

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

**УТВЕРЖДАЮ**

Зав. кафедрой ИСТ

\_\_\_\_\_ Данилов О. Ф.

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

дисциплины: **Теоретические основы программирования**

направление подготовки: **09.04.04 Программная инженерия**

направленность (профиль): **Программная инженерия систем искусственного интеллекта**

форма обучения: **очная**

Фонд оценочных средств рассмотрен на заседании кафедры Интеллектуальных систем и технологий для направления 09.04.04 Программная инженерия направленность (профиль) Программная инженерия систем искусственного интеллекта

## 1. Формы аттестации по дисциплине

1.1. Форма промежуточной аттестации: экзамен

Способ проведения промежуточной аттестации: устный опрос

1.2. Формы текущей аттестации:

Таблица 1.1

№ п/п	Форма обучения
	ОФО
1	Вопросы к защите отчетов по лабораторным работам
2	Тестирование
3	Коллоквиум
4	Защита домашней работы

## 2. Результаты обучения по дисциплине, подлежащие проверке при проведении текущей и промежуточной аттестации

Таблица 2.1

№ п/п	Структурные элементы дисциплины		Код результата обучения по дисциплине	Оценочные средства	
	Номер раздела	Дидактические единицы (предметные темы)		Текущая аттестация	Промежуточная аттестация
1.	1	Начальные сведения о языках программирования, их классификация. Обзор языков высокого уровня. Их сравнительная характеристика. Системы программирования.	31 У1 В1	Вопросы к защите отчета по лабораторным работам, тесты	Вопросы к экзамену
2.	2	Основные понятия и средства алгоритмических языков программирования. Определение переменной. Типы данных и функции вывода. Выражения. Структура программы. Линейные, разветвляющиеся и циклические алгоритмы. Программная реализация алгоритмов на языке Python.	31 У1 В1	Вопросы к защите отчета по лабораторным работам, вопросы к коллоквиуму	Вопросы к экзамену
3.	3	Массивы: понятие одномерных и многомерных массивов, синтаксис описания. Формирование двумерных массивов. Строки, перечисления. Структуры. Файлы. Программная реализация структурированных типов данных на языке Python.	31 У1 В1	Вопросы к защите отчета по лабораторным работам	Вопросы к экзамену
4.	4	Функциональное программирование. Параметры и аргументы функций. Локальные и глобальные переменные. Встроенные функции. Организация библиотек. Модули в Python.	31 У1 В1	Вопросы к защите отчета по лабораторным работам, домашнее задание	Вопросы к экзамену

		Подключение модуля. Использование пакетов. Примеры использования ряда модулей на языке Python.			
5.	5	Основы объектно-ориентированного программирования. Классы объектов. Разработка графического интерфейса. Виджеты и их свойства. Создание приложений с GUI. Обзор графических библиотек: Tkinter, PyQt. Программирование событий. Объекты меню в GUI. Диалоговые окна.	31 У1 В1	Вопросы к защите отчета по лабораторным работам, домашнее задание, тесты	Вопросы к экзамену

### 3. Фонд оценочных средств

3.1. Фонд оценочных средств, позволяющие оценить результаты обучения по дисциплине, включает в себя оценочные средства для текущей аттестации и промежуточной аттестации.

3.2. Фонд оценочных средств для текущей аттестации включает:

- комплект вопросов к защите отчетов по лабораторным работам - 34 шт., размещен в Приложении 1.

- примеры тестовых заданий по теме «Общие понятия программирования» - 10 шт., теме «Основы объектно-ориентированного программирования (ООП)» - 10 шт. размещены в Приложении 2.

- комплект типовых заданий для выполнения домашней работы по теме "Основы модульного и функционального программирования" - 1 шт., «Основы объектно-ориентированного программирования (ООП)» - 1 шт. размещены в Приложении 3.

- перечень вопросов к коллоквиуму по теме «Основы структурного программирования» - 15 шт., размещены в Приложении 4.

3.3. Фонд оценочных средств для промежуточной аттестации включает:

- комплект вопросов к экзамену по дисциплине «Теоретические основы программирования» - 22 шт., размещен в Приложении 5.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Комплект вопросов к защите отчетов по лабораторным работам**  
по дисциплине "Теоретические основы программирования"

**Тема 1. Общие понятия программирования.**

**Лабораторная работа №1.** Знакомство с интерфейсом интегрированной программной среды Python3.

**Цель работы:** Познакомиться со средой разработки Python. Изучить основные типы данных, команды ввода и вывода данных.

**Контрольные вопросы:**

1. Каким образом отражается знак числа при введении значения в переменную в разных системах счисления?
2. Какие операторы в языке Python существуют для того, чтобы округлить значения нецелого числа?
3. Как осуществляется перевод результата арифметического расчета в иную систему счисления?

**Тема 2. Основы структурного программирования.**

**Лабораторная работа №2.** Программная реализация линейных алгоритмов.

**Цель работы:** познакомиться с основными математическими операциями в Python.

**Контрольные вопросы:**

1. Какие типы переменных существуют в Python?
2. Какая библиотека осуществляет округление чисел?
3. Что такое импорт библиотеки?

**Лабораторная работа №3.** Составление алгоритмов и их программная реализация для разветвляющихся процессов.

**Цель работы:** Сформировать умения по разработке простых алгоритмических конструкций разветвляющегося типа.

**Лабораторная работа №4.** Составление алгоритмов и их программная реализация для циклических процессов.

**Цель работы:** Сформировать умения по программированию алгоритмических конструкций циклического типа.

**Контрольные вопросы:**

1. В чем заключается разветвляющийся алгоритм?
2. Какие операторы характеризуют разветвляющийся алгоритм?
3. Какая разница в структурах «цикл-до» и «цикл-после»?
4. Привести по два примера на каждый вид алгоритма.
5. Какие структуры могут встретиться в смешанном типе алгоритма?

**Тема 3. Структурированные типы данных.**

**Лабораторная работа №5.** Составление алгоритмов и их программная реализация для обработки одномерных массивов.

**Цель работы:** Сформировать умения по обработке одномерных массивов данных (списков).

**Лабораторная работа №6.** Составление алгоритмов и их программная реализация для обработки двумерных массивов.

**Цель работы:** Сформировать умения по обработке двумерных массивов данных (списков).

**Контрольные вопросы:**

1. Одномерные массивы в языке Pascal. Их описание. Ввод, вывод. Примеры задач.
2. Как можно осуществить разбиение в Python двумерных массивов на одномерные?
3. Каким образом можно «склеить» массивы?
4. Какие операторы осуществляют нарезку массивов?

**Лабораторная работа №7.** Составление алгоритмов и их программная реализация для работы со строками.

**Цель работы:** Сформировать умения по разработке программ в программной среде Python на реализацию операций со строками.

**Контрольные вопросы:**

1. Символьный тип данных. Стандартные функции с символьными значениями и аргументами.
2. Строковый тип данных. Описание и операции над данными этого типа. Особенности применения.
3. Перечислимые типы данных. Описание и особенности применения.

**Лабораторная работа №8.** Составление алгоритмов и их программная реализация для работы с записями и файлами.

**Цель работы:** отработка навыков работы со строковыми данными, их представлением и форматами, приобретение навыков работы с файловыми структурами.

**Контрольные вопросы:**

1. Какие операции и операторы используются при работе с файлами?
2. Необходимо ли использовать процедуру закрытия файла?
3. Какие библиотеки используются для работы с файлами?

#### **Тема 4. Основы модульного и функционального программирования.**

**Лабораторная работа №9.** Разработка программ с использованием библиотеки NumPy для реализации математических объектов и вычислений.

**Цель работы:**

**Контрольные вопросы:**

1. Какие операции можно реализовать с помощью библиотеки NumPy.
2. Как осуществляется работа с DateTime или линейной алгеброй.
3. Что представляют собой основные нарезки и расширенное индексирование в NumPy Python.

**Лабораторная работа №10.** Использование библиотеки Python для анализа данных и Data Science.

**Цель работы:**

**Контрольные вопросы:**

1. Какие библиотеки используются для анализа данных в Python?
2. Какие показатели можно анализировать с помощью библиотек анализа данных?

**Тема 5. Основы объектно-ориентированного программирования (ООП).**  
**Лабораторная работа №11.** Реализация класса, работа с объектами в Python.

**Цель работы:**

**Контрольные вопросы:**

1. Основные понятия объектно-ориентированного программирования: инкапсуляция, наследование, полиформизм.
2. Создание классов и объектов.
3. Методы класса.
4. Уровни доступа атрибута и метода.

**Лабораторная работа №12.** Разработка графического интерфейса в Python.

**Цель работы:**

**Контрольные вопросы:**

1. Tkinter – что это такое?
2. Какие библиотеки нужны для работы с использованием графики?
3. В каких средах Python можно работать с графическими файлами?

**Требования к выполнению лабораторных работ:**

Результаты выполнения лабораторной работы оформляются в виде текстового отчета.

Содержание отчета включает:

- титульный лист;
- цель и задачи выполнения работы;
- практическая и расчетная часть;
- выводы.

**Критерии оценки**

**5 баллов:**

- выполнены все задания практической (лабораторной) работы,
- обучающийся четко и без ошибок ответил на все контрольные вопросы.

**4 балла:**

- выполнены все задания практической (лабораторной) работы;
- обучающийся ответил на все контрольные вопросы с замечаниями.

**3 балла:**

- выполнены все задания практической (лабораторной) работы с замечаниями;
- обучающийся ответил на все контрольные вопросы с замечаниями.

**2 балла:**

- обучающийся не выполнил или выполнил неправильно задания практической (лабораторной) работы;
- обучающийся ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

Максимальное количество баллов за весь цикл лабораторных работ составляет 60 баллов.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Задания в тестовой форме**  
**по дисциплине «Теоретические основы программирования»**

**Тестовые материалы представлены в системе Eduson.**

**Тема 1. Общие понятия программирования.**

1. Какие характеристики можно отнести к языку программирования Python?
  1. интерпретируемый
  2. с динамической типизацией
  3. использующий препроцессор для макроподстановок
  4. для быстрой разработки приложений
2. Какие характеристики можно отнести к языку программирования Python?
  1. имеет эффективный компилятор в коды процессора
  2. использует раннее связывание
  3. объектно-ориентированный
  4. универсальный язык программирования
3. Какие характеристики можно отнести к языку программирования Python?
  1. для быстрой разработки приложений
  2. богатый и громоздкий синтаксис
  3. поощряет повторное использование кода
4. Какие характеристики можно отнести к языку программирования Python?
  1. большая стандартная библиотека модулей
  2. многоплатформенный
  3. удобен для встраивания в проекты на C/C++
5. Какие из этих утверждений о Python верны?
  1. программы на Python транслируются в машинные коды, которые затем исполняются
  2. Python использует промежуточный код
  3. язык Python применяется для быстрой разработки приложений
  4. Python имеет обширную библиотеку стандартных модулей
6. Какие парадигмы и стили программирования поддерживает Python?
  1. логическое программирование
  2. структурный стиль
  3. модульное программирование
  4. императивное программирование
7. Какие парадигмы и стили программирования поддерживает Python ?
  1. объектно-ориентированный
  2. модульный
  3. программирование в ограничениях

4. функциональное программирование
8. Какие парадигмы и стили программирования Python поддерживает?
  1. объектно-ориентированное программирование
  2. модульное
  3. программирование в ограничениях
  4. функциональное
9. Какие парадигмы и стили программирования Python поддерживает?
  1. императивное программирование
  2. функциональное программирование
  3. объектно-ориентированный подход
  4. логическое
10. Какие парадигмы Python не поддерживает?
  1. логическое программирование
  2. структурный стиль
  3. модульное программирование
  4. гибрид парадигм

## **Тема 5. Основы объектно-ориентированного программирования (ООП)**

1. Что представляет собой объектно-ориентированная программа с точки зрения теории ООП?
  1. набор инструкций для построения объектов
  2. набор объектов, которые посылают друг другу сообщения
  3. декларации типов объектов
  4. декларации классов объектов
2. Выберите правильные (с точки зрения теории ООП) утверждения:
  1. все объекты одного типа могут принимать одни и те же сообщения
  2. все объекты одного типа принадлежат одному классу
  3. каждый объект имеет тип
3. Разбиение программы на объекты называется ...
  1. абстракцией
  2. декомпозицией
  3. инкапсуляцией
  4. полиморфизмом
4. Скрытие информации о внутреннем устройстве объекта, при котором вся работа с объектом ведется только через общедоступный интерфейс называется ...
  1. абстракцией
  2. декомпозицией
  3. инкапсуляцией
  4. агрегацией
5. Что из нижеперечисленного является объектом в Python?
  1. функция
  2. метод
  3. класс
  4. оператор

6. Какие из перечисленных действий можно выполнить над любым объектом `obj` в языке Python 2.4?
1. `print id(obj)`
  2. `print obj`
  3. `hash(obj)`
  4. `not obj`
7. Какие из перечисленных действий можно выполнить над любым объектом `obj` в языке Python 2.4?
1. `print repr(obj)`
  2. `del obj`
  3. `isinstance(w, obj)`
  4. `len(obj)`
8. Как узнать, ссылаются ли два имени (`a` и `b`) на один и тот же объект?
1. `id(a) == id(b)`
  2. `a == b`
  3. `a is b`
  4. `hash(a) == hash(b)`
9. Объекты `a` и `b` принадлежат одному классу `C`. Какие из следующих выражений будут, скорее всего, иметь значение "истина" сразу после создания?
1. `isinstance(a, type(b))`
  2. `issubclass(C, C)`
  3. `dir(a) is dir(b)`
  4. `dir(a) == dir(b)`
10. Сколько общедоступных методов будет иметь экземпляр класса `ABC` и что возвратит вызов метода `a()`?
- ```
class A(object):
    def a(self): return 'a'
class B(object):
    def b(self): return 'b'
class C(object):
    def c(self): return 'c'
class AB(A, B):
    def a(self): return 'ab'
class BC(B, C):
    def a(self): return 'bc'
class ABC(AB, B, C):
    def a(self): return 'abc'
```
1. 3, возвратит `a`
  2. 3, возвратит `ab`
  3. 3, возвратит `abc`
  4. 5, возвратит `a`

#### **Критерии оценки за аттестацию**

**0-2 балла** - менее 60% правильных ответов.

**3 балла** - от 61% до 75% правильных ответов.

**4 балла** - от 76% до 91% правильных ответов.

**5 баллов** - 100% правильных ответов.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Комплект заданий для домашней работы  
по дисциплине «Теоретические основы программирования»**

**Задание 1.**

**Тема 4. Основы модульного и функционального программирования**

**Цель работы:** изучение параметров линейной регрессии с использованием библиотеки Numpy в Python.

**Задание:** Провести анализ структуры анализируемых данных и их визуализацию в случае возможности. В первой части задания предлагается использовать базу данных. Данные представляют собой множество объектов, описываемых одним признаком (численность населения в некотором городе) и меткой (прибыль, которую можно получить при продаже определенного товара в городе с соответствующей численностью населения). Необходимо написать программный код, который позволит выполнять визуализацию данных. Визуализацию данных необходимо осуществлять с использованием функции plotData. Далее необходимо вычислить значение стоимостной функции для линейной регрессии. Выполнить алгоритм градиентного спуска с целью обучения параметров модели линейной регрессии. После завершения предыдущих пунктов выполните предсказание прибыли от продажи товара в городах с численностью населения по данным, представленным преподавателем.

**Задание 2. Создание многопоточных приложений.**

**Тема 5. Основы объектно-ориентированного программирования (ООП)**

**Цель работы:** создание приложения для реализации процесса сортировки данных.

**Задание:**

1. Создать массив заполнен 100000 случайными элементами, целые числа отображенного графическим образом. Определите базовый класс, содержащий, в частности, поле – одномерный массив и методы отображения массива на экране, заполнения массива (разными способами) и абстрактный метод сортировки массива. Определите три класса-наследника, в которых реализованы следующие методы сортировки (по одному – в каждом): один из простых методов (вставки, выбора, обмена), быстрая сортировка и метод, описанный в Вашем варианте

2. Создайте приложение, которое должно не только демонстрировать все три процесса сортировки в «реальном времени», но и делать это одновременно.

3. В приложении должна быть возможность изменять приоритеты потоков (чтобы влиять на скорость сортировки). Разумеется, каждый поток будет работать со своей копией массива. Но было бы желательно показать, что произойдет, если все три потока будут работать с одним и тем же массивом (по очереди). Варианты методов сортировки представлены в таблице.

Таблица. Варианты заданий.

|            |                                             |                                                                               |
|------------|---------------------------------------------|-------------------------------------------------------------------------------|
| Вариант 1  | Сортировка вставками                        | Строго случайные данные и «почти отсортированные» случайные данные            |
| Вариант 2  | Сортировка выбором                          | Строго случайные данные и «отсортированные наоборот» случайные данные         |
| Вариант 3  | Сортировка пузырьком                        | Строго случайные данные и случайные данные с малым числом уникальных значений |
| Вариант 4  | Сортировка Шелла                            | Строго случайные данные и «почти отсортированные» случайные данные            |
| Вариант 5  | Сортировка слиянием                         | Строго случайные данные и «отсортированные наоборот» случайные данные         |
| Вариант 6  | Пирамидальная сортировка                    | Строго случайные данные и случайные данные с малым числом уникальных значений |
| Вариант 7  | Быстрая сортировка                          | Строго случайные данные и «почти отсортированные» случайные данные            |
| Вариант 8  | Быстрая сортировка с разделением на 3 части | Строго случайные данные и «отсортированные наоборот» случайные данные         |
| Вариант 9  | Сортировка вставками                        | Строго случайные данные и случайные данные с малым числом уникальных значений |
| Вариант 10 | Сортировка выбором                          | Строго случайные данные и «почти отсортированные» случайные данные            |
| Вариант 11 | Сортировка пузырьком                        | Строго случайные данные и «отсортированные наоборот» случайные данные         |
| Вариант 12 | Сортировка Шелла                            | Строго случайные данные и случайные данные с малым числом уникальных значений |
| Вариант 13 | Сортировка слиянием                         | Строго случайные данные и «почти отсортированные» случайные данные            |
| Вариант 14 | Пирамидальная сортировка                    | Строго случайные данные и «отсортированные наоборот» случайные данные         |
| Вариант 15 | Быстрая сортировка                          | Строго случайные данные и случайные данные с малым числом уникальных значений |
| Вариант 16 | Быстрая сортировка с разделением на 3 части | Строго случайные данные и «почти отсортированные» случайные данные            |

**Критерии оценки домашнего задания:****Баллы**

|                                                                                                                                                                                                                                                                                       |           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| - соответствие предполагаемым ответам;                                                                                                                                                                                                                                                | 0-3       |
| - правильное использование алгоритма выполнения действий (методики, технологии и т.д.);                                                                                                                                                                                               |           |
| - логика рассуждений;                                                                                                                                                                                                                                                                 |           |
| - неординарность подхода к решению;                                                                                                                                                                                                                                                   |           |
| выставляется студенту, если он владеет категориальным аппаратом, может привести формулы расчета, рассчитать задание;                                                                                                                                                                  | 4-5       |
| выставляется студенту, если он владеет категориальным аппаратом, может привести классификацию факторов явления, решить поставленную задачу и проанализировать полученные результаты;                                                                                                  | 6-7       |
| выставляется студенту, если он владеет категориальным аппаратом, может привести классификацию факторов явления, решить поставленную задачу и проанализировать полученные результаты, объяснить причины отклонений от желаемого результата, отстоять свою точку зрения, приводя факты; | 8-10      |
| <b>Максимальное количество баллов</b>                                                                                                                                                                                                                                                 | <b>10</b> |

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**Федеральное государственное бюджетное**  
**образовательное учреждение высшего образования**  
**«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Перечень вопросов для коллоквиума**  
**по дисциплине «Теоретические основы программирования»**

**Тема 2. Основы структурного программирования**

**Задание:** Продумать и представить графические алгоритмы и программное решение для следующих задач по вариантам.

**Вариант 1.**

В некотором государстве в обращении находятся банкноты определенных номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен.

**ВХОДНЫЕ ДАННЫЕ**

Первая строка входных данных содержит натуральное число  $N$  не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит  $N$  различных натуральных чисел  $x_1, x_2, \dots, x_N$ , не превосходящих 105 — номиналы банкнот. Третья строка содержит натуральное число  $S$ , не превосходящее 105 — сумму, которую необходимо выдать.

**ВЫХОДНЫЕ ДАННЫЕ**

Программа должна найти представление числа  $S$  в виде суммы слагаемых из множества  $x_i$ , содержащее минимальное число слагаемых и вывести это представление на экран (в виде последовательности чисел, разделенных пробелами). Если таких представлений существует несколько, то программа должна вывести любое (одно) из них. Если такого представления не существует, то программа должна вывести строку «No solution».

**ПРИМЕР**

ввод

5

1 3 7 12 32

40

5

10 50 100 500 1000

99

вывод

1 7 32

No solution

**Вариант 2.**

По данному натуральному  $n$  определите число двоичных последовательностей длины  $n$ , не содержащих трех единиц подряд.

**ВХОДНЫЕ ДАННЫЕ**

Программа получает на вход число  $n < 30$ .

**ВЫХОДНЫЕ ДАННЫЕ**

Программа должна вывести число искомых последовательностей.

**ПРИМЕР**

ввод

3

ВЫВОД

7

### Вариант 3.

Дан список чисел. Проверьте, является ли он корректно организованной кучей, в вершине которой находится наибольший элемент.

ВХОДНЫЕ ДАННЫЕ

Первая строка входных данных содержит натуральное число  $n \leq 10000$ . Вторая строка содержит  $n$ -натуральных чисел, не превосходящих 109 каждое.

ВЫХОДНЫЕ ДАННЫЕ

Если данные числа записаны в порядке, образующим корректную кучу, программа должна вывести слово «YES», иначе программа должна вывести слово «NO».

ПРИМЕР

ввод

4

4 3 1 2

4

4 2 1 3

вывод

YES

NO

### Вариант 4.

Постройте матрицу смежности по заданным спискам смежности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных содержится число  $N$  - количество вершин ( $1 \leq N \leq 100$ ). Далее идут  $N$  строк. В  $i$ -й строке содержится описание всех ребер, исходящих из  $i$ -й вершины. Описание начинается количеством исходящих ребер. Далее следуют номера вершин, в которые эти ребра идут. Все вершины нумеруются натуральными числами от 1 до  $N$ .

ВЫХОДНЫЕ ДАННЫЕ

Выведите  $N$  строк, содержащих по  $N$  чисел, равных 0 или 1, разделенных пробелами — матрицу смежности данного ориентированного графа.

ПРИМЕР

ввод

3

2 2 3

0

1 2

вывод

0 1 1

0 0 0

0 1 0

### Вариант 5.

Простой ориентированный граф задан матрицей смежности. Выведите его представление в виде списков смежности.

ВХОДНЫЕ ДАННЫЕ

В первой строке входных данных находится число  $N$  — количество вершин графа ( $1 \leq N \leq 100$ ). Далее идет  $N$  строк по  $N$  чисел в каждой, равных 0 или 1, содержащих матрицу смежности ориентированного графа из  $N$  вершин.

ВЫХОДНЫЕ ДАННЫЕ

Выведите N строк — списки смежности графа. В i-й строке сначала выведите количество исходящих из i-й вершины ребер, а затем — номера вершин, в которые эти ребра идут, упорядоченные по возрастанию.

Нумерация вершин начинается с числа 1.

**ПРИМЕР**

ввод

5

0 0 1 0 0

1 0 1 0 0

0 0 0 0 1

1 1 0 0 0

1 1 0 0 0

вывод

1 3

2 1 3

1 5

2 1 2

2 1 2

### **Вариант 7.**

Дан неориентированный невзвешенный граф. Необходимо посчитать количество его компонент связности.

**ВХОДНЫЕ ДАННЫЕ**

В первой строке входного файла содержится одно натуральное число N ( $N \leq 100$ ) — количество вершин в графе. Далее в N строках по N чисел — матрица смежности графа: в i-й строке на j-м месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

**ВЫХОДНЫЕ ДАННЫЕ**

Вывести одно целое число — количество компонент связности данного графа.

**ПРИМЕР**

ввод

6

0 1 1 0 0 0

1 0 1 0 0 0

1 1 0 0 0 0

0 0 0 0 1 0

0 0 0 1 0 0

0 0 0 0 0 0

вывод

3

### **Вариант 8.**

Дан неориентированный невзвешенный граф. Необходимо определить, является ли он деревом.

**ВХОДНЫЕ ДАННЫЕ**

В первой строке входных данных содержится одно натуральное число N ( $N \leq 100$ ) - количество вершин в графе. Далее в N строках по N чисел - матрица смежности графа: в i-ой строке на j-ом месте стоит 1, если вершины i и j соединены ребром, и 0, если ребра между ними нет. На главной диагонали матрицы стоят нули. Матрица симметрична относительно главной диагонали.

**ВЫХОДНЫЕ ДАННЫЕ**

Программа должна вывести YES, если граф является деревом, NO иначе.

### ПРИМЕР

ВВОД

6

0 1 1 0 0 0

1 0 1 0 0 0

1 1 0 0 0 0

0 0 0 0 1 0

0 0 0 1 0 0

0 0 0 0 0 0

3

0 1 0

1 0 1

0 1 0

ВЫВОД

NO

YES

### Вариант 9.

Дан ориентированный взвешенный граф. Найдите кратчайшее расстояние от одной заданной вершины до другой.

#### ВХОДНЫЕ ДАННЫЕ

В первой строке содержатся три числа:  $N$ ,  $S$  и  $F$  ( $1 \leq N \leq 100$ ,  $1 \leq S, F \leq N$ ), где  $N$  — количество вершин графа,  $S$  — начальная вершина, а  $F$  — конечная. В следующих  $N$  строках вводится по  $N$  чисел, не превосходящих 100 — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы записаны нули.

#### ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести искомое расстояние или -1, если пути между указанными вершинами не существует.

### ПРИМЕР

ВВОД

3 1 2

0 -1 2

3 0 -1

-1 4 0

ВЫВОД

6

### Вариант 10.

Ориентированный взвешенный граф задан так же, как в предыдущей задаче. Найдите кратчайший путь из вершины  $S$  в вершину  $F$ .

#### ВХОДНЫЕ ДАННЫЕ

В первой строке содержатся три числа:  $N$ ,  $S$  и  $F$  ( $1 \leq N \leq 100$ ,  $1 \leq S, F \leq N$ ), где  $N$  — количество вершин графа,  $S$  — начальная вершина, а  $F$  — конечная. В следующих  $N$  строках вводится по  $N$  чисел, не превосходящих 100 — матрица смежности графа, где -1 означает отсутствие ребра между вершинами, а любое неотрицательное число — присутствие ребра данного веса. На главной диагонали матрицы записаны нули.

#### ВЫХОДНЫЕ ДАННЫЕ

Требуется вывести последовательно все вершины одного (любого) из кратчайших путей, или одно число -1, если пути между указанными вершинами не существует.

### ПРИМЕР

ВВОД

3 1 2

0 -1 2  
3 0 -1  
-1 4 0  
ВЫВОД  
1 3 2

### **Вариант 11.**

На одном из вокзалов билеты продают  $N$  касс. Каждая касса работает без перерыва определенный промежуток времени по фиксированному расписанию (одному и тому же каждый день). Требуется определить, на протяжении какого времени в течение суток работают все кассы одновременно.

#### **ВХОДНЫЕ ДАННЫЕ**

Сначала вводится одно целое число  $N$  ( $0 < N \leq 10000$ ).

В каждой из следующих  $N$  строк через пробел расположены 6 целых чисел, первые три из которых обозначают время открытия кассы в часах, минутах и секундах (часы — целое число от 0 до 23, минуты и секунды — целые числа от 0 до 59), оставшиеся три — время закрытия в том же формате. Числа разделены пробелами.

Время открытия означает, что в соответствующую ему секунду касса уже работает, а время закрытия — что в соответствующую секунду касса уже не работает. Например, касса, открытая с 10 ч 30 мин 30 с до 10 ч 35 мин 30 с, ежедневно работает 300 секунд.

Если время открытия совпадает с временем закрытия, то касса работает круглосуточно. Если первое время больше второго, то касса начинает работу до полуночи, а заканчивает — на следующий день.

#### **ВЫХОДНЫЕ ДАННЫЕ**

Требуется вывести одно число — суммарное время за сутки (в секундах), на протяжении которого работают все  $N$  касс.

#### **ПРИМЕР**

ввод

3

1 0 0 23 0 0

12 0 0 12 0 0

22 0 0 2 0 0

2

9 30 0 14 0 0

14 15 0 21 0 0

2

14 00 00 18 00 00

10 00 00 14 00 01

ВЫВОД

7200

0

1

### **Вариант 12.**

В первый день спортсмен пробежал  $x$  километров, а затем он каждый день увеличивал пробег на 70% от предыдущего значения. По данному числу  $u$  определите номер дня, на который пробег спортсмена составит не менее  $u$  километров.

#### **ВХОДНЫЕ ДАННЫЕ**

На вход программа получает два числа  $x$  и  $u$ , записанные в одной строке. Числа положительные, действительные, не превосходят 1000, заданы с точностью до шести знаков после запятой.

#### **ВЫХОДНЫЕ ДАННЫЕ**

Программа должна вывести одно натуральное число — номер дня в который спортсмен

пробежал не менее  $u$  километров.

ПРИМЕР

ввод

10.0 30.0

1 2.89

вывод

4

3

### Вариант 13.

Определите, принадлежит ли точка данному отрезку.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход шесть чисел: координаты точки  $P$  и координаты концов отрезка  $AB$ .

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести YES, если точка принадлежит отрезку, или NO в противном случае.

ПРИМЕР

ввод

3 3

1 2

5 4

вывод

YES

### Вариант 14.

Даны два вектора. Определите, являются ли они коллинеарными (параллельными), перпендикулярными или имеют общее положение.

ВХОДНЫЕ ДАННЫЕ

В двух строках записаны координаты двух ненулевых векторов (два целых числа в каждой строке через пробел).

ВЫХОДНЫЕ ДАННЫЕ

Если данные вектора коллинеарны, выведите 1. Если вектора перпендикулярны, выведите 2. Иначе выведите 0.

ПРИМЕР

ввод

1 1

2 2

0 1

1 0

1 2

2 1

вывод

1

1

0

### Вариант 15.

Полный ориентированный взвешенный граф задан матрицей смежности. Постройте матрицу кратчайших путей между его вершинами. Гарантируется, что в графе нет циклов отрицательного веса.

ВХОДНЫЕ ДАННЫЕ

В первой строке вводится единственное число  $N$  ( $1 \leq N \leq 100$ ) – количество вершин графа. В

следующих N строках по N чисел задается матрица смежности графа (j-е число в i-й строке соответствует весу ребра из вершины i в вершину j). Все числа по модулю не превышают 100. На главной диагонали матрицы стоят нули.

#### ВЫХОДНЫЕ ДАННЫЕ

Выведите N строк по N чисел – матрицу кратчайших расстояний между парами вершин. j-е число в i-й строке должно быть равно весу кратчайшего пути из вершины i в вершину j.

#### ПРИМЕР

ввод

4

0 5 9 100

100 0 2 8

100 100 0 7

4 100 100 0

вывод

0 5 7 13

12 0 2 8

11 16 0 7

4 9 11 0

#### Критерии оценки

**8 – 10 баллов** - выставляется, если обучающийся раскрыл содержание материала в объеме, предусмотренном программой, изложил материал грамотным языком в определенной логической последовательности, точно используя терминологию данного предмета как учебной дисциплины; отвечал самостоятельно без наводящих вопросов преподавателя. Возможны одна – две неточности при освещении второстепенных вопросов или в выкладках, которые обучающийся легко исправил по замечанию преподавателя.

**6-7 баллов** - выставляется, если ответ обучающегося удовлетворяет в основном требованиям на отметку «10-8» баллов, но при этом имеет место один из недостатков: допущены одна - две неточности при освещении основного содержания ответа, исправленные по замечанию преподавателя; допущены ошибка или более двух неточностей при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию преподавателя.

**4-5 баллов** - выставляется в следующих случаях: неполно или непоследовательно раскрыто содержание материала, имеются ошибки при ответах на вопросы, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения материала, определенного учебной программой дисциплины.

**0-3 баллов** - выставляется в случаях, если не раскрыто основное содержание учебного материала; обнаружено незнание или неполное понимание обучающимся большей или наиболее важной части учебного материала; допущены грубые ошибки при ответах на вопросы опроса, допущены ошибки в определении понятий при использовании специальной терминологии в рисунках, схемах, выкладках, которые не исправлены после нескольких наводящих вопросов преподавателя.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

**Вопросы к экзамену**  
**по дисциплине «Теоретические основы программирования»**

1. Понятие алгоритма и программы. Способы записи алгоритмов.
2. Общая характеристика языков программирования и их классификация
3. Понятие о системе программирования. Трансляция программ
4. Основные конструкции языка программирования. Язык программирования Python. Структура программы
5. Понятие величины. Оператор присваивания. Объявление переменных.
6. Линейные программы. Ввод и вывод данных в языке Python.
7. Организация ввода и вывода данных в программах на языке Python.
8. Арифметические и логические операции в Python. Приоритет операций
9. Типы данных. Объявление и инициализация в Python.
10. Разветвляющиеся алгоритмы и программы. Реализация в языке Python. Примеры.
11. Оператор выбора (варианта). Реализация в Python.
12. Циклические алгоритмы и программы. Реализация в языке Python.
13. Понятие подпрограммы. Процедуры и функции. Реализация процедур и функций в языке Python.
14. Вызов метода. Соответствие формальных и фактических параметров. Способы передачи параметров. Примеры
15. Массивы в Python. Ввод и вывод элементов одномерного и двумерного массива.
16. Типовые алгоритмы обработки массивов. Примеры.
17. Парадигма объектно-ориентированного программирования. Поддержка в Python функционального программирования.
18. Объекты. Динамическая типизация. Инкапсуляция.
19. Генерация объекта class. Новое пространство имен. Объект экземпляр класса.
20. Абстрактные методы в Python. Классические классы и классы нового стиля.
21. Наследование. Базовый и производный класс. Построение производного класса. Агрегация. Контейнеры. Иерархия наследования.
22. Полиморфизм. Подмена методов в производном классе. Доступ к методам базового класса.

**Критерии оценки**

**100 – 91 баллов:** студент обладает системными теоретическими знаниями (знает методику выполнения практических навыков, показания и противопоказания, возможные осложнения, нормативы и проч.), без ошибок самостоятельно демонстрирует выполнение практических умений;

90-76 баллов: студент обладает теоретическими знаниями (знает методику выполнения практических навыков, показания и противопоказания, возможные осложнения, нормативы и проч.), самостоятельно демонстрирует выполнение практических умений, допуская некоторые неточности (малозначительные ошибки), которые самостоятельно обнаруживает и быстро исправляет;

**75-61 баллов:** студент обладает удовлетворительными теоретическими знаниями (знает основные положения методики выполнения практических навыков, показания и противопоказания, возможные осложнения, нормативы и проч.), демонстрирует выполнение

практических умений, допуская некоторые ошибки, которые может исправить при коррекции их преподавателем;

**60-0 баллов:** студент не обладает достаточным уровнем теоретических знаний (не знает методики выполнения практических навыков, показаний и противопоказаний, возможных осложнений, нормативы и проч.) и/или не может самостоятельно продемонстрировать практические умения или выполняет их, допуская грубые ошибки.