





МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

СОГЛАСОВАНО Руководитель ОП		ПРОЕКТОР ПО РАЗВИТИЮ
 Е.В. Пустовалов		 Д.И. Земцов
« 24 » июня 2018 г.		« 24 » июня 2018 г.

Р

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»
направления 09.04.01 Информатика и вычислительная техника
Магистерская программа «Технологии виртуальной и дополненной реальности»
Форма подготовки очная

курс 1 семестр 1
лекции 0 час.
практические занятия 0 час.
лабораторные работы 36 час.
всего часов аудиторной нагрузки 36 час.
самостоятельная работа 72 час.
контрольные работы программой не предусмотрены
курсовая работа/проект – не предусмотрено
зачет с оценкой 1 семестр
экзамен – не предусмотрено учебным планом

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.04.01 – Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от 30.10.2014 № 1420

Рабочая программа рассмотрена и утверждена на заседании Дирекции Школы цифровой экономики 24 июня 2018 г., протокол №2

Составитель(и): ст. пр. Кленин А.С.

Оборотная сторона титульного листа РПД

I. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____

(подпись)

(И.О. Фамилия)

II. Рабочая программа пересмотрена на заседании Дирекции Школы цифровой экономики:

Протокол от «_____» _____ 20__ г. № _____

Заместитель директора ШЦЭ

по учебной и воспитательной работе _____

(подпись)

(И.О. Фамилия)

АННОТАЦИЯ

Рабочая программа учебной дисциплины «Языки и методы программирования» предназначена для студентов, обучающихся по направлению подготовки 09.04.01 Информатика и вычислительная техника (уровень магистратуры), профиль «Технологии виртуальной и дополненной реальности».

Рабочая программа разработана на основе макета рабочей программы учебной дисциплины для образовательных программ высшего образования – программ бакалавриата, специалитета, магистратуры ДВФУ, утверждённого приказом ректора ДВФУ от 08.05.2015 № 12-13-824.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единиц или 108 часов. Дисциплина реализуется на 1 курсе в 1 семестре.

Семестр	Аудиторные занятия			Самостоятельная работа	Форма контроля	Всего по дисциплине	
	Лекции и	Лабораторные работы	Всего			Часы	Зачетные единицы
1 семестр	–	36	36	72	зачет с оценкой	108	3

Место дисциплины в структуре ОПОП

Дисциплина «Языки и методы программирования» входит в базовую часть Блока 1. «Дисциплины (модули)» основной профессиональной образовательной программы магистратуры, модуль «Машинное обучение и программирование».

Дисциплина «Языки и методы программирования» логически и содержательно связана с дисциплинами базовой и вариативной частей Блока 1. Дисциплины (модули) и является основой для изучения дисциплин «Распознавание образов и машинное обучение», «Алгоритмическая теория сложности», «Программирование мобильных приложений». Освоение данной дисциплины необходимо для выполнения практической части выпускной квалификационной работы.

Цель изучения дисциплины – формирование общекультурных компетенций будущих специалистов, работающих с технологиями виртуальной и дополненной реальности, через ознакомление с общими принципами построения и использования языков программирования, а также развитие навыков проектирования и реализации алгоритмов решения практических задач на одном из языков программирования.

Задачи:

По окончании курса студент должен знать:

- терминологию дисциплины;
- основные структуры и инструментарий, которые применяются в языках программирования;
- основные структуры и типы данных;
- основные методы при разработке алгоритмов (рекурсия, отход назад, метод ветвей и границ, анализ арифметических выражений);
- базовые алгоритмы на динамических структурах данных;
- библиотеки стандартных программ.

Студент должен уметь:

- применять методы программирования при разработке информационных систем;
- определять структуры данных при проектировании алгоритмов в процессе решения задач;
- разбивать решение сложной задачи на последовательность более простых задач;
- использовать библиотеки стандартных программ, которые включены в язык программирования;
- самостоятельно освоить тот язык программирования, который необходимо использовать при решении задач.

Для успешного освоения дисциплины «Языки и методы программирования» студент должен:

Знать: основы компьютерной грамотности.

Уметь: работать с файлами, ориентироваться в интерфейсе новых программ.

Владеть: основными навыками работы в интегрированной среде.

В результате данной дисциплины у обучающихся формируются следующие общекультурные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-7 – способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности	Знает	информационные технологии, позволяющие приобретать новые знания и умения, в том числе в области современных и перспективных языков программирования
	Умеет	самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения в области современных языков программирования
	Владеет	практическим опытом применения информационных технологий, позволяющих приобретать знания в области современных языков программирования, непосредственно не связанной со сферой деятельности
ОК-8 – способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями магистерской программы	Знает	Особенности функционирования современного оборудованием и приборов.
	Умеет	Пользоваться современным оборудованием и приборами в процессе их эксплуатации.
	Владеет	Методами эксплуатации оборудования и приборов, используемых в информационных технологиях.

Для формирования вышеуказанных компетенций в рамках дисциплины «Языки и методы программирования» применяется следующий метод интерактивного обучения: метод автоматизированного обучения в системе автоматического тестирования программ CATS, предъявляющей задания и позволяющей оценить решение.

При выполнении различных видов работ используются следующие технологии:

1. *Проблемное обучение* – стимулирование обучающихся к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.
3. *Контекстное обучение* – мотивация магистрантов к усвоению знаний путём выявления связей между конкретным знанием и его применением.
4. *Обучение на основе опыта* – активизация познавательной деятельности магистрантов за счёт ассоциации и собственного опыта с предметом обучения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

(нет)

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Лабораторные работы (36 часов)

Лабораторная работа №1. Реализация алгоритмов средней сложности на языке программирования высокого уровня Python

Лабораторная работа №2. Типы данных в Python

Лабораторная работа №3. Особенности и базовые концепции языка программирования высокого уровня Perl.

Лабораторная работа №4. Функции ЯП Perl.

Лабораторная работа №5. Работа со строками на различных языках высокого уровня. Использование стиля программирования.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Распознавание образов и машинное обучение» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Изучение дисциплины «Языки и методы программирования» предусматривает:

- освоение теоретического материала в соответствии с программой, с использованием материала по списку электронных источников для каждой темы;
- выполнение домашних заданий;
- выполнение индивидуальных заданий;
- обязательная проработка материала, который будет разбираться на занятии с подбором дополнительных материалов.

Текущий контроль. Предусматривает учет посещения магистрантами занятий в течение периода обучения и оценку своевременности и качества изучения студентами темы и выполнения домашних заданий.

Итоговый контроль. Предусматривает рейтинговую оценку по учебной дисциплине в течение семестра и экзамен.

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Принципы программирования на языках высокого уровня	ОК-7 ОК-8	знает парадигмы программирования, методы разработки ПО, способы исполнения программ. описания языков программирования, особенности структурного программирования.	Устный опрос	1 - 4
			умеет применять на практике водопадную модель разработки программ, итерационную модель разработки программ	Лабораторная работа (ЛР-1)	Отчет по лабораторной работе
			владеет технологиями структурного программирования, объектно-	Лабораторная работа (ЛР-1)	Отчет по лабораторной работе

			ориентированного программирования, императивного программирования, декларативного программирования		
2	Обзор языков программирования высокого уровня	ОК-7 ОК-8	знает докомпьютерные и компьютерные языки программирования, машинный код, автокод, язык ассемблера, современное развитие, взаимное влияние языков программирования высокого уровня, принципы формализации описания языков программирования, особенности структурного программирования.	Устный опрос (УО-1).	5 - 11
			умеет определять вид и уровень языка программирования, приводить примеры современных языков программирования указанного класса	Лабораторная работа (ЛР-2)	Отчет по лабораторной работе
			владеет языком формализации описания языков программирования	Лабораторная работа (ЛР-2)	Отчет по лабораторной работе
3	Основы программирования на языке Python	ОК-7 ОК-8	Знает модель выполнения, модель памяти ЯП Python, стандартные структуры данных ЯП Python, конструкции языка, элементы ФП: функции высших порядков, замыкания, итераторы, генераторы, стандартные структуры данных, основные виды типов данных в ЯП Python, определение регулярного	Устный опрос (УО-2).	12 – 14

			выражения, понятие виртуального адресного пространства		
			Умеет использовать в ЯП Python ДКА, стандартные структуры данных ЯП Python, реализовать регулярные выражения с использованием НДКА, квантификаторы, множества символов, обратные ссылки средствами работы со строками	Лабораторная работа (ЛР-3)	Отчет по лабораторной работе
			Владеет навыками работы со списками, кортежами, list comprehension, множественным присваиванием, слайсингом, отрицательными индексами списков, ассоциативным массивом словарями, множествами	Лабораторная работа (ЛР-3)	Отчет по лабораторной работе
4	Основы программирования на языке Perl	ОК-7 ОК-8	знает назначение языка ЯП Perl, его классификацию, основные типы данных, понятие контекста, переменные по умолчанию, управляющие конструкции	Устный опрос (УО-2).	15 – 17
			умеет использовать в программе на ЯП Perl управляющие конструкции (if, for, foreach, while, sub), основные типы данных (числа, строки, массивы, ссылки), функции	Лабораторная работа (ЛР-5)	Отчет по лабораторной работе
			владеет навыками программирования на языке Perl, механизмом передачи параметров в	Лабораторная работа (ЛР-5)	Отчет по лабораторной работе

		функциях, навыками работы со строками.		
--	--	--	--	--

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература (электронные и печатные издания)

1. Головин И. Г., Волкова И. А. Языки и методы программирования: учебник для вузов/ И. Г Головин. Москва: Академия, 2012 – 304 с.
2. Молдованова, О. В. Языки программирования и методы трансляции [Электронный ресурс]: учебное пособие / О. В. Молдованова. — Электрон. текстовые данные. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012. — 134 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/54809.html>
3. Кауфман, В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс] / В.Ш. Кауфман. — Электрон. дан. — Москва : ДМК Пресс, 2010. — 464 с. — Режим доступа: <https://e.lanbook.com/book/1270>
4. Серебряков, В.А. Теория и реализация языков программирования [Электронный ресурс] : учебное пособие / В.А. Серебряков. — Электрон. дан. — Москва : Физматлит, 2012. — 236 с. — Режим доступа: <https://e.lanbook.com/book/5294>. — Загл. с экрана.
5. Языки программирования: Учебное пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - 3-е изд., перераб. и доп. - М.: Форум: ИНФРА-М, 2015. - 400 с.: 60x90 1/16. - (Профессиональное образование).

(переплет) ISBN 978-5-91134-744-4 - Режим доступа:

<http://znanium.com/catalog/product/493421>

6. Алгоритмизация и языки программирования [Электронный ресурс] : учебно-методический комплекс / сост. О. С. Ахметова, Б. К. Тульбасова, А. Ж. Абишева. — Электрон. текстовые данные. — Алматы : Нур-Принт, Казахский национальный педагогический университет имени Абая, 2012. — 165 с. — 9965-894-95-7. — Режим доступа: <http://www.iprbookshop.ru/67008.html> ISO/IEC 9075-2:2011 Information technology — Database languages — SQL — Part 2: Foundation (SQL/Foundation). — ISO/IEC, 2011 — 1483 с.

**Дополнительная литература
(печатные и электронные издания)**

1. Формальные языки и компиляторы/МалявкоА.А. - Новосиб.: НГТУ, 2014. - 431 с.: ISBN 978-5-7782-2318-9 - Режим доступа: <http://znanium.com/catalog/product/548152> MariaDB Documentation.
2. Языки программирования : учеб. пособие / О.Л. Голицына, Т.Л. Партыка, И.И. Попов. — 3-е изд., перераб. и доп. — М. : ФОРУМ : ИНФРА-М, 2018. — 399 с. — (Среднее профессиональное образование). - Режим доступа: <http://znanium.com/catalog/product/973007>
3. Фомичев В. С. Формальные языки, грамматики и автоматы, СПбГЭУ "ЛЭТИ". – 2013.

**Перечень ресурсов информационно-телекоммуникационной сети
«Интернет»**

1. Сообщество пользователей SQL [<http://sql.ru>].
2. Утилита для администрирования данных FlameRobin [<http://flamerobin.org>]

Перечень информационных технологий и программного обеспечения

1. Электронная презентация
2. Электронная рассылка

3. Электронный журнал успеваемости
4. Электронные поисковые приложения
5. Электронное тестирование

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

На изучение дисциплины отводится 36 час. аудиторных (лекционных и практических) занятий. На занятиях перед выдачей индивидуальных заданий преподаватель объясняет теоретический материал по заданной теме. Вводит основные требования к его выполнению. Приводит примеры. Необходимо поддерживать непрерывный контакт с аудиторией, отвечать на возникающие у студентов вопросы. На практических занятиях преподаватель разбирает на примерах принципы и аспекты реализации задания по заданной теме.

По ряду тем магистрантам предлагается работать самостоятельно, выполняя полный обзор по теме. Преподаватель контролирует работу студентов, отвечает на возникающие вопросы, предоставляет список литературных источников для освоения темы, а также перечень вопросов для самопроверки. Если знаний, полученных в аудитории, оказалось недостаточно, магистрант может самостоятельно повторно просмотреть методические указания.

После выполнения задания, магистрант оформляет материал в форме программного кода и отправляет его на проверку преподавателю по электронной почте, либо предьявляет на компьютере во время занятия. Магистрант отвечает устно во время занятия по заданной теме.

По данному курсу разработаны учебные материалы. Для успешного достижения учебных целей занятий должны выполняться следующие основные требования:

- соответствие действий обучающихся ранее изученным на лекционных и семинарских занятиях методикам и методам.
- максимальное приближение действий магистрантов к реальным, соответствующим будущим функциональным обязанностям.

- поэтапное формирование умений и навыков, т.е. движение от знаний к умениям и навыкам, от простого к сложному и т.д..

- использование при работе на тренажерах или действующей технике фактических документов, технологических карт, бланков и т.п.

- выработка соответствующих индивидуальных и коллективных умений и навыков.

Магистрант должен:

- научиться работать с книгой, документацией и схемами, пользоваться справочной и научной литературой.

- научиться работать с электронными литературными источниками.

- формировать умение учиться самостоятельно, т.е. овладевать методами, способами и приемами самообучения, саморазвития и самоконтроля.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

<p>Компьютерный класс: Проектор DLP, 3000 ANSI Lm, WXGA 1280x800, 2000:1 EW330U Mitsubishi,; Моноблок HP ProOne 440 G3 23.8" All-in-One, диагональ экрана 23.8", разрешение экрана 1920x1080, Bluetooth, Wi-Fi, операционная система: Windows 10 Enterprise, оптический привод DVD, процессор: Intel Core i5-7500T, размер оперативной памяти: 8 ГБ, видеопроцессор: Intel HD Graphics 630, объем жесткого диска: 1Tb. Беспроводные ЛВС для обучающихся обеспечены системой на базе точек доступа 802.11a/b/g/n 2x2 MIMO(2SS).</p> <p>Специализированное ПО: Visual Studio 2019, Anaconda, Eclipse</p>	<p>690922, Приморский край, г. Владивосток, о. Русский, п. Аякс, 10, г. Владивосток, о. Русский, п. Аякс, корпус G, ауд. G 468</p>
--	--



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**
по дисциплине «ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

**Направление подготовки – 09.04.01 Информатика и вычислительная
техника**

магистерская программа «Технологии виртуальной и дополненной
реальности»

Форма подготовки очная

Владивосток
2018

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Распознавание образов и машинное обучение» включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п, название	Дата/сроки выполнения	Вид СРС	Примерные нормы времени на выполнение	Форма контроля
1. Принципы и методы программирования, использование разных стилей программирования	Третья неделя семестра	ИДЗ	2 недели	Коллоквиум
2. Классификация и описание языков программирования. Управляющие конструкции	Четвертая неделя семестра	ИДЗ	1 неделя	Коллоквиум
3. Особенности и базовые концепции языков Python и Perl.	Пятая неделя семестра	ИДЗ	1 неделя	Коллоквиум
4. Реализация классических алгоритмов на различных языках высокого уровня.	Шестая неделя семестра	ИДЗ	1 неделя	Проверка программы
5. Реализация алгоритмов работы со строками на различных языках высокого уровня.	Седьмая неделя семестра	ИДЗ	3 неделя	Проверка программы
6. Реализация работы со структурами данных. Разработка алгоритмов средней сложности на различных языках высокого уровня.	Десятая неделя семестра	ИДЗ	2 недели	Проверка программы

7. Функции в различных языках высокого уровня	Пятнадцатая неделя семестра	ИДЗ	2 недели	Проверка программы
8. Подготовка к зачету	Последняя неделя	ИДЗ	1 неделя	Итоговое тестирование

Содержание самостоятельной работы студентов по дисциплине

1. Изучение аспектов теории, не уложившихся в лекции.
2. Разработка алгоритмов и программ при выполнении лабораторных работ.
3. Подготовка к тестированию и экзамену.

Текущая СРС

- работа с лекционным материалом, поиск и обзор литературы и электронных источников информации по заданной теме;
- изучение тем, вынесенных на самостоятельную проработку;
- подготовка к лабораторным работам;
- подготовка к зачёту, экзамену.

Творческая проблемно-ориентированная самостоятельная работа (ТСР).

ТСР направлена на развитие интеллектуальных умений, комплекса универсальных (общекультурных) и профессиональных компетенций, повышение творческого потенциала магистрантов и заключается в:

- поиске, анализе, структурировании и презентации информации;
- разработке учебного программного продукта;
- исследовательской работе и участии в научных студенческих семинарах и олимпиадах;
- анализе научных публикаций по заранее определенной преподавателем теме.

Требования к представлению и оформлению результатов самостоятельной работы

1. Устный ответ по указанной теме.
2. Исходный код программ заданий по соответствующим темам .

Критерии оценки выполнения самостоятельной работы

Оценка результатов самостоятельной работы организуется как единство двух форм: самоконтроль и контроль со стороны преподавателя.

Критерии оценки:

1. Обзор литературных источников (10 баллов)
2. Качество устного изложения содержания темы (10 баллов)
3. Выполнение каждого практического задания (10 баллов)



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЦИФРОВОЙ ЭКОНОМИКИ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине «ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

**Направление подготовки – 09.04.01 Информатика и вычислительная
техника**

магистерская программа «Технологии виртуальной и дополненной
реальности»

Форма подготовки очная

**Владивосток
2018**

Фонд оценочных средств по дисциплине «Распознавание образов и машинное обучение» включает в себя:

- типовые контрольные задания,
- методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности,
- а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы

ОПИСАНИЕ ПРАКТИЧЕСКИХ РАБОТ

Лабораторная работа №1 Реализация алгоритмов средней сложности на языке программирования высокого уровня Python,

Модель выполнения. Модель памяти. Стандартные структуры данных ЯП Python: списки, кортежи, словари, множества.

Задания подразумевают использование базовых типов данных, массивов, управляющих конструкций, множественного присваивания

Лабораторная работа №2 Типы данных в ЯП Python.

Список в ЯП Python.

Основные виды типов данных в ЯП Python, list comprehension, множественное присваивание, слайсинг, отрицательные индексы списков в ЯП Python, ассоциативный массив.

Лабораторная работа №3 Особенности и базовые концепции языка программирования высокого уровня Perl.

Основные типы данных ЯП Perl (числа, строки, массивы, ссылки). Понятие контекста. Переменные по умолчанию. Управляющие конструкции (if, for, foreach, while, sub).

Лабораторная работа №4 Функции ЯП Perl.

Передача параметров в функции. Управляющие конструкции.

Реализация классических алгоритмов на различных языках высокого уровня,

Задания требуют работу с моделями динамического распределения памяти.

Темы практической проработки: списки, кортежи, отрицательные индексы списков,

Лабораторная работа №5 Работа со строками на различных языках высокого уровня.

Использование стиля программирования.

Реализация каждого из заданий выполняется одновременно на двух на языках:

Python и Perl.

В заданиях требуется использование регулярных выражений с использованием НДКА, выбор структур данных, квантификаторов, множества символов, средств работы со строками.

Материалы зачета по дисциплине «Языки и методы программирования»

Вопросы к зачету

1. Методы разработки ПО.
2. Способы исполнения программ. Парадигмы программирования.
3. Способы классификации ЯП.
4. Обзор ЯП, классификация некоторых из них (не менее 10).
5. Регулярные выражения - практика: квантификаторы, множества символов, обратные ссылки.
6. Регулярные выражения - теория: определение, проверка принадлежности языку РВ при помощи НДКА.
7. Ручное управление памятью. Куча. Проблемы ручного управления. Автоматическое управление памятью. Сборка мусора. Подсчёт ссылок.
8. ЯП Python. Назначение языка, его классификация. Модель данных. Модель выполнения.
9. ЯП Python. Типы данных (числа, строки, списки, словари, множества, кортежи). Индексация списков, слайсинг.
10. ЯП Python. Конструкции языка (if, for, while, def, list comprehension). Передача параметров в функции.
11. ЯП Python. Итераторы. Генераторы. Функции высших порядков. Замыкания.
12. ЯП Perl. Назначение языка, его классификация. Контекст. Переменные по умолчанию.
13. ЯП Perl. Основные типы данных (числа, строки, массивы, ссылки). Передача параметров в функции.
14. ЯП Perl. Управляющие конструкции (if, for, foreach, while, sub). Работа со строками.

Примеры дополнительных вопросов

Общие вопросы

1. что такое водопадная модель разработки?
2. что такое итерационная модель разработки?

3. что такое тестирование методом белого ящика?
4. что такое тестирование методом чёрного ящика?
5. что такое unit-тест?
6. что такое стресс-тестирование?
7. что такое компиляция?
8. что такое интерпретация?
9. что такое байт-код?
10. что такое структурное программирование?
11. что такое объектно-ориентированное программирование?
12. что такое императивное программирование?
13. что такое декларативное программирование?
14. что такое статическая типизация?
15. что такое динамическая типизация?
16. что такое слабая типизация?
17. что такое сильная типизация?
18. что такое ручное управление памятью?
19. что такое автоматическое управление памятью?

Вопросы по обработке строк и РВ

1. что такое лексикографический порядок сравнения строк?
2. что такое сырая строка?
3. что такое РВ?
4. что такое ДКА?
5. что такое НДКА?
6. что такое квантификатор в РВ?
7. как задать множество символов в РВ?
8. что означает * в РВ?
9. что означает | в РВ?
10. что означает ? в РВ?
11. что означает . в РВ?

12. как составить РВ, принимающее...?

Вопросы по управлению памятью

1. что такое виртуальное адресное пространство?
2. что такое куча?
3. что такое счётчик ссылок?
4. какую проблему подсчёт ссылок решить не может?
5. что такое сборка мусора?

Вопросы по ЯП Python

1. что представляет собой список в ЯП Python?
2. передача аргументов в функции в ЯП Python происходит по ссылке или
ссылке или
3. о значению?
4. какие два основных вида типов данных в ЯП Python?
5. что такое list comprehension?
6. что такое множественное присваивание?
7. что такое слайсинг?
8. что означают отрицательные индексы списков в ЯП Python?
9. что такое функции высших порядков?
10. что такое замыкание для функций?
11. что такое генератор в ЯП Python?
12. что такое итератор?
13. что такое ассоциативный массив?
14. что означает "abc"*4 в ЯП Python

Вопросы по ЯП Perl

1. как конкатенировать строки в ЯП Perl?
2. что такое строковая интерполяция?
3. почему в ЯП Perl умножение строки на число обозначается как x a
не как более естественное *?
4. почему в ЯП Perl операторы сравнения строк отличаются от
операторов сравнения чисел?

5. передача аргументов в функции в ЯП Perl происходит по ссылке или по значению?
6. что означает \$_ в ЯП Perl?
7. что означает @_ в ЯП Perl?

Критерии оценивания

В течение семестра студентам последовательно выдается набор из 5-и лабораторных работ, каждая из которых имеет вес от 15%. Посещаемость занятий также учитывается и имеет вес 10%. Для получения «отлично» необходимо иметь итоговый балл не ниже 80%, «хорошо» – необходимо иметь итоговый балл не ниже 65%, «удовлетворительно» – необходимо иметь итоговый балл не ниже 50%.

Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели
ОК-7 способность самостоятельно приобретать с помощью информационных технологий и использовать в практической деятельности новые знания и умения, в том числе в новых областях знаний, непосредственно не связанных со сферой деятельности	Знает (пороговый уровень)	информационные технологии, позволяющие приобретать новые знания и умения, в том числе в области современных и перспективных языков программирования	демонстрирует теоретическую подготовку	2 из 3 ответов правильны
	Умеет (продвинутой)	самостоятельно приобретать с помощью современных информационных технологий и использовать в практической деятельности новые знания и умения в области современных языков программирования	применяет знания на практике	выполняет, в основном правильно практическое задание
	Владеет (высокий)	практическим опытом применения информационных технологий, позволяющих приобретать знания в области современных языков программирования, непосредственно не связанной со сферой деятельности	быстро выбирает метод решения	качественно выполняет задание аналитического характера
ОК-8 способность к профессиональной эксплуатации современного оборудования и приборов (в соответствии с целями	Знает (пороговый уровень)	инструментальные средства и среды программирования	демонстрирует теоретическую подготовку	2 из 3 ответов правильны
	Умеет (продвинутой)	создавать, редактировать, сохранять и запускать программы на современном компьютере	применяет знания на практике	решает в основном правильно практическое задание

магистерской программы)	Владеет (высокий)	навыками работы на компьютере, как в плане использования, так и создания программных приложений	быстро выбирает метод решения	качественно выполняет задание аналитического характера
-------------------------	-------------------	---	-------------------------------	--

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Задания к зачету

Задача А. Hello, world!

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая печатает "Hello, world!" (без кавычек)

Задача В. Радуга

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая печатает все цвета радуги не русском языке. Каждый цвет вывести в отдельной строке с большой буквы.

Примеры тестов

№	Стандартный вход	Стандартный выход
1		Красный Оранжевый Желтый Зеленый Голубой Синий Фиолетовый

Задача С. Представьтесь

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая считывает имя и фамилию и выводит их в формате

Меня зовут [фамилия]

[имя] [фамилия]

Формат входных данных

Входные данные содержат два слова в разных строках

Формат выходных данных

Выходные данные должны содержать текст в описанном формате

Примеры тестов

№	Стандартный вход	Стандартный выход
1	Джеймс Бонд	Меня зовут Бонд Джеймс Бонд

Задача D. Электронная почта

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

В ДВФУ у каждого студента и сотрудника есть персональный адрес электронной почты в домене `dvfu.ru`.

Требуется написать программу, которая по заданным фамилии, имени и роли пользователя создает адрес его электронной почты в формате `[фамилия].[имя]@[роль].dvfu.ru`

Формат входных данных

Входные данные содержат три слова в разных строках.

Формат выходных данных

Выходные данные должны содержать одну строку — электронный адрес в описанном формате.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	sporyshev ms students	sporyshev.ms@students.dvfu.ru

Задача E. Солнечный день

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Ваша лодка потерпела крушение! Но вы и не против, сидите себе спокойно и каждые 10 минут выливаете за борт XX литров воды. При этом каждую минуту через пробоину поступает YY литров воды. При условии, что в таком темпе вы никогда не вычерпаете воду полностью, сколько литров воды будет в вашей лодке через TT минут?

Формат входных данных

Входные данные содержат целые числа XX , YY и TT , по одному числу в строке.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	10 1	5

	15	
2	1000 9 9	81

Задача F. Олины торты

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Сегодня к Оле в гости придет NN одноклассников, из-за чего она решила приготовить NN тортов. В ее распоряжении есть MM грамм сахара. Поскольку Оля не хочет никого обидеть, в каждом торте должно быть одинаковое количество сахара. Какое максимальное количество сахара может содержать каждый торт?

Формат входных данных

Входные данные содержат числа MM и NN, каждое на новой строке.

Формат выходных данных

Необходимо вывести единственное число — максимальное количество сахара.

Ограничения

$1 < N, M < 100001 < N, M < 10000$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	10 2	5
2	6 4	1.5

Задача G. Простое

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Условный оператор — это просто! Верно?

Напишите программу, которая принимает на вход от пользователя одну строку. Если пользователь ответил "Просто!", "Easy!" или "Einfach!", напечатайте в ответ улыбающийся смайл ":)". В ином случае — грустный смайл ":(".

Примеры тестов

№	Стандартный вход	Стандартный выход
1	Просто!	:)
2	Как-то не очень	:(

Задача N. Сложное

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Требуется написать программу, которая определит тип предложения. Если в предложении есть вопросительный знак, то оно вопросительное. Если есть восклицательный, то предложение восклицательное. Если есть и то и другое, то оно вопросительно-восклицательное. Если ни того ни другого в предложении нет, то оно повествовательное. Если ко всему прочему в предложении есть хотя бы одна запятая, то оно еще и сложное.

Формат входных данных

Входные данные содержат предложение

Формат выходных данных

Выходные данные должны содержать его тип. Если предложение сложное, слово "сложное" должно стоять первым и отделяться пробелом от восклицательности предложения (смотри пример).

Примеры тестов

№	Стандартный вход	Стандартный выход
1	Что тут происходит?!?!	вопросительно-восклицательное
2	Скажите мне, пожалуйста!	сложное восклицательное
3	Ничего	повествовательное

Задача I. FizzBuzz

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Требуется написать программу, которая считывает число и выводит `Fizz`, если число делится на 3, `Buzz`, если число делится на 5, и `FizzBuzz`, если оно делится и на 3, и на 5. Если число не делится ни на 3, ни на 5, вывести пустую строку.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	23	
2	25	Buzz
3	30	FizzBuzz

Задача J. Кто первый?

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Помогите трем студентам понять, кто из них первый пойдет решать диффуры, если преподаватель вызывает к доске в алфавитном порядке.

Формат входных данных

Входные данные содержат три фамилии студентов, разделенные переносом строки.

Формат выходных данных

Программа должна напечатать фамилию студента, который пойдет к доске первым.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	Петров Иванов Сидоров	Иванов

Задача K. Вынутый разворот

Автор: Владивостокская городская олимпиада школьников по информатике 2002/2003

Входной файл: input.txt

Выходной файл: output.txt

Условие

Брошюра составлена из листов. На каждой стороне листа напечатано по две страницы. Страницы пронумерованы начиная с первой. Из брошюры был вынут один лист. Требуется по двум номерам страниц, напечатанным на одной из сторон этого листа, определить общее количество страниц в брошюре.

Формат входного файла

Во входном файле содержатся два целых числа AA и BB — номера страниц на стороне листа, в произвольном порядке

Формат выходного файла

В выходном файле должно содержаться единственное число:

- 00 — если такой брошюры не существует
- Число страниц в брошюре — в противном случае

Ограничения

$1 \leq A, B \leq 10^6$

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
---	--------------------------	----------------------------

1	3 14	16
2	9 1	0

Задача L. Время в пути

Автор: А. Кленин

Входной файл: input.txt

Выходной файл: output.txt

Условие

Время отправления и время прибытия поезда задаются в виде Ч М, где Ч - час от 0 до 23, М - минута от 0 до 59. Время в пути задаётся аналогично в формате Ч М, где Ч - количество часов от 0 до 999, а М - количество минут от 0 до 59.

Требуется по данному времени отправления и времени в пути вычислить время прибытия поезда (возможно, в другие сутки).

Формат входного файла

В первой строке входного файла содержится время отправления, во второй — время в пути.

Формат выходного файла

В выходном файле должна быть единственная строка, содержащая время прибытия.

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	18 25 7 37	2 2

Задача M. Вася и мысли

Автор: А. Кленин

Входной файл: input.txt

Выходной файл: output.txt

Условие

Юный программист Вася любит прогуливаться по тропинке длиной DD метров, проходящей неподалёку от его школы, и размышлять о том о сём.

В начале прогулки Вася размышляет о Важных Вещах, и ноги несут его с постоянной скоростью, равной либо v_1v_1 , либо v_2v_2 м/с. Затем Вася отвлекается и начинает думать о Всякой Ерунде. В этот момент его скорость становится равной либо w_1w_1 , либо w_2w_2 м/с.

Если Вася затратил на прогулку TT секунд, каково максимальное возможное расстояние, пройденное им с Важными Вещами в голове?

Формат входного файла

Первая строка входного файла содержит числа $D TD T$. Во второй строке записаны числа $v_1 v_2v_1 v_2$. В третьей строке записаны числа $w_1 w_2w_1 w_2$. Каждое число имеет не более трёх знаков после десятичной точки.

Формат выходного файла

Выходной файл должен содержать одно вещественное число dd — максимальное возможное расстояние с точностью не менее 3 десятичных знаков.

Входные данные таковы, что решение всегда существует.

Ограничения

$$1 \leq D, T \leq 10001 \leq D, T \leq 1000$$

$$10^{-3} \leq v_1 < v_2 \leq 10210^{-3} \leq v_1 < v_2 \leq 102$$

$$10^{-3} \leq w_1 < w_2 \leq 10210^{-3} \leq w_1 < w_2 \leq 102$$

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	10.0 5.0 1.0 3.0 0.5 1.0	9.0
2	15.0 3.0 1.0 2.5 4.25 6.1	2.291
3	25.0 5.0 1.0 3.0 4.0 5.0	0.0

Задача N. Вокруг дома

Автор: А. Кленин

Входной файл: input.txt

Выходной файл: output.txt

Условие

Васин дом имеет в плане форму прямоугольника. Северная и южная стороны имеют длину по AA метров каждая, а западная и восточная — по BB метров каждая.

Вася прогуливается вокруг дома со скоростью 1 метр в секунду, начиная с угла между западной и южной стороной, и проходит вдоль южной, восточной, северной и западной сторон в указанном порядке. Обойдя полностью вокруг дома, Вася снова повторяет свой маршрут.

Требуется написать программу, которая определит, вдоль какой стены будет идти Вася через t секунд после начала прогулки. Будем считать, что дойдя то угла, Вася немедленно за него поворачивает, поэтому на секундах с нулевой по $A-1A-1$ -ю он находится у южной стены, на секунде номер AA — у восточной, и так далее.

Формат входного файла

Входной файл содержит целые числа $A B tA B t$.

Формат выходного файла

Выходной файл должен содержать единственный символ, обозначающий стену, вдоль которой прогуливается Вася:

- N — северная стена,

- W — западная стена,
- S — южная стена,
- E — восточная стена.

Ограничения

$$1 \leq A, B, t \leq 5 \cdot 10^8$$

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	10 5 18	N
2	1 8 18	S

Задача А. Горшочек каши

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая считывает строки из стандартного ввода. Если считана строка "Stop!", программа печатает "Ок" (кириллицей) и останавливает работу. В ином случае она печатает "Каша" и считывает следующую строку.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	больше вари, ленивец больше! хмммм перестань остановись я приказываю тебе прекрати остановись же Stop!	Каша Каша Каша Каша Каша Каша Каша Каша Ок

Задача В. Кто первый?-2

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Помогите группе из NN студентов понять, кто из них первый пойдет решать диффуры, если преподаватель вызывает к доске в алфавитном порядке.

Формат входных данных

Первая строка содержит число NN — количество студентов в группе. Следующие NN строк содержат фамилии студентов.

Формат выходных данных

Выходные данные должны содержать единственную строку — фамилию студента, который первым пойдет к доске.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	4 Петров Иванов Сидоров Щуров	Иванов

Задача С. Грибы

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Аня очень хорошо разбирается в грибах. Грибы бывают разные. Опята, лисички, сыроежки, подосиновики, подберезовики, обабки, маслята, олени рожки, поганки (ядовитые). *Boletus edulis* (белые грибы) — самые хорошие. Нужно посчитать, сколько *Boletus edulis* собрала Аня.

Формат входных данных

В первой строке содержится число NN — количество грибов. В следующих NN строках — названия грибов.

Формат выходных данных

Выходные данные должны содержать одно число — количество *Boletus edulis*. Регистр не имеет значения, т.е. к примеру *BoLeTuS eDuLiS* и *boLETUS edulIS* — один и тот же гриб.

Ограничения

$1 \leq N \leq 1001 \leq N \leq 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	7 подосиновик камень <i>boletus edulis</i> обабок обабок обабок <i>Boletus edulis</i>	2
2	5 шампиньон лисичка маслёнок сыроежка	0

опёнок	
--------	--

Задача D. Космический корабль

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Запуск космического корабля это очень торжественное событие, требующее обратного отсчета.

Формат входных данных

На вход программа получает одно число NN — количество секунд до запуска.

Формат выходных данных

Программа должна вывести обратный отсчет от NN до 1, завершающийся командой "Start". Каждый шаг отсчета сопровождается восклицательными знаками. С каждой секундой их количество увеличивается на единицу, так что команда "Start" сопровождается NN восклицательными знаками.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5	5 4! 3!! 2!!! 1!!!! Start!!!!!!

Задача E. Таблица умножения

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая выводит таблицу умножения чисел от 1 до nn.

Формат входных данных

Входные данные содержат одно целое число nn.

Формат выходных данных

Выходные данные должны содержать таблицу умножения числа 11 на числа от 11 до nn, затем числа 22 на числа от 11 до nn и т. д. Формат вывода см. в примере.

Ограничения

$1 \leq N \leq 1001 \leq N \leq 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3	1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 3 * 1 = 3 3 * 2 = 6 3 * 3 = 9

Задача F. Игра в предложения

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Паша, Аня, Оля и Оргрим играют в игру, в которой нужно восстановить предложение по его словам, произнесенным в обратном порядке.

Формат входных данных

Входные данные содержат количество слов в предложении N и затем само предложение, разбитое по словам, записанное в обратном порядке, каждое слово в своей строке.

Формат выходных данных

Выходные данные должны содержать исходное предложение, разбитое по словам, каждое слово в своей строке.

Ограничения

$1 \leq N \leq 100$
 $1 \leq N \leq 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	4 cha g'thazag mog Bin	Bin mog g'thazag cha

Задача H. Циклический сдвиг

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Циклический сдвиг — это преобразование массива, при котором все элементы смещаются вправо или влево на несколько позиций. При этом, в случае сдвига влево, элементы из начала массива циклически перемещаются в конец (аналогично переходу из начала в конец очереди).

Формат входных данных

Первая строка содержит числа NN и KK. Следующая строка содержит NN чисел — элементы массива.

Формат выходных данных

Необходимо вывести NN чисел — исходный массив, циклически сдвинутый влево на KK.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	4 1 1 2 3 4	2 3 4 1
2	5 2 7 1 3 6 10	3 6 10 7 1

Задача I. Поэлементное сложение

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая поэлементно складывает два массива.

Формат входных данных

Первая строка входных данных содержит число NN — длину массивов. Следующие 2 строки содержат по NN чисел каждая — элементы массивов.

Формат выходных данных

Выходные данные должны содержать NN разделенных пробелом чисел — результат поэлементного сложения.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 1 2 3 4 5 6	5 7 9

Задача J. Форум

Входной файл: Стандартный вход

файл:

Выходной Стандартный выход
файл:

Условие

Гости Всемирного Форума собрались по домам. Мы не знаем, представители каких делегаций были на форуме, но знаем, что делегаций не больше 197 (по количеству стран в мире). Всё, что может сказать каждый отдельный гость о себе, — номер делегации, которой он принадлежит. Необходимо рассадить гостей по автобусам так, чтобы каждой присутствующей делегации достался ровно один автобус. Какое минимальное количество автобусов нужно, чтобы рассадить гостей?

Формат входных данных

Входные данные содержат список чисел через пробел — номер делегации каждого отдельного гостя.

Формат выходных данных

Выходные данные должны содержать одно число — минимальное количество автобусов.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	100 100 23 197 100 23 197	3

Задача К. Количество слов-0

Входной Стандартный вход
файл:

Выходной Стандартный выход
файл:

Условие

Во входном файле записана строка текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак.

Слово — это последовательность подряд идущих латинских букв. Вопреки правилам, все знаки препинания в этой задаче окружены пробелами.

Напишите программу, определяющую, сколько слов в данной строке текста.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	As God is my witness , I will never be hungry again .	11

Задача L. Количество каждой цифры (курс Python)

Автор: Г. Гренкин

Входной input.txt
файл:

Выходной
файл: output.txt

Условие

Петя написал на заборе NN цифр. Когда Вася увидел то, что натворил Петя, он решил посчитать, сколько раз написана каждая цифра.

Напишите программу, принимающую на вход список цифр, которые написал Вася, и выводящую для каждой цифры, сколько раз она встречается в списке.

Формат входного файла

Входной файл содержит целое число NN, за которым следуют NN целых чисел (цифр), каждое с новой строки.

Формат выходного файла

Выходной файл должен содержать 1010 чисел — количество цифр 1, количество цифр 2, и т.д., количество цифр 9, количество цифр 0.

Ограничения

$1 \leq N \leq 1000000$

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	4 1 2 1 0	2 1 0 0 0 0 0 0 0 1
2	7 9 1 3 2 3 2 7	1 2 2 0 0 0 1 0 1 0

Задача M. Количество слов

Автор: Московская олимпиада для 7-9 кл., 2005

Входной
файл: d.in

Выходной
файл: d.out

Условие

Во входном файле записана строка текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак,
- символ —, обозначающий в некоторых случаях тире, а в некоторых — дефис.

Слово — это последовательность подряд идущих латинских букв и знаков дефис, ограниченная с обоих концов. В качестве ограничителей могут выступать начало строки, конец строки, пробел, знак препинания, тире. Тире отличается от дефиса тем, что слева

и справа от знака дефис пишутся буквы, а хотя бы с одной стороны от тире идет либо начало строки, либо конец строки, либо пробел, либо какой-либо знак препинания, либо еще одно тире.

Напишите программу, определяющую, сколько слов в данной строке текста.

Формат входного файла

Во входном файле записана строка.

Формат выходного файла

В выходной файл выведите одно число — количество слов, которые содержатся в исходной строке.

Ограничения

Входная строка имеет длину не более 200 символов.

Примеры тестов

№	Входной файл (d.in)	Выходной файл (d.out)
1	Hello , world!	2
2	www.olympiads.ru	3
3	Gyro-compass - this is a ...	4

Задача N. Разделимся на группы

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Не знаем, зачем, но заказчику необходимо разделить персонажей на две группы.

Формат входных данных

Входные данные содержат список имён, каждое в своей строке.

Формат выходных данных

Выходные данные должны содержать две строки: в первой — персонажи на нечетных позициях списка, во второй — на четных. Имена должны быть разделены запятой с пробелом.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	Крош Бараш Человек-паук Иванов Петров Сидоров	Крош, Человек-паук, Петров Бараш, Иванов, Сидоров

Задача A. Излишек

Входной файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

У Влада возник излишек — 1000 рублей. Ему необходимо его срочно потратить, возможно, даже, с перерасходом. Для этого он зашел в супермаркет и кладет в корзину товары по порядку, пока их суммарная стоимость не достигнет излишка. Сколько рублей потратит Влад?

Формат входных данных

Входные данные содержат число NN — количество доступных товаров. Далее NN строк, в каждой из которых цена отдельного товара, в порядке, в котором Влад кладет их в корзину. Гарантируется, что товаров достаточно, чтобы покрыть излишек.

Формат выходных данных

Выходные данные должны содержать одно число — суммарную итоговую стоимость товаров в корзине.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	6 10 10 100 2000 10 10	2120
2	4 500 500 500 500	1000

Задача В. Энтное слово

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Вам даны фраза и положительное число — позиция слова. Необходимо вывести слово, находящееся в фразе на этой позиции. Если данное число больше, чем количество слов в фразе, то вывести последнее слово.

Формат входных данных

В первой строке входных данных дана строка текста, содержащая символы латинского алфавита, пробелы и цифры. Слово — это последовательность идущих подряд латинских букв и/или цифр. Во второй строке дано целое число — позиция искомого слова при нумерации слов с единицы.

Формат выходных данных

Выходные данные должны содержать искомое слово.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	FEFU was established in 1899 as the Eastern Institute 5	1899
2	FEFU was established 5	established

Задача C. Будь как все

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Дан массив чисел. Нужно заменить в нем максимальный элемент на минимальный из этого же массива. Если максимальных несколько — заменить каждый из них.

Формат входных данных

Входные данные содержат несколько чисел, разделенных пробелами.

Формат выходных данных

Выходные данные должны содержать массив, в котором максимальные элементы заменены на минимальный. Элементы массива должны быть разделены пробелами.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 -1 2 4 9 6	3 -1 2 4 -1 6
2	5 4 1 5 2	1 4 1 1 2

Задача D. Палиндром?

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Палиндром — это слово или фраза, которые одинаково читаются слева направо и справа налево (без учета пробелов и регистра символов).

Формат входных данных

Входные данные содержат фразу, не содержащую знаков препинания, в которой слова разделены пробелами.

Формат выходных данных

Выходные данные должны содержать `True`, если фраза — палиндром, и `False` в противном случае.

Примеры тестов

№	Стандартный вход	Стандартный выход
---	------------------	-------------------

1	А роза упала на лапу Азора	True
2	Роза не упала на лапу Азора	False

Задача F. Норма L1

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Дан вектор произвольной размерности. Необходимо посчитать так называемую норму L_1 для этого вектора. Норма L_1 — это сумма абсолютных величин компонент вектора (абсолютная величина вычисляется функцией `abs`).

Формат входных данных

Входные данные содержат целые числа — компоненты вектора, записанные в одной строке через пробел.

Формат выходных данных

Выходные данные должны содержать одно число — норму L_1 этого вектора.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 4	7
2	10 -20 3	33

Задача G. Транспонирование квадратной матрицы

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Дана квадратная матрица, нужно ее транспонировать. Транспонированная матрица — это матрица, для получения которой из исходной нужно элементы каждой строки исходной матрицы записать в виде столбца в том же порядке.

Формат входных данных

Первая строка содержит число NN — количество строк (и столбцов) в матрице. Следующие NN строк содержат NN чисел каждая — элементы матрицы.

Формат выходных данных

Выходные данные должны содержать NN строк по NN чисел каждая — элементы транспонированной матрицы.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
---	------------------	-------------------

1	3	1 4 7
	1 2 3	2 5 8
	4 5 6	3 6 9
	7 8 9	

Задача N. Норма L2

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Вам дан вектор произвольной размерности (массив вещественных чисел). Необходимо посчитать его норму L_2 (евклидову норму). Евклидова норма — это квадратный корень из суммы квадратов компонент вектора.

Формат входных данных

Входные данные содержат вещественные числа — компоненты вектора, записанные в одной строке через пробел.

Формат выходных данных

Выходные данные должны содержать одно вещественное число — евклидову норму этого вектора.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3.0 4.0	5.0
2	10.5 20.1 3.1	22.8882065701968

Задача I. Убрать соседа

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Дан массив чисел. Необходимо удалить элементы, за которыми в этом массиве следует ноль.

Формат входных данных

Входные данные содержат ряд чисел, разделенных пробелом.

Формат выходных данных

Выходные данные должны содержать преобразованный массив.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	1 2 0 3 4 0 5 10	1 0 3 0 5 10
2	11 0 0 12	0 12

Задача К. Увеличить по позициям

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Даны массив и список индексов (номеров элементов). Нумерация начинается с 1. Необходимо прибавить единицу к каждому элементу массива, индекс которого есть в списке индексов.

Формат входных данных

Первая строка входных данных содержит два числа NN и MM — длины массива и списка индексов соответственно. Вторая строка содержит NN целых чисел — элементы массива. Третья строка содержит MM целых чисел — индексы.

Формат выходных данных

Выходные данные должны содержать преобразованный массив.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5 3 1 2 3 4 5 1 3 5	2 2 4 4 6

Задача А. Add

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать на языке Python функцию `Add(x, y)`, которая принимает два числа и возвращает их сумму. Пример использования функции в примерах тестов.

Формат выходных данных

Код решения должен содержать только определение и реализацию функции. Он не должен ничего выводить.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<code>x, y = 2, 3</code> <code>print(Add(x, y))</code>	5
2	<code>print(Add(10, 11))</code>	21

Задача В. Key sort

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Дан массив слов, нужно отсортировать его по второй букве. Гарантируется, что в каждом слове не меньше двух букв, а все вторые буквы слов различны.

Формат входных данных

Входные данные содержат слова, разделённые пробелами.

Формат выходных данных

Выходные данные должны содержать строку слов, разделённых пробелом, отсортированных по второй букве.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	azx zab ddd	zab ddd azx

Задача C. GCD

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Дано 3 числа: AA, BB, CC. Необходимо посчитать наибольший общий делитель (НОД) каждой из пар AA и BB, AA и CC, BB и CC

Формат входных данных

Входные данные содержат 3 числа — AA, BB, CC.

Формат выходных данных

Выходные данные должны содержать 3 числа — НОД AA и BB, НОД AA и CC и НОД BB и CC

Примеры тестов

№	Стандартный вход	Стандартный выход
1	2 6 3	2 1 3

Задача D. PrintMatrix

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python функцию `PrintMatrix(mat)`, которая принимает двумерный массив и печатает его. Пример использования функции в примерах тестов.

Формат выходных данных

Код решения должен содержать только определение и реализацию функции.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>mat = [[1, 2, 3], [4, 5, 6], [7, 8, 9]] PrintMatrix(mat)</pre>	<pre>1 2 3 4 5 6 7 8 9</pre>
2	<pre>PrintMatrix([[1, 2, 3], [4, 5, 6]])</pre>	<pre>1 2 3 4 5 6</pre>

Задача E. Map

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python функцию `Map(func, l)`, которая принимает два параметра — функцию и список — и возвращает новый список, полученный применением функции `func` к элементам списка `l`. Пример использования функции в примерах тестов.

В решении запрещено использовать стандартную функцию `map`.

Формат выходных данных

Код решения должен содержать только определение и реализацию функции. Он не должен ничего выводить на экран.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>ary = [-1, 2, -3, 4] print(Map(abs, ary))</pre>	<pre>[1, 2, 3, 4]</pre>
2	<pre>print(sum(Map(lambda x: x**2, [2, 3])))</pre>	<pre>13</pre>

Задача F. Однострочник

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая выведет квадраты положительных элементов списка в обратном порядке. Решение должно быть написано в одну строку.

Формат входных данных

Входные данные содержат исходный список целых чисел, разделенных пробелами.

Формат выходных данных

Выходные данные должны содержать преобразованный список.

Примеры тестов

№	Стандартный вход	Стандартный выход
---	------------------	-------------------

1	-1 2 -3 4 0 -5 6	36 16 4
---	------------------	---------

Задача N. Двустрочник

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Необходимо составить новый список из существующего по такому правилу: крайние элементы в начале и в конце списка оставить на своих местах (по два элемента с каждой стороны), а элементы между ними записать в обратном порядке. Решение должно быть написано в две строки.

Формат входных данных

Входные данные содержат список целых чисел, разделенных пробелами. Гарантируется, что исходный список содержит как минимум 4 элемента.

Формат выходных данных

Выходные данные должны содержать список чисел, разделенных пробелами.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	1 2 30 40 50 6 7	1 2 50 40 30 6 7

Задача I. Join

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Требуется реализовать на языке Python функцию `Join(array, separator)`, которая принимает два параметра — список строк и разделитель — и возвращает строку, полученную соединением элементов переданного списка, при этом между элементами списка вставляется разделитель (`separator`). Если не передать функции разделитель, то она должна использовать в качестве разделителя пробел. Пример использования функции в примерах тестов.

В решении запрещено использовать стандартную функцию `join`.

Формат выходных данных

Код решения должен содержать только определение и реализацию функции. Он не должен ничего выводить.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>ary = ['aa', 'bb', 'cc'] print(Join(ary, '!'))</pre>	aa!bb!cc
2	<pre>ary = ['aa', 'bb', 'cc'] print(Join(ary))</pre>	aa bb cc

Задача А. Норма L1

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Эту задачу следует решать с использованием библиотеки `numpy`.

Дан вектор произвольной размерности. Необходимо посчитать так называемую норму L_1 для этого вектора. Норма L_1 — это сумма абсолютных величин компонент вектора (абсолютная величина вычисляется функцией `abs`).

Формат входных данных

Входные данные содержат целые числа — компоненты вектора, записанные в одной строке через пробел.

Формат выходных данных

Выходные данные должны содержать одно число — норму L_1 этого вектора.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 4	7
2	10 -20 3	33

Задача В. Прогрессия с плавающей точкой

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Требуется написать программу, которая получает на вход начало SS , шаг прогрессии dd и количество необходимых элементов NN и выводит NN членов заданной арифметической прогрессии.

Ограничения

$1 < N < 1001$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	1 0.5 3	1 1.5 2

Задача С. Reshape

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Дан вектор, состоящий из pp чисел. Необходимо преобразовать вектор в матрицу размера $n \times m = pn \times m = p$, состоящую из nn строк и mm столбцов. Матрица заполняется последовательно, то есть элементы вектора с 11 по mm попадут в первую строку матрицы, с $m+1m+1$ по $2m2m$ — во вторую строку и так далее.

Формат входных данных

Первая строка входных данных содержит целое число pp — количество элементов в входном векторе. Вторая строка содержит pp целых чисел — элементы вектора. Третья строка содержит два целых числа nn и mm — размеры итоговой матрицы.

Формат выходных данных

Выходные данные должны содержать итоговую матрицу. Строки матрицы разделяются переводами строки, числа в каждой строке — пробелами.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	6 1 2 3 4 5 6 3 2	1 2 3 4 5 6
2	12 1 2 3 4 5 6 1 2 3 4 5 6 2 6	1 2 3 4 5 6 1 2 3 4 5 6

Задача D. Норма L2

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Эту задачу следует решать с использованием библиотеки `numpy`.

Вам дан вектор произвольной размерности (массив вещественных чисел). Необходимо посчитать его норму L_2L_2 (евклидову норму). Евклидова норма — это квадратный корень из суммы квадратов компонент вектора.

Формат входных данных

Входные данные содержат вещественные числа — компоненты вектора, записанные в одной строке через пробел.

Формат выходных данных

Выходные данные должны содержать одно вещественное число — евклидову норму этого вектора.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3.0 4.0	5.0
2	10.5 20.1 3.1	22.8882065701968

Задача E. Dot первого и последнего столбцов матрицы

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая скалярно перемножает первый и последний столбец матрицы. Скалярным произведением двух векторов будет скалярная величина, равная сумме попарного произведения координат векторов (Например $(1,2,3) \cdot (3,4,5) = 1 \cdot 3 + 2 \cdot 4 + 3 \cdot 5 = 26$ $(1,2,3) \cdot (3,4,5) = 1 \cdot 3 + 2 \cdot 4 + 3 \cdot 5 = 26$).

Формат входных данных

Первая строка содержит числа NN и MM — количество строк и столбцов матрицы. Следующие NN строк содержат MM чисел каждая — элементы матрицы.

Формат выходных данных

Выходные данные должны содержать единственное число — скалярное произведение первого и последнего столбца матрицы.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	2 2 1 3 4 5	23

Задача I. Обменять строки в матрице

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется написать программу, которая меняет строки в матрице местами.

Формат входных данных

Первая строка содержит 4 числа N,M,I,J,N,M,I,J, где NN — количество строк матрицы, MM — количество столбцов матрицы, I,J,I,J — индексы строк, которые нужно поменять местами. Индексы начинаются с 0. Следующие NN строк содержат по MM чисел каждая — элементы матрицы.

Формат выходных данных

Выходные данные должны содержать NN строк по MM чисел в каждой — матрица, в которой строки II и JJ поменяны местами.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	2 2 0 1 1 2 3 4	3 4 1 2

Задача J. Чтение csv файлов

Входной
файл: input.csv

Выходной
файл: Стандартный выход

Условие

Дан файл в формате `csv`, содержащий в первой колонке имя питона, а во второй колонке его длину. Необходимо считать файл, найти в нём всех питонов максимальной длины и вывести их имена по одному в строке.

Формат входного файла

Входной файл содержит данные в формате `csv`.

Формат выходных данных

Выходные данные должны содержать имена искомых питонов по одному в строке, в соответствии с их порядком во входном файле.

Примеры тестов

№	Входной файл (input.csv)	Стандартный выход
1	Igor Olegovich, 185 Sasha, 170 Lesha, 180	Igor Olegovich
2	Platon, 45 Baton, 39 Pahom, 13 Ashot, 45	Platon Ashot

Задача K. Поэлементное сложение

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Эту задачу следует решать с использованием библиотеки `numpy`.

Требуется написать программу, которая поэлементно складывает два массива.

Формат входных данных

Первая строка входных данных содержит число `NN` — длину массивов. Следующие 2 строки содержат по `NN` чисел каждая — элементы массивов.

Формат выходных данных

Выходные данные должны содержать `NN` разделенных пробелом чисел — результат поэлементного сложения.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 1 2 3 4 5 6	5 7 9

Задача L. Транспонирование квадратной матрицы

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Эту задачу следует решать с использованием библиотеки `numpy`.

Дана квадратная матрица, нужно ее транспонировать. Транспонированная матрица — это матрица, для получения которой из исходной нужно элементы каждой строки исходной матрицы записать в виде столбца в том же порядке.

Формат входных данных

Первая строка содержит число NN — количество строк (и столбцов) в матрице. Следующие NN строк содержат NN чисел каждая — элементы матрицы.

Формат выходных данных

Выходные данные должны содержать NN строк по NN чисел каждая — элементы транспонированной матрицы.

Ограничения

$1 < N < 1001 < N < 100$

Примеры тестов

№	Стандартный вход	Стандартный выход
1	3 1 2 3 4 5 6 7 8 9	1 4 7 2 5 8 3 6 9

Задача M. Увеличить по позициям

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Эту задачу следует решать с использованием библиотеки `numpy`.

Даны массив и список индексов (номеров элементов). Нумерация начинается с 1. Необходимо прибавить единицу к каждому элементу массива, индекс которого есть в списке индексов.

Формат входных данных

Первая строка входных данных содержит два числа NN и MM — длины массива и списка индексов соответственно. Вторая строка содержит NN целых чисел — элементы массива. Третья строка содержит MM целых чисел — индексы.

Формат выходных данных

Выходные данные должны содержать преобразованный массив.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	5 3 1 2 3 4 5 1 3 5	2 2 4 4 6

Задача A. dir

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Вывести сначала все публичные методы класса `str` в алфавитном порядке, а затем все приватные методы, тоже в алфавитном порядке, каждый с новой строки. Названия приватных методов начинаются с символа '_' (ascii 95).

Задача B. Hello world

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс `Hello_world`. У класса должен быть следующий интерфейс:

```
class Hello_world:
    # Конструктор
    def __init__(self):
        pass

    # Функция, по умолчанию возвращающая "Hello world!". В случае
    # использования set_new_hello_world возвращаемое значение изменяется.
    def get_hello_world(self):
        pass

    # Функция, изменяющая возвращаемое функцией get_hello_world значение на
    new_hello_world.
    def set_new_hello_world(new_hello_world):
        pass
```

Формат выходных данных

Код решения должен содержать только определение и реализацию класса.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>hello = Hello_world() print(hello.get_hello_world())</pre>	Hello world!
2	<pre>hello = Hello_world() print(hello.get_hello_world()) hello.set_new_hello_world('zbc') print(hello.get_hello_world()) hello.set_new_hello_world('Python') print(hello.get_hello_world())</pre>	Hello world! zbc Python

Задача C. Time

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс Time.

У класса должен быть следующий интерфейс:

```
class Time:
    # Конструктор, принимающий четыре целых числа: часы, минуты, секунды и
    # миллисекунды.
    # В случае, если передан отрицательный параметр, вызвать исключение
    ValueError.
    # После конструирования, значения параметров времени должны быть
    корректными:
    # 0 <= GetHour() <= 23
    # 0 <= GetMinute() <= 59
    # 0 <= GetSecond() <= 59
    # 0 <= GetMillisecond() <= 999
    def __init__(self, hours=0, minutes=0, seconds=0, milliseconds=0):
        pass
    def GetHour(self):
        pass
    def GetMinute(self):
        pass
    def GetSecond(self):
        pass
    def GetMillisecond(self):
        pass
    # Прибавляет указанное количество времени к текущему объекту.
    # После выполнения этой операции параметры времени должны остаться
    корректными.
    def Add(self, time):
        pass
    # Оператор str должен представлять время в формате
    # HH:MM:SS.milliseconds
    def __str__(self):
        pass
```

Формат выходных данных

Код решения должен содержать только определение и реализацию класса.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>time = Time(25, 11, 12, 1) print(str(time)) time.Add(Time(0, 0, 0, 1010)) print(str(time))</pre>	<pre>01:11:12.1 01:11:13.11</pre>

Задача D. ShiftableList

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс `ShiftableList(list)`, который наследуется от `list` и реализует операторы циклического сдвига влево (`<<`) и вправо (`>>`). Результатом выполнения этих операций должен быть новый объект, при этом исходный не изменяется. Если операнд справа от оператора сдвига имеет тип, отличный от `int`, необходимо вызвать исключение `TypeError`. В случае, если величина сдвига отрицательна, должен выполняться сдвиг в противоположную сторону на количество позиций, равное модулю этой величины.

Формат выходных данных

Код решения должен содержать только определение и реализацию класса.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>lst = ShiftableList(['a', 'b', 'c']) lst2 = lst <<< 1 print(', '.join(lst2))</pre>	<pre>b, c, a</pre>
2	<pre>lst = ShiftableList(['a', 'b', 'c']) lst2 = lst >>> -2 lst[0] = 'z' print(''.join(lst), ''.join(lst2))</pre>	<pre>zbc cab</pre>

Задача E. Chromosome

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс `Chromosome`, который предполагается использовать при построении генетического алгоритма.

У класса должен быть следующий интерфейс:

```
class Chromosome:
```

```

# Chromosome class constructor
# Actual functionality is to set up an array called genes.
# If boolean flag fillGenes is set to True, genes must be
# filled with random values between 0 and 1, otherwise
# it must be filled with 0.
# Length of array genes must be equal to length
# constructor parameter.
# Also initializes local variable mutationRate
# with corresponding parameter.
def __init__(self, length, mutationRate, fillGenes=False):
    pass

# Creates two offspring children using a single crossover point.
# The basic idea is to first pick a random position, create two
# children and then swap their genes starting from the randomly
# picked position point.
# Children genes must be different from both of parents.
#
# Returning type: (Chromosome, Chromosome)
def Crossover(self, another):
    pass

# Mutates the chromosome genes in place by randomly switching them
# depending on mutationRate. More precisely, mutation
# of i-th gene happens with probability of mutationRate.
def Mutate(self):
    pass

```

У класса должны быть следующие поля:

- genes — array of float. The value of `len(list(genes))` must be defined and equal to length;
- length — int;
- mutationRate — float.

Формат выходных данных

Код решения должен содержать только определение и реализацию класса.

Задача F. dir*

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

То же для произвольного встроенного класса (Вывести сначала все публичные методы класса в алфавитном порядке, а затем все приватные методы, тоже в алфавитном порядке, каждый с новой строки.)

Формат входных данных

Входные данные содержат одну строку — название класса, список методов которого необходимо вывести.

Задача A. Простой декоратор

Входной
файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python функции-декораторы.

Функция `time_decorator` должна принимать декорируемую функцию и возвращать функцию выводящую время начала выполнения функции, до выполнения декорируемой функции. После выполнения декорируемой функции должно выводиться время окончания выполнения функции и общее время выполнения в миллисекундах.

```
def time_decorator (decorated_func):  
    pass
```

Функция `counter_decorator` должна принимать декорируемую функцию и возвращать функцию выводящую количество вызовов декорируемой функции. И выполнять декорируемую функцию.

```
def counter_decorator(decorated_func):  
    pass
```

Функция `method_decorator` должна принимать метод класса с одним аргументом, и возвращать функцию от 3-х аргументов (x, y, z), и вызывать декорируемый метод от аргумента $3 * x - 7 * y + 15 * z + 18$.

```
def method_decorator(decorated_method):  
    pass
```

Формат выходных данных

Код решения должен содержать только определение и реализацию функций.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>@time_decorator def floyd_warshall (n): dp = [[0] * n for i in range(n)] for i in range(n): for j in range(n): if (i == j): dp[i][j] = 0 else: dp[i][j] = random.randint(1, n) for k in range(n): for i in range(n): for j in range(n): dp[i][j] = min(dp[i][j], dp[i][k] + dp[k][j]) sum = 0 for i in range(n): for j in range(n): sum += dp[i][j] print(sum) floyd_warshall(100)</pre>	<pre>Start time: 1545320524.501671 70943 End time: 1545320524.830567 Execution time: 0.32890844345092773</pre>

2	<pre>def sum (a, b): print(a + b) fn = counter_decorator(sum) fn(1, 2) fn(4, 5) fn(-4, 5)</pre>	<pre>Function calls count: 1 3 Function calls count: 2 9 Function calls count: 3 1</pre>
3	<pre>class do_it: def __init__(self): pass @method_decorator def calc(self, x): print(x * x) obj = do_it() obj.calc(1, 2, 3)</pre>	2704

Задача B. n-th element

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Реализовать функцию `nth_element(n)`, которая принимает на вход целое число `n` и возвращает функцию `f`. Функция `f` принимает на вход индексруемую коллекцию и возвращает `n`-й элемент этой коллекции (нумерация с нуля).

Формат выходных данных

Решение должно содержать только реализацию функции.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<pre>f = nth_element(2) print(f(['a', 'b', 'c', 'd', 'e']))</pre>	c

Задача C. groupby group

Входной файл: input.txt

Выходной файл: output.txt

Условие

Необходимо написать программу, которая группирует студентов по их группам.

Формат входного файла

В первой строке входного файла дано число `nn` — количество студентов. Далее следует `nn` строк, в каждой из которых записаны группа и имя студента.

Группа и имя студента разделены символом табуляции.

Формат выходного файла

Выходной файл должен содержать список студентов, сгруппированный по группам. Для каждой группы необходимо вывести имя группы, а затем все имена студентов, которые принадлежат этой группе в алфавитном порядке, каждое в новой строке.

Сами группы следуют также в алфавитном порядке.

Ограничения

$1 \leq n \leq 10$; $1 \leq m \leq 105$

Примеры тестов

№	Входной файл (input.txt)	Выходной файл (output.txt)
1	3 M8103 Sidorov Sidor M8888 Petrov Petr M8103 Ivanov Ivan	M8103 Ivanov Ivan Sidorov Sidor M8888 Petrov Petr

Задача D. Property

Входной файл: Стандартный вход

Выходной файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс `king_lion` со свойством `x`.

При создании экземпляра класса `king_lion` `x` должно быть равно 47. Должна иметься возможность получить значение свойства `x`, и присвоить значение. При этом свойство `x` должно всегда оставаться целыми числом в диапазоне от -100 до 100. В случае если `x` пытаются присвоить значение с типом отличным от `int`, то такое присвоение следует игнорировать. Если же присваиваемое число лежит вне диапазона -100 до 100, значение свойства `x` следует установить 0.

```
class king_lion():
    def __init__(self):
        pass

    def get_x (self):
        pass

    def del_x (self):
        pass

    def set_x(self, value):
        pass
```

Формат выходных данных

Код решения должен содержать только определение и реализацию класса `king_lion`.

Примеры тестов

№	Стандартный вход	Стандартный выход
1	<code>obj = king_lion()</code> <code>obj.x = asf</code>	47 47

<pre>print(obj.x) obj.x = ffffffff print(obj.x) obj.x = 10 print(obj.x)</pre>	10
---	----

Задача E. @property

Входной
файл: Стандартный вход

Выходной
файл: Стандартный выход

Условие

Требуется реализовать на языке Python класс `Student` со свойством `name` с геттером и сеттером.

```
class Student:
    def __init__(self, name=''):
        pass

    @property
    def name(self)
```

Сеттер свойства `name` должен позволять присваивать только строки, состоящие из пробелов и букв английского алфавита (в любом регистре), в остальных случаях требуется бросать исключение `ValueError`.