**Практическая работа**

В тех задачах, где нет начального формата данных (например, в задаче 1 уже есть готовый словарь), для решения необходимо использовать словари и (или) множества.

Задание 1.

Приложение для удобного прослушивания музыки, песни хранятся в виде словаря. Каждая песня состоит из названия и продолжительности с точностью до долей минут.

violator\_songs = {  
'World in My Eyes': 4.86,  
'Sweetest Perfection': 4.43,  
'Personal Jesus': 4.56,  
'Halo': 4.9,  
'Waiting for the Night': 6.07,  
'Enjoy the Silence': 4.20,  
'Policy of Truth': 4.76,  
'Blue Dress': 4.29,  
'Clean': 5.83  
}

Напишите программу, которая запрашивает у пользователя количество песен из списка и их названия, а на экран выводит общее время их звучания.

**Пример**

Сколько песен выбрать? 3

Название первой песни: Halo

Название второй песни: Enjoy the Silence

Название третьей песни: Clean

Общее время звучания песен: 14,93 минуты

Задание 2. Криптовалюта

При работе с API (application programming interface) сайта биржи по криптовалюте, вы получили такие данные в виде словаря:

data = {  
"address": "0x544444444444",  
"ETH": {  
"balance": 444,  
"totalIn": 444,  
"totalOut": 4  
},  
"count\_txs": 2,  
"tokens": [  
{  
"fst\_token\_info": {  
"address": "0x44444",  
"name": "fdf",  
"decimals": 0,  
"symbol": "dsfdsf",  
"total\_supply": "3228562189",  
"owner": "0x44444",  
"last\_updated": 1519022607901,  
"issuances\_count": 0,  
"holders\_count": 137528,  
"price": False  
},  
"balance": 5000,  
"totalIn": 0,  
"total\_out": 0  
},  
{  
"sec\_token\_info": {  
"address": "0x44444",  
"name": "ggg",  
"decimals": "2",  
"symbol": "fff",  
"total\_supply": "250000000000",  
"owner": "0x44444",  
"last\_updated": 1520452201,  
"issuances\_count": 0,  
"holders\_count": 20707,  
"price": False  
},  
"balance": 500,  
"totalIn": 0,  
"total\_out": 0  
}  
]  
}

Теперь необходимо обработать эти данные.

Напишите программу, которая выполняет следующий алгоритм действий:

1. Вывести списки ключей и значений словаря.
2. В ETH добавить ключ total\_diff со значением 100.
3. Внутри fst\_token\_info значение ключа name поменять с fdf на doge.
4. Удалить total\_out из tokens и присвоить его значение в total\_out внутри ETH.
5. Внутри sec\_token\_info изменить название ключа price на total\_price.

После выполнения алгоритма выводить результат.

Советы

Если вы достали из словаря список по ключу, то можете применять к нему методы списка.

Например: словарь[“список”].append(123)

Python возьмёт из словаря объект по ключу «список» и применит к нему метод append. Эта же логика работает с другими типами данных. Например, если вы достали из словаря словарь, то к нему можно применять методы словаря, а если достали строку — методы строк.

Задание 3. Товары

В базе данных магазина вся необходимая информация по товарам делится на два словаря: первый отвечает за коды товаров, второй — за списки количества разнообразных товаров на складе:

goods = {  
'Лампа': '12345',  
'Стол': '23456',  
'Диван': '34567',  
'Стул': '45678',  
}  
store = {  
'12345': [  
{'quantity': 27, 'price': 42},  
],  
'23456': [  
{'quantity': 22, 'price': 510},  
{'quantity': 32, 'price': 520},  
],  
'34567': [  
{'quantity': 2, 'price': 1200},  
{'quantity': 1, 'price': 1150},  
],  
'45678': [  
{'quantity': 50, 'price': 100},  
{'quantity': 12, 'price': 95},  
{'quantity': 43, 'price': 97},  
],  
}

Каждая запись второго словаря отображает, сколько и по какой цене закупалось товаров. Цена указана за одну штуку.

Напишите программу, которая рассчитывает общую стоимость позиций для каждого товара на складе и выводит эту информацию на экран.

**Результат работы программы:**

Лампа — 27 штук, стоимость 1134 рубля.

Стол — 54 штуки, стоимость 27 860 рублей.

Диван — 3 штуки, стоимость 3550 рублей.

Стул — 105 штук, стоимость 10 311 рублей.

Задание 4. Словарь синонимов

Одна библиотека поручила вам написать программу для оцифровки словарей синонимов. На вход в программу подаётся N пар слов. Каждое слово является синонимом для своего парного слова.

Реализуйте код, который составляет словарь синонимов (все слова в словаре различны), затем запрашивает у пользователя слово и выводит на экран его синоним. Обеспечьте контроль ввода: если такого слова нет, выведите ошибку и запросите слово ещё раз. При этом проверка не должна зависеть от регистра символов.

**Пример**

Введите количество пар слов: 3

Первая пара: Привет — Здравствуйте

Вторая пара: Печально — Грустно

Третья пара: Весело — Радостно

Введите слово: интересно

Такого слова в словаре нет.

Введите слово: здравствуйте

Синоним: Привет

Задание 5. Пицца

В базе данных интернет-магазина PizzaTime хранятся сведения о том, кто, что и сколько заказывал у них в определённый период. Вам нужно структурировать эту информацию и определить, сколько всего пицц купил каждый заказчик.

На вход в программу подаётся N заказов. Каждый заказ представляет собой строку вида «Покупатель — название пиццы — количество заказанных пицц». Реализуйте код, который выводит список покупателей и их заказов по алфавиту. Учитывайте, что один человек может заказать одну и ту же пиццу несколько раз.

**Пример**

Введите количество заказов: 6

Первый заказ: Иванов Пепперони 1

Второй заказ: Петров Де-Люкс 2

Третий заказ: Иванов Мясная 3

Четвёртый заказ: Иванов Мексиканская 2

Пятый заказ: Иванов Пепперони 2

Шестой заказ: Петров Интересная 5

Иванов:

Мексиканская: 2

Мясная: 3

Пепперони: 3

Петров:

Де-Люкс: 2

Интересная: 5

Задание 6. Три списка

Даны три списка.

array\_1 = [1, 5, 10, 20, 40, 80, 100]

array\_2 = [6, 7, 20, 80, 100]

array\_3 = [3, 4, 15, 20, 30, 70, 80, 120]

Нужно выполнить две задачи:

1. найти элементы, которые есть в каждом списке;
2. найти элементы из первого списка, которых нет во втором и третьем списках.

Каждую задачу нужно выполнить двумя способами:

1. без использования множеств;
2. с использованием множеств.

Пример выполнения на других данных:

array\_1 = [1, 2, 3, 4]

array\_2 = [2, 4]

array\_3 = [2, 3]

Вывод:

Задача 1:

Решение без множеств: 2

Решение с множествами: 2

Задача 2:

Решение без множеств: 1

Решение с множествами: 1

Задача 7. Ревью кода

Ваня работает middle-разработчиком на Python в IT-компании. Один кандидат на позицию junior-разработчика прислал ему код тестового задания.

В задании был словарь из трёх студентов. Необходимо:

1. Вывести на экран список пар «ID студента — возраст».
2. Написать функцию, которая принимает в качестве аргумента словарь и возвращает два значения: полный список интересов всех студентов и общую длину всех фамилий студентов.

Далее в основном коде вызывается функция, значения присваиваются отдельным переменным и выводятся на экран.

Ваня — очень придирчивый программист, и после просмотра кода он понял, что этого кандидата на работу не возьмёт, хотя код выдаёт верный результат. Вот код кандидата:

students = {  
1: {  
'name': 'Bob',  
'surname': 'Vazovski',  
'age': 23,  
'interests': ['biology, swimming']  
},  
2: {  
'name': 'Rob',  
'surname': 'Stepanov',  
'age': 24,  
'interests': ['math', 'computer games', 'running']  
},  
3: {  
'name': 'Alexander',  
'surname': 'Krug',  
'age': 22,  
'interests': ['languages', 'health food']  
}  
}  
  
def f(dict):  
lst = []  
string = ''  
for i in dict:  
lst += (dict[i]['interests'])  
string += dict[i]['surname']  
cnt = 0  
for s in string:  
cnt += 1  
return lst, cnt  
  
pairs = []  
for i in students:  
pairs += (i, students[i]['age'])  
  
my\_lst = f(students)[0]  
l = f(students)[1]  
print(my\_lst, l)

Перепишите этот код так, чтобы он был максимально pythonic и Ваня мало к чему мог придраться (только если очень захочется). Убедитесь, что программа верно работает. Проверки на существование записей в словаре не обязательны, но приветствуются.

Результат работы программы:

Список пар «ID студента — возраст»: [(1, 23), (2, 24), (3, 22)]

Полный список интересов всех студентов: {'running', 'computer games', 'math', 'languages', 'biology, swimming', 'health food'}

Общая длина всех фамилий студентов: 20

* Обратите внимание на нейминг — имена переменных и функций должны быть полезными и понятными (не стоит использовать одиночные буквы, непонятные сокращения). Названия не должны пересекаться с уже существующими в Python объектами (например, лучше не называть свою переменную print или list).
* Попробуйте найти лишние действия в коде. Если вы сможете получить нужный результат меньшим количеством действий, то не нужно заставлять Python выполнять лишние действия. Также нет необходимости заставлять Python выполнять одни и те же действия над одним и тем же объектом (например, вызывать функцию с одними и теми же входными данными несколько раз).

Задача 8. Универсальная программа

Напишите функцию, возвращающую список элементов итерируемого объекта (кортежа, строки, списка, словаря), у которых индекс — это простое число.

Для проверки на простое число напишите отдельную функцию is\_prime.

Необязательное усложнение: сделайте так, чтобы основная функция состояла только из оператора return и так же возвращала список.

**Пример вызова функции:**

print(crypto([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))

Ответ в консоли: [2, 3, 5, 7]

**Пример вызова функции:**

print(crypto('О Дивный Новый мир!'))

Ответ в консоли: ['Д', 'и', 'н', 'й', 'в', 'й', 'р']

Для нумерации элементов используйте функцию enumerate. Это позволит работать одинаково со всеми структурами данных.

Задача 9. Игроки

У вас есть словарь игроков, которые участвовали в трёх видах спорта. В словаре хранятся пары «ФИ — очки»:

players = {  
("Ivan", "Volkin"): (10, 5, 13),  
("Bob", "Robbin"): (7, 5, 14),  
("Rob", "Bobbin"): (12, 8, 2)  
}

Один программист попросил нас прислать другой вариант хранения этой информации для его базы.

Напишите программу, которая объединяет ключ словаря со значением в один кортеж, и выведите результат на экран. Постарайтесь использовать как можно более эффективное решение.

Результат работы программы:

[('Ivan', 'Volkin', 10, 5, 13), ('Bob', 'Robbin', 7, 5, 14), ('Rob', 'Bobbin', 12, 8, 2)]