**Федеральное государственное образовательное бюджетное учреждение**

**высшего образования**

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**(Финансовый университет)**

Департамент анализа данных и машинного обучения

Дисциплина **«Алгоритмы и структуры данных на языке Python»**

Факультет информационных технологий и анализа больших данных

Форма обучения: **очная** Семестр: **1**

Направление подготовки: **09.03.03 Прикладная информатика**

Профиль: **ИТ-сервисы и технологии обработки данных в экономике и финансах**

**Список вопросов к экзамену «Алгоритмы и структуры данных на языке Python»**

**Введение**

1. Присвоение по ссылке и по значению. Специфика создания объектов и присвоения в Python, особенности Python в связи с распространенностью использования неизменяемых типов.
2. Специфика типизации в языках программирования (различные аспекты типизации). Реализация типизации в Python.

**Управление исполнением и списки**

1. Организация условий в языке Python, тернарный оператор.
2. Циклы в Python, работа и устройство цикла for, типичное применение range и enumerate в цикле for.
3. Циклы в Python, работа и устройство цикла while.
4. Списки в Python. Обращение к элементам списка и создание срезов.
5. Обход списка и поиск элементов в списке. Ключевые операции, проводящие к изменению списка и порождающие измененные списки, копирование списков.

**Словари, множества, кортежи**

1. Словари в Python. Итерирование по словарям, преобразование между словарями и списками в Python.
2. Многомерные списки.
3. Операции с представлениями словарей.
4. Операции со словарями, учитывающие возможное отсутствие ключа. Операции многоэлементного изменения словарей. Операции поэлементного извлечения из словаря и их использование.
5. Множества в Python. Основные способы создания, получения и изменения значений. Обход множеств.
6. Выполнение основных операций с парой множеств в Python.
7. Кортежи в Python. Отличия кортежей от списков. Распаковка и частичная распаковка кортежей.
8. Выражения генераторы и генераторы списков в Python. Использование условий в генераторах.
9. Генераторы множества и словарей. Использование условий в генераторах.
10. Функции стандартной библиотеки для работы с контейнерами.

**Функции**

1. Объявление и вызов функции в Python. Параметры функции со значением по умолчанию и комментирование функции. Получение информации о функции. Способы передачи параметров при вызове функции.
2. Передача переменного количества параметров (именованных и не именованных) в функции Python. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.
3. Анонимные функции.

**Исключения и файлы**

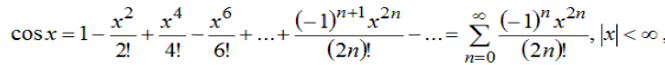
1. Синтаксис и семантика обработки исключительных ситуаций в Python.
2. Создание пользовательских исключений и инструкция assert.
3. Базовые операции для работы с файлами в Python. Использование инструкции with … as на примере работы с файлами.
4. Файлы с представлением информации в текстовом виде.
5. Модуль Pickle, создание и работа с бинарными файлами.
6. Модуль CSV, методы создания и работы с CSV файлами.

**Модули и пакеты**

1. Модули в Python и их отличие от скриптов Python. Варианты синтаксиса импорта модуля и объектов модуля. Применение импортированных объектов. Порядок поиска модулей и специфика их загрузки. Загрузка модулей из глобального репозитория.

**Список задач к эазамену «Алгоритмы и структуры данных на языке Python»**

1. Собрать статистику использования слов для текста revisor.txt и сохранить результаты в отдельных файлах. Написать функцию, которая на основе файлов статистики рассчитает статистку использования слов в тексте. Сохранить результат в виде отдельного текстового файла, в котором на каждой строке располагается информация по отдельному слову в формате: . Список слов должен быть отсортирован по частоте слов (сначала идут наиболее распространенные слова). (20 баллов)
2. В квадратной матрице, элементами которой являются случайные целые числа, размера n x n найти среднее арифметическое положительных элементов главной диагонали и среднее арифметическое отрицательных элементов побочной диагонали. Матрицу вывести на экран. (20 баллов)
3. Считать данные из текстового файла myfile.txt, в котором расположен массив. Отсортировать числа полученного массива по возрастанию, потом остальные элементы нечислового типа данных. Перезаписать результирующий массив в myfile.txt. Конечный десериализированный результат из файла выведите на экран. Стартовый файл содержит данные: ['hbac', [5,6,7], 1, 4, 8, True, 1398, 61, {'rrr':3}, -4, 24]. (20 баллов)
4. Разработать лямбда-функции вычисления по известной площади круга его радиуса и длины окружности. (20 баллов)
5. Создать три модуля. В первом расположена функция от трех параметров длин сторон параллелепипеда, она вычисляет объем параллелепипеда. Во втором модуле функция вычисляет площадь параллелепипеда. В третьем модуле функция вычисляет периметр сторон параллелепипеда. В основном файле вызывать функции из каждого модуля. Например, для значений 10, 20, 30 результаты вызова функций будет Объем = 6 000, Периметр = 200, Площадь = 2200 (20 баллов)
6. Разработать программу вычисления дня недели для произвольной даты, например, 9 мая 1945 года, 12 апреля 1961 года. (20 баллов)
7. Дана дробь 𝑛/ 𝑚 , n и m - натуральные числа. Напишите 2 функции, которые сокращают эту дробь, то есть находят числа p и q такие, что 𝑛 /𝑚 = 𝑝/ 𝑞 , и дробь 𝑝/ 𝑞 — несократимая: - аргументами функции являются числа n, m, функция возвращает кортеж (p, q); - аргументом функции является список [n, m], функция не возвращает значения, а изменяет этот список на [p, q]. (20 баллов)
8. Для произвольного значения x найти сумму первых пяти элементов убывающего ряда:



Сравнить результат с точным значением косинуса. (20 баллов)

1. Напишите функцию, которая принимает имя файла. Если файла нет, то возвращает текст "404. File (filename) not found", иначе возвращает содержимое файла. (20 баллов)
2. С помощью генератора создать в цикле список значений х от -π до +π (с небольшим произвольным шагом). Для заданного списка рассчитать значения функций y1=2sinx и y2=cos2x. (20 баллов)
3. Дано: phones\_list = [{'name':'Ivan', 'city':'Moscow', 'phones':['232-19-55', '+7 (916) 230-00-75']}, {'name':'Anna', 'city':'Samara', 'phones':['200-11-15']}, {'name':'Anna', 'city':'Vologda', 'phones':['+7 (931) 711-00-75']}, {'name':'Nikolay', 'city':'Moscow', 'phones':['+7 (916) 778-71-05', '331-66-11', '783-33- 85']}, {'name':'Ivan', 'city':'Moscow', 'phones':['+7 (916) 205-41-05', '232-19-55']}, {'name':'Anna', 'city':'Samara', 'phones':['+7 (916) 105-13-56']} ]. Переменная phones\_list хранит структуру данных со списком контактов, содержащим номера телефонов. Преобразовать этот список в словарь, ключами в котором являются города, а значениями - словари, в которых номерам телефонов людей, проживающих в данном городе, сопоставлены имена этих людей. Пример результата преобразования для данных из переменной phones\_list: {'Moscow': {'+7 (916) 205-41-05': 'Ivan', '+7 (916) 230-00-75': 'Ivan', '+7 (916) 778-71-05': 'Nikolay', '232-19-55': 'Ivan', '331-66-11': 'Nikolay', '783-33-85': 'Nikolay'}, 'Samara': {'+7 (916) 105-13-56': 'Anna', '200-11-15': 'Anna'}, 'Vologda': {'+7 (931) 711-00-75': 'Anna'}} {'Moscow': {'+7 (916) 205-41-05': 'Ivan', '+7 (916) 230-00-75': 'Ivan', '+7 (916) 778-71-05': 'Nikolay', '232-19-55': 'Ivan', '331-66-11': 'Nikolay', '783-33-85': 'Nikolay'}, 'Samara': {'+7 (916) 105-13-56': 'Anna', '200-11-15': 'Anna'}, 'Vologda': {'+7 (931) 711-00-75': 'Anna'}}. (20 баллов)
4. Для произвольного значения x найти сумму первых пяти элементов убывающего ряда:



Сравнить результат с точным значением синуса (20 баллов)

1. Дано: phones\_list = [{'name':'Ivan', 'city':'Moscow', 'phones':['232-19-55', '+7 (916) 230-00-75']}, {'name':'Anna', 'city':'Samara', 'phones':['200-11-15']}, {'name':'Anna', 'city':'Vologda', 'phones':['+7 (931) 711-00-75']}, {'name':'Nikolay', 'city':'Moscow', 'phones':['+7 (916) 778-71-05', '331-66-11', '783-33- 85']}, {'name':'Ivan', 'city':'Moscow', 'phones':['+7 (916) 205-41-05', '232-19-55']}, {'name':'Anna', 'city':'Samara', 'phones':['+7 (916) 105-13-56']} ]. Переменная phones\_list хранит структуру данных со списком контактов, содержащим номера телефонов. Из списка необходимо удалить дубликаты (записи о людях с совпадающим именем и городом). При удалении дубликатов необходимо телефоны из удаляемой записи добавить в сохраняемую запись (если их там еще нет). То есть номера телефонов, которые уже есть в сохраняемой записи, переносить из дубликата не надо (чтобы избежать дублирования номеров телефонов). (20 баллов)
2. Пользователь поочередно вводит координаты точки в декартовой системе координат. Определить, какой четверти принадлежит данная точка или на какой оси она находится. Расположение точки вывести на экран. Найти произведение номера четверти на расстояние от этой точки до начала координат и вывести его на экран. Если точка лежит на оси, считать, что номер четверти равен 0. (20 баллов)
3. Задан список с вложенными списками (элементы - числа). Для всех списков с длиной больше 3: оставить только три первых элемента, прибавив к третьему все удаленные элементы. Например, из [[1,2], [3,4,4,3,1], [4,1,4,5]] получить [[1, 2], [3, 4, 8], [4, 1, 9]]. (20 баллов)
4. Дана строка, состоящая из русских слов, набранных заглавными буквами и разделенных пробелами (одним или несколькими). Найти количество слов, которые начинаются и заканчиваются одной и той же буквой. (20 баллов)
5. Имеется переменная listlist, содержащая список списков. Заранее неизвестно, сколько в переменной списков и какой они длины (их длины в общем случае различны). Из этих списков составить список, содержащий поочередно элементы каждого из вложенных списков, при этом, порядок этих элементов сохраняется. Процедура создания списка останавливается, когда хотя бы в одном из исходных вложенных списков будут использованные все элементы. Кроме составленного списка получить список из всех элементов, не вошедших в составленный список. Пример: listlist =[[1,2,3],['a', 'b'], [30, 40, 50, 60]] -> [1, 'a', 30, 2, 'b', 40], [3, 50, 60]. (20 баллов)
6. Дана строка-предложение. Зашифровать ее, поместив вначале все символы, расположенные на четных позициях строки, а затем, в обратном порядке, все символы, расположенные на нечетных позициях (например, строка «Программа» превратится в «ргамамроП»). (20 баллов)
7. Из произвольной строки создать список. Вместо каждой буквы "s" (без учета регистра), стоящей не на первом и не на последнем месте, вставить в список строку из двойного предыдущего символа и одного следующего. Например, из строки "test\_Stringss" должен получиться список ['t', 'e', 'eet', 't', '\_', '\_\_t', 't', 'r', 'i', 'n', 'g', 'ggs', 's']. (20 баллов)
8. Задана строка, в которой через запятую перечислены слова. Создать словарь, в котором ключами будут слова из строки, а значениями - текст "номер {номер-слова-в-строке} в строке". Например, 'ten,one,five,two,three,four' преобразовать в {'three': 'номер 5 в строке', 'one': 'номер 2 в строке', 'ten': 'номер 1 в строке', 'two': 'номер 4 в строке', 'five': 'номер 3 в строке', 'four': 'номер 6 в строке'}. (20 баллов)
9. Функция func(2, 4), в качестве аргумента принимает другую функцию (не встроенную в Python). В результате работы она выводит следующие данные: название переданной функции, наименование всех принимаемых ею параметров и их типы (позиционные, ключевые, целые вещественные, строковые). Например, для вызова функции с данными параметрами func(subfunс (17, a=9.5)), должно быть выдано: имя функции subfunс, первый параметр позиционный целого типа, второй параметр ключевой вещественного типа. (20 баллов)
10. Дана матрица размера M×N. Найти номер ее строки с наибольшей суммой элементов и вывести данный номер, а также значение наибольшей суммы. (20 баллов)
11. Заданы две заранее неизвестные строки в которых слова разделены пробелами. Из двух строк составить одну, в котрой слова из первой и второй строки будут чередоваться при этом порядок слов будет сохранен. Для преобразований использовать списки слов. Пример: 'один два три', 'альфа бетта гамма' -> 'один альфа два бетта три гамма'. (20 баллов)
12. Реализовать функцию repl, которая принимает на вход строку и набор заранее неизвестных параметров. Результатом функции является строка, в которой слова совпадающие с именами параметров заменены на значения параметров. Пример: строка: 'replace my val abc', параметры my='s1', abc='fff' -> Результат: 'replace s1 val fff'. (20 баллов)
13. Выполнить циклический сдвиг слов в произвольной строке (слова разделены пробелами) на заднное пользователем число слов. Решить с помощью списков слов. Пример "один два три четыре пять шесть семь", 3 -> "четыре пять шесть семь один два три". (20 баллов)
14. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список имен параметров, отсортированный по значениям параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [b, ac, c, a]. (20 баллов)
15. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список значений параметров отсортированный по именам параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [22, 17, 16, 21]. (20 баллов)
16. Посчитать сколько раз в текстовом файле встречается символ «!». Вывести строки, в которых встречается данный символ. (20 баллов)
17. Реализовать функцию par\_val, которая принимает на вход заранее неизвестное количество именованных параметров (значения параметров - строки) и возвращает список имен параметров, которым соответствуют строки, содержащие более двух слов. Пример: par\_val(pp='abba war', fan='oneword', zr='a x') -> [pp, zr]. (20 баллов)
18. Написать функцию подсчета суммы главной диагонали. Если матрица не является квадратной, должна вызываться ошибка. Использовать конструкцию try..except. (20 баллов)
19. Преобразовать строку 'Eeny, meeny, miney, moe; Catch a tiger by his toe.' в связный список символов строки и удалить из него все элементы содержащие гласные буквы (20 баллов)
20. Напишите функцию, которая принимает имя файла. Обработать исключительную ситуацию при условии, что такого файла не существует, или не верно задан путь к файлу. Использовать конструкцию try..except. (20 баллов)
21. На основе строки, представляющей из себя предложение, построить вложенный список, содержащий символы всех слов в предложении. Пример: строка 'Eeny, meeny, miney, moe; Catch a tiger by his toe.' будет преобразована в: [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e'], ['C', 'a', 't', 'c', 'h'], ['a'], ['t', 'i', 'g', 'e', 'r'], ['b', 'y'], ['h', 'i', 's'], ['t', 'o', 'e']] (20 баллов)
22. Реализовать калькулятор для выражений с обыкновенными дробями. Число может быть записано как правильная, неправильная или смешанная дробь (вид записи выражений приведен в примерах). Должны поддерживаться операции: +, \*. Выражение включает только одну операцию. Результат представить как правильную дробь. Пример 1: 1 3/5 + 2/3 -> 2 4/15. Пример 2: 2/9 \* 6/5 -> 4/15. (20 баллов)
23. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменнымии. Приемер: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)] (20 баллов)
24. Результат сессии, состоящей из 3 экзаменов (История, Математика, Информатика), для студента задается в виде списка, содержащего фамилию студента и 3 оценки по пятибалльной системе (0-неявка, 2-неудовл., 3-удовл., 4-хорошо, 5-отлично). Результаты группы сохраняются в виде словаря. Для группы выведите на экран: таблицу с результатами экзаменов; средний балл по каждой дисциплин; средний балл для каждого студента. (20 баллов)
25. Реализовать функцию nam\_par, которая принимает на вход заранее неизвестное количество параметров и необязательный параметр name в который можно передать строку. Функция возвращает словарь в котором переданные параметры являются значениями, ключами для них являются соответсвующие (сопоставленные по порядку следования) символы из строки name. Если строка name не задана, то значения присваиваются по порядку английского алфавита. Пример 1: nam\_par(7, 3, 1, 8, 10, 13, name='xyzafg') -> {'x':7, 'y':3, 'z':1, 'a':8, 'f':10, 'g':13} Пример 2: nam\_par(21, 'val', -3.5) -> {'a':21, 'b':'val', 'c':-3.5} (20 баллов)
26. Пользователь поочередно вводит координаты точки в декартовой системе координат. Определить, какой четверти принадлежит данная точка или на какой оси она находится. Расположение точки вывести на экран. Найти произведение номера четверти на расстояние от этой точки до начала координат и вывести его на экран. Если точка лежит на оси, считать, что номер четверти равен 0. (20 баллов)
27. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа в исходной строки. Пример: 'abcd' -> ['a', 'bb', 'ccc', 'dddd']. (20 баллов)
28. Функция func(2, 4), в качестве аргумента принимает другую функцию (не встроенную в Python). В результате работы она выводит следующие данные: название переданной функции, наименование всех принимаемых ею параметров и их типы (позиционные, ключевые, целые вещественные, строковые). Например, для вызова функции с данными параметрами func(subfunс (17, a=9.5)), должно быть выдано: имя функции subfunс, первый параметр позиционный целого типа, второй параметр ключевой вещественного типа. (20 баллов)
29. Используя генератор списков (и не используя код вне него) преобразовать строку по следующей логике: для каждого символа исходной строки создать в итоговом списке строку, содержащую копии символа в количестве, равном номеру символа рассчитанному с конца исходной строки. Пример: 'abcd' -> ['aaaa', 'bbb', 'cc', 'd'] (20 баллов)
30. Написать функцию сложения двух положительных чисел. Вызвать исключение AssertionError при вводе пользователем отрицательных чисел. (20 баллов)
31. Используя генератор списков (и не используя код вне него) преобразовать список содержащий положительные целые числа в список, элементами которого являются списки с длиной равной соответствующему числу в первом списке. Содержимым вложенных списков являются последовательно идущие целые числа начиная с 1. Пример: [3, 1, 4] -> [[1, 2, 3], [1], [1, 2, 3, 4]]. (20 баллов)
32. Написать функцию подсчета суммы главной диагонали. Если матрица не является квадратной, должна вызываться ошибка. Использовать конструкцию try..except. (20 баллов)
33. Используя генератор списков (и не используя код вне него) преобразовать два списка (в первом содержатся целые числа, во втором строки, содержащие один символ) в словарь, в котором соответствующие друг другу пары значений из исходных списков преобразованы в целочисленный ключ и строку состоящую из повторенных символов (количество повтарений равно значению ключа). Пример [2, 4, 1, 3], ['a', 'b', 'c', 'd'] -> {2:'aa', 4:'bbbb', 1:'c', 3:'ddd'}. (20 баллов)
34. Написать функцию сложения двух положительных чисел. Вызвать исключение AssertionError при вводе пользователем отрицательных чисел. (20 баллов)
35. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменнымии. Приемер: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)]. (20 баллов)
36. Функция func(2, 4), в качестве аргумента принимает другую функцию (не встроенную в Python). В результате работы она выводит следующие данные: название переданной функции, наименование всех принимаемых ею параметров и их типы (позиционные, ключевые, целые вещественные, строковые). Например, для вызова функции с данными параметрами func(subfunс (17, a=9.5)), должно быть выдано: имя функции subfunс, первый параметр позиционный целого типа, второй параметр ключевой вещественного типа. (20 баллов)
37. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключем является минимальное значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,2,3):'b', (12,3):'c'} -> {2:'a', 1:'b', 2:'c'}. (20 баллов)
38. Напишите функцию, которая принимает имя файла. Обработать исключительную ситуацию при условии, что такого файла не существует, или не верно задан путь к файлу. Использовать конструкцию try..except. (20 баллов)
39. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключом является максимальное значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,11,1):'b', (2,3):'c'} -> {4:'a', 11:'b', 3:'c'}. (20 баллов)
40. Реализовать функцию find\_by\_name, в которую можно передать произвольное количество параметров, в результате функция вернет данные (в виде списка списков) только по людям, у которых в столбце 'First Name' содержатся указанные имена. Заголовок csv-файла выглядит следующим образом: Id, First Name, Last Name, Age, Phone. (20 баллов)
41. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами и значениями являются целые числа в список, в котором содержатся суммы исходных пар ключей и значений, причем, в список включаются только суммы, являющиеся нечетными числами. Пример: {2:4, 3:2, 11:6, 5:4, 1:3} -> [5, 17, 9]. (20 баллов)
42. Дана строка, состоящая из русских слов, набранных заглавными буквами и разделенных пробелами (одним или несколькими). Найти количество слов, которые начинаются и заканчиваются одной и той же буквой. (20 баллов)
43. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами и значениями являются целые числа в список, в котором содержатся суммы исходных пар ключей и значений, причем, в список включаются только суммы, являющиеся четными числами. Пример: {2:4, 3:2, 12:6, 5:4, 1:3} -> [6, 18, 4]. (20 баллов)
44. Реализовать функцию repl, которая принимает на вход строку и набор заранее неизвестных параметров. Результатом функции является строка, в которой слова совпадающие с именами параметров заменены на значения параметров. Пример: строка: 'replace my val abc', параметры my='s1', abc='fff' -> Результат: 'replace s1 val fff'. (20 баллов)
45. Используя генератор словарей (и не используя код вне него) преобразовать словарь в котором ключами являются кортежи из целых чисел в словарь в котором ключом является среднее значение из чисел исходного ключа, значение оставить прежним. Пример: {(2,4):'a', (1,1,1):'b', (2,3):'c'} -> {3.0:'a', 1.0:'b', 2.5:'c'}. (20 баллов)
46. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список имен параметров, отсортированный по значениям параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [b, ac, c, a] . (20 баллов)
47. Используя генератор словарей (и не используя код вне него) инвертировать словарь, т.е. сделать ключи словарая, его значениями и наоборото. Значения, которые в исходном словаре повторяются не добавлять в итоговый словарь. Пример: {'a':1, 'b':3, 'c':4, 'd':3} -> {1:'a', 4:'c'}. (20 баллов)
48. Реализовать функцию psort, которая принимает на вход набор заранее неизвестных поименованных параметров. Функция возвращает список значений параметров отсортированный по именам параметров. Пример: psort(c=21, a=22, ac=17, b=16) -> [22, 17, 16, 21]. (20 баллов)
49. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку ‘Eeny,meeny,miney,moe’. (20 баллов)
50. Реализовать функцию par\_val, которая принимает на вход заранее неизвестное количество именованных параметров (значения параметров - строки) и возвращает список имен параметров, которым соответствуют строки, содержащие более двух слов. Пример: par\_val(pp='abba war', fan='oneword', zr='a x') -> [pp, zr]. (20 баллов)
51. В строке содержащей последовательность слов, разделенных запятыми удалить все нечетные слова. Ответ представить в виде строки. Пример: строка 'SIX,SEVEN,EIGHT,NINE,TEN' будет преобразована в: 'SIX,EIGHT,TEN'. (20 баллов)
52. Реализовать функцию nam\_par, которая принимает на вход заранее неизвестное количество параметров и необязательный параметр name в который можно передать строку. Функция возвращает словарь в котором переданные параметры являются значениями, ключами для них являются соответсвующие (сопоставленные по порядку следования) символы из строки name. Если строка name не задана, то значения присваиваются по порядку английского алфавита. Пример 1: nam\_par(7, 3, 1, 8, 10, 13, name='xyzafg') -> {'x':7, 'y':3, 'z':1, 'a':8, 'f':10, 'g':13} Пример 2: nam\_par(21, 'val', -3.5) -> {'a':21, 'b':'val', 'c':-3.5}. (20 баллов)
53. Используя генератор списков (и не используя код вне него) преобразовать список кортежей в список кортежей по следующему правилу: если в кортеже четное количество элементов, то из него нужно удалить последний элемент. В остальных случаях кортежи оставить неизменнымии. Приемер: [(1,3,4), (2,1), (6,), (2,2,2,1)] -> [(1,3,4), (2,), (6,), (2,2,2,)]. (20 баллов)