

1. (**Значення для ключа**) Написати код, який для заданого словника і заданого ключа перевіряє, чи міститься ключ в словнику. Якщо ключ є серед ключів словника, то вивести на екран відповідне йому значення.

**Приклад:**

```
letters = {  
  "a": 1,  
  "b": 2,  
  "c": 3  
}  
key = "d"
```

*Output: Such a key does not exist.*

**Джерело:** Projekt MmF

2. (**Заміна літер в тексті**) Створити словник для заміни літер в тексті і використати його для створення нової версії деякого тексту.

Чи можна розв'язати дану задачу за допомогою функції *replace()*?

**Приклад:**

```
change = {  
  "a": "b",  
  "b": "c",  
  "c": "a"  
}
```

*"aaabbbc" → "bbbccca"*

**Джерело:** Projekt MmF

3. (**Кількість літер a, b, c**) Створити словник з ключами "a", "b", "c" і значенням 0 для кожного ключа. Написати програму, яка обчислює кількість літер a, b, c в деякому тексті і записує кількість у словник. Вивести значення на екран.

**Джерело:** Projekt MmF

4. (**Всі ключі для заданого значення**) Для заданого словника і заданого значення скласти список всіх ключів словника, які відповідають заданому значенню.

**Приклад:**

```
letters = {  
    "a": 2,  
    "b": 3,  
    "c": 4,  
    "d": 4,  
    "e": 2  
}
```

value = 2

Output: ["a", "e"]

**Джерело:** Projekt MmF

5. (**Максимальне значення в словнику**) Нехай задано словник із числовими значеннями. Визначити, яке значення є найбільшим і які ключі йому відповідають.

**Приклад:**

```
letters = {  
    "a": 2,  
    "b": 3,  
    "c": 4,  
    "d": 4,  
    "e": 2  
}
```

Вивід:

4

"c" "d"

**Джерело:** Projekt MmF

6. (**Аббревіатури**) Створити словник з аббревіатурами. Написати код, який зчитує з клавіатури текст, видаляє в ньому всі знаки пунктуації (див. `replace()` та `string.ascii_punctuation`), розбиває текст на слова і шукає серед них аббревіатури зі словника. Вивести на екран розшифрування аббревіатур.

**Приклад:**

```
abbrevs = {  
    "AI": "Artificial intelligence",
```

```
"EU": "European Union"  
}
```

"AI is one of the most popular fields in modern science" → "AI = Artificial Interlligence"

**Джерело:** Projekt MmF

7. (**Алфавіт**) Створити словник, ключами якого є (малі) літери алфавіту, а значеннями є їх порядкові номери.

**Вказівка:** рядок з літерами латинського алфавіту можна отримати як `string.ascii_lowercase`

**Джерело:** Projekt MmF

8. (**Словник з двох списків**) Для заданих двох списків створити словник, ключами якого є елементи першого списку, а значеннями - елементи другого списку.

**Приклад:**

```
keys = [1, 2, 3, 4]  
values = ["a", "bc", "def", "ghij"]
```

**Вивід:**

```
{  
  1: "a",  
  2: "bc",  
  3: "def",  
  4: "ghij"  
}
```

**Джерело:** <https://www.codewars.com/kata/5533c2a50c4fea6832000101>

9. (**Dictionary of factors**) Створити словник, ключі якого це числа від 2 до n, а значення - це списки дільників. При цьому до списків не включати числа 1 і саме число. Якщо число просте, то список має бути ["None"].

**Приклад:**

```
{  
  2: ['None'],  
  3: ['None'],  
  4: [2],  
  5: ['None'],  
  6: [2, 3]  
}
```

Джерело: <https://www.codewars.com/kata/58bf3cd9c4492d942a0000de>

10. (**Функція, точки на графіку і прообрази**) Для заданої математичної функції (напр.  $f(x) = x^3 - 9x$ ) написати код, який створює словник з ключами, що дорівнюють точкам між числами -10 і 10 з інтервалом 0.1 і значеннями, що дорівнюють значенням функції в цих точках.

Вивести на екран нулі функції

Вивести на екран ключі, при яких значення більші за 1

Джерело: Projekt MmF

11. (**Літери в тексті**) Створити словник, ключами якого є символи тексту, а значеннями є кількість відповідних літер в тексті.

**Приклад**

`text = "aabababacc"`

Вивід:

```
{  
  "a": 5,  
  "b": 3,  
  "c": 2  
}
```

Джерело: <https://www.codewars.com/kata/529eef7a9194e0cbc1000255>

12. (**Основна теорема арифметики**) Основна теорема арифметики стверджує, що кожне натуральне число можна розкласти на добуток степенів простих чисел, причому єдиним способом. Наприклад:

$$36 = 2^2 \times 3^2$$

$$60 = 2^2 \times 3 \times 5$$

Написати функцію, яка приймає як параметр натуральне число і повертає словник з простими дільниками числа і їх степенями

`theorem(36) → {2: 2, 3: 2}`

`theorem(60) → {2: 2, 3: 1, 5: 1}`

Джерело: Projekt MmF

13. (**Dictionary merge**) Написати функцію, яка об'єднує два словники за наступними правилами. Якщо ключ є рівно в одному словнику, то його значення переноситься в об'єднання словників. Якщо ключ є в обох словниках, то в об'єднанні словників ключу відповідатиме список із значеннями з обох словників.

**Приклад:**

```
dict1 = {  
  1: "a",  
  2: "b",  
  3: "c"  
}
```

```
dict2 = {  
  1: "f",  
  2: "g",  
  4: "h",  
  5: "j"  
}
```

*Output:*

```
{  
  1: ["a", "f"],  
  2: ["b", "g"],  
  3: "c",  
  4: "h",  
  5: "j"  
}
```

**Джерело:** <https://www.codewars.com/kata/5ae840b8783bb4ef79000094>