон) в четвертом квартале.
2	1. продажи компьютеров в последнем квартале; 2. продажи печатных устройств (принтер, МФУ) и устройств связи (офисная АТС) в первом полугодии.

Задача 2. Реконструктивный уровень.

На основе данных, содержащихся в исходной электронной таблице (файл *Вариант_N.xlsx*), требуется выполнить следующие действия.

1. Построение сводной таблицы и сводной диаграммы.
 - Создать сводную таблицу, которая отражает заданные в варианте агрегированные показатели (суммарная масса груза, средний балл и т.п.) с указанной точностью (например, один знак после запятой).
 - В сводной таблице должны быть учтены фильтры по определённым критериям (например, конкретные населённые пункты и даты; определённые округа и предметы).
 - В макете сводной таблицы должна быть предусмотрена возможность выбора значения одного из полей через срез (или фильтр отчёта): например, выбор «Пункта отправления» или «Фамилии ученика».
 - На основе полученной сводной таблицы построить сводную диаграмму.
2. Оформление диаграммы.
 - Добавить название диаграммы (содержание названия определяется вариантом).
 - Произвести форматирование диаграммы в соответствии с требованиями варианта, включая: формат области диаграммы; формат области построения диаграммы; формат стенок и основания диаграммы (для объёмных диаграмм); формат рядов данных: фигура (форма столбцов, секторов и т.п.), фронтальный и боковой зазоры, эффекты (тени, свечение, сглаживание), заливка (цвет, градиент, текстура) и другие параметры.

Примеры вариантов

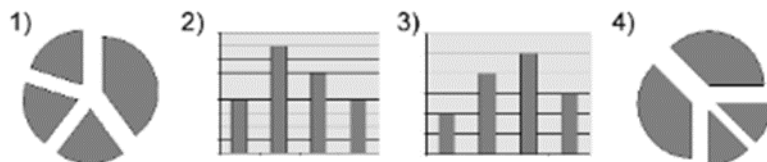
№	Постановка задачи
1	В электронную таблицу занесли информацию о грузоперевозках, совершенных некоторым автопредприятием с 1 по 9 октября (файл <i>Вариант_1.xlsx</i>). На основании данных, содержащихся в исходной таблице, постройте сводную таблицу и сводную диаграмму, отражающие сведения о суммарной массе груза, доставленного в населённые пункты <i>Березки, Буково, Осинки</i> 8 и 9 октября. При создании макета сводной таблицы должна быть предусмотрена возможность выбора <i>Пункта отправления</i> .
2	В электронную таблицу занесли данные о тестировании учеников. На основании данных, содержащихся в этой таблице, постройте сводную таблицу и сводную диаграмму, отражающие сведения о среднем балле по <i>литературе и русскому языку</i> учащихся четырех округов: <i>Северного, Восточного, Южного и Западного</i> . Средний балл в таблице должен отражаться с точностью – один знак после запятой. При создании макета сводной таблицы должна быть предусмотрена возможность выбора <i>Фамилии ученика</i> .

Контрольные вопросы

1. Перечислите основные элементы диаграммы и приведите их краткое описание.
2. Перечислите стандартные типы диаграмм табличного процессора и дайте их краткое описание.
3. Укажите допустимые варианты размещения сводной таблицы и диаграммы.
4. Опишите основные возможности при форматировании рядов трехмерной гистограммы.
5. Какой тип диаграмм наиболее удобен для представления информации в процентах?
6. Каким образом возможно изменить расположение легенды в диаграмме?
7. Как понимать сообщение # знач! при вычислении формулы?
8. Дан фрагмент электронной таблицы в режиме отображения формул.

	A	B	C	D
1	=C2-1	=A1*2	=C2+B1-2	=(B1+D2)/2
2			3	2

После выполнения вычислений построили диаграмму по значениям диапазона A1:D1. Укажите полученную диаграмму:

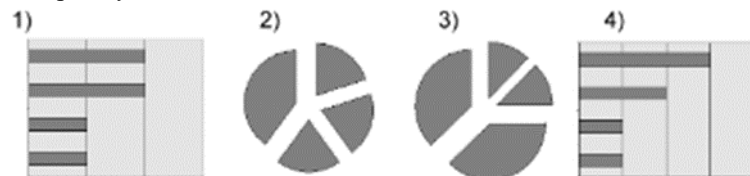


Ответ обоснуйте.

9. Дан фрагмент электронной таблицы в режиме отображения формул.

	A	B	C	D
1	$=C2-B1$	$=B2-C2$	$=B1+C2$	$=(C1-C2)*3$
2		3	2	

После выполнения вычислений построили диаграмму по значениям диапазона A1:D1. Укажите полученную диаграмму:



Ответ обоснуйте.

- Опишите инструменты табличного процессора, позволяющие настраивать интерактивные фильтры для сводной таблицы или диаграммы.
- Какие элементы диаграммы соответствуют числовым данным рабочего листа и как они изменяются при изменении исходных данных?
- Какие основные параметры необходимо указать при построении новой диаграммы?
- Как открыть панель «Формат» для тонкой настройки дизайна конкретного элемента диаграммы?
- Что такое сводная таблица и для каких целей она используется?
- Почему сводные таблицы не обновляются автоматически при изменении исходных данных? Как выполнить их обновление?
- Как можно изменить стиль оформления среза (цветовую гамму, внешний вид)?

Лабораторная работа «Математическое моделирование средствами электронных таблиц» (УК-1-32, УК-1-33, УК-1-У3, УК-1-В2, УК-1-В3, УК-3-У2)

Практическое задание

Задача 1. Репродуктивный уровень.

Дана готовая математическая модель (система ограничений, целевая функция и направление оптимизации), представленная в виде системы линейных неравенств и целевой функции. Требуется реализовать эту модель в среде электронных таблиц и найти оптимальное решение с использованием надстройки «Поиск решения».

Примеры вариантов

№ варианта	Математическая модель
1	$L(X) = 5x_1 + 7x_2 - 6x_3 + 9x_4 + 8x_5 \rightarrow \max;$ $\begin{cases} 0,7x_1 + 0,9x_2 + 1,5x_3 + 2,3x_4 + 1,8x_5 \leq 50000 \\ 0,4x_1 + 1,1x_2 - 0,5x_3 + 1,3x_4 - 2,8x_5 \geq 32000 \\ 0,5x_1 + 1,8x_3 + 0,7x_4 + 2x_5 \leq 40000 \\ 2,2x_1 - 1,4x_2 - 0,8x_3 + 0,9x_4 = 15000 \\ x_j \geq 0 (j = \overline{1,5}). \end{cases}$
2	$L(X) = x_1 + 4x_3 + 8x_4 - 12x_5 \rightarrow \min;$ $\begin{cases} x_1 + 9x_2 + 2x_3 - 4x_4 = 250, \\ 0,4x_1 + x_2 - 5x_3 + 3x_4 + 8x_5 \leq 460, \\ 0,5x_1 + 10x_2 - 8x_3 + 6x_4 + 2x_5 \leq 190, \\ 11x_2 - 8,5x_3 + 3x_4 + 2x_5 = 210, \\ x_j \geq 0 (j = \overline{1,5}). \end{cases}$

Задача 2. Реконструктивный уровень.

Дано текстовое описание реальной оптимизационной задачи (производство, транспорт, распределение ресурсов, назначение и т.п.) с числовыми данными (таблицы затрат, норм расхода, объемов ресурсов и т.д.). Требуется:

- составить математическую модель задачи, которая должна содержать:

- описание переменных X_i (например, количество продукции вида А, объём перевозок из пункта i в пункт j и т.п.);
- целевую функцию и направление её оптимизации (максимизация прибыли, минимизация затрат, минимизация времени и т.п.);
- систему ограничений с пояснением смысла каждого ограничения (например, ограничения на ресурсы, на спрос, на пропускную способность);
- граничные условия (неотрицательность переменных, целочисленность, если требуется);

2. реализовать модель среде электронных таблиц и найти оптимальное решение с помощью надстройки «Поиск решения»;

3. изменить параметры задачи согласно варианту (например, увеличить ресурс, изменить стоимость, повысить производительность и т.п.) и найти новое решение;

4. сравнить значения целевых функций для исходного и изменённого варианта и сделать выводы о влиянии изменения параметров на оптимальный план и значение целевой функции.

Примеры вариантов

Вариант 1

1. На предприятии имеется 3 типа взаимозаменяемых станков и требуется изготовить 3 вида изделий. Известно время изготовления (мин) каждого изделия на каждом оборудовании. Требуется определить: какое изделие и на каком оборудовании необходимо изготавливать, чтобы суммарное время изготовления всех изделий было минимально.

2. Увеличить производительность 3-го станка при изготовлении изделия В в 2 два раза, найти решение и сравнить с предыдущим.

Изделия \ Станки	1	2	3
А	30	70	50
Б	20	40	40
В	40	70	20

Вариант 2

1. Цех выпускает три вида изделий, используя три вида затрат производственных ресурсов. Определить оптимальный план производства трех видов продукции, обеспечивающий максимум товарной продукции.

2. Увеличить количество сырья в два раза, найти решение и сравнить с предыдущим.

Ресурсы	Нормативы затрат			Общий объем ресурсов
	1	2	3	
Оборудование (машин/час)	2	3	4	78
Сырье (кг)	1	4	5	85
Электроэнергия (кВт.ч)	3	4	2	81
Оптовая цена (руб.)	8	7	6	

Контрольные вопросы

1. Дайте определение понятия «Математическое моделирование».
2. Дайте определение понятия «Математическая модель».
3. Назовите обязательные компоненты задачи оптимизации.
4. Каковы основные этапы решения задач оптимизации в MS Excel?
5. В чем смысл использования символа \$ в формулах MS Excel?
6. В чем состоит отличие результата, полученного при вычислении суммы по указанным ниже формулам:
 $=\text{СУММ}(\text{Январь:Март!C5})$
 $=\text{СУММ}(\text{Январь!C5; Март!C5})$
7. Почему при вводе формул в ячейки ЦФ и левых частей ограничений в них отображаются нулевые значения?
8. Каким образом в MS Excel задается направление оптимизации ЦФ?
9. Поясните общий порядок работы с окном «Поиск решения».
10. Каким образом можно изменять, добавлять, удалять ограничения в окне «Поиск решения»?
11. Какие сообщения выдаются в MS Excel в случаях: успешного решения задачи ЛП; несовместности системы ограничений задачи; неограниченности ЦФ?
12. В чём заключается назначение параметра «Относительная погрешность» в окне «Параметры поиска решения»? Как его изменение влияет на процесс поиска оптимального решения?
13. Какие типы задач оптимизации можно решать с помощью надстройки «Поиск решения»? Какие ограничения задает флажок «Линейная модель»?

14. Как в Excel задаётся требование целочисленности переменных в задаче оптимизации? В каких случаях это требование необходимо?.

Лабораторная работа «Работа с базой данных. Язык QBE» (УК-1-32, УК-1-33, УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, УК-3-У2, УК-3-В1, ОПК-6-32)

Практическое задание

Задача 1. Репродуктивный уровень.

- На основании группы атрибутов, приведенных в варианте задания, спроектируйте две таблицы со справочной информацией и одну таблицу учетного характера, для этого:
 - определите типы значений атрибутов и их размер;
 - определите первичные и внешние ключи в таблицах и необходимые связи между ними.
- Создайте файл БД.
- Создайте таблицы БД:
 - при создании справочных таблиц укажите ключевые поля;
 - при создании учетной таблицы для внешних ключей используйте столбцы подстановок (код и характеризующий его столбец) из справочных таблиц, при этом следует отменить опцию: «скрыть ключевой столбец» и выбрать основной столбец подстановки;
 - откорректируйте, при необходимости, схему данных со связями между таблицами.
- Внесите данные в таблицы (не менее 10 кортежей в каждой таблице) и убедитесь, что система поддержки целостности БД функционирует.
- Используя язык запросов QBE, выполните запросы согласно варианту задания.
- Создайте 2 простые формы для каждой справочной таблицы и выполните следующие действия:
 - Выполните коррекцию форм при помощи Конструктора;
 - Просмотрите созданные формы по записям;
 - Введите новые записи в таблицы через формы согласно заданию;
 - Внесите изменения в таблицы через формы согласно заданию;
 - Отсортируйте записи в таблицах согласно заданию;
 - Создайте и примените фильтр согласно заданию;
 - Измените простые формы на ленточные и просмотрите их по записям.
- Создайте сводный отчет, в который должны войти все поля за исключением одинаковых полей:
 - Предусмотрите в отчете текущие итоги по какой-либо агрегатной функции;
 - Просмотрите отчет и, если необходимо, отредактировать его в режиме Конструктора.

Примеры вариантов

№ варианта	Атрибуты таблиц и задания
1	СКЛАДСКОЙ УЧЁТ ПРОКАТА Таблицы - Код_проката, Наименование_проката, Марка_стали, ГОСТ, Цена_за_тонну - Код_кладовщика, ФИО, Смена, Телефон - Код_проката, Код_кладовщика, Приход_тонн, Расход_тонн, Дата_операции Запросы - Удалить информацию о прокате с заданным наименованием. - Увеличить на 10% цену проката заданной марки стали. - Создать новую таблицу с наименованиями проката и его ценой, превышающей заданное значение. - Выбрать коды и наименования проката с кодами от 2 до 6. - Выбрать все сведения о кладовщиках, работающих в заданной смене. - Выбрать коды, названия и цены проката, оприходованного в заданном месяце. - Определить количество позиций проката, цена которых не превышает заданную. Формы - Включить информацию о новом кладовщике. - Изменить приход проката для заданного кода проката. - Просмотреть таблицы после изменений. - Отсортировать записи по фамилиям кладовщиков. - Отфильтровать данные, показав только прокат с выбранной маркой стали. Отчеты Создать отчет, демонстрирующий текущие итоги по количеству прихода проката.

2	ОТГРУЗКА ГОТОВОЙ ПРОДУКЦИИ Таблицы - Код_изделия, Наименование_изделия, Тип_проката, Масса_кг, Цена - Код_экспедитора, ФИО, Автомобиль, Номер_путевого_листа - Код_изделия, Код_экспедитора, Отгружено_штук, Дата_отгрузки, Покупатель Запросы - Удалить данные об изделии с заданным наименованием. - Увеличить цену изделий заданного типа проката на 15%. - Создать таблицу с названиями изделий и ценами, превышающими заданную. - Выбрать коды и наименования изделий с кодами от 4 до 9. - Выбрать все сведения об экспедиторах, работающих на заданном автомобиле. - Выбрать коды, названия и цены изделий, отгруженных в заданном месяце. - Подсчитать количество изделий, цена которых не выше заданной. Формы - Добавить нового экспедитора. - Изменить количество отгруженных изделий для заданного кода изделия. - Просмотреть таблиц после изменений. - Отсортировать записи по фамилиям экспедиторов. - Отфильтровать данные, показав только информацию по выбранному покупателю. Отчёты Создать отчет, демонстрирующий, текущие итоги по количеству отгруженных изделий.
---	---

Задача 2. Реконструктивный уровень.

- Используя язык запросов QBE, выполните запросы согласно варианту задания.
- Создайте 2 составные формы: подчиненную и связанную, в которые должны войти все поля за исключением одинаковых полей. В режиме Конструктора отредактируйте одну из составных форм и добавьте в нее вычисляемое поле согласно заданию. Отформатируйте форму произвольным образом.
- Внести изменения в макет отчета:
 - добавить в отчет вычисляемое поле согласно заданию;
 - добавить рисунок в верхний колонтитул и примечание с датой создания отчета и фамилией исполнителя.

Примеры вариантов

№ варианта	Задания
1	Запросы - Определить ФИО кладовщика, принявшего наибольшее количество проката. - Определить наименование проката с минимальной ценой за весь период. - Определить общую стоимость оприходованного проката в заданном месяце. - Определить общий приход проката в разрезе наименований и кладовщиков. - Определить общую стоимость прихода в разрезе проката и кладовщиков. Формы Добавить вычисляемое поле: $\text{Стоимость_прихода} = \text{Цена_за_тонну} * \text{Приход_тонн}$. Отчёты Добавить вычисляемое поле: Стоимость_прихода. В верхний колонтитул – рисунок, в примечание – дату и фамилию.
2	Запросы - Определить экспедитора с максимальным количеством отгрузок. - Найти изделие с максимальной ценой за весь период. - Общая стоимость отгруженных изделий в заданном месяце. - Общее количество отгрузок в разрезе изделий и экспедиторов. - Общая стоимость отгрузок в разрезе изделий и экспедиторов. Формы Добавить вычисляемое поле: $\text{Стоимость_отгрузки} = \text{Цена} * \text{Отгружено_штук}$. Отчёты Добавить вычисляемое поле – $\text{Стоимость отгрузки}$. В верхний колонтитул – рисунок, в примечание – дату и фамилию.

Контрольные вопросы

- Приведите примеры нескольких моделей данных и дайте их краткое описание.
- В чем основные преимущества и недостатки реляционной модели данных?

3. Опишите основные виды межтабличных связей в реляционной БД.
4. Дайте определений понятий «первичный ключ» и «внешний ключ». Приведите пример.
5. Что подразумевается под целостностью данных (структурной, ссылочной)?
6. Перечислите допустимые свойства полей таблиц базы данных (на примере СУБД Microsoft Access)).
7. Перечислите базовые типы данных (на примере СУБД Microsoft Access).
8. Приведите краткое описание основных объектов базы данных (на примере СУБД Microsoft Access).
9. Какое влияние на алгоритм удаления записей из таблиц БД оказывает опция «каскадное удаление связанных полей»?
10. Какое влияние на алгоритм обновления записей в таблицах БД оказывает опция «каскадное обновление связанных полей»?
11. Приведите пример отношения, не ключевой атрибут которого находится в частичной функциональной зависимости от первичного ключа.
12. Приведите пример отношения, не ключевой атрибут которого находится в транзитивной зависимости от первичного ключа.
13. Каким требованиям должны удовлетворять отношения, находящиеся в 1 нормальной форме (1 НФ), 2 НФ и 3НФ?

Лабораторная работа «Работа с базой данных. Язык SQL» (УК-1-32, УК-1-33, УК-1-У3, УК-1-В1, УК-1-В2, УК-1-В3, ОПК-6-32, ОПК-6-У2, ОПК-6-В3, УК-3-В1, УК-3-У3)

Практическое задание

Задача 1. Репродуктивный уровень.

1. Создайте указанную в варианте таблицу и заполните ее исходными данными (не менее 5 кортежей).
2. Создайте указанные в варианте задания запросы SQL:
 - 2.14 запроса на изменение данных (добавление, удаление, обновление, создание таблиц);
 - 2.2 5 запросов на выборку данных, соответствующих заданным критериям.

Примеры вариантов

№ варианта	Задания																																										
1	База данных «Серверное оборудование»: уникальный идентификатор сервера; название/модель сервера; IP-адрес сервера; установленная ОС; количество ядер процессора; объём оперативной памяти (ГБ). Пример таблицы. <table><tr><th>id_сервера</th><th>название</th><th>ip_адрес</th><th>операционная система</th><th>количество ядер</th><th>объём ram</th><th>дата установки</th></tr><tr><td>1</td><td>Dell PowerEdge R740</td><td>192.168.1.10</td><td>Ubuntu 22.04 LTS</td><td>16</td><td>64</td><td>10.05.2023</td></tr><tr><td>2</td><td>HP ProLiant DL380</td><td>192.168.1.20</td><td>Windows Server 2022</td><td>12</td><td>48</td><td>15.08.2023</td></tr><tr><td>3</td><td>IBM Power S822LC</td><td>192.168.1.30</td><td>Red Hat 9.0</td><td>20</td><td>128</td><td>20.01.2024</td></tr><tr><td>4</td><td>Supermicro 6029P</td><td>192.168.1.40</td><td>Ubuntu 20.04 LTS</td><td>8</td><td>32</td><td>01.11.2022</td></tr><tr><td>5</td><td>Cisco UCS B200 M5</td><td>192.168.1.50</td><td>Windows Server 2019</td><td>10</td><td>64</td><td>05.12.2023</td></tr></table> Запросы на изменение данных: - Удалить из базы данных информацию о сервере с указанным IP-адресом. - Изменить для данного сервера (по id сервера) операционную систему. - Увеличить объём оперативной памяти на 16 ГБ для всех серверов с количеством ядер менее 12. - Создать новую таблицу Серверы_Бэкап, содержащую id_сервера, название и ip_адрес серверов с объёмом RAM ≥ 64 ГБ. Запросы на выборку данных: - Выбрать название, ip_адрес и операционную систему сервера по указанному id_сервера. - Выбрать все серверы с объёмом RAM ≥ 64 ГБ, отсортированные по количеству ядер (по убыванию). - Выбрать серверы с операционной системой Linux (Ubuntu, Red Hat), у которых количество ядер больше 10. - Выбрать названия и ip-адреса серверов, установленных в 2024 году. - Подсчитать общее количество ядер у всех серверов под управлением Windows.	id_сервера	название	ip_адрес	операционная система	количество ядер	объём ram	дата установки	1	Dell PowerEdge R740	192.168.1.10	Ubuntu 22.04 LTS	16	64	10.05.2023	2	HP ProLiant DL380	192.168.1.20	Windows Server 2022	12	48	15.08.2023	3	IBM Power S822LC	192.168.1.30	Red Hat 9.0	20	128	20.01.2024	4	Supermicro 6029P	192.168.1.40	Ubuntu 20.04 LTS	8	32	01.11.2022	5	Cisco UCS B200 M5	192.168.1.50	Windows Server 2019	10	64	05.12.2023
id_сервера	название	ip_адрес	операционная система	количество ядер	объём ram	дата установки																																					
1	Dell PowerEdge R740	192.168.1.10	Ubuntu 22.04 LTS	16	64	10.05.2023																																					
2	HP ProLiant DL380	192.168.1.20	Windows Server 2022	12	48	15.08.2023																																					
3	IBM Power S822LC	192.168.1.30	Red Hat 9.0	20	128	20.01.2024																																					
4	Supermicro 6029P	192.168.1.40	Ubuntu 20.04 LTS	8	32	01.11.2022																																					
5	Cisco UCS B200 M5	192.168.1.50	Windows Server 2019	10	64	05.12.2023																																					
2	База данных «Электронная библиотека IT-литературы»: международный номер книги; название книги; автор(ы) книги; издательство; год издания; количество страниц; формат (PDF, EPUB, FB2, DOCX); доступна для скачивания (TRUE/FALSE).																																										

	Пример таблицы.							
	isbn	название	автор	издательство	год издания	количество страниц	формат	доступна
	978-5-604-12345-6	Python для начинающих	Иванов И.И.	Издательство IT	2023	350	PDF	TRUE
	978-5-604-54321-0	Базы данных SQL	Петров П.П.	ТехЛит	2022	280	EPUB	TRUE
	978-5-604-98765-4	Алгоритмы и структуры данных	Сидоров С.С.	Наука	2024	420	FB2	FALSE
	978-5-604-11111-1	Веб-разработка на JavaScript	Кузнецова А.А.	Издательство IT	2023	310	PDF	TRUE
	978-5-604-22222-2	Кибербезопасность	Смирнов В.В.	ТехЛит	2022	390	DOCX	FALSE
Запросы на изменение данных: 1. Удалить из библиотеки книгу по указанному ISBN. 2. Изменить для данной книги (по ISBN) формат на PDF. 3. Установить значение доступна (TRUE) для всех книг издательства «Издательство IT», изданных после 2022 года. 4. Создать новую таблицу Новые_Книги, содержащую isbn, название и автора книг, изданных в 2024 году. Запросы на выборку данных: 1. Выбрать все сведения о книге по указанному ISBN. 2. Выбрать названия и авторов книг, доступных для скачивания (доступна = TRUE). 3. Выбрать книги с количеством страниц более 300, отсортированные по году издания (сначала новые). 4. Выбрать названия, авторов и форматы книг издательства «Издательство IT», изданных в 2022–2024 годах. 5. Подсчитать количество книг каждого формата в библиотеке.								

Задача 2. Реконструктивный уровень.
Выберите указанную в варианте информацию, используя агрегатные функции и подзапросы в запросах SQL.

Примеры вариантов

№ варианта	Задания
1	1. Определить количество серверов с операционной системой Ubuntu. 2. Определить название, IP-адрес и объём RAM сервера с максимальным количеством ядер. 3. Определить название, IP-адрес и количество ядер сервера с минимальным объёмом RAM. 4. Определить общее количество ядер у всех серверов под управлением Windows. 5. Выбрать названия, IP-адреса и операционные системы серверов, установленных в том же году, что и сервер с идентификатором 3.
2	1. Определить количество книг издательства «Издательство IT». 2. Определить название, автора и количество страниц книги с максимальным количеством страниц. 3. Определить название, автора и год издания книги с минимальным количеством страниц. 4. Определить общее количество страниц всех книг, доступных для скачивания. 5. Выбрать названия, авторов и форматы книг, изданных в том же году, что и книга с ISBN 978-5-604-12345-6.

Контрольные вопросы

1. Из каких подмножеств состоит SQL?
2. Какой оператор используется для удаления строки в таблице? Приведите пример оператора, реализующего удаление строки.
3. Какой оператор используется для выборки данных из таблицы? Приведите пример оператора, реализующего выборку данных из таблицы.
4. Какой оператор используется для обновления данных в таблице? Приведите пример оператора, реализующего обновление данных в таблице.
5. В чем разница между типом данных CHAR и VARCHAR в SQL?
6. Что такое первичный ключ (Primary key)?
7. Какие ограничения (Constraints) допустимо использовать при создании новой таблицы в базе данных? Приведите несколько примеров.
8. Что такое внешний ключ (Foreign key)?
9. Совпадают ли значения NULL со значениями нуля или пробела?
10. Что такое подзапрос в SQL? Приведите пример запроса, включающего подзапрос.
11. Приведите пример запроса, позволяющего определить:

- количество записей в таблице;
- количество записей, удовлетворяющих заданному критерию;
- количество записей, в каждой группе (при использовании фразы GROUP BY в операторе SELECT).

Лабораторная работа «Программирование разветвляющихся вычислительных процессов» (ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У3, ОПК-6-В2, ОПК-6-В3)

Практическое задание

Задача 1. Репродуктивный уровень.

Для заданного в варианте графика функции $y=f(x)$ требуется разработать программу на языке Python, которая:

- определяет аналитическое выражение для каждого фрагмента графика (прямые линии, дуги окружностей и т.п.) на основе геометрических данных, представленных на рисунке;
- запрашивает у пользователя значение аргумента x (вещественное число);
- определяет, какому интервалу области определения принадлежит x ;
- вычисляет значение y по соответствующей формуле;
- выводит результат на экран с поясняющим текстом (например, « $X=... Y=...$ »).

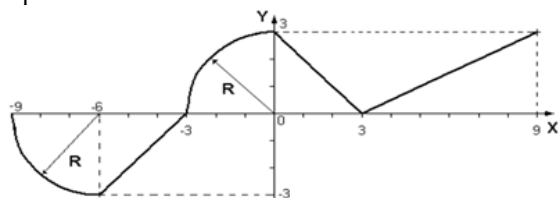
Программа должна:

- содержать все аналитические выражения, полученные при анализе графика (в виде формул в коде);
- проверять область допустимых значений для каждого интервала (включая граничные точки, которые должны принадлежать области определения);
- выдавать сообщение об ошибке, если x не входит в область определения функции (например, «Значение x вне области определения функции»);

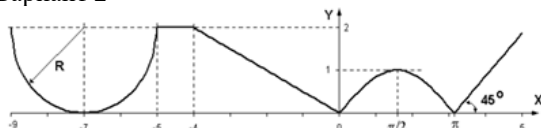
выводить результат с точностью до двух знаков после запятой.

Примеры вариантов

Вариант 1



Вариант 2



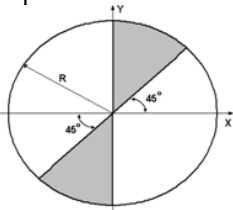
Задача 2. Реконструктивный уровень.

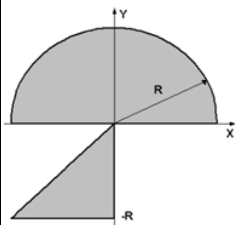
Для заданного в варианте графика (заштрихованной области на координатной плоскости) требуется разработать программу на языке Python, которая определяет, попадает ли точка с заданными координатами в указанную область. Требования к реализации.

- Анализирует геометрическую фигуру (заштрихованную область), представленную на рисунке в варианте задания.
- Определяет аналитическое описание границ области (уравнения прямых, окружностей, парабол и т.п.) на основе данных, представленных на рисунке (координаты точек, радиусы, размеры).
- Запрашивает у пользователя координаты точки – x и y (вещественные числа).
- Проверяет, принадлежит ли точка заштрихованной области, с учётом того, что точки на границе считаются принадлежащими области.
- Выводит результат для каждой точки в виде текстового сообщения: «Точка (x, y) попадает в заданную область» или «Точка (x, y) не попадает в заданную область».

Примеры вариантов

Вариант 1





Контрольные вопросы

1. Перечислите базовые алгоритмические структуры.
2. Приведите краткое описание базовых инструментов системы программирования.
3. Перечислите базовые операции языка Python с указанием их приоритета.
4. Приведите краткое описание типов данных языка Python.
5. Приведите синтаксис и опишите алгоритм работы условного оператора языка Python. Приведите примеры использования различных форм данного оператора (полная, неполная, каскадная).

6. Какие типы констант существуют в Python. Привести примеры констант разных типов.

7. К какому типу относятся константы *192345*, *0x56*, *0xCB*, *016*, *0.7865*, *.0045*, *"one"*, *5*, *5.?*

8. Какие операции называются унарными? Привести примеры.

9. Какие операции называются бинарными? Привести примеры.

10. Изучите пункты приведенного ниже списка и выясните, что означают два числа, записанные после двоеточия перед буквой *f*. Приведите результаты каждого пункта для $x = 16/7$ (с указанием количества выводимых пробелов).

a) `print("x={:10.3f}".format(x))`

b) `print("x={:8.2f}".format(x))`

c) `print("x={:2.2f}".format(x))`

d) `print("x={:0.2f}".format(x))`

e) `print("x={:0.1f}".format(x))`

11. Укажите результат работы следующего кода

```
print("1", end="")
```

```
print("23", end="")
```

```
print(" 456")
```

Поясните ответ.

12. Приведите несколько примеров допустимых и недопустимых идентификаторов. Поясните ответ.

13. В каком из языков программирования память для переменной выделяется автоматически: Паскаль, С, Python, Java.

14. Каким образом транслятор Python определяет тип переменной?

15. Укажите результат работы следующего кода, если, например, пользователь введет значения 12 и 34

```
num1 = input()
```

```
num2 = input()
```

```
summa = num1 + num2
```

```
print(summa)
```

Поясните ответ.

16. Как организовать ввод нескольких значений в одной строке. Приведите пример.

17. В каком порядке выполняются операции, если они имеют одинаковый приоритет?

18. Вычислите значение целочисленной переменной *c* при $a = 26$ и $b = 6$. Поясните ответ.

a) $c = a \% b + b$

b) $c = a // b + a$

c)

$b = a // b$

$c = a // b$

d)

$b = a // b + b$

$c = a \% b + a$

e)

$b = a \% b + 4$

$c = a \% b + 1$

f)

$b = a // b$

$c = a \% (b+1)$

19. Что выведет эта программа при $x = -3$? $x = 0$? $x = 123$? Поясните ответ.

```
if x >= 0:
```

```
    if x > 0: print(1)
```

```
    else: print(0)
```

```
else:
```

```
print(-1)
```

20. Определите порядок операций при определении истинности условия:
`not (a > 10) or not (a < 10) and (a < b)`

Определите, истинно или ложно это выражения при $a = 5$, $b = 10$. Поясните ответ.

21. Напишите условное выражение (тернарный оператор), которое присваивает переменной `status` значение "positive", если число n больше нуля, "negative", если n меньше нуля, и "zero" в противном случае. Можно ли записать такую логику одним тернарным оператором?

22. Напишите фрагмент кода, который запрашивает у пользователя оценку (целое число от 0 до 100) и выводит соответствующую буквенную оценку по правилу:

90–100 → "A"

80–89 → "B"

70–79 → "C"

60–69 → "D"

ниже 60 → "F"

Используйте конструкцию `if-elif-else`. Обработайте случай ввода числа вне диапазона (вывести сообщение об ошибке).

23. Что выведет код? Как можно переписать ту же логику с помощью `if-elif-else`? В чём преимущество `match-case` для таких структур?

```
value = (1, 2)
match value:
    case (0, 0):
        print("Начало координат")
    case (x, 0):
        print(f"На оси X, x={x}")
    case (0, y):
        print(f"На оси Y, y={y}")
    case (x, y):
        print(f"Точка ({x}, {y})")
```

24. Опишите, что произойдёт при выполнении кода в Python:

```
if value:
    print("Истинно")
else:
    print("Ложно")
```

при следующих значениях переменной:

a) `value = 0`

b) `value = []`

c) `value = ""`

d) `value = None`

f) `value = "Hello"`

g) `value = [1, 2]`

Какие значения в Python считаются False?

25. Сравните два фрагмента кода и ответьте на вопрос: в каких случаях результаты выполнения будут различаться? Приведите пример значений условий, при которых разница проявится.

```
# Фрагмент А
if condition1:
    блок_кода1
elif condition2:
    блок_кода2
elif condition3:
    блок_кода3
```

```
# Фрагмент Б
if condition1:
    блок_кода1
if condition2:
    блок_кода2
if condition3:
    блок_кода3
```

Практическое задание

Задача 1.1. Репродуктивный уровень.

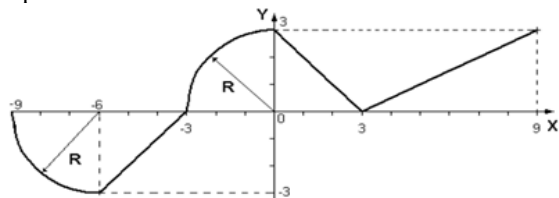
Для заданной в варианте функции $y = f(x)$, промежутка $[a; b]$ и шага h требуется разработать программу на языке Python, которая выполняет табулирование функции – вычисляет и выводит на экран значения функции для всех значений аргумента x из указанного диапазона с заданным шагом.

Требования к программе:

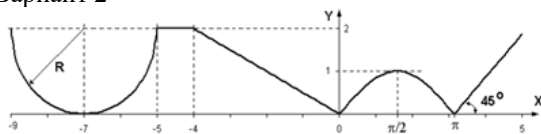
- запрашивает у пользователя:
 - левую границу интервала a ;
 - правую границу интервала b ;
 - шаг табулирования h .
- вычисляет значения функции $y=f(x)$ для каждого x , начиная с a и заканчивая b (включительно), увеличивая x на h на каждой итерации;
- выводит результаты в виде таблицы, содержащей:
 - столбец значений аргумента x ;
 - столбец соответствующих значений функции y ;
 - таблица должна быть отформатирована (выравнивание по правому краю, фиксированное число знаков после запятой, наличие «шапки» и разделительных линий).
- запрещается использование списков, функций, библиотек (кроме `math` при необходимости) – только базовые конструкции: ввод/вывод, арифметические операции, циклы и условные операторы.

Примеры вариантов

Вариант 1



Вариант 2



Задача 1.2. Репродуктивный уровень.

Для заданной в варианте последовательности $\{a_i\}_{i=1}^{\infty}$ требуется разработать программу на языке Python, вычисляющую значение члена последовательности a_i , для которого выполняется условие $|a_i - a_{i-1}| < \varepsilon$, где $\varepsilon = 0.0001$. В качестве результата на экран следует вывести номер требуемого члена последовательности, его значение, а также значение предыдущего члена последовательности.

Примеры вариантов

№ варианта	Задания
1	$a_i = 1/(i^2 + 2i + 10)$
2	$a_i = 1/(i^3 + 3i + 2)$

Задача 2. Реконструктивный уровень.

Для заданного числового отрезка $[A; B]$ (где A и B — целые положительные числа, $A < B$) требуется найти числа, удовлетворяющие определённому критерию, указанному в варианте. Требуется разработать программу на языке Python, которая:

- перебирает все целые числа из заданного интервала $[A; B]$ (включая границы);
- проверяет каждое число на соответствие критерию, указанному в варианте.
- выполняет вычисления согласно варианту:
 - подсчёт количества найденных чисел;
 - нахождение суммы, среднего арифметического, максимального или минимального значения;
 - вычисление суммы цифр, произведения цифр, суммы делителей и т.п.;
 - вывод найденных чисел с дополнительной информацией (делители, порядковые номера, суммы цифр и т.д.).
- выводит результат на экран в формате, указанном в варианте (одна строка, таблица, список с пояснениями).

Требования к реализации.

- Использовать циклическую конструкцию (for или while) для перебора чисел в заданном диапазоне.
- Использовать условные операторы (if-elif-else) для проверки критериев.
- Запрещается использование сторонних библиотек (NumPy, SymPy и т.п.) – только стандартные средства Python.
- Обрабатывать особые случаи:
 - если подходящих чисел нет – выводить соответствующее сообщение;
 - если требуется среднее арифметическое, а чисел нет – выводить сообщение об отсутствии данных;
 - корректно обрабатывать границы интервала (включительно).
- Оптимизация: при работе с большими диапазонами (например, до 1 300 000) следует использовать эффективные алгоритмы проверки делителей (например, проверка до $\sqrt{n}$).

Примеры вариантов

№ варианта	Задания
1	Среди целых чисел, принадлежащих числовому отрезку [264871; 322989], найдите числа, которые представляют собой произведение двух различных простых делителей, заканчивающихся на одну и ту же цифру. Запишите в ответе количество таких чисел и их среднее арифметическое.
2	Среди целых чисел, принадлежащих числовому отрезку [1686; 13276], найдите числа, все цифры которых нечетные. Ответом будет сумма цифр найденных чисел.

Контрольные вопросы

1. Приведите синтаксис и опишите алгоритм работы цикла с предусловием языка Python. Приведите примеры использования различных модификаций данного оператора.
2. Приведите синтаксис и опишите алгоритм работы цикла с параметром языка Python. Приведите примеры использования различных модификаций данного оператора.
3. Перечислите операторы прерывания циклических вычислительных процессов, опишите алгоритм их работы. Приведите примеры.

4. В каком случае программа, содержащая цикл с предусловием, может заиклиться?

5. В каком случае цикл с предусловием не выполняется ни разу?

6. В каких случаях цикл с параметром не выполняется ни разу?

7. Верно ли, что любой цикл с параметром можно заменить циклом с условием? Верно ли обратное утверждение?

8. В каком случае можно заменить цикл с предусловием на цикл с параметром?

9. Значения переменных a и b равны a = 4 и b = 6. Определите, сколько раз выполнится цикл, и чему будут равны значения этих переменных после его завершения:

a. `while a < b: a += 1`

b. `while a < b: a += b`

c. `while a > b: a = a + 1`

d. `while a < b: b = a - b`

e. `while a < b: b = a - b`

10. Значение переменной a равно 1. Определите, сколько раз выполнится цикл и чему будет равно значение этой переменной после его завершения:

a. `for i in range(3): a += 1`

b. `for i in range(3,0): a += 1`

c. `for i in range(1,3,-1): a += i`

d. `for i in range(3,0,-1): a += i`

11. Опишите пошаговое выполнение приведенных ниже циклов: укажите значения переменных на каждой итерации, объясните условие остановки цикла.

a.

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

b.

```
k = 1
for i in range(1,6):
    print( k * k, end = " " )
    k += 2
```

c.

```
k = 8
for i in range(5,0,-1):
```

```
print( 2 * i - k, end = " " )
k -= 2
```

d.

```
x = 1
for i in range(1, 6):
    x = x * i + (i % 2)
print(x, end = " ")
```

12. Напишите фрагмент кода, содержащего циклическое вычисление суммы квадратов натуральных чисел в интервале от a до b.

13. Выполнится ли в приведенном ниже коде блок *else*. Поясните ответ. В каких случаях в циклах полезен блок *else*? Приведите пример.

```
i = 0
while i < 5:
    print(i, end=' ')
    if i == 3:
        break
    i += 1
else:
    print("else")
```

14. Напишите фрагмент кода с одним циклом *while*, который вычисляет сумму цифр целого положительного числа *num* (например, для 12345 сумма = 15). Запрещается преобразовывать число в строку.

15. В чём заключается ошибка в следующем цикле? Как её исправить, чтобы программа вывела числа от 1 до 10?

```
i = 1
while i <= 10:
    print(i)
```

16. Укажите количество итераций, которые проделает данный цикл. Объясните ответ.

```
n = 0
s = 0
while s <= 35:
    s = s + 4
    n = n + 1
print(n)
```

17. Определите, при каком наименьшем положительном введённом значении переменной *s* программа выведет число *s* без изменения его значения. Объясните ответ.

```
s = int(input())
n = 40
while s + n < 100:
    s = s + 25
    n = n - 5
print(s)
```

18. Определите, при каком наименьшем введённом значении переменной *s* программа выведет число 10. Объясните ответ.

```
s = int(input())
n = 0
while s < 1000:
    s = s * 2
    n = n + 5
print(n)
```

Лабораторная работа «Одномерные коллекции Python: списки и генераторы» (ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У1, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3)

Практическое задание

Задача 1. Репродуктивный уровень.

Сформировать одномерный список, состоящий из *N* целых неотрицательных чисел. Количество элементов списка (*N*) и их значения вводятся пользователем. Требуется выполнить указанные в варианте операции. Алгоритм решения задачи должен быть реализован двумя способами:

- 1) без использования встроенных функций и методов списков Python. Все действия (поиск, подсчёт, вставка, удаление, перебор) реализовать через циклы и условные операторы;
- 2) с использованием всех допустимых методов списков (*len*, *in*, *count*, *index*, *insert*, *pop*, *max*, *min*, *sum*), генераторов списков и т.д.

Примеры вариантов	
№ варианта	Задания
1	1. Проверить, являются ли все элементы списка трёхзначными. 2. Проверить, содержится ли в списке хотя бы один элемент, оканчивающийся 5. 3. Вычислить среднее арифметическое нечётных элементов, сумма цифр которых кратна 3. 4. Вставить на позицию с индексом 5 (шестой элемент) сумму минимального и максимального элементов исходного списка. 5. Удалить последний элемент списка и вывести его значение. 6. Найти максимальный трёхзначный элемент, кратный 7, но не кратный 14. 7. Найти минимальный трёхзначный элемент, у которого сумма цифр больше 15. 8. Поменять местами найденные максимальный (из п.6) и минимальный (из п.7) элементы. 9. Проверить, содержит ли список число, введённое пользователем. 10. Подсчитать, сколько раз в списке встречается число, введённое пользователем.
2	1. Проверить, все ли элементы списка четные. 2. Проверить, содержится ли в списке хотя бы один элемент, кратный 8. 3. Вычислить среднее арифметическое элементов, у которых первая цифра равна последней (для двузначных элементов списка). 4. Вставить на позицию с индексом 3 (четвертый элемент) модуль разности минимального и максимального элементов исходного списка. 5. Удалить последний элемент списка и вывести его значение. 6. Найти максимальный элемент, который делится на 4, но не делится на 8. 7. Найти минимальный элемент, у которого цифра в младшем разряде не менее 5. 8. Поменять местами найденные максимальный (из п.6) и минимальный (из п.7) элементы. 9. Проверить, содержит ли список число, введённое пользователем. 10. Подсчитать, сколько раз в списке встречается число, введённое пользователем.

Задача 2. Реконструктивный уровень.

В каждом варианте задан набор операций, которые необходимо выполнить над одномерным списком целых чисел. Список генерируется случайным образом (значения в диапазоне, например, от 0 до 10000). Требуется разработать программу на языке Python, которая реализует указанную последовательность действий и выводит результаты в удобочитаемом виде.

Примеры вариантов

№ варианта	Задания
1	Дан список из N случайных чисел. Требуется: 1. удалить все вхождения элемента X; 2. вставить новый элемент Y на позицию с индексом, равным первому вхождению ранее удаленного элемента X в исходный список; 3. определить, упорядочены ли элементы преобразованного списка в порядке возрастания (используя функцию <code>all()</code>); 4. вывести итоговый список и его отсортированную версию (порядок сортировки указывает пользователь). <i>Значения X, Y, N – определяются пользователем при запуске программы.</i> В результатах тестирования программы, приводимых в отчете, должны быть представлены тесты всех ветвей работы алгоритма программы.
2	Дан список из N случайных чисел. Требуется: 1. найти разницу между максимальным и минимальным элементами списка; 2. удалить все элементы, превосходящие эту разницу; вывести количество удаленных элементов; 3. определить, упорядочены ли элементы преобразованного списка в порядке убывания (используя функцию <code>all()</code>); 4. вывести итоговый список и его отсортированную версию (порядок сортировки указывает пользователь). <i>Значение N – определяются пользователем при запуске программы.</i> В результатах тестирования программы, приводимых в отчете, должны быть представлены тесты всех ветвей работы алгоритма программы.

Контрольные вопросы

1. Что такое массив?
2. Как сгенерировать целочисленный одномерный массив, состоящий из 10 элементов, образующих арифметическую прогрессию, первый элемент которой равен 1, а шаг 2?
3. Как написать следующее выражение «Второму элементу массива *array* присвоено значение 10»?
4. Дан список: `arr = [5, 10, 15, 20, 25, 30]`. Что вернут следующие выражения?
 - a) `arr[2]`
 - b) `arr[-1]`
 - c) `arr[1:4]`
 - d) `arr[::2]`
 - e) `arr[::]`
5. Что выводит на экран представленный ниже фрагмент кода:

```
x = [i for i in range(10, 31, 1)]
for i in range(21):
    if x[i] % 5: continue
    print(x[i], end = " ")
```

Ответ следует обосновать.

6. Что произойдет с массивом `[1, 2, 3, 4]` ($N = 4$) при выполнении следующего фрагмента программы:
 - a)

```
for i in range(N-1):
    A[i] = A[i+1]
```
 - b)

```
for i in range(N-1):
    A[i+1] = A[i]
```

Ответ следует обосновать.

7. Чем отличаются методы `append()` и `extend()`? Что выведет следующий код? Ответ следует обосновать.

- a)

```
a = [1, 2]
a.append([3, 4])
print(a)
```
- b)

```
b = [1, 2]
b.extend([3, 4])
print(b)
```

8. Объясните, что такое генератор списка (list comprehension). Преобразуйте следующий цикл в генератор списка:

```
arr = []
for x in range(10):
    if x % 2:
        arr.append(x**2)
    else:
        arr.append(x**3)
```

9. Опишите преимущества использования генераторов списков.
10. К каких случаям генераторы списков использовать не следует?
11. Объясните принципиальное отличие двух ниже приведенных фрагментов кода:

```
squares_list = [x**2 for x in range(10000000)]

squares_gen = (x**2 for x in range(10000000))
```

12. Дан список чисел *nums*. Как с помощью генератора списка создать новый список, содержащий только положительные числа из *nums*?
13. Напишите одну строку кода, предназначенную для ввода с клавиатуры последовательности целых чисел, разделённых пробелами и преобразования их в список целых чисел.
14. Напишите одну строку кода, которая подсчитывает и выводит на экран количество четных чисел в произвольном списке *numbers* (используя генератор списка и функцию `len()`).
15. Опишите назначение и приведите пример использования функции `enumerate()`.
16. Чем список (list) в Python принципиально отличается от массивов в таких языках, как C++ или Java?
17. В чем состоит отличие между `sort()` и `sorted()`?
18. Объясните смысл фразы «функция `sorted()` гарантирует устойчивость».
19. Приведите пример сортировки строк по их длине.
20. Напишите фрагмент кода на Python, который с помощью одного цикла находит **второе по величине** число в списке целых чисел (без использования `sort`, `max`, `heapq` и готовых функций). Список может содержать дубликаты. Обработайте случаи, когда второго максимума не существует (длина списка < 2 или все элементы одинаковы).

Практическое задание

Задача 1. Репродуктивный уровень.

Требуется разработать программу на языке Python, которая:

- запрашивает у пользователя:
 - количество строк N и количество столбцов M матрицы (целые положительные числа);
 - диапазон значений элементов матрицы $[a,b]$ (целые числа, $a < b$).
- генерирует целочисленную матрицу размером $N \times M$, заполняя её случайными числами из заданного диапазона $[a,b]$ (для генерации используется функция `randint(a,b)` из модуля `random`);
- выводит исходную матрицу на экран в табличном виде (каждая строка матрицы выводится на отдельной строке экрана);
- выполняет обработку матрицы согласно варианту задания:
 - вычисляет некоторый агрегатный показатель (среднее арифметическое, сумму, количество и т.п.) для элементов, удовлетворяющих заданному условию (например, положительные двузначные, кратные 3, не оканчивающиеся на 1 и т.п.);
 - производит преобразование исходной матрицы (заменяет элементы, удовлетворяющие определённому условию, на вычисленное значение или на его целую часть).
- выводит преобразованную матрицу на экран в том же табличном виде.

Требования к реализации. Код должен быть реализован в двух вариантах.

Вариант А (с использованием списков Python)

- Матрица создаётся как список списков.
- Для заполнения, обработки и вывода матрицы используются вложенные циклы.
- Для проверки условий и замены элементов используются условные операторы.
- Все вычисления выполняются вручную, без использования функций `sum()`, `max()`, `min()`.
- Запрещается использование сторонних библиотек (NumPy, Pandas и т.п.).
- Вывод матрицы осуществляется построчно с форматированием, например, `"{:5d}".format(value)`.

Вариант Б (с использованием библиотеки NumPy)

- Матрица создаётся с помощью функции `np.random.randint(a, b+1, (N, M))`.
 - Для работы с матрицей используются векторизованные операции, а не явные циклы.
 - Для фильтрации элементов по условию используется булева индексация (маски).
 - Для вычисления агрегатных показателей применяются методы массивов NumPy: `.mean()`, `.sum()`, `np.where()`.
 - Преобразование матрицы выполняется с помощью `np.where()` или других векторных операций.
- Вывод матрицы осуществляется с помощью `print(matrix)` или с использованием методов форматирования NumPy.

Примеры вариантов

№ варианта	Задания
1	Требуется вычислить среднее арифметическое всех положительных четных элементов матрицы, не оканчивающихся на 0. Заменить все отрицательные элементы исходной матрицы полученным значением.
2	Требуется вычислить среднее арифметическое всех положительных элементов матрицы, кратных 5, но не кратных 10. Заменить все двухзначные элементы исходной матрицы полученным значением.

Задача 2. Реконструктивный уровень.

В каждом варианте задана прямоугольная (в некоторых случаях квадратная) матрица – двумерный массив чисел, отражающий данные некоторой предметной области (температуры, оценки, расстояния, схема проезда, показания датчиков, изображение, характеристики комплектующих, транзакции, результаты ЕГЭ, показатели металлургического комбината). Матрица задана явно в виде списка списков с конкретными числовыми значениями. Требуется разработать программу на языке Python, которая выполняет последовательность операций над этой матрицей в соответствии с вариантом задания. Типовые операции включают:

- Вычисление агрегатных характеристик по всей матрице, по строкам или по столбцам: среднее арифметическое, сумма, максимум, минимум, медиана, произведение, дисперсия и т.п.
- Поиск элементов с заданными свойствами (например, минимальный элемент, максимальный элемент) с указанием их координат (индексов строки и столбца).
- Преобразование матрицы:
 - замена элементов по условию (например, все значения ниже среднего заменить на 0);
 - транспонирование;
 - поворот на 90° (по или против часовой стрелки);
 - удаление строк или столбцов;
 - добавление новых строк или столбцов;
 - нормировка (по строкам или столбцам);

- бинаризация;
- сглаживание и другие операции.
- Построение новых структур на основе исходной матрицы: векторов (списков) из строк или столбцов, новых матриц (например, нормированная, бинарная, транспонированная).
- Вывод результатов каждого шага в понятном формате с пояснениями.

Требования к реализации.

- Программа должна использовать только стандартные средства Python (списки, циклы, условные операторы) или библиотеку NumPy для векторизованных операций. Выбор инструмента не ограничен (если иное не указано в варианте).
- Исходная матрица задаётся в коде как константа (ввод с клавиатуры не требуется).
- Все вычисления и преобразования должны быть реализованы алгоритмически – с использованием вложенных циклов (для списков) или векторизованных операций (для NumPy).
- Вывод результатов должен быть понятным: указывать, что именно вычисляется, и отображать промежуточные и итоговые данные в удобочитаемом формате (матрицы – в табличном виде, векторы – в строку).
- Обращать особые случаи: если операция невозможна (например, удаление несуществующего столбца), программа должна выдавать соответствующее сообщение.

Примеры вариантов

Вариант 1

Температурная матрица:

- строки – 5 городов: Москва, Санкт-Петербург, Новосибирск, Екатеринбург, Казань.
- столбцы – 5 дней недели (понедельник – пятница).
- значения – среднедневная температура (°C) в соответствующем городе и день.

Исходная матрица:

```
[[22, 24, 23, 21, 20],
 [18, 19, 20, 19, 18],
 [25, 26, 27, 28, 26],
 [15, 16, 15, 14, 15],
 [30, 31, 29, 30, 32]]
```

На основании данных исходной матрицы требуется:

1. вычислить среднюю температуру по всем городам за всю неделю (в целом по матрице);
2. найти, в каком городе и в какой день была самая низкая температура за неделю (минимальный элемент). Вывести город и день;
3. для каждого дня определить город с максимальной температурой в этот день (максимум по столбцу);
4. заменить все температуры, которые ниже среднего значения по матрице, на 0 (моделирование аномально холодных дней, которые не учитываются);
5. транспонировать матрицу: получить таблицу, где строки – дни, столбцы – города;
6. найти сумму температур на главной диагонали (соответствует температуре в каждом городе в «свой» день: Москва в понедельник, СПб во вторник и т.д.);
7. найти произведение температур на побочной диагонали (Москва в пятницу, СПб в четверг, ..., Казань в понедельник).

Вариант 2

Успеваемость студентов:

- Строки – 6 студентов (студент 1 ... студент 6).
- Столбцы – 4 предмета: математика, физика, программирование, английский язык.
- Значения – оценки по 5-балльной шкале.

Исходная матрица:

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

На основании данных исходной матрицы требуется:

1. для каждого студента вычислить средний балл. Также для каждого студента вывести список предметов, по которым его оценка ниже среднего балла по этому предмету в группе;
2. определить, есть ли студент, у которого все оценки отличные;
3. найти предмет с самым высоким средним баллом в группе и предмет с самым низким средним баллом;
4. поменять местами оценки студента с минимальным средним баллом и студента с максимальным средним баллом;
5. удалить из списка студента с самой низкой суммой баллов;
6. добавить новый столбец – новый предмет «физкультура» – со значениями [4,5,3,4,5,4] в конец матрицы;
7. повернуть матрицу на 90° по часовой стрелке: теперь строки – предметы, столбцы – студенты.

Контрольные вопросы

1. Для чего в коде программы лабораторной работы присутствует проверка `if len(suitable) == 0`? Что произойдёт, если её убрать и подходящих элементов не окажется? Напишите, какую ошибку вызовет программа в таком случае.
2. Что выводит на экран представленный ниже фрагмент кода

```
A = [[1, 0, 1], [-1, 0, 1], [0, 1, 1]]
summa = 0
for row in A:
    summa += sum(row)
print(summa)
```

Ответ следует обосновать.

3. Опишите, какие элементы матрицы будут выбраны посредством приведенной ниже маски

```
mask = (matrix >= 10) & (matrix < 100) & \
       (matrix % 3 == 0) & (matrix % 10 != 1)
```

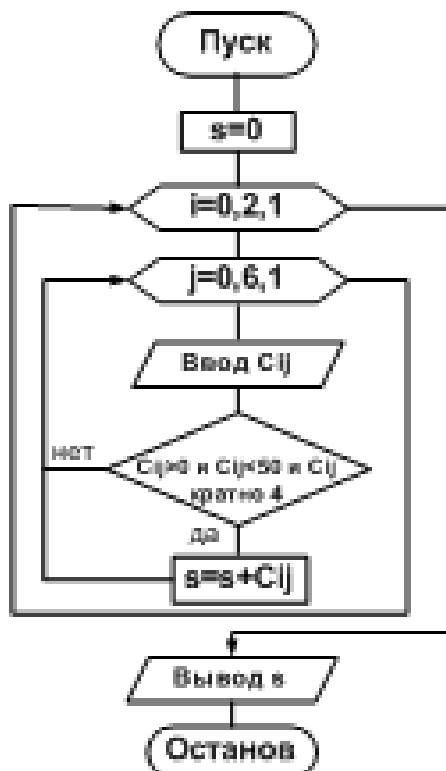
Почему здесь используются операторы `&`, а не логический `and`?

4. Сравните два способа обхода всех элементов матрицы M размера $n \times m$. Какие преимущества и недостатки у каждого способа? В каком случае способ 1 остаётся единственным возможным?

```
# Способ 1
for i in range(len(M)):
    for j in range(len(M[i])):
        print(M[i][j])
```

```
# Способ 2
for row in M:
    for elem in row:
        print(elem)
```

5. Что вычисляет алгоритм, представленный следующей блок-схемой.



6. Что выводит на экран представленный ниже фрагмент кода:

```
A = []
for i in range(3):
    A.append([0]*4)
print(A)
```

Ответ следует обосновать.

7. Что выводит на экран представленный ниже фрагмент кода:

```
A = [[0]*5 for i in range(2)]  
print(A)
```

Ответ следует обосновать.

8. Объясните назначение представленного ниже фрагмента кода:

```
from random import randint  
A = [[0]*3 for i in range(2)]  
for i in range(2):  
    for j in range(3):  
        A[i][j] = randint(20,80)  
print(A)
```

9. Что выводит на экран представленный ниже фрагмент кода:

```
A = [[0]*5 for i in range(3)]  
for i in range(3):  
    print(len(A[i]))  
print(len(A))
```

Ответ следует обосновать.

10. Объясните назначение представленного ниже фрагмента кода:

```
from random import randint  
A = [[0]*3 for i in range(2)]  
for i in range(2):  
    for j in range(3):  
        A[i][j] = randint(20,80)  
print(A)
```

11. Объясните назначение представленного ниже фрагмента кода:

```
for i in range(3):  
    A[i][2], A[i][4] = A[i][4], A[i][2]
```

12. Какие результаты выведет на экран приведенный ниже фрагмент кода. Поясните ответ.

```
X = np.array([[1, 1, 1],[5, 5, 5],[7, 7, 7]])  
print(X.sum())  
print(X.sum(axis = 0))  
print(X.sum(axis = 1))
```

13. Объясните принцип работы функции

```
np.where(matrix < 0, 1000, matrix)
```

14. Почему суммирование элементов главной диагонали матрицы требует одиночного цикла, а суммирование элементов под главной диагональю – вложенного?

15. Дан фрагмент кода, который создаёт «единичную матрицу» размером $n \times n$ (на главной диагонали 1, остальные 0). Допишите пропущенные строки.

```
n = 4  
eye = [[0] * n for _ in range(n)]  
for i in range(n):  
    # здесь нужно добавить код
```

16. Как изменится код в вопросе 14, если нужно заполнить не главную, а побочную диагональ?

17. Что произойдёт при выполнении следующих операций с двумерным списком? Объясните, почему B и C не являются независимыми копиями.

```
A = [[1, 2], [3, 4]]  
B = A[:]  
C = [row[:] for row in A]  
B[0][0] = 99  
C[1][1] = 88  
print(A)  
print(B)  
print(C)
```

18. Дан двумерный список размером $n \times m$. Напишите циклы, которые вычисляют среднее арифметическое каждой строки и сохраняют эти средние в отдельный список `row_averages`. Как изменить код, чтобы вычислить среднее по каждому столбцу?

Практическое задание

Задача 1.1. Репродуктивный уровень.

Требуется разработать программу на языке Python, которая позволяет пользователю:

1. Ввести с клавиатуры одномерный список целых чисел (сначала количество элементов, затем сами элементы).
2. Выполнять над этим списком следующие операции (каждая операция выбирается из меню):
 - Вывод текущего списка на экран.
 - Проверка, все ли элементы удовлетворяют заданному критерию. Результат – «Да» или «Нет».
 - Проверка, содержит ли список хотя бы один элемент, удовлетворяющий заданному критерию. Результат – «Да» или «Нет».
 - Вычисление среднего арифметического элементов, удовлетворяющих заданному критерию. Если таких элементов нет – вывести сообщение.
 - Поиск максимального элемента, удовлетворяющего заданному критерию. Вывести значение и его индекс (первое вхождение). Если элемента нет – сообщение.
 - Поиск минимального элемента, удовлетворяющего заданному критерию. Вывести значение и его индекс. Если элемента нет – сообщение.
 - Обмен местами найденных максимального (из предыдущего пункта) и минимального (из предыдущего пункта) элементов. Если хотя бы один из них не найден – вывести сообщение о невозможности обмена.
3. Программа должна быть реализована в виде набора функций (каждая функция решает одну подзадачу). В основной функции **main** организуется бесконечный цикл с меню, вызывающий соответствующие функции. Список хранится в глобальной переменной **arr** внутри **main**.

Обязательные функции:

- **digit_sum(n)** – вычисление суммы цифр целого числа (работает с отрицательными числами).
- **input_list()** – ввод списка (возвращает список).
- **print_list(arr)** – вывод списка.
- **all_criterion(arr)** – проверка, все ли элементы удовлетворяют заданному критерию (использовать **all**).
- **any_criterion(arr)** – проверка наличия элемента, удовлетворяющего заданному критерию (использовать **any**).
- **average(arr)** – вычисление среднего арифметического элементов, удовлетворяющих заданному критерию (возвращает **float** или **None**).
- **max_criterion(arr)** – поиск максимального элемента по условию (возвращает кортеж (значение, индекс) или **(None, -1)**).
- **min_criterion(arr)** – поиск минимального элемента по условию (возвращает кортеж (значение, индекс) или **(None, -1)**).
- **swap_elements(arr, idx1, idx2)** – обмен элементов по индексам.

Требования к реализации.

- Код должен быть структурирован, каждая функция должна иметь строку документации (**docstring**).
- Меню должно содержать пункты, соответствующие всем операциям (включая ввод/обновление списка, вывод списка, проверки, вычисления, поиски, обмен) и пункт выхода.

Обращать случаи, когда подходящие элементы отсутствуют (для среднего, для поиска максимума/минимума).

Примеры вариантов

№ варианта	Задания
1	1. Проверить, являются ли все элементы списка трёхзначными. 2. Проверить, содержится ли в списке хотя бы один элемент, оканчивающийся 5. 3. Вычислить среднее арифметическое нечётных элементов, сумма цифр которых кратна 3. 4. Найти максимальный трёхзначный элемент, кратный 7, но не кратный 14. 5. Найти минимальный трёхзначный элемент, у которого сумма цифр больше 15.
2	1. Проверить, все ли элементы четные. 2. Проверить, содержится ли в списке хотя бы один элемент, кратный 8. 3. Вычислить среднее арифметическое элементов, у которых первая цифра равна последней (для двузначных элементов списка). 4. Найти максимальный элемент, который делится на 4, но не делится на 8. 5. Найти минимальный элемент, у которого цифра в младшем разряде не менее .

Задача 1.2. Репродуктивный уровень.

Требуется выполнить сортировку данного в варианте списка кортежей по нескольким полям, используя функцию **sorted()** и лямбда-функцию. В качестве результата следует вывести три отсортированных списка.

Примеры вариантов

№ варианта	Задания
1	Дан список сотрудников компании в виде кортежей: (имя, отдел, зарплата, стаж). Требуется отсортировать список, используя функцию <code>sorted()</code> и лямбда-функцию, по следующим правилам: 1. по возрастанию зарплата; 2. по убыванию стажа, при равном стаже – по имени (по алфавиту); 3. по сумме зарплата и стажа (чем больше, тем выше). В качестве результата следует вывести три отсортированных списка.
2	Дан список книг в библиотеке в виде кортежей: (название, автор, год_издания, количество_страниц). Требуется отсортировать список, используя функцию <code>sorted()</code> и лямбда-функцию, по следующим правилам: 1. по возрастанию года издания; 2. по убыванию количества страниц, при равном количестве – по названию (по алфавиту); 3. по разности (год_издания - количество_страниц) (чем меньше разность, тем выше). В качестве результата следует вывести три отсортированных списка.

Задача 2.1. Реконструктивный уровень.

Требуется написать код программы, осуществляющей фильтрацию списка чисел с помощью функции `filter()` и лямбда-функции.

Примеры вариантов

№ варианта	Задания
1	Дан список целых чисел. С помощью <code>filter()</code> и лямбда-функции отберите: 1. все числа, кратные 3 и 5 одновременно; 2. все числа, являющиеся простыми (для проверки на простоту написать отдельную функцию); 3. все числа, у которых сумма цифр больше 10.
2	Дан список целых чисел. С помощью <code>filter()</code> и лямбда-функции отберите: 1. все чётные числа, кратные 4; 2. все числа, являющиеся полными квадратами (например, 1, 4, 9, 16...); для проверки написать отдельную функцию; 3. все числа, у которых произведение цифр больше 20.

Задача 2.2. Реконструктивный уровень.

Дан список целых чисел. Требуется с помощью функции `reduce()` из модуля `functools` и лямбда-функции вычислить указанные величины.

Примеры вариантов

№ варианта	Задания
1	Дан список чисел. Вычислите с помощью <code>reduce</code>: 1. произведение всех чисел; 2. сумму только нечётных чисел; 3. наибольшее число в списке.
2	Дан список чисел. Вычислите с помощью <code>reduce</code>: 1. сумму всех чисел, кратных 3; 2. произведение всех положительных чисел; 3. наименьшее число в списке.

Задача 2.3. Реконструктивный уровень.

Требуется написать рекурсивную функцию согласно варианту задания. В теле функции допускается использование только условного оператора `if-elif-else` и рекурсивный вызов функции самой себя, использование циклов `for` и `while` запрещается.

Примеры вариантов

№ варианта	Задания
1	Напишите рекурсивную функцию <i>digit_sum(n)</i>, которая вычисляет сумму цифр целого неотрицательного числа <i>n</i>. <i>Пример: digit_sum(1234) → 10.</i> Запрещается преобразовывать число в строку и использовать циклы.
2	Напишите рекурсивную функцию <i>power(a, n)</i>, которая возводит число <i>a</i> в целую неотрицательную степень <i>n</i>. Запрещается использовать оператор <i>**</i> и циклы.

Контрольные вопросы

1. Что такое функция? В чем смысл использования функций?
2. Какую функцию называют рекурсивной? Приведите пример рекурсивной функции.
3. Что называется интерфейсом функции?
4. В каких случаях необходимо использовать оператор *return* в теле функции?
5. Опишите область действия локальных и глобальных переменных.
6. Дайте определение понятий «формальные параметры» и «фактические параметры». Приведите пример, иллюстрирующий данные понятия.
7. Может ли оператор *return* возвращать более одного значения?
8. Можно ли использовать список в качестве формального параметра функции? Если возможно, приведите пример.
9. Как в функции прочитать и изменить значение глобальной переменной?
10. Напишите процедуру, которая принимает параметр – натуральное число *N* – и выводит на экран линию из *N* символов '- '.
11. Какие функции называются логическими? Зачем они нужны?
12. Напишите функцию, которая вычисляет максимальное из трёх чисел, не используя встроенную функцию *max*.
13. Напишите два варианта функции, вычисляющей факториал натурального числа *N* (с использованием рекурсии, без использования рекурсии).
14. Что выведет следующий код? Объясните, почему получен именно такой результат.

```
def compute(x, y):  
    result = x * y + x - y  
    return result  
compute(5, 3)  
print(result)
```

15. Каким будет результат вызова функции *greet("Иван")*, *greet()*, *greet("Иван", "Привет")*? Объясните механизм подстановки значений по умолчанию.

```
def greet(name, greeting = "Здравствуйте"):  
    return f"{greeting}, {name}!"  
print(greet("Иван"))  
print(greet())  
print(greet("Иван", "Привет"))
```

16. Дан вызов функции *display(a, b, c)*. Что произойдёт при вызове *display(c = 10, a = 5, 20)*? Почему возникнет ошибка? Как правильно передать аргументы?
17. Напишите фрагмент кода, в котором функция возвращает одновременно минимальное, максимальное и среднее арифметическое списка чисел. Продемонстрируйте, как присвоить эти три значения разным переменным при вызове.
18. Как изменится значение глобальной переменной *total* после выполнения функции? Объясните, что произойдёт, если внутри функции написать *total = total + x* без объявления *global total*.

```
total = 100  
def add_to_total(x):  
    global total  
    total = total + x  
  
add_to_total(50)  
print(total)
```

19. Дан список строк *words = ["яблоко", "банан", "апельсин", "груша"]*. Используя *sorted()* и лямбда-функцию, отсортируйте список по последней букве каждого слова. Напишите код и объясните, как работает лямбда-функция в этом случае.
20. Что такое документирующая строка (docstring)? Как ее добавить в функцию? Как посмотреть документацию функции из интерактивной оболочки Python? Приведите пример правильно оформленной функции с docstring.

Лабораторная работа «Обработка структурированных данных с помощью словарей Python» (ОПК-6-32, ОПК-6-33, ОПК-6-34, ОПК-6-У2, ОПК-6-У3, ОПК-6-У4, ОПК-6-В2, ОПК-6-В3, ОПК-6-В4, УК-3-В1, УК-3-У3)

Практическое задание

Задача 1.1. Репродуктивный уровень.

Базовые операции со словарем

1. Создайте словарь с ключами, указанными в варианте.
2. Выведите на экран требуемую информацию.
3. Добавьте в словарь новую пару «ключ-значение».
4. Измените значение, соответствующее указанному ключу.
5. Удалите элемент словаря, используя функцию *del()*, предусмотрев предварительную проверку наличия ключа перед вызовом функции для предотвращения ошибки **KeyError**.

6. Выведите итоговый словарь на экран.

Примеры вариантов

№ варианта	Задания
1	1 Создайте словарь book с ключами: title, author, year и значениями: «Преступление и наказание», «Фёдор Достоевский», 1866. 2 Выведите на экран год издания книги. 3 Добавьте в словарь новый ключ genre со значением «Роман». 4. Измените автора книги на «Ф. М. Достоевский». 5. Удалите ключ title, предусмотрев проверку наличия ключа в словаре.
2	1. Создайте словарь book с ключами: title, author, year и значениями: «1984», «Джордж Оруэлл», 1949. 2. Выведите на экран название книги. 3. Добавьте в словарь новый ключ country со значением «Великобритания». 4. Измените год издания на 1948. 5. Удалите ключ author, предусмотрев проверку наличия ключа в словаре.

Задача 1.2. Репродуктивный уровень.

Работа с методами словарей

Требуется разработать программу на языке Python, которая позволяет пользователю:

1. Ввести с клавиатуры словарь (сначала количество пар, затем для каждой пары ключ (строка) и значение (целое число)).
2. Выполнять над этим словарём следующие операции (каждая операция выбирается из меню):
 - Вывод текущего словаря на экран.
 - Получение значения по заданному ключу (метод get()). Если ключа нет, возвращается 0. Запрос ключа и вывод значения повторяются в цикле, пока пользователь не решит прекратить (вводом 0).
 - Вычисление суммы всех значений словаря (метод values()).
 - Вывод количества элементов в словаре (len()).
 - Обновление словаря (метод update()): пользователь в цикле вводит новые пары «ключ–значение» до тех пор, пока не оставит ключ пустым; существующие ключи обновляются, новые – добавляются.
 - Удаление элемента по ключу (метод pop()). Если ключ найден – вывести удалённое значение, иначе сообщить об отсутствии.
 - Удаление нескольких последних элементов (метод popitem()). Количество удаляемых элементов указывает пользователь. Если запрошенное количество превышает размер словаря – вывести сообщение об ошибке, иначе удалить указанное число последних элементов и вывести обновлённый словарь.
 - Очистка словаря (метод clear()) с сохранением объекта словаря.

Программа должна быть реализована в виде набора функций (каждая функция решает одну подзадачу). В основной функции *main* организуется бесконечный цикл с меню, вызывающий соответствующие функции. Словарь хранится в локальной переменной внутри *main*.

Обязательные функции:

input_dict() -> dict – ввод словаря с клавиатуры.

print_dict(d: dict) -> None – вывод словаря.

get_value(d: dict) -> None – многократный запрос ключа и вывод значения через get() (с возвратом 0 при отсутствии), выход по вводу 0 после каждого запроса.

sum_values(d: dict) -> None – вывод суммы всех значений.

dict_length(d: dict) -> None – вывод количества элементов.

update_dict(d: dict) -> None – интерактивное обновление словаря: цикл ввода пар до пустого ключа.

pop_key(d: dict) -> None – удаление элемента по ключу, введённому пользователем.

pop_multiple(d: dict) -> None – удаление указанного пользователем количества последних элементов (проверка на превышение размера словаря).

clear_dict(d: dict) -> None – полная очистка словаря.

Требования к реализации.

- Код должен быть структурирован, каждая функция снабжена строкой документации (docstring).
- Меню должно содержать пункты, соответствующие всем перечисленным операциям (ввод словаря, вывод, получение значения, сумма, длина, обновление, удаление по ключу, удаление последних N элементов, очистка, выход).
- Обращать внимание на ситуации: отсутствие ключа при get() и pop(), превышение запрошенного количества при удалении последних элементов.

При вводе словаря, обновлении и удалении обеспечивать корректное преобразование вводимых значений в целые числа (предполагается, что пользователь вводит корректные данные; проверки не требуются, кроме указанной для pop_multiple).

Примеры вариантов

№ варианта	Задания
1	- Создайте словарь goods, который будет хранить информацию о количестве товаров на складе: мониторы – 120, клавиатуры – 200, мыши – 180. - Определите: - количество мониторов; - количество веб-камер. - Выведите на экран: - суммарное количество всех товаров; - количество элементов в словаре. - Измените количество мониторов и добавьте: наушники – 160; колонки – 90. - Удалите из словаря информацию: - количество клавиатур; - количество микрофонов. - Удалите из словаря информацию о двух последних из ранее добавленных товаров. Выведите значения, которые вернул метод, и словарь, полученный в результате удаления указанных элементов. - Удалите все элементы словаря, сохранив словарь.
2	- Создайте словарь goods, который будет хранить информацию о количестве товаров на складе: роутеры – 80, модемы – 110, кабели – 250. - Определите: - количество роутеров; - количество коммутаторов. - Выведите на экран: - суммарное количество всех товаров; - количество элементов в словаре. - Измените количество роутеров и добавьте: патч-панели – 70; коннекторы – 300. - Удалите из словаря информацию: - количество модемов; - количество кросс-панелей. - Удалите из словаря информацию о двух последних из ранее добавленных товаров. Выведите значения, которые вернул метод, и словарь, полученный в результате удаления указанных элементов. - Удалите все элементы словаря, сохранив словарь.

Задача 2. Реконструктивный уровень.

Словарь в словаре. Анализ данных.

- Разработать программу на языке Python, которая:
 - Содержит словарь, структура которого задана в варианте.
 - Реализует интерактивное меню с тремя операциями.
 - Первая операция выполняет агрегацию данных (например, топ категорий, суммарное значение, группировка).
 - Вторая операция – поиск экстремума (максимум или минимум) по одному из числовых полей.
 - Третья операция – вычисление значения с обязательным использованием метода `get()` для безопасного доступа к элементам словаря. Вычисление включает ввод пользователем некоторого ключа и подсчёт суммы, среднего или общего количества по отфильтрованным данным.
 - Программа должна содержать вспомогательную функцию для вывода всего словаря. Меню должно содержать пункт «Показать словарь».
 - Все функции должны иметь строки документации (docstring). Код должен быть структурирован, глобальные переменные не используются, словарь хранится в локальной переменной внутри `main()`.

Примеры вариантов

№ варианта	Задания
1	Структура словаря <pre>{название_книги: {"автор": str, "год": int, "жанр": str, "кол_во_экземпляров": int}}</pre> Операции меню - Топ-2 самых популярных жанров – по суммарному количеству экземпляров книг каждого жанра. - Найти книгу с максимальным количеством экземпляров (вывести название, автора, жанр, количество). Вычислить количество экземпляров заданной книги (используя <code>get()</code>) и общее количество экземпляров всех книг (при суммировании использовать <code>get()</code> для каждого значения)

2	Структура словаря <pre>{id_ товара: {"название": str, "категория": str, "цена": float, "количество": int, "скидка": int (проценты)}}</pre> Операции меню 1. Общая стоимость всех товаров на складе с учётом скидки (формула: $\text{цена} * \text{количество} * (100 - \text{скидка})/100$). 2. Найти товар с минимальной ценой до скидки. Вычислить общую стоимость товаров заданной категории (с учётом скидки), используя get() для доступа к полям цена, количество, скидка.
---	---

Контрольные вопросы

1. Что такое словарь (dict) в Python и для хранения каких данных он предназначен? Чем словарь принципиально отличается от списка (list) и кортежа (tuple)?
2. Перечислите все известные вам способы создания словаря в Python. Приведите примеры для каждого способа.
3. Какие типы данных могут использоваться в качестве ключей словаря? Может ли ключом быть список? Почему?
4. Какими способами можно получить значение по ключу в словаре? В чём разница между обращением через квадратные скобки dict[key] и использованием метода get()?
5. Как проверить, существует ли ключ в словаре? Напишите соответствующий код.
6. Как добавить новую пару «ключ-значение» в уже существующий словарь? Как изменить значение по уже существующему ключу?
7. Какие методы позволяют удалить элемент из словаря? В чём разница между оператором del, методом pop() и методом popitem()?
8. Как удалить все элементы из словаря, не удаляя сам объект словаря? Какой метод для этого используется?
9. Какие методы позволяют получить отдельно все ключи, все значения и все пары «ключ-значение» словаря? Какого типа объекты они возвращают?
10. Как объединить два словаря, добавив все пары из одного словаря в другой? Какой метод для этого используется и как он обрабатывает совпадающие ключи?
11. Напишите цикл for, который перебирает все пары «ключ-значение» словаря и выводит их на экран.
12. Как узнать количество элементов (пар «ключ-значение») в словаре?
13. Напишите инструкцию, которая доказывает, что созданный объект является словарём.
14. Что такое вложенный словарь? Приведите пример структуры данных, где словарь содержит другие словари в качестве значений.
15. Как преобразовать ключи словаря в список? Как преобразовать значения словаря в список? Как преобразовать весь словарь в список кортежей?
16. Будут ли равны два словаря, если они содержат одинаковые пары «ключ-значение», но в разном порядке? Почему?
17. Какая ошибка возникает при попытке обратиться к несуществующему ключу словаря через квадратные скобки? Как этого избежать?
18. Ниже записана программа. Укажите минимальное натуральное число, при вводе которого этот алгоритм напечатает число 4. Поясните ответ.

```
x = int(input())
d = {}
while x > 0:
    digit = x % 10
    d[digit] = d.get(digit, 0) + 1
    x //= 10
print(sum(d.values()))
```
19. Ниже записана программа. Укажите максимальное натуральное число, при вводе которого этот алгоритм напечатает число 3. Поясните ответ.

```
x = int(input())
d = {}
while x > 0:
    digit = x % 10
    d[digit] = d.get(digit, 0) + 1
    x //= 10
print(len(d))
```
20. Приведите краткое описание следующих характеристик ключа словаря: хешируемость (hashable); уникальность; неизменяемость; порядок вставки.

5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)

В семестре I предусмотрен экзамен.

Экзаменационный билет из 4 вопросов. Типовые вопросы экзамена приведены в вопросах самоподготовки.

Вопрос 1- вопрос по теории разделов 1-5.

Вопрос 2 - вопрос по теории разделов 1-5.

Вопрос 3 - задача по основам работы с базами данных – раздел 2.

Вопрос 4 - задача по основам алгоритмизации и программирования – раздел 3.

Экзаменационные билеты хранятся на кафедре и утверждаются ее заведующим.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного образовательного учреждения
высшего образования «Национальный исследовательский
технологический университет «МИСИС»

Кафедра «ВЫСШЕЙ МАТЕМАТИКИ И ИНФОРМАТИКИ»

15.03.04 – Автоматизация технологических процессов и производств

Профиль – Автоматизация технологических процессов и производств (горно-металлургическое производство)

Дисциплина «**Информатика**»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Проблема полисемии термина «информация».
2. Базовые типы данных языка Python. Динамическая типизация.
3. Задача (основы работы с базами данных). Имеются данные о товарах на складе:

- productID /*код товара*/
- number /*номер пролета*/
- rack /*номер стеллажа*/
- group /*группа*/
- name /*наименование*/

Используя операторы языка SQL создайте следующие запросы:

- запрос на создание таблицы в соответствии с указанной выше информацией;
- запрос на внесение записей в таблицу;
- запрос, изменяющий номер стеллажа для товаров трех заданных групп;
- запрос, определяющий количество товаров каждой группы.

4. Задача (основы алгоритмизации и программирования).
 - a. Дан одномерный список, состоящий из N целых неотрицательных чисел. Количество элементов списка (N) и их значения вводятся пользователем. Требуется вставить на позицию с заданным пользователем индексом значение среднего арифметического положительных элементов этого массива, восьмеричная запись которых оканчивается цифрой 0. В случае отсутствия указанных элементов, на позицию, указанную пользователем, следует вставить значение, равное индексу максимального элемента данного списка.
 - b. Дан словарь *servers*, где ключом является ID сервера (например, "SRV-001"), а значением – другой словарь, содержащий информацию о сервере: *name* (имя сервера (строка)); *ip* (IP-адрес (строка)); *response_times* (список времен отклика в миллисекундах (целые или вещественные числа), полученных в результате последовательных ping-запросов). Например, "SRV-001": {"name": "Web1", "ip": "192.168.1.1", "response_times": [25, 30, 28, 32]}
Требуется написать функцию *calculate_average_response_time(servers_dict)*, которая возвращает новый словарь, где ключ – ID сервера (из исходного словаря); значение – среднее арифметическое времени отклика для этого сервера (тип float), округлённое до двух знаков после запятой.

« ____ » _____ 20__ г.

Экзаменатор _____ Н.И. Некрасова

Утверждено на заседании кафедры ВМиИ

Протоколом №__ от _____ 202__ г.

Заведующий кафедрой ВМиИ _____ Е.Г. Кабулова

5.4. Методика оценки освоения дисциплины (модуля, практики. НИР)

Балльно-рейтинговая система оценивания, используемая преподавателем для текущей оценки успеваемости.

В течение семестра студент выполняет:

1) 11 лабораторных работ, каждая из которых оценивается от 2 до 4 баллов, вычисляемых как сумма следующих баллов:

1 балл – выполнение практического задания репродуктивного уровня;

2 балла – выполнение практического задания реконструктивного уровня;

1 балл – ответы на контрольные вопросы;

2) 1 домашнее задание, которое оценивается от 8 до 16 баллов:

8 баллов – при защите домашнего задания обучающийся показывает знания в объеме пройденной программы по разделу 3, ответы излагает хотя и с ошибками, но уверенно исправляемыми после дополнительных и наводящих вопросов, правильно действует по применению знаний на практике;

12 баллов – при защите домашнего задания обучающийся показывает твердые и достаточно полные знания в объеме пройденной программы по разделу 3, допускает незначительные ошибки при освещении заданных вопросов, правильно действует по применению знаний на практике;

16 баллов – при защите домашнего задания обучающийся показывает глубокие, исчерпывающие знания в объеме пройденной программы по разделу 3, уверенно действует по применению полученных знаний на практике.

По итогам работы в семестре рейтинг студента может составить от 30 (минимум, необходимый для допуска к экзамену) до 60 баллов.

Шкала оценивания знаний обучающихся на экзамене:

0-20 баллов– студент не знает и не понимает значительную или основную часть программного материала в пределах поставленных вопросов; имеет слабо сформированные и неполные знания и не умеет применять их к решению заданий по образцу и ответам на конкретные вопросы;

21-28 баллов– студент обнаруживает отдельные пробелы в усвоении существенных вопросов курса информатики, не препятствующие дальнейшему усвоению программного материала; испытывает затруднения в применении знаний, необходимых для решения практических задач различных типов;

29-36 – баллов обучающийся показывает знания в объеме пройденной программы, ответы излагает хотя и с ошибками, но уверенно исправляемыми после дополнительных и наводящих вопросов, правильно действует по применению знаний на практике;

37-40 – баллов обучающийся показывает глубокие, исчерпывающие знания в объеме пройденной программы, уверенно действует по применению полученных знаний на практике, грамотно и логически стройно излагает материал при ответе, умеет формулировать выводы из изложенного теоретического материала, знает дополнительно рекомендованную литературу.

Методика расчета оценки на экзамене:

Оценка «отлично» – итоговый рейтинг составил от 91 до 100 баллов;

Оценка «хорошо» – итоговый рейтинг составил от 71 до 90 баллов;

Оценка «удовлетворительно» - итоговый рейтинг составил от 54 до 70 баллов;

Оценка «неудовлетворительно» – итоговый рейтинг составил от 0 до 53 баллов;

Оценка «не явился» – обучающийся на экзамен не явился.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

6.1. Рекомендуемая литература

6.1.1. Основная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л1.1	Волк В.К.	Базы данных : учебник	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=725631	Москва ; Вологда : Инфра-Инженерия,
Л1.2	Гусева Е.Н., Ефимова И.Ю., Коробков Р.И. и др.	Информатика : учебное пособие	ЭБС "Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=83542	Москва : Флинта, 2021
Л1.3	Калитвин В. А.	Введение в программирование на Python : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=714538	Липецк : Липецкий государственный

Л1.4	Карчевская М.П.	Основы алгоритмизации инженерных задач : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: URL: https://biblioclub.ru/index.php?page=book&id=725669	Москва ; Вологда : Инфра-Инженерия, 2025
Л1.5	Муромцев В.В.	Информационные технологии в профессиональной деятельности : учебник и практикум	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=725647	Москва ; Вологда : Инфра-Инженерия, 2023
Л1.6	Нагаева И.А.	Основы алгоритмизации и программирования : практикум : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=722929	Москва : Директ-Медиа, 2025
Л1.7	Пролубников А.В.	Сети передачи данных : учебное пособие : в 2 частях	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=614062	Омск : Омский государственный университет им. Ф.М.Достоевского (ОмГУ), 2020
Л1.8	Пушкарёва Т.П.	Информатика : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=706616	Красноярск : Сибирский федеральный университет (СФУ), 2021

6.1.2. Дополнительная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л2.1	Волков М. А.	Информационные технологии : учебное пособие	ЭБС "Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=725646	Москва ; Вологда : Инфра-Инженерия, 2023
Л2.2	Завозкин С.Ю.	Архитектура вычислительных систем : практикум	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=719802	Кемерово : Кемеровский государственный университет, 2023
Л2.3	Калинина С. Д.	Анализ и визуализация данных : учебное пособие : в 2 частях	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=730629	Москва : МГИМО-Университет, 2023
Л2.4	Колокольникова А.И.	Информатика : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=596690	Москва ; Берлин : Директ-Медиа, 2020
Л2.5	Кузьменко И.П.	Базы данных и SQL : учебник	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=721011	Ставрополь : АГРУС, 2024
Л2.6	Лужков А.А.	Основы работы в Python : учебный компьютерный практикум : учебно-методическое пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=709623	Санкт-Петербург : Российский государственный педагогический университет им. А.И. Герцена
Л2.7	Сергеева О.А.	Программирование на Python : учебно-методическое пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=733177	Кемерово : Кемеровский государственный университет, 2024

Л2.8	Титов А. Н.	Python. Обработка данных : учебно-методическое пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=702252	Казань : Казанский национальный исследовательский технологический университет (КНИТУ), 2022
Л2.9	Сергеева О.А.	Программирование на Python : учебно-методическое пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book_red&id=733177	Кемерово : Кемеровский государственный университет, 2024
Л2.10	Титов А. Н.	Python. Обработка данных : учебно-методическое пособие	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=702252	Казань : Казанский национальный исследовательский технологический университет (КНИТУ), 2022
Л2.11	Шуваев А.В.	Информационные технологии : учебное пособие	ЭБС «Университетская библиотека" ONLINE URL: URL: https://biblioclub.ru/index.php?page=book&id=721339	Ставрополь : АГРУС, 2024
Л2.12	Щерба А. В.	Программирование на Python : первые шаги	ЭБС «Университетская библиотека" ONLINE URL: https://biblioclub.ru/index.php?page=book&id=686587	Москва : Лаборатория знаний, 2022

6.1.3. Методические разработки

Обозна	Авторы,	Заглавие	Библиотека	Издательство, год
Л3.1	Некрасова Н.И.	Информатика. Практикум по электронным таблицами и базам данных: учебное пособие	НТБ СТИ НИТУ «МИСИС»	СТИ НИТУ «МИСИС», 2026
Л3.2	Некрасова Н.И.	Основы программирования на языке Python : учебное пособие.	НТБ СТИ НИТУ «МИСИС»	СТИ НИТУ «МИСИС», 2026

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Э1	Научная электронная библиотека. Монографии, изданные в издательстве Российской Академии Естествознания	https://www.monographies.ru/
Э2	Национальный Открытый Университет «ИНТУИТ»	https://www.intuit.ru
Э3	ЭБС «Университетская библиотека ONLINE»	http://biblioclub.ru
Э4	Электронная библиотека НИТУ «МИСИС»	http://elibrary.misis.ru
Э5	Образовательная платформа Stepik	https://stepik.org/
Э6	Яндекс.Образование	https://education.yandex.ru/
Э7	Яндекс Практикум	https://practicum.yandex.com/
Э8	Яндекс Практикум — SQL	https://practicum.yandex.ru/sql-database-basics
Э9	VK Education	https://education.vk.company/
Э10	Хекслет (Hexlet)	https://ru.hexlet.io/
Э11	«Код будущего» (Академия «Просвещение»)	https://academy.prosv.ru/
Э12	D-Link — Портал обучения	https://www.dlink.ru/
Э13	СПбГУ — Анализ данных в Excel	https://spbu.ru/

Э14	Яндекс Практикум — Основы Excel	https://practicum.yandex.ru/
Э15	Видеоуроки по Microsoft Excel : сайт	https://www.planetaexcel.ru/video/

6.3 Перечень программного обеспечения	
П.1	Microsoft Windows
П.2	Microsoft Office
П.3	Интегрированная среда разработки и обучения на языке Python – IDLE
П.4	Система управления учебным процессом SMART Sync
П.5	7- Zip (свободно распространяемое программное обеспечение)
П.6	Kaspersky Endpoint Security
6.4. Перечень информационных справочных систем и профессиональных баз данных	
	Полнотекстовые российские научные журналы и статьи:
И.1	Научная электронная библиотека статей и публикаций «eLibrary.ru» : российский информационно-аналитический портал в области науки, технологии, медицины, образования. – URL: eLIBRARY https://elibrary.ru/
И.2	КиберЛенинка : научная электронная библиотека [научные журналы в полнотекстовом формате свободного доступа] : сайт. – URL: http://cyberleninka.ru
И.3	Национальный доступ к журналам РАН (РЦНИ). – URL: https://journals.rcsi.science/
	Профессиональные базы данных по информатике и ИТ
И.4	Планета информатики [профессиональная база данных по информатике и информационным технологиям]. – URL: http://plani.kz/
И.5	GitHub [профессиональная база данных и платформа для разработки и совместной работы над программными проектами, хранения кода и документации]. – URL: https://github.com
	Иностранные базы данных (доступ с IP адресов МИСИС):
И.6	аналитическая база (индексы цитирования) Web of Science https://apps.webofknowledge.com
И.7	аналитическая база (индексы цитирования) Scopus https://www.scopus.com/
И.8	научометрическая система InCites https://apps.webofknowledge.com
И.9	научные журналы издательства Elsevier https://www.sciencedirect.com/

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ	
7.1	Аудитория №205 «Компьютерный класс» Перечень основного оборудования, учебно-наглядных пособий: - моноблок – 17 шт.; - проектор; - экран настенный.
7.2	Аудитория №206 «Компьютерный класс» Перечень основного оборудования, учебно-наглядных пособий: - моноблок – 17 шт.; - проектор; - экран настенный.
7.3	Аудитория №318 «Лекционная аудитория» Перечень основного оборудования, учебно-наглядных пособий: • усилитель-распределитель; • монитор; • панель аудио; • монитор планшетный; • компьютер; • настенный экран; • микшерный пульт; • мультимедиа проектор; • усилитель звука; • документ – камера; • система видеоконференции связи; • контроллер; • коммутатор; • звуковые колонки; • вокальная радиосистема; • комплект учебной мебели на 70 посадочных мест.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

При изучении дисциплины «Информатика» самостоятельная работа студентов предусматривает два направления: изучение и освоение теоретического лекционного материала; освоение методики решения практических задач, предусмотренных лабораторным практикумом и домашним заданием. При всех формах самостоятельной работы студент должен обращаться к рекомендуемым преподавателем учебникам и учебным пособиям, информационным справочным системам и профессиональным базам данных, образовательным ресурсам информационно-телекоммуникационной сети «Интернет». Для работы с электронными материалами имеется возможность использования электронных библиотек, компьютеров в читальных залах библиотеки, локальной компьютерной сети ВУЗа. Кроме того, студент может получить разъяснения по непонятным вопросам у преподавателя на индивидуальных консультациях в соответствии с графиком консультаций. Консультация является активной формой учебной деятельности, поэтому каждую консультацию должно предварять самостоятельное изучение студентом литературы по определенной теме. Качество консультации зависит от степени подготовки студента и его умения анализировать и формулировать возникшую в результате самостоятельной работы проблему.

Основной частью самостоятельной работы студента является его систематическая подготовка к лабораторным работам. Для качественной подготовки к каждой лабораторной работе студент должен вначале освоить теоретический материал, необходимый для решения практических задач, предусмотренных лабораторным практикумом. Затем разобрать образец решения практического задания репродуктивного уровня, представленный в учебном пособии. Если некоторые задания вызвали затруднения при решении, попросить объяснить преподавателя на очередном лабораторном занятии. Для закрепления теоретического материала и освоенной во время выполнения лабораторной работы методики решения практических задач рекомендуется ответить на контрольные вопросы, представленные в методических указаниях к выполнению каждой лабораторной работы.

Ряд вопросов курса отведены для самостоятельной проработки студентами, что позволяет расширить круг изучаемых тем и глубину их проработки. Студенту предлагается освоить данные темы, самостоятельно изучив рекомендуемые преподавателем разделы учебников и учебных пособий, проанализировав актуальную информацию по данным темам, представленную в информационных справочных системах и профессиональных базах данных, обратившись к образовательным ресурсам информационно-телекоммуникационной сети «Интернет». Преподаватель проверяет качество усвоения самостоятельно проработанных вопросов на лабораторных занятиях, при защите домашнего задания и во время экзамена. Затем корректирует изложение материала и нагрузку на студентов.

Для получения практического опыта решения задач по дисциплине «Информатика» во внеаудиторное время предлагается самостоятельная работа в форме домашнего задания. Контроль над выполнением и оценка домашней работы осуществляется в форме собеседования.

Таким образом, использование всех рекомендуемых видов самостоятельной работы дает возможность значительно активизировать работу студентов над материалом курса и повысить уровень их усвоения.