



# Задачник Python. Функциональное программирование на Python. Модуль itertools

**Задача 1:** Создание списка бесконечных натуральных чисел: Используйте `itertools.count` для создания итератора бесконечных натуральных чисел, начиная с 1. Выведите первые 5 чисел.

**Задача 2:** Создайте список, содержащий символ "A" повторенный 10 раз, используя `itertools.repeat`.

**Задача 3:** Бесконечное чередование "да" и "нет": Используйте `itertools.cycle` для создания итератора, бесконечно чередующего строки "да" и "нет". Выведите первые 5 элементов.

**Задача 4:** Произведение двух списков: Найдите декартово произведение списков `[1, 2]` и `['a', 'b']` с помощью `itertools.product`. Выведите результат.

**Задача 5:** Перестановки чисел в списке: Создайте список всех перестановок чисел `[1, 2, 3]` с помощью `itertools.permutations`. Выведите список перестановок.

**Задача 6:** Комбинации чисел без повторений: Генерируйте все комбинации длиной 2 из списка `[1, 2, 3, 4]` без повторений с помощью `itertools.combinations`. Выведите список комбинаций.

**Задача 7:** Комбинации чисел с повторениями: Генерируйте все комбинации длиной 2 из списка `[1, 2, 3]` с повторениями с помощью `itertools.combinations_with_replacement`. Выведите список комбинаций.

**Задача 8:** Фильтрация элементов по условию: Используя `itertools.filterfalse`, создайте список всех чисел от 1 до 10, кроме тех, что делятся на 3. Выведите результат.



**Задача 9:** Группировка соседних элементов: Используйте `itertools.groupby` для группировки соседних элементов в списке `[1, 2, 2, 3, 3, 3, 4]` по их значению. Выведите результат в формате списка кортежей (число, количество повторений).

**Задача 10:** Сжатие списка по условию: Создайте список из первых 5 чисел натурального ряда, используя `itertools.compress`, где только числа, которые делятся на 2, будут включены. Выведите результат.



### Решение 1:

Python

```
def square_list(nums):  
    return list(map(lambda x: x**2, nums))
```

### Решение 2:

Python

```
def filter_odd(nums):  
    return list(filter(lambda x: x % 2 != 0, nums))
```

### Решение 3:

Python

```
def to_upper(strings):  
    return list(map(lambda x: x.upper(), strings))
```

### Решение 4:

Python

```
def sum_numbers(nums):  
    return sum(nums)
```

### Решение 5:

Python

```
def find_min(nums):  
    return min(nums)
```



#### Решение 6:

Python

```
def join_words(words):  
    return " ".join(words) + "."
```

#### Решение 7:

Python

```
def filter_by_length(strings):  
    return list(filter(lambda x: len(x) > 3, strings))
```

#### Решение 8:

Python

```
def multiply_nums(nums):  
    from functools import reduce  
    return reduce(lambda x, y: x * y, nums)
```

#### Решение 9:

Python

```
def add_prefix(names):  
    return list(map(lambda name: "Mr. " + name, names))
```

#### Решение 10:



Python

```
def sort_descending(nums):  
    return sorted(nums, reverse=True)
```