Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный индустриальный университет»

Кафедра прикладных информационных технологий и программирования

УТВЕРЖДАЮ
Директор института
информационных технологий и
автоматизированных систем
_____ А.В. Зимин
подпись
«____» _____ 20___ г.

# РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Python для анализа данных

09.03.02 «Информационные системы и технологии»
(направленность (профиль): «Разработка и эксплуатация информационных систем и технологий»)

Квалификация выпускника
Бакалавр

Форма обучения
Очная форма

Срок обучения: 4 года

Год начала подготовки 2025

Новокузнецк
2025

СибГИУ
Подлинник электронного документа хранится в системе электронного согласования документов СибГИУ
ДОКУМЕНТ ПОДПИСАН ЭП
Сертификат: daba56d7b892efca3bf5c04ed0251c98b19b099c
Владелец: Директор ИИТиАС Зимин Алексей Валерьевич
Действителен: с 15 января 2025 г. по 10 апреля 2026 г.
Дата и время: 07-05-2025 11:53:50 ПРОВЕРИТЬ ПОДПИСЬ

## 1 Цели и задачи освоения учебной дисциплины

Целями учебной дисциплины являются:

– формирование у обучающихся навыков составления программ на языке Python для решения различных задач.

Задачами учебной дисциплины являются:

– изучение базовых алгоритмических конструкций и синтаксиса языка Python;

– освоение навыков работы с пакетами,строками, списками и коллекциями данных;

– приобретение обучающимися навыков работы над решением системных задач.

## 2 Место учебной дисциплины в структуре ООП по направлению подготовки (специальности)

Учебная дисциплина относится к учебным дисциплинам обязательной части **Блока 1 «Дисциплины (модули)»** ООП по направлению подготовки (специальности) 09.03.02 «Информационные системы и технологии».

Учебная дисциплина базируется на предварительном усвоении обучающимися учебных дисциплин:

– Информатика.

Учебная дисциплина дополняет знания и умения, получаемые по одновременно изучаемым и последующим учебным дисциплинам:

– Численные методы решения инженерных задач;

– Проектная деятельность 1;

– Проектная деятельность 2.

## 3 Планируемые результаты обучения по учебной дисциплине

Процесс изучения учебной дисциплины направлен на формирование следующих компетенций:

### – Общепрофессиональные компетенции

| Наименование категории (группы) ОПК | Код и наименование ОПК | Код и наименование индикатора достижения ОПК | Планируемые результаты обучения |
|---|---|---|---|
| | ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и | ОПК-6.1 Понимает теоретические основы алгоритмизации и программирования | – знать: теоретические основы алгоритмизации и программирования, структуры данных, библиотеки для анализа данных (NumPy, pandas), основы ООП, алгоритмы, статистику, |

| | | | |
|---|---|---|---|
| | технологий | | машинное обучение, визуализацию данных.<br>– уметь: разрабатывать программы на Python, писать читаемый код, использовать библиотеки, предобрабатывать данные, проводить статистический анализ, визуализировать данные, обучать модели. |
| | | ОПК-6.3 Разрабатывает программное обеспечение | – знать: основы Python, алгоритмы, структуры данных, ключевые библиотеки для анализа данных, ООП, статистику, машинное обучение, визуализацию данных, работу с файлами.<br>– уметь: разрабатывать программы, писать читаемый код, использовать библиотеки, предобрабатывать данные, проводить статистический анализ, визуализировать данные. |

## 4 Объем и содержание учебной дисциплины

Учебные занятия по учебной дисциплине проводятся в форме контактной работы и в форме самостоятельной работы обучающихся.

Контактная работа включает в себя занятия лекционного типа (лекции), занятия семинарского типа (семинары, практические занятия, практикумы), промежуточную аттестацию обучающихся и иные формы взаимодействия обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации ООП на иных условиях, в том числе при проведении промежуточной аттестации обучающихся. Контактная работа может проводиться с применением электронного обучения, дистанционных образовательных технологий.

## Объем учебной дисциплины

| Семестр / курс | | ***ИТОГО*** | ***3 семестр*** |
|---|---|---|---|
| Форма промежуточной аттестации | | | *экзамен* |
| Трудоёмкость | *академ. час.* | ***144*** | *144* |
| | *зачетных единиц* | ***4*** | *4* |
| Лекции, *академ. час.* | | ***8*** | *8* |
| в форме практической подготовки | | ***0*** | *0* |
| Лабораторные работы, *академ. час.* | | ***0*** | *0* |
| в форме практической подготовки | | ***0*** | *0* |
| Практические занятия, *академ. час.* | | ***48*** | *48* |
| в форме практической подготовки | | ***0*** | *0* |
| Курсовая работа / проект, *академ. час.* | | ***0*** | *0* |
| в форме практической подготовки | | ***0*** | *0* |
| Консультации, *академ. час.* | | ***0*** | *0* |
| в форме практической подготовки | | ***0*** | *0* |
| Самостоятельная работа, *академ. час.* | | ***52*** | *52* |
| в форме практической подготовки | | ***0*** | *0* |
| Контроль, *академ. час.* | | ***36*** | *36* |
| в форме практической подготовки | | ***0*** | *0* |

## Содержание учебной дисциплины

Раздел 1 Язык Python и его особенности;

Тема 1.1 Типы и структуры данных (Числовые типы данных. Логические значения. Работа со строками, кортежами, списками);

Тема 1.2 Ввод и вывод данных в Python (Диалоговый режим. Чтение из файла и запись в файл);

Раздел 2 Основные алгоритмы и их реализация на Python;

Тема 2.1 Линейные алгоритмы (Базовые блоки линейного алгоритма. Типовые задачи с линейной структурой. Операции с числами и строками);

Тема 2.2 Операторы ветвления и выбора (Базовая структура операторов. Типовой алгоритм с оператором ветвления/выбора. Вложенные, полные и не полные алгоритмы ветвления);

Тема 2.3 Циклические алгоритмы (Базовые блоки циклического алгоритма. Циклы с предусловием, постусловием, параметром. Нисходящий цикл с параметром. Обработка последовательностей и одномерных массивов. Сортировка массивов);

Тема 2.4 Обработка двумерных массивов (Структура матриц. Типовой алгоритм обработки двумерных массивов);

Раздел 3 Двумерная графика в Python;

Тема 3.1 Основные команды для вывода двумерной графики (Команды отрисовки линий, окружностей, изменения цвета, заливки и штриховки);

Тема 3.2 Построение графика функции средствами Python (Модуль Tkiner. Работа с "холстом". Вывод точечного графика функции);

Раздел 4 Интерактивные вычисления и среда разработки Jupiter Notebook и основы NumPy;

Тема 4.1 Основы Jupiter Notebook (Основы синтаксиса и особенности среды разработки. Интроспекция. Исключения и обратная трассировка. Интеграция с matplotlib и режим pylab);

Тема 4.2 Средства разработки программ (Команды оболочки и псевдонимы. Интерактивный отладчик. Хронометраж программы: %time и %timeit.);

Тема 4.3 NumPy ndarray (Тип данных для ndarray. Операции между массивами и скалярами. Индексирование и вырезание. Булево индексирование. Транспонирование массивов и перестановка осей);

Тема 4.4 Обработка данных с применением массивов (Запись логических условий в виде операций с массивами. Математические и статистические операции. Методы булевых массивов. Сортировка. Устранение дубликатов и другие теоретико-множественные операции);

Тема 4.5 Линейная алгебра и генерация случайных чисел (Операции линейной алгебры. Работа с матрицами. Генерация случайных чисел: случайное блуждание. Моделирование сразу нескольких случайных блужданий).

## 5 Перечень тем лекций

| № раздела / темы дисциплины | Темы лекций | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| Раздел 1. | Язык Python и его особенности | 2 | |
| Раздел 2. | Основные алгоритмы и их реализация на Python | 2 | |
| Раздел 3. | Двумерная графика в Python | 2 | |
| Раздел 4. | Интерактивные вычисления и среда разработки Jupiter Notebook и основы NumPy | 2 | |
| **Итого:** | | **8** | **0** |

## 6 Перечень тем практических занятий (семинаров)

| № раздела / темы дисциплины | Темы практических занятий (семинаров) | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| Тема 1.1. | Числовые типы данных. Логические значения. Работа со строками, кортежами, списками. | 2 | |

| | | | |
|---|---|---|---|
| | Типовая структура программы | | |
| Тема 1.2. | Диалоговый режим. Чтение из файла и запись в файл | 2 | |
| Тема 2.1. | Базовые блоки линейного алгоритма. Типовые задачи с линейной структурой. Операции с числами и строками | 2 | |
| Тема 2.2. | Базовая структура операторов. Типовой алгоритм с оператором ветвления/выбора. Вложенные, полные и не полные алгоритмы ветвления | 4 | |
| Тема 2.3. | Базовые блоки циклического алгоритма. Циклы с предусловием, постусловием, параметром. Нисходящий цикл с параметром. Обработка последовательностей и одномерных массивов. Сортировка массивов | 4 | |
| Тема 2.4. | Структура матриц. Типовой алгоритм обработки двумерных массивов | 4 | |
| Тема 3.1. | Команды отрисовки линий, окружностей, изменения цвета, заливки и штриховки | 4 | |
| Тема 3.2. | Модуль Tkiner. Работа с "холстом". Вывод точечного графика функции | 6 | |
| Тема 4.1. | Основы Jupiter Notebook | 4 | |
| Тема 4.2. | Средства разработки программ | 4 | |
| Тема 4.3. | NumPy ndarray | 4 | |
| Тема 4.4. | Обработка данных с применением массивов | 4 | |
| Тема 4.5. | Линейная алгебра и генерация случайных чисел | 4 | |
| **Итого:** | | **48** | **0** |

## 7 Перечень тем лабораторных работ

| **№ раздела / темы дисциплины** | **Темы лабораторных работ** | **Трудоемкость, *академ.час*** | |
|---|---|---|---|
| | | **всего** | **в форме практической подготовки** |
| | *Отсутствуют* | | |
| **Итого:** | | **0** | **0** |

## 8 Перечень тем курсовых работ (проектов)

| № раздела / темы дисциплины | Темы курсовых работ (проектов) | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | всего | в форме практической подготовки |
| | *Отсутствуют* | | |
| **Итого:** | | **0** | **0** |

## 9 Виды самостоятельной работы

| № раздела / темы дисциплины | Виды самостоятельной работы | Трудоемкость, *академ.час* | |
|---|---|---|---|
| | | всего | в форме практической подготовки |
| Раздел 1. | 1. Изучение лекционного материала;<br>2. Оформление отчета о практической работе;<br>3. Подготовка к практическому занятию;<br>4. Прохождение тестирования. | 14 | |
| Раздел 2. | 1. Изучение лекционного материала;<br>2. Оформление отчета о практической работе;<br>3. Подготовка к практическому занятию;<br>4. Прохождение тестирования. | 12 | |
| Раздел 3. | 1. Изучение лекционного материала;<br>2. Оформление отчета о практической работе;<br>3. Подготовка к практическому занятию;<br>4. Прохождение тестирования. | 12 | |
| Раздел 4. | 1. Изучение лекционного материала;<br>2. Оформление отчета о практической работе;<br>3. Подготовка к практическому занятию;<br>4. Прохождение тестирования. | 14 | |
| *Контроль* | *Подготовка к экзамену* | 36 | |
| **Итого:** | | **88** | **0** |

## 10 Матрица компетентностных задач по учебной дисциплине

| Контролируемые разделы (темы) дисциплины | Контролируемые компетенции | Оценочные средства |
|---|---|---|
| Раздел 1. | ОПК-6 | 1. Наблюдение за работой на практическом занятии;<br>2. Отчет о практической работе;<br>3. Тестовые задания |
| Раздел 2. | ОПК-6 | 1. Наблюдение за работой на практическом занятии;<br>2. Отчет о практической работе;<br>3. Тестовые задания |
| Раздел 3. | ОПК-6 | 1. Наблюдение за работой на практическом занятии;<br>2. Отчет о практической работе;<br>3. Тестовые задания |
| Раздел 4. | ОПК-6 | 1. Наблюдение за работой на практическом занятии;<br>2. Отчет о практической работе;<br>3. Тестовые задания |
| Экзамен. | ОПК-6 | 1. Вопросы к экзамену;<br>2. Задачи к экзамену |

## 11 Входной контроль

**11.1 Цель входного контроля** − определить начальный уровень подготовленности обучающихся и выстроить индивидуальную траекторию обучения. В условиях личностно-ориентированной образовательной среды результаты входного оценивания обучающегося используются как начальные значения в индивидуальном профиле академической успешности обучающегося.

### 11.2 Описание оценочных средств

Форма проведения входного контроля – тестирование.

Длительность тестирования – 30 минут.

Тестирование проводится в течение первых двух недель изучения учебной дисциплины в СУО «Moodle».

Для проведения тестирования используются тестовые задания для входного контроля (для ответа на 1 вопрос, как правило, выделяется 1 минута).

### Шкала оценивания результатов тестирования

За каждый правильный ответ выставляется 1 балл. Оценка выставляется в зависимости от процента правильных ответов:

Оценка формируется в соответствии с критериями таблицы:

| Процент результативности (правильных ответов) | Оценка |
|---|---|
| 86 до 100 | «отлично» |
| от 67до 85 | «хорошо» |
| от 50 до 66 | «удовлетворительно» |
| менее 50 | «неудовлетворительно» |

Основываясь на результатах тестирования, формируется индивидуальная траектория обучающегося (группы обучающихся), исходя из следующих соображений:

– 86 – 100 % – обучающийся готов к изучению учебной дисциплины без дополнительного повторения изученного ранее материала;

– 50 – 67 % – обучающийся в целом готов к изучению учебной дисциплины, требуется повторение отдельных вопросов(тем) изученного ранее материала, необходимого для успешного освоения учебной дисциплины;

– 49% и менее – обучающийся имеет слабую подготовку к изучению учебной дисциплины, необходимы корректирующие занятия и консультации.

**Тестовые задания для входного контроля знаний обучающихся приведены в приложении.**

## 12 Текущий контроль

**12.1 Текущий контроль знаний** используется для оперативного и регулярного управления учебной деятельностью (в том числе самостоятельной) обучающихся. Данный вид контроля стимулирует у обучающихся стремление к систематической самостоятельной работе по изучению учебной дисциплины и позволяет оценить степень освоения изученного материала.

Текущий контроль успеваемости осуществляется в течение семестра, в ходе повседневной учебной работы по индивидуальной инициативе педагогического работника, в том числе посредством использования СУО «Moodle».

Формы проведения текущего контроля – наблюдение за работой на практическом занятии; отчет о практической работе; тестовые задания.

### 12.2 Описание оценочных средств

### Шкалы оценивания (методики оценки)

#### 12.2.1 Рекомендации по оцениванию результатов тестирования обучающихся

Тестирование проводится по изученным разделам (темам) учебной дисциплины для выявления уровня знаний и умений обучающегося,

мотивирования обучающегося к активизации самостоятельной работы по усвоению учебного материала, выявления и устранения пробелов в знаниях, повышения дисциплины и организации учебной деятельности обучающегося.

Тестирование проводится в СУО «Moodle» в рамках часов, отведенных на самостоятельную работу обучающихся. Для тестирования используется 20 вопросов из базы типовых тестовых заданий по изученным разделам (темам) учебной дисциплины. Обучающийся выбирает один ответ из предложенных вариантов ответа. В случае выбора неправильного ответа обучающийся получает за данный вопрос 0 баллов; в случае правильного ответа – 1 балл.

Оценка формируется в соответствии с критериями таблицы:

| Максимальный балл | Проходной балл | Оценка |
|---|---|---|
| 20 | не менее 18 | отлично |
| 17 | не менее 14 | хорошо |
| 13 | не менее 10 | удовлетворительно |
| 9 | – | неудовлетворительно |

### 12.2.2 Рекомендации по оцениванию в процессе наблюдения за работой на практическом занятии

Наблюдение за работой на практическом занятии – важнейший компонент контроля деятельности обучающихся и качества их обучения.

**Критерии оценки в процессе наблюдения за работой на практическом занятии:**

– самостоятельность выполнения задания и работы с методическими указаниями (учитывается работа в течение занятия, быстрота и способность нахождения в методических указаниях нужной информации);

– правильность выполнения работы (учитывается последовательность выполняемых действий, умение работать с инструментами и/или средствами вычислительной техники);

– рациональность использования времени, отведенного на задание (не одобряется затянутость выполнения работы с учетом индивидуальных особенностей обучающихся);

– активность обучающегося при выполнении практической работы (учитывается индивидуальная работа и работа в группе).

Критерии оценки по результатам наблюдения за работой на практическом занятии

| Наименование параметра оценки | Максимальное количество баллов |
|---|---|
| Самостоятельно выполняет задания; умеет работать с методическими указаниями. | 2,5 |

| Наименование параметра оценки | Максимальное количество баллов |
|---|---|
| Указанные в методических указаниях действия выполняет в правильной последовательности; умеет работать с инструментами и/или средствами вычислительной техники. | 1,25 |
| Рационально использует время, отведенное на задание; проявляет активность при выполнении практической работы. | 1,25 |
| **Итого** | **5** |

Максимальное количество – 5 баллов соответствует оценке «отлично», 4 балла – «хорошо», 3 балла – «удовлетворительно», 2 балла – «неудовлетворительно».

### 12.2.3 Рекомендации по оцениванию отчета о практической работе

Оформление отчета о проделанной практической работе обучающиеся осуществляют в рамках самостоятельной работы.

**Критерии оценки отчета о практической работе:**

– правильность выполнения работы (учитывается полнота отчета о практической работе, наличие всех необходимых данных, наличие выводов, наличие ошибок и неточностей);

– аккуратность оформления отчета о практической работе.

Критерии оценки отчета о практической работе

| Наименование параметра оценки | Максимальное количество баллов |
|---|---|
| Отчет о практической работе содержит все необходимые данные и выводы. | 2,5 |
| Отчет о практической работе не содержит ошибок и неточностей. | 1,25 |
| Отчет о практической работе оформлен аккуратно. | 1,25 |
| **Итого** | **5** |

Максимальное количество – 5 баллов соответствует оценке «отлично», 4 балла – «хорошо», 3 балла – «удовлетворительно», 2 балла – «неудовлетворительно».

**Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины приведены в приложении.**

## 13 Промежуточная аттестация

**13.1 Оценочные материалы для промежуточной аттестации** обучающихся по учебной дисциплине предназначены для оценки степени достижения запланированных результатов обучения по завершению изучения учебной дисциплины в установленной учебным

планом форме и позволяют определить качество усвоения изученного материала.

Итоговой формой контроля сформированности компетенций у обучающихся по учебной дисциплине является: экзамен.

Оценочные материалы для промежуточной аттестации обучающихся включают: вопросы к экзамену; задачи к экзамену.

### 13.2 Оценивание обучающегося на экзамене

| Оценка | Требования к знаниям |
|---|---|
| отлично | Оценка «отлично» выставляется обучающемуся, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически его излагает, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, использует в ответе материал различной литературы, правильно обосновывает принятое нестандартное решение, обладает приемами выполнения практических задач по формированию компетенций. |
| хорошо | Оценка «хорошо» выставляется обучающемуся, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, обладает необходимыми умениями и приемами их выполнения. |
| удовлетворительно | Оценка «удовлетворительно» выставляется обучающемуся, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, испытывает сложности при выполнении практических работ и затрудняется связать теорию вопроса с практикой. |
| неудовлетворительно | Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, неуверенно отвечает, допускает серьезные ошибки, не имеет представлений по методике выполнения практической работы. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по данной учебной дисциплине. |

**Задания и вопросы для промежуточной аттестации обучающихся приведены в приложении.**

## 14 Учебно-методическое и информационное обеспечение учебной дисциплины

**а) литература:**

1 Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю.

Федоров. – Москва : Юрайт, 2020. – 235 с. – ISBN 978-5-534-02816-4. – URL: https://urait.ru/bcode/450999 (дата обращения: 29.04.2025);

2 Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие / В. М. Шелудько. – Москва : ЮФУ, 2017. – 146 с. – ISBN 978-5-9275-2649-9. – URL: https://www.studentlibrary.ru/book/ISBN9785927526499.html (дата обращения: 29.04.2025);

3 Рамальо, Л. Python. К вершинам мастерства / Л. Рамальо. Пер. с англ. А. А. Слинкин. - Москва : ДМК Пресс, 2016. - 768 с. - ISBN 978-5-97060-384-0. – URL: https://www.studentlibrary.ru/book/ISBN9785970603840.html (дата обращения: 29.04.2025);

4 Хахаев, И. А. Практикум по алгоритмизации и программированию на Python: курс / И. А. Хахаев. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 179 с. : ил. – URL: https://biblioclub.ru/index.php?page=book&id=429256 (дата обращения: 29.04.2025).

**б) ресурсы информационно-телекоммуникационной сети «Интернет»:**

1 Электронный каталог : сайт / Научно-техническая библиотека СибГИУ. – Новокузнецк, [199 – ]. – URL: http://libr.sibsiu.ru. – URL: https://libr.sibsiu.ru;

2 Электронная библиотека // Научно-техническая библиотека СибГИУ : сайт. – Новокузнецк, [200 – ]. – URL: http://library.sibsiu.ru/LibrELibraryFullText.asp. – Режим доступа: для авторизир. пользователей. – URL: https://library.sibsiu.ru/LibrELibraryFullText.asp;

3 Консультант студента : электронно-библиотечная система / ООО «КОНСУЛЬТАНТ СТУДЕНТА». – Москва, [200 – ]. – URL: http://www.studentlibrary.ru. – Режим доступа: для авторизир. пользователей;

4 ЛАНЬ : электронно-библиотечная система : [коллекция «Единая профессиональная база знаний для технических вузов - Издательство Лань" ЭБС»] / ООО «Издательство ЛАНЬ». – Санкт-Петербург, [200 – ]. – URL: http://e.lanbook.com. – Режим доступа: для авторизир. пользователей;

5 Образовательная платформа ЮРАЙТ / ООО «Электронное издательство ЮРАЙТ». – Москва, [200 – ]. – URL: https://urait.ru. – Режим доступа: для авторизир. пользователей;

6 Университетская библиотека онлайн : электронно-библиотечная система / ООО «Директ-Медиа». – Москва, [200 – ]. – URL: https://biblioclub.ru. – Режим доступа: для авторизир. пользователей. – URL: http://www.biblioclub.ru;

7 Электронная библиотека УМЦ ЖДТ : [коллекция «Эксплуатация железных дорог»] / ФГБУ ДПО «Учебно-методический центр по

образованию на железнодорожном транспорте». – Москва, [2013 – ]. – URL: https://umczdt.ru/books/. – Режим доступа: для авторизир. пользователей;

8 НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА eLIBRARY.RU : база данных / ООО «НЭБ». – Москва, [200 – ]. – URL: http://elibrary.ru. – Режим доступа: по подписке;

9 Электронные периодические издания ИВИС : универсальная база данных / ООО «ИВИС». – Москва, [200 – ]. – URL: http://eivis.ru. – Режим доступа: по подписке.

**в) лицензионное и свободно распространяемое программное обеспечение:**

– 7-Zip;
– Adobe Acrobat Reader;
– Astra Linux Special Edition;
– Kaspersky Endpoint Security;
– Microsoft Office;
– Microsoft Windows;
– Notepad++;
– OnlyOffice;
– Р7-Офис.

**г) базы данных и информационно-справочные системы:**

1 ГАРАНТ : справочно-правовая система / ООО «Правовой центр «Гарант». – Кемерово, [200 – ]. – Режим доступа: компьютерная сеть Сиб. гос. индустр. ун-та.;

2 КонсультантПлюс : справочно-правовая система / ООО «Информационный центр АНВИК». – Новокузнецк, [199 – ]. – Режим доступа: компьютерная сеть библиотеки Сиб. гос. индустр. ун-та.;

3 Техэксперт : информационно-справочная система / ООО «Группа компаний «Кодекс». – Кемерово, [200 – ]. – Режим доступа: компьютерная сеть Сиб. гос. индустр. ун-та.

## 15 Материально-техническое обеспечение учебной дисциплины

Материально-техническое обеспечение учебной дисциплины включает учебные аудитории, оснащенные оборудованием, компьютерной техникой, и техническими средствами обучения, в том числе:

- учебную аудиторию для проведения занятий лекционного типа, оборудованную учебной доской, экраном и мультимедийным проектором;
- учебную аудиторию для проведения занятий семинарского типа (практических занятий), оснащенную персональными компьютерами и выходом в сеть «Интернет»;
- учебную аудиторию (помещения) для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации;

- помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду, научно-техническую библиотеку СибГИУ.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) 09.03.02 «Информационные системы и технологии».

Составитель(и):

старший преподаватель Гусев Максим Михайлович (кафедра прикладных информационных технологий и программирования).

Рабочая программа дисциплины рассмотрена и утверждена на заседании кафедры.

Согласована:

Системный аналитик группы
системной архитектуры департамента
информационных технологий
ООО "СИ СОФТ"  Е.А. Шапошникова

## Приложение

## Тестовые задания для входного контроля знаний обучающихся

**Выберите один правильный ответ.**

1 Алфавит языка программирования – это…

a. набор символов русского языка, используемых при составлении любых текстов, называемых алфавитом языка

b. набор цифр и знаков препинания, используемых при составлении текстов, называемых алфавитом языка

c. разрешенный к использованию набор символов, с помощью которого могут быть образованы слова и величины данного языка

2 Переменная характеризуется

a. именем, типом, значением

b. именем, значением

c. значением, типом

3 Величина, значение которой меняется в процессе выполнения программы, называется

a. константой

b. переменной

c. меткой

4 Операции DIV и MOD допустимы над величинами

a. целочисленного типа

b. вещественного типа

c. символьного типа

5 Какое значение примет переменная X после выполнения оператора X=17 mod 4?

a. 1

b. 4

c. 4.25

6 Дано трехзначное число x=abc. Операция x div 10 mod 10 определяет

a. единицы

b. десятки

c. сотни

7 Цикл FOR используется в том случае, если

a. необходимо, чтобы тело цикла выполнялось хотя бы один раз

b. точно знаете, сколько раз должно быть выполнено тело цикла

c. хотите, чтобы проверка была произведена прежде, чем выполнено тело цикла

8 Массив - это

a. совокупность конечного числа данных различных типов

b. совокупность конечного числа данных одного типа

c. скалярный тип данных

9 Индекс массива определяет

a. положение элементов массива относительно друг друга

b. номер элемента в массиве

c. количество элементов

10 Разветвляющейся называется алгоритмическая конструкция,…

a. в которой выбирается один из нескольких возможных путей (вариантов) вычислительного процесса;

b. в которой все этапы решения задачи выполняются строго последовательно

c. организующая многократное выполнение указанного действия

11 Алгоритм – это...

a. абстрактная или реальная система, способная выполнить действия, предписанные исполнителю

b. это понятное и точное предписание исполнителю выполнить определенную последовательность действий для решения некоторой задачи за конечное число шагов

c. инструкция для решения поставленной задачи

12 Свойства алгоритма включают в себя:

a. понятность, дискретность

b. массовость, результативность

c. неоднозначность,

d. определенность

13 Укажите базовые алгоритмические структуры:

a. линейная

b. цикл

c. сортировка

d. ветвление

e. массив

14 Укажите виды циклов

a. с условием, с заданным числом повторений

b. для нахождения суммы, для обработки массивов

c. для ввода данных в программу, для обработки значений элементов массива

15 Прочитайте код. Что будет выведено на консоль?

```
int firstNumber = 5;
double secondNumber = 5.0;
Console.WriteLine(firstNumber == secondNumber);
```

a. False

b. True

c. Возникнет ошибка

16 Типы данных в языке C# принято классифицировать как:

a. фигурные

b. определенные программистом

c. простые

d. встроенные (базисные)

e. сложные (структурные)

f. правильных ответов нет

17 Что используют все языки программирования высокого уровня для предоставления программисту простого и легкого доступа к различным объектам?

a. идентификаторы

b. коммутаторы

c. спецификаторы

d. правильных ответов нет

18 Что понимают под языком программирования (ЯП)?

a. язык, предназначенный для решения определенного класса задач (проблем)

b. правила представления данных и записи алгоритмов их обработки, которые автоматически выполняются ЭВМ

c. язык, предназначенный для создания пакетов прикладных программ, в том числе для современных операционных систем

d. правильных ответов нет

19 Что понимается под наследованием типов?

a. однородная масса разрядов, имеющая какую-либо структуру

b. возможность дисциплинированного создания новых типов на основе уже определенных

c. многовходовой программный модуль, точки входа которого соответствуют набору операций реализуемого типа

d. правильных ответов нет

20 Что понимается под классом в объектно-ориентированном программировании (ООП)?

a. процедуры и функции любого языка программирования

b. структурный тип данных, который включает описание полей данных, процедур и функций, работающих с этими полями данных

c. списки передаваемых параметров, типы функции

d. правильных ответов нет

## Задания и вопросы для текущего контроля знаний обучающихся по разделам (темам) учебной дисциплины

### 1 Тестирование по результатам освоения лекционного материала и самостоятельной работы

Тестирование проводится для определения степени усвоения материалов лекций, а также выполнения самостоятельной работы в рамках часов, отведенных на самостоятельную работу обучающихся, в СУО «Moodle».

**Раздел 1. Язык Python и его особенности**

Перечень вопросов для тестирования:

1 Как получить данные от пользователя?

a) Использовать метод read()

b) Использовать метод readLine()

c) Использовать метод cin()
d) Использовать метод get()
e) Использовать метод input()

2 Какие из приведенных стилей программирования поддерживает язык Python?

a) Процедурный
b) Объектно-ориентированный
c) Функциональный
d) Смешанный

3 Каким способом можно объявлять переменные в Python:

a) a=5
b) a=int (5)
c) int a=5

4 Какая функция отвечает за вывод на экран?

a) cout<<a
b) out (a)
c) print (a)

5 Какая функция отвечает за открытие файла?

a) file()
b) open()
c) open_file()

6 В каком из вариантов присутствует ошибка?

a) a=5
   print ('a')
b) while True
   print(a)
c) a=open("file.txt")

7 Что делает команда import

a) импортирует файл модуля
b) создает функцию
c) удаляет файл

8 Выберите вариант правильного удаления переменной a

a) del(a)
b) delete(a)
c) delete=a

9 Какое значение 1//2 вернет выражение в среде IDLE?

a) 0
b) 0.5
c) 0.50

10 Язык Python:

a) имеет возможность интеграции с другими языками программирования
b) является низкоуровневым языком программирования
c) является высокоуровневым языком программирования
d) имеет статическую типизацию

e) имеет динамическую типизацию

11 Язык Python:

a) подходит для написания сценариев, связывающих остальные компоненты проекта

b) предназначен для решения задач требовательных к производительности кода

c) предназначен для быстрой разработки приложений

d) является компилируемым языком

e) является интерпретируемым языком

12 Язык Python:

a) сильно типизирован

b) слабо типизирован

c) поощряет повторное использование кода

d) является низкоуровневым языком программирования

e) является высокоуровневым языком программирования

13 Привести переменную x к типу числа с плавающей точкой можно следующим способом:

a) (float)x

b) x.float()

c) x.__float__()

d) float(x)

e) x.__class__=float

14 Python поддерживает следующие простые скалярные типы:

a) целое

b) беззнаковое целое

c) булево (логическое)

d) число с плавающей точкой

e) строка

15 Из приведенных ниже высказываний укажите истинное:

a) перед использованием переменной она должна быть определена с указанием типа и инициализирована каким-либо значением

b) перед использованием переменной она должна быть определена с указанием типа

c) перед использованием переменной она должна быть инициализирована каким-либо значением

d) переменные не обязаны быть инициализированы каким-либо значением перед использованием, а тип переменной определяется в зависимости от контекста

e) перед использованием переменной она будет автоматически проинициализирована значением по умолчанию

16 Укажите результат выполнения скрипта:

```
foo = (1,)
bar = foo
bar += (1,)
```

print (foo)
a) (1,)
b) (1,1)
c) (1)
d) скрипт не будет выполнен, так как содержит ошибки
17 Укажите результат выполнения скрипта:
x = 'Printing'
x[0] = 'p'
print (x)
a) Printing
b) printing
c) p
d) скрипт не будет выполнен, так как содержит ошибки
18 Укажите результат выполнения скрипта:
x = {0:'P',1:'r',2:'i',3:'n',4:'t'}
x[0] = 'p'
print (x)
a) {0: 'P', 1: 'r', 2: 'i', 3: 'n', 4: 't'}
b) {0: 'p', 1: 'r', 2: 'i', 3: 'n', 4: 't'}
c) {0: 'p'}
d) скрипт не будет выполнен, так как содержит ошибки
19 В языке Python существуют следующие побитовые операции:
a) побитовое или ( | )
b) побитовое или ( || )
c) отрицание ( ! )
d) исключающее или (^ )
e) сдвиг вправо ( > )
20 В языке Python существуют следующие арифметические операции:
a) деление с остатком ( % )
b) целочисленное деление ( % )
c) деление по модулю ( % )
d) нахождение дробной части от результата деления ( % )
e) возведение в степень ( % )
21 Введите на месте пропуска текст (регистр не учитывается).
Какая функция используется для получения длины строки или списка? _____
22 Введите на месте пропуска текст (регистр не учитывается).
Какая функция используется для вывода текста на экран? _____
23 Введите на месте пропуска текст (регистр не учитывается).
Какой метод используется для добавления элемента в конец списка? _____
24 Введите на месте пропуска текст (регистр не учитывается).

Какой оператор используется для проверки равенства двух значений? _____

25 Введите на месте пропуска текст (регистр не учитывается).

Какой метод используется для добавления элемента в конец списка? _____

**Раздел 2. Основные алгоритмы и их реализация на Python**

Перечень вопросов для тестирования:

1 Как называется встроенный в языке Python тип данных неупорядоченной коллекции из нуля или более пар ключ-значение?

a) dict

b) set

c) list

d) frozenset

2 Если предположить, что класс Mydict наследует класс dict, то каким класс dict является по отношению к классу Mydict?

a) дочерним

b) подклассом

c) базовым

3 Дан массив >>> c = array([[1,2], [2,3], [4,5]])

Чему равен срез c[:,1]:

a) array([1, 2, 4])

b) array([2, 3])

c) array([2, 3, 5])

4 Какие новые имена появятся в текущем модуле после выполнения следующего кода:

import sre as re

from re import compile

a) имена sre, re и compile

b) только имена re и compile +

c) только имена sre и compile

5 Чему будет равен результат выполнения:

urlparse.urlsplit(“http://google.com/search?q=Python#1″)

a) (‘http’, ‘google.com’, ‘/search’, ”, ‘q=Python’, ‘1’)

b) (‘http://’, ‘google.com/’, ‘search?’, ‘q=Python#’, ‘1’)

c) (‘http’, ‘google.com’, ‘/search’, ‘q=Python’, ‘1’)

6 Для чего применяется метод nextset() объекта-курсора:

a) для перехода к следующему набору записей результата запроса

b) для перехода к следующей записи результата запроса

c) для получения следующего набора записей результата запроса

7 В каком модуле нужно искать функции, помогающие тестировать программу:

a) dictutils

b) profile

c) unittest

8 Каким образом в модуле poplib представлен сеанс работы с POP3-сервером:

a) набор функций

b) экземпляр класса POP3 +

c) список кортежей

9 Какой из перечисленных обработчиков mod_python выполняется раньше других:

a) PythonPostReadRequestHandler

b) PythonHandler

c) PythonFixupHandler

10 Что будет получено в результате вычисления следующего выражения:

(0 < 5 <= 3) and (0 / 0)

a) True (или 1)

b) False (или 0)

c) синтаксическая ошибка

11 Какое из приведенных ниже регулярных выражений некорректно:

a) a+b++

b) (?P(ac))

c) (a+b+)+

12 Зачем в XML пространства имен:

a) пространства имен позволяют указывать опции для приложения, обрабатывающего XML

b) пространства имен позволяют включать однотипные XML-документы друг в друга

c) для сочетания в одном документе XML с различными DTD

13 Что такое регулярное выражение:

a) шаблон, описывающий множество строк

b) синтаксически правильное выражение на языке Python

c) шаблон для поиска файлов в каталоге

14 Для чего нужны функции модуля gettext:

a) для получения текста от пользователя

b) для обеспечения интернационализации программы

c) для чтения строки со стандартного ввода

15 Экземпляры какого класса сочетают замок и средство коммуникации между потоками:

a) Event

b) Lock

c) Condition

16 Как средствами самого Python определить имена формальных аргументов функции func(), если известно, что функция написана на Python:

a) func.func_globals

b) inspect.getargspec(func)

c) func.func_locals

17 Какие кодировки исходного текста программы поддерживает интерпретатор Python:

a) большинство кодировок, распространенных сегодня

b) ASCII, Latin-1, UTF-8

c) ASCII

18 Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

a) items()

b) get_main_type()

c) get_payload()

19 Какой метод позволяет узнать, имеет ли данное сообщение несколько частей:

a) items()

b) get_payload()

c) is_multipart()

20 Сокрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс называется:

a) абстракцией

b) инкапсуляцией

c) агрегацией

**Раздел 3. Двумерная графика в Python**

Перечень вопросов для тестирования:

1 Размер экрана в графическом режиме:

a) 640*400

b) 680*480

c) 600*400

d) 800*600

2 Наименьший элемент экрана:

a) Линия

b) Точка

c) Прямоугольник

d) Окружность

3 Где находится начало координат?

a) Нижний левый угол экрана

b) Центр экрана

c) Верхний правый угол экрана

d) Верхний левый угол экрана

4 Центр экрана имеет следующие координаты:

a) 400*300

b) 320*200

c) 320*240

d) 330*250

5 Процедура для построения точки:

a) Line
b) Circle
c) Rectangle
d) PutPixel
6 Процедура для построения линии:
a) Line
b) Circle
c) Rectangle
d) PutPixel
7 Процедура для построения окружности:
a) RoundRect
b) Arc
c) Circle
d) Ellipse
8 Сколько параметров имеет процедура Circle?
a) 5
b) 4
c) 2
d) 3
9 Количество параметров в процедуре Line?
a) 2
b) 4
c) 3
d) 5
10 Укажите верное написание цвета в программе:
a) красный
b) red
c) clred
d) ColorRed
11 Какая процедура рисует закрашенный эллипс?
a) Circle (100, 120, 50)
b) FillRectangle
c) FillEllipse
d) FillEllipse (100, 120, 160,210)
12 Укажите верный вариант рисования прямоугольника:
a) Rectangle (100, 200, 100, 200)
b) Rectangle (100, 200, 200, 400)
c) Rectangle (100, 200.5, 200, 400.5)
d) Rectangle (100, 200, 100)
13 Процедура, выводящая текст в левый верхний угол экрана:
a) TextOut(10,10, ‘Privet’)
b) TextOut(10,10, “Privet”)
c) TextOut(10,10, Privet)
d) TextOut(10,10, ‘Privet’, clblack)
14 Процедура установки цвета переднего фона

a) Setfillstyle
b) Setcolor
c) Clrscr
d) Floodfill

15 Процедура закраски в выбранном стиле внутри замкнутой фигуры

a) Setfillstyle
b) Setcolor
c) Clrscr
d) Floodfill

16 Процедура установки стиля заполнения и цвета

a) Setfillstyle
b) Setcolor
c) Clrscr
d) Floodfill

17 Процедура построения закрашенного сектора

a) Fillellipse
b) Pieslice
c) Piesector
d) Fillsector

18 Что обозначает в процедуре «SetFillStyle(1,7);» число 1

a) синий цвет штриховки
b) серый цвет штриховки
c) серый цвет контура
d) сплошная штриховка

19 Что выводится на экран при выполнении операторов

```
 ...
Y:=90;
Repeat Y:=Y-10;
FillEllipse(5,Y,5,5);Until Y<0;
...
```

a) шарики по диагонали
b) шарики по вертикали
c) шарики по горизонтали

20 Для вывода линии синего цвета используются процедуры

a) Line(10,10,7,1);SetColor(1);
b) SetFillStyle(1,1);Line(10,10,7,1);
c) SetColor(1);Line(10,10,7,1);
d) Line(10,10,7,1);FloodFill(10,10,1);

**Раздел 4. Интерактивные вычисления и среда разработки Jupiter Notebook и основы NumPy**

1 Сопоставьте термин и определение:

| Термин | Определение |
|---|---|
| 1) cell | A) «вычислительный движок», который выполняет код, |

| | |
|---|---|
| | содержащийся в документе ноутбука |
| 2) markdown | Б) контейнер для текста, который будет отображаться в записной книжке, или код, который будет выполняться ядром записной книжки |
| 3) kernel | В) маркировка ячейки как контейнера для текста |

Выберите вариант ответа:

a) 1-Б, 2-В, 3-А;

b) 1-А, 2-Б, 3-В;

c) 1-В, 2-Б, 3-А;

d) 1-Б, 2-А, 3-В;

2 Укажите код, который успешно скомпилируется и выведет приветствие:

1)

```
def say_hello(recipient):
return 'Hello, {}!'.format(recipient)
say_hello('Tim')
```

2)

```
def say_hello(recipient):
    return 'Hello, {}!'.format(recipient)
say_hello('Tim')
```

Выберите один ответ:

a) 1;

b) 2;

c) оба скомпилируются;

d) оба не скомпилируются;

3 Каким термином обозначается вывод общей информации об объекте по специальному запросу (например, In [545]: b?)?

Выберите один ответ:

a) Интроспекция;

b) Инспекция;

c) Справка;

d) Ревью;

4 Установите соответствие между командой IPython и ее описанием:

| Команда | Описание |
|---|---|
| 1) %alias alias_name cmd | А) Выполнить команду в оболочке системы |
| 2) !cmd | Б) Определить псевдоним команды оболочки |

| | |
|---|---|
| 3) %pwd | В) Напечатать историю посещения каталогов |
| 4) %dhist | Г) Вернуть текущий рабочий каталог |
| 5) %env | Д) Вернуть переменные среды в виде словаря |

Выберите вариант ответа:

a) 1-Б, 2-А, 3-Г, 4-В, 5-Д;

b) 1-А, 2-Б, 3-В, 4-Г, 5-Д;

c) 1-В, 2-Б, 3-А, 4-Д, 5-Г;

d) 1-Д, 2-В, 3-Г, 4-А, 5-Б;

5 Каким термином обозначается вывод общей информации об объекте по специальному запросу (например, In [545]: b?)?

Выберите один ответ:

a) Интроспекция;

b) Инспекция;

c) Справка;

d) Ревью;

6 Какой командой можно вызвать отладчик?

Выберите один ответ:

a) %debug;

b) !cmd;

c) %dirs;

d) !del;

e) %pwd;

7 Для чего нужен объект ndarray в Numpy?

Выберите один ответ:

a) для представления N-мерного массива;

b) для генерации псевдослучайных чисел;

c) для реализации функционала линейной алгебры;

d) для представления векторов;

8 Установите соответствие между функцией создания объекта ndarray и его описанием:

| Функция создания объекта | Описание |
|---|---|
| 1) array | А) Аналогична встроенной функции range, но возвращает массив, а не список |
| 2) arange | Б) Преобразует входные данные в ndarray |
| 3) ones | В) Порождает массив, состоящий из одних нулей, с заданными атрибутами shape и dtype |
| 4) zeros | Г) Порождает массив, состоящий из одних единиц, с заданными |

| | |
|---|---|
| | атрибутами shape и dtype |
| 5) eye | Д) Создает новые массивы, выделяя под них память, но не инициализирует ее |
| 6) empty | Е) Создает единичную квадратную матрицу N × N |

Выберите вариант ответа:

a) 1-Б, 2-А, 3-Г, 4-В, 5-Е, 6-Д;

b) 1-А, 2-Б, 3-В, 4-Г, 5-Д, 6-Е;

c) 1-В, 2-Е, 3-А, 4-Д, 5-Г, 6-Б;

d) 1-Е, 2-Б, 3-Г, 4-В, 5-А, 6-Д;

9 Для чего нужен атрибут shape в типе данных ndarray??

Выберите один ответ:

a) кортеж, описывающий размер по каждому измерению;

b) объект, описывающий тип данных в массиве;

c) обобщенный многомерный контейнер для однородных данных;

d) объект, преобразующий входные данные в ndarray;

10 Прочтите код, сделайте предположение, что будет выведено после его компиляции:

```
import numpy as np
data = np.array([['0.9526', '-0.246' , '-0.8856'], ['0.5639', '0.2379', '0.9104']])
data.dtype
```

Выберите один ответ:

a) dtype('float64');

b) dtype('<U7');

c) dtype('str');

d) dtype('int32');

11 Какой командой можно преобразовать массив строк в массив вещественных чисел?

```
numeric_strings = np.array(['1.25', '-9.6', '42'], dtype=np.string_)
```

Выберите один ответ:

a) numeric_strings.astype(float);

b) numeric_strings.dtype(float);

c) numeric_strings(float);

d) dtype(float);

12 Прочитайте код, предположите, что произойдет при его компилировании:

```
from numpy.random import randn
names = np.array(['Bob', 'Joe', 'Will', 'Bob', 'Will', 'Joe', 'Joe'])
data = randn(7, 4)
data[~(names == 'Joe')]
```

Выберите один ответ:

a) из массива data будут выведены все значения, кроме соответствующих имени Joe;

b) из массива data будут выведены все значения, соответствующие имени Joe;

c) из массива names будут выведены все значения, соответствующие имени Joe;

d) из массива data будут удалены все значения, соответствующие имени Joe;

13 Какая из строк кода заменит в массиве данные на 3 при условии соответствия строкам с именем 'Will'?

Выберите один ответ:

a) data[names != 'Joe'] = 3;

b) data[names = 'Will'] = 3;

c) data[names != 'Will'] = 3;

d) data[names = 'Will'];

14 Что такое прихотливое индексирование в NumPy?

Выберите один ответ:

a) индексирование с помощью целочисленных массивов;

b) обращения к нужному элементу массива по его индексу;

c) обращения к нужному элементу массива через связный индекс булева массива;

d) индексирование с помощью логических типов;

15 Каким атрибутом можно транспонировать массив в Numpy?

Выберите один ответ:

a) arr.axes(1, 2);

b) arr.reshape(2,3);

c) arr.ix_;

d) arr.T;

16 Каким методом можно изменить порядок осей в массиве Numpy?

Выберите один ответ:

a) arr.swapaxes(1, 2);

b) arr.arange(16);

c) arr.reshape(2, 3);

d) arr.ix_;

17 Что делает функция np.meshgrid?

Выберите один ответ:

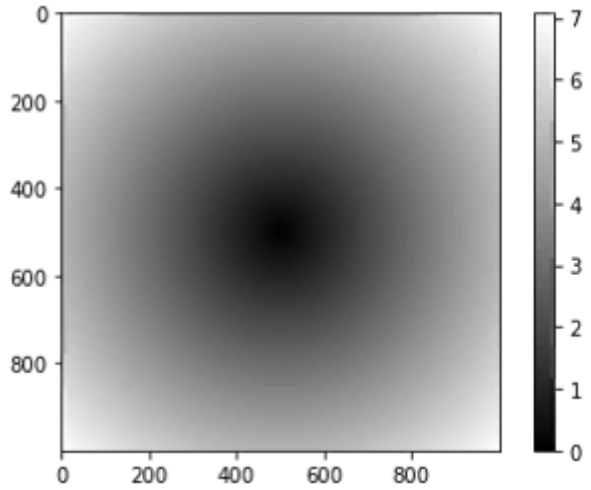
a) транспонирует массив;

b) порождает две двумерные матрицы, соответствующие всем парам (x, y) элементов;

c) индексирует с помощью целочисленных массивов;

d) производит перестановку осей массива;

18 При помощи какой команды создан график на изображении?




Выберите один ответ:

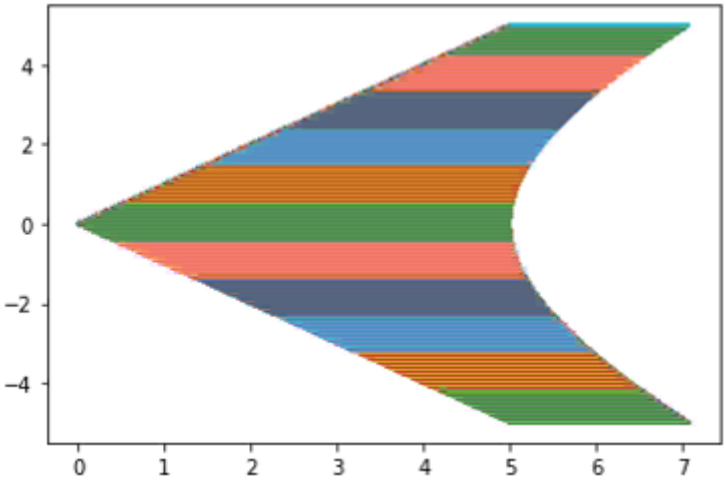
a) plt.imshow(z, cmap=plt.cm.gray);

b) plt.plot(z, xs);

c) x = np.linspace(start =-4, stop =4, num=1000);

d) plt.grid(z, xs);

19 При помощи какой команды создан график на изображении?




Выберите один ответ:

a) plt.imshow(z, cmap=plt.cm.gray);

b) plt.plot(z, xs);

c) x = np.linspace(start =-4, stop =4, num=1000);

d) plt.grid(z, xs);

20 Какому выражению соответствует функция numpy.where(x, y, 0)?

Выберите один ответ:

a) x if condition else y;

b) x else y;

c) y if condition else x;

d) x xor y;

21 Приведите в соответствие статистический метод массива и его описание:

| Метод массива | Описание |
|---|---|
| 1) sum | A) Дисперсия |

| | |
|---|---|
| 2) var | Б) Стандартное отклонение |
| 3) mean | В) Сумма элементов всего массива или вдоль одной оси |
| 4) std | Г) Среднее арифметическое |

Выберите вариант ответа:

a) 1-В, 2-А, 3-Г, 4-Б;

b) 1-А, 2-Б, 3-В, 4-Г;

c) 1-Б, 2-Г, 3-А, 4-В;

d) 1-Г, 2-В, 3-Б, 4-А;

22 Введите на месте пропуска текст (регистр не учитывается).

Укажите название встроенного метода для сортировки массива (без скобок) _____

23 Что делает операция np.unique с массивом?

Выберите один ответ:

a) возвращает отсортированное множество уникальных значений в массиве;

b) возвращает отсортированное множество значений в массиве;

c) транспонирует массив вдоль оси;

d) проверяет, есть ли в массиве хотя бы одно значение, равное True;

e) проверяет, что все значения в массиве равны True;

24 Что делает операция np.in1d с массивом?

Выберите один ответ:

a) возвращает отсортированное множество уникальных значений в массиве;

b) возвращает отсортированное множество значений в массиве;

c) проверяет, присутствуют ли значения из одного массива в другом;

d) проверяет, есть ли в массиве хотя бы одно значение, равное True;

e) проверяет, что все значения в массиве равны True;

25 Дополните код, вставьте в пропуски […] команды. Код должен генерировать объект типа ndarray:

```
import numpy as np
a = […].[…]([1, 2, 3])
a
```

Выберите несколько ответов:

a) np;

b) numpy;

c) array;

d) range;

e) dtype;

f) shape;

## 2 Выполнение практических работ

Задания для практических работ, теоретический и методический материал для самостоятельного изучения и вопросы для собеседования представлены в методических указаниях к практическим занятиям

**Раздел 1: Язык Python и его особенности**

**1. Числовые типы данных и логические значения**

1.1. Создайте переменные различных числовых типов (целые числа, числа с плавающей запятой) и произведите над ними следующие операции:

- Сложение
- Вычитание
- Умножение
- Деление
- Возведение в степень
- Целочисленное деление
- Остаток от деления

1.2. Создайте несколько переменных логического типа (bool) и проведите операции:

- Логическое И (and)
- Логическое ИЛИ (or)
- Логическое отрицание (not)
- Сравнение (>, <, ==, !=)

Часть 2. Работа со строками

2.1. Создайте строковую переменную и выполните следующие операции:

- Изменение регистра (в верхний и нижний)
- Конкатенация (сложение строк)
- Получение длины строки
- Извлечение подстроки (срез)
- Поиск подстроки в строке

Часть 3. Работа с кортежами и списками

3.1. Создайте кортеж и список с элементами различных типов данных (числа, строки, логические значения).

3.2. Произведите следующие операции:

- Добавление элементов в список
- Удаление элементов из списка
- Изменение элементов по индексу
- Получение среза списка и кортежа
- Изменение кортежа (возможно ли?)

Часть 4. Типовая структура программы

4.1. Напишите программу, которая принимает на вход два числа и выполняет операции:

- Сложение
- Вычитание
- Умножение
- Деление

- Вывод результата на экран

Инструкции:

Каждый пункт задания выполняйте в отдельных ячейках кода.

Комментируйте свой код, объясняя, что делает каждая часть.

Проверяйте результаты выполнения каждого шага перед переходом к следующему.

**2 Диалоговый режим. Чтение из файла и запись в файл**

Часть 1. Чтение из файла

1.1. Создайте текстовый файл с несколькими строками текста.

1.2. Напишите программу, которая открывает файл на чтение и выводит содержимое файла на экран.

1.3. Используйте цикл для построчного чтения файла и вывода каждой строки.

Часть 2. Запись в файл

2.1. Напишите программу, которая запрашивает у пользователя ввод текста с клавиатуры.

2.2. Затем программа открывает файл на запись и записывает введенный текст в файл.

2.3. Проверьте содержимое файла, чтобы убедиться, что текст был записан правильно.

Часть 3. Диалоговый режим

3.1. Напишите программу, которая предлагает пользователю выбрать действие: прочитать файл или записать в файл.

3.2. В зависимости от выбора пользователя, программа либо открывает файл на чтение и выводит его содержимое, либо запрашивает у пользователя ввод текста и записывает его в файл.

**Раздел 2: Основные алгоритмы и их реализация на Python**

**3 Базовые блоки линейного алгоритма. Типовые задачи с линейной структурой. Операции с числами и строками**

Часть 1. Операции с числами

1.1. Напишите программу, которая принимает на вход два числа от пользователя и выводит их сумму, разность, произведение и частное.

1.2. Проверьте программу на различных парах чисел.

Часть 2. Операции со строками

2.1. Напишите программу, которая запрашивает у пользователя строку и выводит ее длину.

2.2. Затем программа выводит строку в верхнем и нижнем регистрах.

2.3. Программа также выводит первый и последний символы строки.

Часть 3. Типовые задачи с линейной структурой

3.1. Напишите программу, которая находит сумму первых N натуральных чисел.

3.2. Напишите программу, которая находит факториал числа N.

3.3. Решите задачу: даны три числа a, b, c. Найдите их среднее арифметическое.

**4 Базовая структура операторов. Типовой алгоритм с оператором ветвления/выбора. Вложенные, полные и не полные алгоритмы ветвления**

Часть 1. Операторы ветвления

1.1. Напишите программу, которая запрашивает у пользователя его возраст.

1.2. Если возраст больше или равен 18, программа выводит сообщение "Совершеннолетний", иначе программа выводит сообщение "Несовершеннолетний".

Часть 2. Вложенные алгоритмы ветвления

2.1. Напишите программу, которая запрашивает у пользователя его возраст и проверяет:

- Если возраст меньше 0, программа выводит сообщение "Ошибка: отрицательный возраст".
- Если возраст от 0 до 18 лет включительно, программа выводит сообщение "Несовершеннолетний".
- Если возраст больше 18 лет, программа спрашивает у пользователя, работает ли он (ответ 'да' или 'нет'). Если ответ 'да', программа выводит сообщение "Работающий взрослый", иначе "Неработающий взрослый".

Часть 3. Полные и неполные алгоритмы ветвления

3.1. Напишите программу, которая запрашивает у пользователя число.

3.2. Если число делится на 2 без остатка, программа выводит сообщение "Число четное", иначе "Число нечетное".

**5 Базовые блоки циклического алгоритма. Циклы с предусловием, постусловием, параметром. Нисходящий цикл с параметром. Обработка последовательностей и одномерных массивов. Сортировка массивов**

Часть 1. Циклы

1.1. Напишите программу, используя цикл с предусловием, которая запрашивает у пользователя ввод числа и выводит все числа от 1 до введенного числа включительно.

1.2. Напишите программу, используя цикл с постусловием, которая запрашивает у пользователя ввод числа и выводит все числа от введенного числа до 1 включительно.

1.3. Напишите программу, используя цикл с параметром, которая запрашивает у пользователя количество чисел, а затем считывает указанное количество чисел и выводит их сумму.

Часть 2. Обработка последовательностей и массивов

2.1. Напишите программу, которая создает список чисел от 1 до 10 и выводит его на экран.

2.2. Напишите программу, которая создает список чисел от 1 до N (N вводится с клавиатуры) и выводит его на экран.

2.3. Напишите программу, которая находит сумму всех элементов списка.

2.4. Напишите программу, которая находит среднее арифметическое всех элементов списка.

Часть 3. Сортировка массивов

3.1. Напишите программу, которая создает список чисел и сортирует его по возрастанию.

3.2. Напишите программу, которая создает список чисел и сортирует его по убыванию.

**6 Структура матриц. Типовой алгоритм обработки двумерных массивов**

Часть 1. Создание и вывод матрицы

1.1. Напишите программу, которая создает двумерный массив (матрицу) размером 3x3 и выводит его на экран.

1.2. Модифицируйте программу, чтобы пользователь мог вводить размер матрицы (количество строк и столбцов), а затем программа создавала и выводила матрицу.

Часть 2. Обработка матрицы

2.1. Напишите программу, которая находит сумму всех элементов матрицы.

2.2. Напишите программу, которая находит сумму элементов каждой строки матрицы и выводит результат для каждой строки.

2.3. Напишите программу, которая находит сумму элементов каждого столбца матрицы и выводит результат для каждого столбца.

2.4. Напишите программу, которая транспонирует матрицу (меняет местами строки и столбцы) и выводит результат.

**Раздел 3: Двумерная графика в Python**

**7 Команды отрисовки линий, окружностей, изменения цвета, заливки и штриховки**

Часть 1. Отрисовка линий и окружностей

1.1. Напишите программу, которая отрисовывает горизонтальную линию заданной длины и цвета.

1.2. Напишите программу, которая отрисовывает вертикальную линию заданной длины и цвета.

1.3. Напишите программу, которая отрисовывает окружность заданного радиуса и цвета.

Часть 2. Изменение цвета

2.1. Напишите программу, которая отрисовывает несколько прямоугольников разных цветов на одном изображении.

2.2. Напишите программу, которая отрисовывает окружности разного радиуса и цвета.

Часть 3. Заливка и штриховка

3.1. Напишите программу, которая отрисовывает прямоугольник и заливает его заданным цветом.

3.2. Напишите программу, которая отрисовывает окружность и заполняет ее штриховкой.

**8 Модуль Tkiner. Работа с "холстом". Вывод точечного графика**

**Функции**

Часть 1. Создание холста и вывод точек

1.1. Напишите программу, используя модуль Tkinter, которая создает окно с холстом заданного размера.

1.2. На холсте отобразите координатные оси (Ox и Oy).

1.3. Напишите функцию, которая находит значения функции y=f(x) для заданных значений x.

1.4. Выведите на холст точки с координатами (x,f(x)) для нескольких значений x.

**Раздел 4: Интерактивные вычисления и среда разработки Jupiter Notebook и основы NumPy**

**9 Основы Jupiter Notebook**

Часть 1. Знакомство с интерфейсом Jupyter Notebook

1.1. Запустите Jupyter Notebook.

1.2. Создайте новый блокнот.

1.3. Ознакомьтесь с основными элементами интерфейса: ячейки кода, ячейки Markdown, панель инструментов.

Часть 2. Работа с ячейками

2.1. Создайте новую ячейку и напишите в ней простую программу на языке Python (например, вывод "Hello, world!").

2.2. Запустите ячейку и убедитесь, что программа выполняется корректно.

2.3. Создайте еще одну ячейку и напишите в ней небольшое описание вашей программы, используя язык разметки Markdown.

Часть 3. Работа с данными

3.1. Загрузите набор данных (например, csv-файл) в блокнот.

3.2. Используя библиотеку Pandas, выполните базовый анализ загруженных данных (например, выведите первые несколько строк, информацию о структуре данных, сделайте простой график).

Часть 4. Документирование кода

4.1. Добавьте комментарии к вашему коду, объясняющие его работу и основные шаги.

4.2. Вставьте изображение или ссылку на источник данных, если таковые имеются.

**10 Средства разработки программ**

Часть 1. Знакомство с интегрированными средами разработки (IDE)

1.1. Запустите выбранную IDE (например, PyCharm, Visual Studio Code, Atom).

1.2. Создайте новый проект или файл.

1.3. Ознакомьтесь с основными элементами интерфейса: редактор кода, панель инструментов, навигационные панели, окно вывода.

Часть 2. Создание и отладка программы

2.1. Напишите простую программу на выбранном языке программирования (например, Python, Java, C++).

2.2. Используйте функции и возможности выбранной IDE для выполнения отладки программы: установка точек останова, пошаговое выполнение, просмотр значений переменных и т. д.

2.3. Используйте функции автодополнения и подсказок, предоставляемых IDE, для ускорения написания кода.

**11 NumPy ndarray**

Часть 1. Установка библиотеки NumPy и знакомство с документацией

1.1. Установите библиотеку NumPy, если она еще не установлена (pip install numpy).

1.2. Ознакомьтесь с документацией NumPy и изучите основные функции и возможности библиотеки.

Часть 2. Создание массивов ndarray

2.1. Создайте одномерный массив из списка чисел.

2.2. Создайте двумерный массив из списка списков.

2.3. Создайте массив заданной формы, заполненный нулями.

2.4. Создайте массив заданной формы, заполненный случайными числами.

Часть 3. Основные операции с массивами

3.1. Используя созданные массивы, выполните базовые операции: сложение, вычитание, умножение, деление.

3.2. Выполните операции транспонирования и изменения формы массивов.

3.3. Примените функции для вычисления суммы, среднего, минимального и максимального значений массивов.

Часть 4. Индексирование и срезы

4.1. Обратитесь к элементам массивов по индексам и выполните изменение их значений.

4.2. Используйте срезы для получения подмассивов.

**12 Обработка данных с применением массивов**

Часть 1. Загрузка данных

1.1. Загрузите набор данных (например, CSV-файл) с помощью библиотеки Pandas или NumPy.

1.2. Ознакомьтесь с данными: выведите первые несколько строк, оцените структуру данных, типы признаков и т. д.

Часть 2. Предварительная обработка данных

2.1. Проведите предварительную обработку данных: удалите или заполните пропущенные значения, преобразуйте категориальные признаки в числовые и т. д.

2.2. Проведите анализ данных: выведите статистические характеристики, постройте графики распределения признаков.

Часть 3. Работа с массивами

3.1. Создайте массивы NumPy на основе обработанных данных.

3.2. Используйте массивы для выполнения различных операций: вычисление средних, медиан, минимумов и максимумов, корреляций и т. д.

Часть 4. Визуализация данных

4.1. Используйте библиотеки Matplotlib или Seaborn для визуализации данных: постройте гистограммы, диаграммы рассеяния, ящики с усами и т. д.

4.2. Анализируйте полученные графики, пытаясь выявить зависимости и закономерности в данных.

**13 Линейная алгебра и генерация случайных чисел**

Часть 1. Линейная алгебра

1.1. Создайте несколько векторов и матриц различных размеров с использованием библиотеки NumPy.

1.2. Выполните основные операции линейной алгебры: сложение векторов, умножение вектора на число, сложение матриц, умножение матрицы на число и на другую матрицу.

1.3. Выполните операции транспонирования и обращения матрицы.

1.4. Решите систему линейных уравнений Ax=b, где A - матрица коэффициентов, x - вектор неизвестных, b - вектор свободных членов.

Часть 2. Генерация случайных чисел

2.1. Используйте функции библиотеки NumPy для генерации случайных чисел различных распределений: равномерного, нормального, экспоненциального и т. д.

2.2. Сгенерируйте случайную выборку заданного размера и выведите ее статистические характеристики (среднее, дисперсию и т. д.).

2.3. Постройте графики распределения случайных чисел.

Часть 3. Написание собственного генератора случайных чисел

3.1. Напишите функцию для генерации равномерно распределенных случайных чисел в заданном диапазоне.

3.2. Используйте вашу функцию для генерации случайной выборки заданного размера.

3.3. Сравните результаты, полученные с помощью вашего генератора, с результатами, полученными с помощью библиотечных функций NumPy.

## 3 Вопросы к экзамену для промежуточной аттестации

1 Условный оператор и каскадная условная конструкция в Python

2 Строки в Python. Методы find, count, replace, strip, upper, lower и другие.

3 Строки в Python. Срезы с двумя и тремя параметрами

4 Кортежи в Python. Создание и использование. Кортежи переменных

5 Списки в Python. Методы списков и операции со списками

6 Срезы списков. Присваивание в срез. Проблема копирования списка

7 Списки в Python. List comprehensions: генерация списков.

8 Цикл for и его особенности в Python. Функция range()

9 Двумерные массивы (списки списков). Вложенная генерация.

10 Классы в Python. Перегрузка операторов.

11 Классы в Python. Конструктор класса.

12 Классы в Python. Наследование.

13 Оператор yield и функции-генераторы.

14 Команды отрисовки линий, окружностей, изменения цвета, заливки и штриховки.

15 Работа с "холстом".

16 Вывод точечного графика функции

## 4 Задачи к экзамену для промежуточной аттестации

### 1 Тысяча чертят

Зима самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Саша любит сказочные миры и истории про драконов, спасающих принцев от ужасных принцесс, поэтому он постоянно ходит в кино на мультфильмы.

Проблема в том, что на такие сеансы всегда приходит куча детей. Если заранее занять своё место, то каждый проходящий мимо тебя ребёнок может облить тебя колой. А если зайти в зал самым последним, то каждый ребёнок, мимо которого ты будешь протискиваться к своему месту, будет норовить пнуть тебя. Поэтому в последнее время Саша, заходя в зал, смотрит, какие места уже заняты, и в зависимости от этого принимает решение: ждать в проходе или идти к своему месту.

Саша знает, что

- если в момент, когда он заходит в зал, какое-то место ещё свободно, то с вероятностью $p$% до начала сеанса его никто так и не займёт;
- зрители заходят в зал по одному и в совершенно случайном порядке;
- все зрители, в том числе и Саша, проходят к своему месту с левого края ряда;
- опоздавших зрителей в зал не пускают. Поэтому если в момент начала сеанса Саша всё ещё стоит в проходе, то он понимает, что пропускать больше никого не придётся и садится на своё место.

Учитывая всё это и видя, какие места в его ряду уже заняты, Саша хочет выработать стратегию, которая **минимизировала бы сумму количества зрителей, через которых он пройдёт к своему месту, и количества зрителей, которые пройдут к своему месту через него**. Определите, чему равняется матожидание искомой суммы при оптимальной стратегии.

**Исходные данные**

В первой строке даны целые числа $n$ и $p$ – количество мест в ряду и вероятность в процентах, что свободное место до начала сеанса никто не займёт ($1 \le n \le 1\,000$; $0 \le p \le 100$). Во второй строке записаны $n$ символов, описывающих занятые места в ряду в тот момент, когда Саша заходит в зал. $i$-й символ равен 0, если $i$-е слева место свободно, 1, если оно занято, и 2, если это место Саши. Гарантируется, что в строке есть ровно один символ 2.

**Результат**

Выведите искомое матожидание с абсолютной или относительной погрешностью не более $10^{-6}$.

**Примеры**

| исходные данные | результат |
|---|---|
| 5 0<br>01120 | 2.500 |
| 3 50<br>020 | 0.375 |

**Замечания**

В первом примере оптимальная стратегия – ждать, пока не сядет пятый зритель (считая слева), и занимать своё место сразу после этого. Если пятый зритель придёт раньше первого, то Саше придётся пройти через двух зрителей, если позже – через трёх зрителей. Вероятность каждого из этих событий составляет 0.5, поэтому искомое матожидание равняется $0.5 \cdot (2 + 3) = 2.5$.

Во втором примере нужно ждать, пока не придёт третий зритель или не начнётся сеанс. С вероятностью 0.375 следующим в зал зайдёт первый зритель (и Саше придётся пройти через него), с вероятностью 0.375 – третий зритель, и с вероятностью 0.25 никто из них не придёт до начала сеанса.

**2 Все дороги ведут в сугроб**

Зима самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Для Фёдора сегодняшний вечер особенный – впервые за много лет в городе даёт свой концерт легендарная немецкая рок-группа Decline. И, как назло, именно в этот вечер повалил снег, основательно замедлив движение по всем дорогам. А ехать до концертного зала Фёдору нужно через весь город. Немного спасает ситуацию снегоуборочная техника, после проезда которой движение по

дорогам заметно оживляется. Правда, во время уборки дороги проезд по ней закрывают.

В нормальную погоду $i$-ю дорогу можно проехать за время $t_i$. В снегопад время проезда по ней зависит от количества снега на ней. Если выехать на $i$-ю дорогу через время $T$ после окончания последней её уборки (или после начала снегопада, если эту дорогу ещё не убирали), то время проезда по ней равняется $\min(\lceil(1 + T/100) \cdot t_i\rceil, 100500 \cdot t_i)$ ($\lceil x\rceil$ – округление числа $x$ вверх).

Фёдору удалось найти в интернете полный план уборки дорог. Он хочет выстроить свой маршрут так, чтобы никогда не заезжать на дорогу во время её уборки и успевать съехать с неё до начала следующей уборки. Фёдор может ждать на перекрёстках момента, когда он сможет проехать. Теперь ему нужно выбрать маршрут, при котором он попадёт на концерт как можно раньше, чтобы успеть занять места в первых рядах толпы фанатов.

**Исходные данные**

В первой строке даны целые числа $n$ и $m$ – количество перекрёстков и дорог между ними соответственно ($2 \le n \le 10^5$; $1 \le m \le 10^5$). Дом Фёдора находится на первом перекрёстке, а концертный зал – на $n$-м. В $i$-й из следующих $m$ строк описана $i$-я дорога в виде целых чисел $a_i$, $b_i$ и $t_i$ – номеров перекрёстков, которые она соединяет, и времени проезда по ней в нормальную погоду ($1 \le a_i, b_i \le n$; $1 \le t_i \le 10^6$). Все дороги в городе двусторонние, между любыми двумя перекрёстками есть не более одной дороги, никакая дорога не соединяет перекрёсток сам с собой. По дорогам можно доехать от дома Фёдора до концертного зала.

В следующей строке записано целое число $k$ ($1 \le k \le 10^5$). Следующие $k$ строк содержат план уборки дорог. В $i$-й из них записаны целые числа $p_i$, $s_i$ и $f_i$ – номер дороги и время начала и окончания её уборки ($1 \le p_i \le m$; $0 \le s_i < f_i \le 10^9$). За вечер одну и ту же дорогу могут убирать несколько раз, но одну её уборку всегда заканчивают строго раньше, чем начинают следующую.

Фёдор выезжает из дома в момент начала снегопада. Все времена измеряются в минутах и отсчитываются от времени начала снегопада.

**Результат**

Выведите единственное целое число – минимальное время, за которое Фёдор сможет доехать до концертного зала.

**Пример**

| исходные данные | результат |
|---|---|
| 4 3<br>1 2 10<br>2 3 10<br>3 4 10<br>1 | 38 |

| | |
|---|---|
| 2 10 15 | |

**Замечания**

В примере Фёдор проезжает по первой дороге за 10 минут, затем ждёт 5 минут на втором перекрёстке, пока закончат убирать вторую дорогу, после чего проезжает эту дорогу также за 10 минут. Через 25 минут после старта он выезжает на третью дорогу, которую проезжает за 13 минут (из-за снега, выпавшего на неё за 25 минут снегопада).

**3 Естественный отбор**

Зима в городе самое длинное время года. И каждый коротает долгие зимние вечера по-своему. Никите же в этом году не до развлечений. Ещё бы – ведь ему всего через полгода поступать на матмех. Поэтому Никита уже сейчас выбирает для себя специальность, куда пойти учиться. В последнее время их появилось на матмехе столько, что очень трудно сравнить их все между собой и сделать правильный выбор.

Для начала Никита узнал, какие предметы преподают на каждой из специальностей. После этого он хочет выбрать пару предметов *A* и *B* и разбить все специальности на четыре множества (некоторые из которых могут быть пустыми):

1) Специальности, на которых преподают как предмет *A*, так и предмет *B*.

2) Специальности, на которых преподают предмет *A*, но не преподают предмет *B*.

3) Специальности, на которых не преподают предмет *A*, но преподают предмет *B*.

4) Специальности, на которых не преподают ни предмет *A*, ни предмет *B*.

Никите хочется, чтобы размер самого большого множества из этих четырёх был как можно меньше. Какие предметы *A* и *B* для этого он должен выбрать?

**Исходные данные**

В первой строке даны целые числа $n$ и $m$ – количество специальностей на матмехе и количество преподаваемых предметов ($4 \le n, m \le 100$). В следующих $n$ строках записано по $m$ чисел 0 или 1. $j$-е число в $i$-й строке равно единице, если на $i$-й специальности преподают $j$-й предмет, и нулю в противном случае.

**Результат**

В первой строке выведите размер максимального множества из описанных четырёх при оптимальном разбиении. Во второй строке выведите два различных целых числа в пределах от 1 до $m$ – номера предметов, которые должен выбрать Никита. Если задача имеет несколько оптимальных решений, можно вывести любое из них.

**Пример**

| **исходные данные** | **результат** |
|---|---|

| | |
|---|---|
| 6 4<br>0 0 0 1<br>0 0 1 0<br>0 1 1 1<br>1 1 1 1<br>0 0 0 0<br>1 0 0 1 | 2<br>1 3 |

**Замечания**

Выбор предметов 1 и 3 разбивает все специальности на следующие множества: {4}, {6}, {2, 3}, {1, 5}.

**4 Паровозики**

Юля в университете предпочитала математику программированию, поэтому максимум может написать сортировку пузырьком на паскале, но зато она – серьёзный менеджер в крупной ИТ-компании. Как и у любого менеджера, у Юли бывают совещания. Не все совещания одинаково полезны, поэтому Юля очень ценит игры на телефоне, в которые можно сыграть за 1-2 минуты.

Так в Юлином телефоне поселились «Паровозики». Игра очень простая: на экране изображена карта железных дорог, все дороги оканчиваются разноцветными станциями или депо, на пересечениях дорог – стрелки. Из единственного депо один за другим выходят паровозики. Вовремя переводя стрелки, нужно направить каждый паровозик на станцию соответствующего цвета. Стоит на секунду отвлечься, и вот уже паровозики, как разноцветные тараканы, разбегаются в разные стороны, а ты не понимаешь, за что схватиться. В общем, нужно очень быстро и сосредоточенно тыкать в экран.

Как-то выглянув у Юли из-за плеча, паровозики увидел Дэнчик. Дэнчик, в отличие от Юли, опытный АСМ-щик, поэтому он не хочет бездумно тыкать в экран, он хочет **алгоритм**. Сказано – сделано, через полчаса и три шота Б-52 в паровозики уже успешно играл написанный Дэнчиком бот.

Будь как Дэнчик! Реализуй алгоритм игры в паровозики. При этом помни, что если Юля как менеджер умеет планировать наперёд и переключает стрелки заранее, то Дэнчика мотивирует только приближающийся дедлайн, так что его бот переводит стрелку в тот момент, когда на ней оказывается паровозик.

**Исходные данные**

В первой строке даны два числа $N$ и $M$ ($2 \le N, M \le 500$), далее идут $2 \times N-1$ строк по $2 \times M-1$ символов каждая – описание карты. На карте приняты следующие обозначения:

- ’S’ – депо;
- ’X’ – станция;
- ’F’, ’L’, ’R’ – стрелка;

- ’|’ (вертикальная черта, десятичный ASCII-код 124), ’-’ (дефис, десятичный ASCII-код 45) – прямые участки пути;

- ’.’ (точка, десятичный ASCII-код 46) – точка вне железной дороги.

Гарантируется, что карта представляет из себя дерево, внутренними вершинами которого являются стрелки, листьями – депо или станции, причём депо ровно одно. Рёбрами дерева являются прямые участки пути (одно ребро – это один символ ’|’ или ’-’).

Положение стрелки задаёт направление, в котором через нее поедет поезд, стрелка не может направить поезд в точку вне железной дороги. Буква, указанная на карте на месте стрелки, показывает её начальное положение. Буква ’F’ означает, что поезд по дороге из депо, пересекая стрелку, поедет прямо, ’R’ – повернёт направо, ’L’ – повернёт налево.

В следующей строке дано одно число $Q$ ($1 \le Q \le 2 \times 10^5$) – количество поездов, выезжающих из депо. Каждая из следующих $Q$ строк содержит 3 числа $T_i$, $X_i$, $Y_i$ ($1 \le T_i \le 10^9$, $1 \le X_i \le N$, $1 \le Y_i \le M$) – время выхода $i$-го поезда из депо и координаты станции назначения. Поезда упорядочены в порядке возрастания времени выхода из депо.

**Результат**

В первой строке выведите единственное число $R$ ($0 \le R \le 2 \times 10^5$) – минимальное количество необходимых переключений стрелок.

Каждая из следующих $R$ строк должна содержать описание одного переключения – три числа $T_i$, $X_i$, $Y_i$ и один символ $C_i$ ($C_i \in$ {’F’, ’L’, ’R’}), отделённый от $Y_i$ ровно одним пробелом. Строка $T_i\ X_i\ Y_i\ C_i$ указывает на то, что в момент времени $T_i$ стрелку с координатой ($X_i$, $Y_i$) нужно переключить в сторону, соответствующую движению поезда: ’F’ – прямо, ’R’ – вправо, ’L’ – влево. Допускается переключение нескольких стрелок в один момент времени.

Гарантируется, что существует хотя бы одно решение с количеством переключений не более $2 \times 10^5$. Если существует несколько решений – выведите любое из них.

**Примеры**

| **исходные данные** | **результат** |
|---|---|
| 3 3<br>S-F-X<br>..\|..<br>L-R-R<br>\|...\|<br>X.X-R<br>4<br>1 1 3<br>2 3 1<br>4 1 3<br>6 3 2 | 4<br>3 1 2 R<br>5 1 2 F<br>7 1 2 R<br>8 2 2 L |

| | |
|---|---|
| 2 3<br>S-F-X<br>..\|..<br>..X..<br>3<br>1 2 2<br>2 1 3<br>4 1 3 | 2<br>2 1 2 R<br>3 1 2 F |

**Замечания**

Рассмотрим второй пример подробнее. На карте расположено депо, две станции, три участка пути и стрелка. Положение стрелки в начальный момент времени таково, что поезд через нее поедет прямо. Всего у стрелки два возможных положения – прямо и направо. В момент времени 1 из депо покажется поезд. За единицу времени поезд пройдёт единицу пути и в момент времени 2 окажется на стрелке. Чтобы попасть в пункт назначения, поезду нужно на стрелке повернуть направо, так что стрелку нужно переключить в положение «направо». Получаем первое переключение: 2 1 2 R.

В тот же момент времени 2 из депо покажется второй поезд. В момент времени 3 он будет находиться на стрелке, причём ему нужно ехать прямо. Так что стрелку придётся снова переключить. Получаем второе переключение: 3 1 2 F.

Третий поезд будет двигаться в том же направлении, что и второй, и стрелку переключать ему не нужно.

**5 Space Invader**

Женя создаёт по вечерам компьютерные игры. Нынешняя его игра имеет весьма простой игровой процесс – пользователь управляет космическим кораблём, летящим в пространстве. Для тестирования механики движения корабля Женя хочет решить такую задачу: может ли его корабль, двигаясь по прямой, пролететь через точки *A* и *B*, после чего совершить поворот на 90° и, продолжая двигаться по прямой, пролететь ещё через точки *C* и *D*. Точки *A*, *B*, *C*, *D* должны быть посещены кораблём именно в таком порядке, точка поворота может совпадать с точкой *B* или точкой *C*. Корабль следует считать материальной точкой.

**Исходные данные**

В четырёх строках дано по три целых числа $x_i$, $y_i$, $z_i$ – координаты точек *A*, *B*, *C*, *D*, соответственно ($-10^6 \le x_i, y_i, z_i \le 10^6$). Все точки попарно различны.

**Результат**

Если корабль может пролететь через данные точки, выведите «Valid», иначе выведите «Invalid».

**Пример**

| **исходные данные** | **результат** |
|---|---|

| | |
|---|---|
| -2 0 0<br>-1 0 0<br>0 1 0<br>0 2 0 | Valid |

**6 Гусеницы**

Юный садовод давно не приезжал в сад, и поэтому дела там плохи: в саду появилось $n$ гусениц.

Приехал Кирилл в сад и решил повеселиться с гусеницами: устроить соревнование – «гусеничный заполз».

По команде Кирилла все гусеницы стартуют из своих домиков на земле и ползут по дереву. Но гусеницы довольно быстро устают. Каждая гусеница после того, как проползёт $t_i$ см, отдыхает $t_i$ минут, в это время гусеница сползает вниз по дереву. Скорость любой гусеницы – 1 см/мин, скорость сползания – тоже 1 см/мин.

Кириллу интересно, на какой высоте находится гусеница-лидер в фиксированные моменты времени.

**Исходные данные**

В первой строке записано одно целое число $n$ – количество гусениц ($1 \le n \le 10^6$).

Во второй строке записаны $n$ целых чисел $t_i$, характеризующих гусениц ($1 \le t_i \le 10^9$).

В третьей строке записано одно число $q$ – количество моментов времени, интересующих Кирилла ($1 \le q \le 10^6$).

Затем, по одному в строке, описаны запросы Кирилла. Каждый запрос описывается одним числом $x_i$ – количество минут, прошедших с начала соревнования ($1 \le x_i \le 10^6$).

**Результат**

На каждый запрос выведите одно число в отдельной строке – высоту, на которой находится наивысшая гусеница, в соответствующие моменты времени.

**Пример**

| **исходные данные** | **результат** |
|---|---|
| 4<br>1 3 2 1<br>12<br>1<br>2<br>3<br>4<br>5<br>6<br>7<br>8 | 1<br>2<br>3<br>2<br>1<br>2<br>1<br>2<br>3<br>2<br>1 |

| | |
|---|---|
| 9<br>10<br>11<br>12 | 0 |

**7 Тяжёлый рок**

Илья – лидер самой популярной хард-рок группы на Земле. И к выходу нового сингла группа решила записать самый крутой клип из когда-либо созданных: Илья проедет по Манхэттену, стоя на крыше огромного грузовика и играя потрясающие воображение соло-партии. А в это время восторженные жители острова будут подпевать и качать головой в ритм песни. Есть только одна проблема: далеко не на всех улицах любят рок...

Напомним, что Манхэттен состоит из $n$ вертикальных и $m$ горизонтальных улиц, образующих сетку из $(n - 1)\times(m - 1)$ кварталов. Продюсер Ильи провёл исследование и выяснил, что популярность группы, во-первых, постоянна на протяжении всей улицы, и, во-вторых, её можно задать целым числом от 1 до $10^9$. Так, если рокеры проедут по улице, где их популярность равна $10^9$, то их встретят градом оваций, салютом, лазерным шоу и ящиками... предположим, апельсинового сока. Однако если они проедут по улице, где их популярность равняется 1, они немедленно будут закиданы помидорами и тухлыми яйцами. Что, разумеется, не пойдёт на пользу крутости клипа!

Итак, по задумке маршрут должен проходить из самого верхнего левого перекрёстка в самый нижний правый. Определим крутость маршрута как минимальную популярность группы среди всех улиц, по которым рокеры проедут ненулевое расстояние. Как вы уже, наверное, догадались, музыканты хотят найти маршрут с самой большой крутостью. Если поможете им, Илья даже даст вам свой автограф!

**Исходные данные**

В первой строке через пробел даны целые числа $n$ и $m$ ($2 \le n, m \le 10^5$) – количества вертикальных и горизонтальных улиц соответственно.

В следующих $n$ строках по одному в строке даны значения популярности группы на вертикальных улицах в порядке слева направо.

В следующих $m$ строках по одному в строке даны значения популярности группы на горизонтальных улицах в порядке сверху вниз.

Гарантируется, что все значения популярности группы – целые числа от 1 до $10^9$.

**Результат**

Выведите целое число – максимально возможную крутость маршрута.

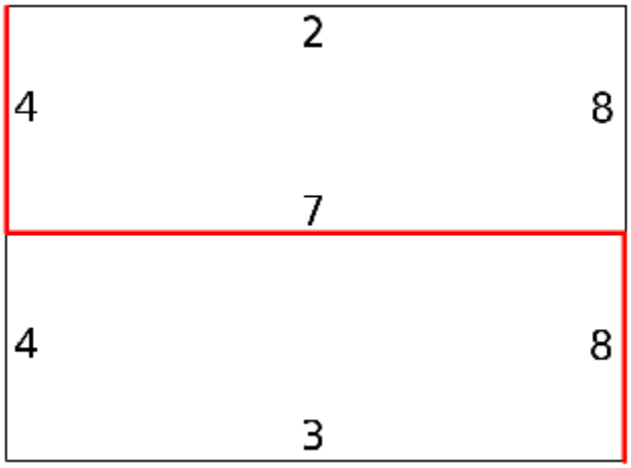
**Примеры**

| **исходные данные** | **результат** |
|---|---|
| 2 3 | 4 |

| | |
|---|---|
| 4<br>8<br>2<br>7<br>3 | |
| 4 3<br>12<br>4<br>12<br>3<br>21<br>5<br>16 | 12 |

**Замечания**

Пояснение к первому примеру (выделен самый «крутой» маршрут):




**8 Интересные числа**

В свободное время Коля и Ася вместе исследуют числа. Коля считает число интересным, если число простое, а Ася считает число интересным, если простым является количество положительных делителей числа (например, у числа 1 один делитель, а у числа 10 – четыре делителя).

Коля и Ася радуются, когда по поводу какого-то числа их вкусы совпадают, и расстраиваются, когда одному из них число кажется интересным, а другому – нет. В эти выходные они собрались исследовать числа на отрезке [*L*; *R*] и спрашивают вас, сколько там чисел, которые окажутся либо интересными, либо неинтересными одновременно для них обоих?

**Исходные данные**

В единственной строке ввода записаны два целых числа *L* и *R* ($2 \le L \le R \le 10^{12}$).

**Результат**

Выведите одно целое число – количество чисел на отрезке от *L* до *R* (включительно), по поводу которых вкусы Коли и Аси совпадают.

**Примеры**

| исходные данные | результат |
|---|---|
| 3 7 | 4 |
| 2 2 | 1 |
| 77 1010 | 924 |

**9 Классификатор Тимуса**

Вова составил классификатор задач с сайта Timus Online Judge по темам. Все темы структурированы в виде дерева – то есть каждая тема может подразделяться ещё на несколько тем. Например, тему «геометрия» Вова разбил на две темы: «геометрия на идею» и «геометрия на технику». Для каждой темы указаны примеры задач на неё. Одна задача может относиться сразу к нескольким темам. Скажем, задача 1065 относится сразу к темам «графы» и «геометрия на технику».

Требуется по заданному классификатору для каждой задачи выдать список всех тем, к которым она относится.

**Исходные данные**

В этой задаче не будет строгого описания формата ввода и вывода. Вы можете понять формат, изучив пример в условии. Мы укажем лишь следующие ограничения на входные данные.

1) Всего входные данные содержат не более 20 тем и не более 20 различных номеров задач. К некоторым темам могут относиться сразу все задачи, а к некоторым – не относиться ни одной задачи.

2) Все номера задач – числа в пределах от 1000 до 9999. Список задач, относящихся к теме, может быть записан в несколько строк, но в каждой из этих строк есть хотя бы один номер задачи. В списке не фигурирует дважды номер одной и той же задачи.

3) Названия тем – непустые строки длиной не более 20, состоящие из строчных латинских букв и пробелов. Первый и последний символ названия – не пробел, двух пробелов подряд в названии также быть не может.

5) Код темы – последовательность вида <число>.<число>. ... <число>. Все числа в коде лежат в пределах от 1 до 20 и не содержат ведущих нулей. Коды всех тем различны.

6) Первая тема имеет код «1.»

7) Если тема *A* расположена до темы *B*, то либо код *B* содержит код *A* как префикс, либо первое слева число, в котором различаются коды *A* и *B*, меньше у кода *A*.

8) Если код темы имеет вид «X.» либо оканчивается на «.X.», где в обоих случаях X больше единицы, то обязательно где-то ранее описана

тема с кодом, который отличается от кода данной только заменой суффикса «X.» на суффикс «Y.», где Y = X − 1.

9) Если код темы оканчивается на «.1.», то непосредственно перед ней описана тема, код которой получается из кода данной отбрасыванием «1.» в конце.

**Результат**

Перечислите для каждой задачи список её тем, как показано в примере. Описание темы должно содержать её собственное название и название всех тем, являющихся предками этой темы в дереве. Если задача относится к нескольким темам, то описание этих тем нужно вывести через запятую и пробел. Если у нескольких задач абсолютно одинаковый набор тем, то нужно склеить их описания, перечислив номера задач через запятую и пробел.

Номера задач слева от двоеточия должны быть перечислены в порядке возрастания, описания тем справа от двоеточия также должны быть выведены в лексикографически возрастающем порядке. Все строки в ответе нужно также упорядочить лексикографически по возрастанию.

**Пример**

| **исходные данные** | **результат** |
|---|---|
| 1. graphs<br>1022<br>1065<br>2. geometry<br>1020<br>2.1. idea<br>1130<br>2.2. technique<br>1111 1065<br>1265<br>2.2.1. convex hull<br>1271<br>3. dynamic programming<br>1244 | 1020 : geometry<br>1022 : graphs<br>1065 : geometry.technique, graphs<br>1111, 1265 : geometry.technique<br>1130 : geometry.idea<br>1244 : dynamic programming<br>1271 : geometry.technique.convex hull |

**10 Волшебный программист**

В одной известной ИТ-компании есть множество проектов. К сожалению, во многих из них дела в разработке складываются не лучшим образом: срываются сроки, система не выдерживает нагрузку, пользователи недовольны. К счастью, в компании работает Волшебный программист Иван. Он способен прийти в проект и за короткий срок вывести разработку на принципиально другой уровень, и всё начинает работать как часы. После этого ему нет смысла оставаться в текущем проекте, и его направляют туда, где больше всего проблем.

Но у Ивана есть и свои требования: он хочет не только спасать проекты, но и изучать новые технологии. На данный момент Ивана интересуют $m$ ещё не изученных им технологий. Иван играл в ACM ещё в те времена, когда программе выделялся 1Мб оперативной памяти, а в отведённую секунду времени работы можно было успеть совсем немногое, поэтому он привык ценить время. Иван решил, что раз уж ему приходится путешествовать из проекта в проект, то за время работы в компании он, во-первых, должен поработать со всеми интересующими его технологиями, а во-вторых, не должен работать с одной и той же технологией в двух разных проектах – это скучно и лишило бы его столь необходимых в работе эмоций новизны. Также, разумеется, он не пойдёт в один и тот же проект дважды.

Всего в компании $n$ проектов. В силу различных причин существует лишь $n-1$ пара проектов, между которыми возможен прямой переход (как в одну, так и в другую сторону). При этом из любого проекта можно попасть в любой другой посредством нескольких прямых переходов, не посещая один и тот же проект дважды, но только единственным способом. В каждом проекте используется некоторое подмножество интересующих Ивана технологий. Помогите Ивану выбрать первый проект и составить план переходов между проектами таким образом, чтобы он не возвращался в один и тот же проект дважды, а также смог ровно один раз попробовать каждую из $m$ технологий.

**Исходные данные**

В первой строке через пробел записаны два целых числа $n$ и $m$ ($1 \le n, m \le 10^5$).

Следующие $n-1$ строк содержат описания разрешённых прямых переходов между проектами. В $j$-й строке через пробел записаны два различных числа $u_j$ и $v_j$ ($1 \le u_j, v_j \le n$) – номера проектов, между которыми возможен прямой переход. Гарантируется, что заданная система переходов удовлетворяет условиям, описанным выше.

Следующие $n$ строк описывают интересующие Ивана технологии, применяемые в каждом из проектов. $i$-я строка начинается с числа $a_i$ ($0 \le a_i \le m$), равного количеству применяемых интересных Ивану технологий в проекте с номером $i$. Затем в этой же строке через пробел записаны $a_i$ различных целых чисел от 1 до $m$ – номера этих технологий.

Гарантируется, что сумма всех $a_i$ не превосходит $10^6$.

**Результат**

Если не существует плана переходов между проектами, удовлетворяющего требованиям Ивана, выведите единственную строку «No solution» (без кавычек).

В противном случае выведите два числа $s$ и $t$ – номера первого и последнего проектов в подходящем плане переходов.

Если решений несколько, выведите любое.

**Примеры**

| исходные данные | результат |
| --- | --- |
| 5 3<br>1 2<br>1 3<br>3 4<br>3 5<br>1 1<br>1 3<br>0<br>2 1 3<br>1 2 | 2 5 |
| 3 3<br>1 2<br>1 3<br>2 2 3<br>3 1 2 3<br>1 3 | 2 2 |
| 2 3<br>1 2<br>2 1 2<br>2 1 3 | No solution |

**Замечания**

Во втором примере Ивану нет смысла делать ни одного перехода, поэтому всю свою карьеру в компании он проработает в проекте 2, где и попробует все технологии. Соответственно, номера первого и последнего проектов совпадают.