

Задание 1

Вводится строка из целых чисел, разделённых пробелами. Сформировать список из этих чисел.

С помощью **генератора списков** сформируйте второй список из чисел первого списка, удовлетворяющих заданному условию.

С помощью функции **filter(...)** сформируйте третий список из чисел первого списка, удовлетворяющих заданному условию.

Выведите элементы второго списка на экран.

Выведите элементы третьего списка в текстовый файл "output.txt".

Посмотрите на числа, выведенные на экран, и сравните их с числами, выведенными в файл.

Пример

Элементы, являющиеся положительными числами.

```
numbers = list(map(int, input('Введите числа, разделённые пробелами: ').split()))

numbers_2 = [number for number in numbers if number > 0]

def is_positive(number):
    return number > 0

numbers_3 = list(filter(is_positive, numbers))

print(' '.join(map(str, numbers_2)))

with open('output.txt', 'w') as f:
    f.write(' '.join(map(str, numbers_3)))
```

Индивидуальные задания

1. Элементы, являющиеся нечетными числами.
2. Кратные 4 элементы.
3. Не кратные 7 элементы.
4. Элементы, делящиеся на 3 и на 5.
5. Делящиеся на 7 и не делящиеся на 4.
6. Элементы являющиеся нечетными, не кратными 7 числами.

Задание 2

Вводится строка из целых чисел, разделённых пробелами. Сформировать список из этих чисел.

С помощью **генератора списков** сформируйте второй список, состоящий из строк "Да" на тех позициях, на которых элементы первого списка удовлетворяют заданному условию и "Нет" в противном случае.

Выведите элементы второго списка, разделённые пробелами.

С помощью **генератора словарей** и функции **zip()** сформируйте словарь, в котором ключи – уникальные элементы из первого списка, а значения – соответствующие им элементы из второго списка.

Выведите пары ключ-значение полученного словаря по одной в строке.

Пример

Элементы, являющиеся положительными числами.

```
numbers = list(map(int, input('Введите числа, разделённые пробелами: ').split()))

answers = ['Да' if number > 0 else 'Нет' for number in numbers]
numbers_and_answers = {k: v for k, v in zip(numbers, answers)}

print('\n'.join(map(str, numbers_and_answers.items())))

# Введите числа, разделённые пробелами: 1 -1 1 2 -2 2 3 -3 3 -3
# (1, 'Да')
# (-1, 'Нет')
# (2, 'Да')
# (-2, 'Нет')
# (3, 'Да')
# (-3, 'Нет')
```

Индивидуальные задания

1. Элементы, являющиеся нечетными числами.
2. Кратные 4 элементы.
3. Не кратные 7 элементы.
4. Элементы, делящиеся на 3 и на 5.
5. Делящиеся на 7 и не делящиеся на 4.
6. Элементы являющиеся нечетными, не кратными 7 числами.

Задание 3

Индивидуальные задания

1. Вводится строка. С помощью **генератора словарей** сформировать словарь, в котором ключами являются символы строки, а значениями – числа, сколько раз соответствующий символ встречается в строке. Вывести все пары ключ-значение по одному в строке.
2. Вводится строка. С помощью **генератора словарей** сформировать словарь, в котором ключами являются гласные буквы строки, а значениями – числа, сколько раз соответствующий символ встречается в строке. Вывести все пары ключ-значение по одному в строке.
3. Вводится строка. С помощью **генератора словарей** сформировать словарь, в котором ключами являются символы строки, не являющиеся гласными буквами; а значениями словаря являются числа, сколько раз соответствующий символ встречается в строке. Вывести все пары ключ-значение по одному в строке.
4. Вводятся две строки. С помощью **генератора словарей** сформировать словарь, в котором ключами являются символы, которые встречаются одновременно и в первой, и во второй строке; а значениями словаря являются числа, сколько раз соответствующий символ встречается в первой строке. Вывести все пары ключ-значение по одному в строке.
5. Вводятся две строки. С помощью **генератора словарей** сформировать словарь, в котором ключами являются символы, которые встречаются или в первой, или во второй строке; а значениями словаря являются числа, сколько раз соответствующий символ встречается в суммарно в двух строках. Вывести все пары ключ-значение по одному в строке.
6. Вводятся две строки. С помощью **генератора словарей** сформировать словарь, в котором ключами являются символы, которые встречаются в первой и не встречаются во второй строке; а значениями словаря являются числа, сколько раз соответствующий символ встречается в первой строке. Вывести все пары ключ-значение по одному в строке.

Задание 4*

1. Написать функцию **groupby_year_sort_title(d)** для решения задачи.
У вас есть список словарей, каждый из которых содержит информацию о книге, включая ее заголовок, автора и год издания. Ваша задача – **используя генераторы списков, генераторы множеств и генераторы словарей**, создать новый словарь, в котором ключами будут годы издания, а значениями – списки книг, опубликованных в соответствующем году. При этом книги в каждом списке должны быть отсортированы по алфавиту по заголовку.

Например, для такого списка словарей:

```
books = [
    {"title": "Python for Beginners", "author": "John Smith", "year": 2010},
    {"title": "Advanced Python Programming", "author": "Alice Johnson", "year": 2013},
    {"title": "Python Data Analysis", "author": "Emily Brown", "year": 2010},
    {"title": "Python Machine Learning", "author": "David Jones", "year": 2013},
    {"title": "Python Cookbook", "author": "Michael White", "year": 2010}
]
```

Должен получиться такой результат:

```
{
    2010: [
        {"title": "Python Cookbook", "author": "Michael White", "year": 2010},
        {"title": "Python Data Analysis", "author": "Emily Brown", "year": 2010},
        {"title": "Python for Beginners", "author": "John Smith", "year": 2010}
    ],
    2013: [
        {"title": "Advanced Python Programming", "author": "Alice Johnson", "year": 2013},
        {"title": "Python Machine Learning", "author": "David Jones", "year": 2013}
    ]
}
```

2. Написать функцию **groupby_author_sort_year(d)** для решения задачи. У вас есть список словарей, представляющих информацию о книгах, как и в предыдущей задаче. Теперь вам нужно, **используя генераторы списков, генераторы множеств и генераторы словарей**, создать новый словарь, где ключами будут имена авторов, а значениями - словари, содержащие информацию о книгах, написанных этим автором. При этом книги в каждом словаре должны быть отсортированы по году издания в порядке убывания.

Должен получиться такой результат:

```
{
    "John Smith": [
        {"title": "Python Cookbook", "author": "John Smith", "year": 2010},
        {"title": "Python for Beginners", "author": "John Smith", "year": 2010}
    ],
    "Alice Johnson": [
        {"title": "Advanced Python Programming", "author": "Alice Johnson", "year": 2013}
    ],
    "Emily Brown": [
        {"title": "Python Data Analysis", "author": "Emily Brown", "year": 2010}
    ],
    "David Jones": [
        {"title": "Python Machine Learning", "author": "David Jones", "year": 2013}
    ]
}
```