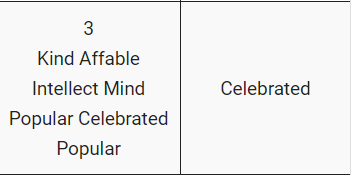
**Практика. Словари**

**Задача 1.**

Есть словарь, состоит из слов, слова разные по написанию, но имеющие схожее или тождественное значение. Повторений слов нет. Напишите программу, которая принимает на вход слово определяет по словарю его тождественное слово и выводит его.

**Задача 2.**

Существуют слова, которые состоят из одинаковых букв: кот-ток, трос-рост.

Есть два слова на входе. Создайте программу для определения, состоят эти слова из одинаковых букв или нет.

**Формат входных данных**  
Две строки по одной.

**Формат выходных данных**  
Программа должна вывести Да, если буквы в строках совпадают или Нет, если не совпадает

**Примечание.** Только буквы, других символов в словах нет.

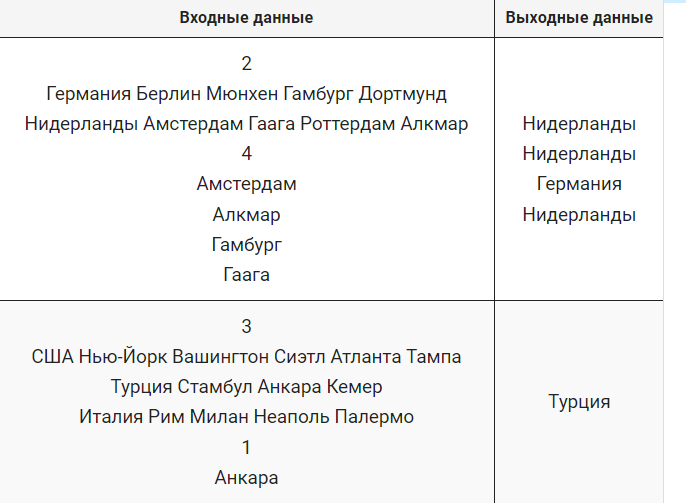


**Задача 3.**

Есть необычный список, где первый элемент страна, второй и последующие элементы это населенные пункты из этой страны. В программу поступает запрос с наименованием пункта из числа перечисленных в списке. Напишите программу, которая для каждого населенного пункта выводит, в какой стране он находится.

**Формат входных данных**  
Программа получает на вход списки (неизвестно сколько списков), где первым элементом идет страна, потом идут названия населенных пунктов этой страны. Затем вводится несколько строковых значений- пунктов из перечисленных выше.

**Формат выходных данных**  
Программа должна вывести название страны, где находится каждый населенный пункт как в примере:



**Задача 4.**

Джон записал пароли своих программ, чтобы было легче ~~вспоминать~~ находить нужный пароль. Программы Джона принимают один или несколько паролей. Нужна программа, которая будет находить в записях все пароли для определенной программы.

**Формат входных данных**  
В программу вводится несколько списков, каждый из двух символьных элементов: первый пароль, второй название программы Джона.

Затем вводится запрос в программу:имя программы, для которой будем искать пароли.

**Формат выходных данных**  
Для каждого запроса от Джона выведите в отдельной строке все пароли, принадлежащие запрашиваемой программе. Если в записной книге нет паролей для запрашиваемой программы, выведите в соответствующей строке «Программа не найдена» (без кавычек).

**Sample Input:**

ParolOne ProgramOne

Parol2 ProgramTwo

ParolTwo ProgramTwo

ParolThree ProgramThree

ProgramTwo

ProgramThree

Program4

**Sample Output:**

ParolTwo Parol2

ParolThree

Программа не найдена

**Задача 5.**

В магазине FixPrice товары продаются по фиксированным ценам: 49 руб, 99 руб, 249руб, 89руб.

Вася пришел в магазин, увидел ценник и спросил у продавца: «Какие еще товары я могу у вас купить по этой цене?» Напишите программу для помощи продавцу, которая определяет товары, которые продаются по запрошенной фиксированной цене.

**Формат входных данных**  
В программу вводится 4 списка, каждый из нескольких элементов: первый элемент-фиксированная цена, второй, третий и т.д. это названия товаров.

Затем в программу вводится цена, которую Вася озвучил продавцу.

**Формат выходных данных**

Программа должна вывести названия товаров, которые продаются в магазине по запрашиваемой цене.

**Задача 6.**

У блогеров есть две площадки: Телеграм и YouTube.

На каждой площадке зарегистрированы Pupsik , Smorchok и Shushera.

В Телеграм блогеры имеют подписчиков: Pupsik 900, Smorchok 300 и Shushera 350

В YouTube блогеры имеют подписчиков: Pupsik 1900, Smorchok 700 и Shushera 2350

Необходимо определить, какой из блогеров имеет большее количество подписчиков.

Для этого создайте программу, которая будет принимать кортежи или списки с данными о блогерах и их подписчиках, а потом формировать словари по каждой из площадок с данными блогеров. Функционал программы также заключается в том, что

а)при вводе наименования площадки программа выводит имя блогера с наибольшим числом подписчиков по данной площадке

б) при запросе «самого популярного» блогера программа выводит блогера с максимальным числом подписчиков по всем площадкам

Учтите, что программу будут использовать по большому количеству площадок и блогеров (более 100), нужен универсальный алгоритм.

**Задача 7.**

Дан словарь с правами доступа к файлам:

properties={“write” : “w”, “read” : “r”, “execute” : “e”}

От операционной системы с блока ввода-вывода поступает запрос на действия с неким файлом. В данном запросе есть характеристика прав доступа к файлу (w,r,e) .

Вам необходимо написать алгоритм-программу нахождения ключей словаря properties по запросу, содержащему значения w,r,e

Например: Вводим: e, w Получаем: execute write

**Задача 8.**

Дана база данных игроков, участвующих в шутере, где юниты уничтожаются разными видами оружия.

Записи в базе данных представлены в виде кортежей следующего вида:

(игрок, оружие, количество юнитов)

С помощью словаря создайте программу, создающую список всех видов оружия и суммарное количество юнитов, уничтоженное всеми игроками данным видом оружия.

**На вход** в программу поступает список (list) кортежей (tuple)

**На выходе** программа выдает на печать список всех видов оружия и суммарное количество юнитов, уничтоженное всеми игроками по каждому виду оружия, например, Базука 666

Арбалет 15

Ружье 34 и т.д.

**Задача 9.**

Дана символьная строка, состоящая из нескольких слов, состоящих только из строчных букв, например, «мы с другом зашли сегодня в порт».

Напишите программу, которая из введенной строки будет составлять словарь, где в качестве ключей будут все встречающиеся в строке буквы, а в качестве значений этих ключей будет количество вхождений этих букв в строке (количество раз, когда эта буква встречается). Словарь надо вывести на печать.

**Задача 10. Управление базой данных лекарств**

Вам необходимо создать программу для управления базой данных лекарств. База данных представлена в виде словаря, где ключами являются названия лекарств, а значениями — информация о лекарстве в виде вложенного словаря.

**Требования:**

1. Создайте словарь medicines, содержащий следующие данные:

medicines = {

"Парацетамол": {"Форма выпуска": "Таблетки", "Количество": 20, "Цена за штуку": 30, "Срок хранения": "11.08.2027"},

"Амоксициллин": {"Форма выпуска": "Капсулы", "Количество": 15, "Цена за штуку": 50, "Срок хранения": "11.08.2026"},

"Ибупрофен": {"Форма выпуска": "Сироп", "Количество": 100, "Цена за штуку": 400, "Срок хранения": "11.02.2026"},

"Гидрокортизон": {"Форма выпуска": "Мазь", "Количество": 30, "Цена за штуку": 250, "Срок хранения": "11.08.2026"},

"Глицерин": {"Форма выпуска": "Капсулы", "Количество": 10, "Цена за штуку": 80, "Срок хранения": "11.08.2025"},

"Цефтриаксон": {"Форма выпуска": "Мазь", "Количество": 10, "Цена за штуку": 300, "Срок хранения": "11.08.2026"},

"Левофлоксацин": {"Форма выпуска": "Глазные капли", "Количество": 10, "Цена за штуку": 500, "Срок хранения": "11.08.2025"},

"Азитромицин": {"Форма выпуска": "Таблетки", "Количество": 50, "Цена за штуку": 650, "Срок хранения": "11.02.2026"},

"Фентанил": {"Форма выпуска": "Пластырь", "Количество": 5, "Цена за штуку": 200, "Срок хранения": "11.08.2026"},

"Ксилометазолин": {"Форма выпуска": "Спрей для носа", "Количество": 20, "Цена за штуку": 320, "Срок хранения": "11.08.2025"}

}

**Напишите** функции для выполнения следующих операций:

* + **Добавление нового лекарства**: Функция должна принимать название лекарства и его данные, добавлять их в словарь medicines.
  + **Удаление лекарства**: Функция должна принимать название лекарства и удалять его из словаря medicines.
  + **Обновление информации о лекарстве**: Функция должна принимать название лекарства и обновленные данные, изменять соответствующую запись в словаре medicines.
  + **Поиск лекарства по названию**: Функция должна принимать название лекарства и возвращать его данные.
  + **Поиск лекарства форме выпуска**: Функция должна принимать форму выпуска лекарства и возвращать все лекарства с данной формой выпуска в виде словаря.

Сделайте корректный вызов всех функций для проверки их работоспособности.

**Задача 11.**

В библиотеке “Читай-город” книги выдаются по фиксированным срокам: 7 дней, 14 дней, 21 день, 30 дней. Маша пришла в библиотеку и спросила у библиотекаря: «Какие книги я могу взять на 14 дней?» Напишите программу для помощи библиотекарю, которая определяет, какие книги можно взять на запрошенный срок, учитывая, что у каждой книги есть определенное количество экземпляров, и некоторые книги могут быть зарезервированы.

**Формат входных данных:**

1. В программу вводится словарь, где ключами являются фиксированные сроки, а значениями - списки кортежей, каждый из которых содержит название книги, количество экземпляров и количество зарезервированных экземпляров.
2. Затем вводится срок, который Маша озвучила библиотекарю.

**Формат выходных данных:** Программа должна вывести названия книг, которые можно взять на запрашиваемый срок, учитывая доступные экземпляры.

Пример:

**Входные данные:**

books = {

7: [("Книга1", 3, 1), ("Книга2", 2, 0)],

14: [("Книга3", 5, 2), ("Книга4", 4, 4)],

21: [("Книга5", 2, 1), ("Книга6", 1, 0)],

30: [("Книга7", 6, 3), ("Книга8", 3, 1)]

}

requested\_period = 14

**Выходные данные:**

Книги на 14 дней: "Книга3"

В данном примере “Книга4” не выводится, так как все экземпляры зарезервированы.

**Задача 12. Подсчет количества четных и нечетных чисел**

Напишите программу, которая принимает словарь, где ключи - это имена, а значения - списки чисел. Программа должна подсчитать количество четных и нечетных чисел для каждого списка, используя словарь (dict).

**Формат входных данных:**

* Один словарь, где ключи - строки (имена), а значения - списки чисел.

**Формат выходных данных:**

* Для каждого имени программа должна вывести количество четных и нечетных чисел в формате “имя: четные: количество, нечетные: количество”.

**Пример:**

**Входные данные:**

{

"Алиса": [1, 2, 3, 4, 5],

"Боб": [6, 7, 8, 9, 10]

}

**Выходные данные:**

Алиса: четные: 2, нечетные: 3

Боб: четные: 3, нечетные: 2

Понял, давайте создадим задачу, которая требует использования различных методов словарей.

**Задача 13. Анализ оценок студентов**

Напишите программу, которая принимает словарь, где ключи - это имена студентов, а значения - списки их оценок. Программа должна выполнить следующие действия:

1. Подсчитать среднюю оценку каждого студента.
2. Найти студента с самой высокой средней оценкой.
3. Создать новый словарь, где ключи - это имена студентов, а значения - их средние оценки.

**Формат входных данных:**

* Один словарь, где ключи - строки (имена студентов), а значения - списки чисел (оценки).

**Формат выходных данных:**

* Программа должна вывести имя студента с самой высокой средней оценкой и новый словарь со средними оценками всех студентов.

**Пример:**

**Входные данные:**

{

"Алиса": [85, 90, 78],

"Боб": [92, 88, 84],

"Чарли": [70, 75, 80]

}

**Выходные данные:**

Лучший студент: Боб

Средние оценки: {'Алиса': 84.33, 'Боб': 88.0, 'Чарли': 75.0}

**Задача 14. Анализ продаж товаров с учетом скидок**

Напишите программу, которая принимает словарь, где ключи - это названия товаров, а значения - списки чисел, представляющих количество проданных единиц за каждый день. Программа должна выполнить следующие действия:

1. Подсчитать общее количество проданных единиц каждого товара.
2. Найти товар с наибольшим количеством продаж.
3. Применить скидку 10% к товарам, у которых общее количество продаж превышает 50 единиц.

**Формат входных данных:**

* Один словарь, где ключи - строки (названия товаров), а значения - списки чисел (количество проданных единиц за каждый день).
* Один словарь, где ключи - строки (названия товаров), а значения - числа (цена за единицу товара).

**Формат выходных данных:**

* Программа должна вывести название товара с наибольшим количеством продаж и новый словарь с итоговой стоимостью товаров с учетом скидок.

**Пример:**

**Входные данные:**

sales\_data = {

"яблоки": [10, 20, 30],

"бананы": [5, 15, 25],

"апельсины": [8, 12, 18]

}

prices = {

"яблоки": 2,

"бананы": 1.5,

"апельсины": 3

}

**Выходные данные:**

Самый продаваемый товар: яблоки

Итоговая стоимость: {'яблоки': 108.0, 'бананы': 67.5, 'апельсины': 114}

**Задача 15. Анализ оценок студентов по предметам**

Напишите программу, которая принимает словарь, где ключи - это имена студентов, а значения - словари, в которых ключи - это названия предметов, а значения - списки оценок по этим предметам. Программа должна выполнить следующие действия:

1. Подсчитать среднюю оценку каждого студента по каждому предмету.
2. Найти студента с наибольшей средней оценкой по каждому предмету.
3. Создать новый словарь, где ключи - это имена студентов, а значения - словари с их средними оценками по каждому предмету.

**Формат входных данных:**

* Один словарь, где ключи - строки (имена студентов), а значения - словари, где ключи - строки (названия предметов), а значения - списки чисел (оценки).

**Формат выходных данных:**

* Программа должна вывести имя студента с наибольшей средней оценкой по каждому предмету и новый словарь со средними оценками всех студентов по каждому предмету.

**Пример:**

**Входные данные:**

{

"Алиса": {

"математика": [85, 90, 78],

"физика": [80, 85, 88]

},

"Боб": {

"математика": [92, 88, 84],

"физика": [70, 75, 80]

},

"Чарли": {

"математика": [70, 75, 80],

"физика": [90, 95, 85]

}

}

**Выходные данные:**

Лучший студент по математике: Боб

Лучший студент по физике: Чарли

Средние оценки: {'Алиса': {'математика': 84.33, 'физика': 84.33}, 'Боб': {'математика': 88.0, 'физика': 75.0}, 'Чарли': {'математика': 75.0, 'физика': 90.0}}