

# Lab: Decorators

Problems for in-class lab for the [Python OOP Course @SoftUni](#).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1946>.

## 1. Number Increment

Having the following code:

```
def number_increment(numbers):  
  
    def increase():  
  
        # TODO: Implement  
  
    return increase()
```

Complete the code, so it works as expected.

### Examples

Test Code	Output
print(number_increment([1, 2, 3]))	[2, 3, 4]

## 2. Vowel Filter

Having the following code:

```
def vowel_filter(function):  
  
    def wrapper():  
  
        # TODO: Implement  
  
    return wrapper
```

Complete the code, so it works as expected:

### Examples

Test Code	Output
@vowel_filter def get_letters(): return ["a", "b", "c", "d", "e"]  print(get_letters())	["a", "e"]

### 3. Even Numbers

Having the following code:

```
def even_numbers(function):  
  
    def wrapper(numbers):  
  
        # TODO: Implement  
  
    return wrapper
```

Complete the code, so it works as expected.

#### Examples

Test Code	Output
@even_numbers def get_numbers(numbers): return numbers print(get_numbers([1, 2, 3, 4, 5]))	[2, 4]

### 4. Multiply

Having the following code:

```
def multiply(times):  
  
    def decorator(function):  
  
        # TODO: Implement  
  
    return decorator
```

Complete the code, so it works as expected.

#### Examples

Test Code	Output	Comment
@multiply(3) def add_ten(number): return number + 10  print(add_ten(3))	39	First, we add 3 to 10 = 13, and then we multiply the result by 3: 13 * 3 = 39
@multiply(5) def add_ten(number): return number + 10  print(add_ten(6))	80	(6 + 10) * 5 = 80