Lab: Iterators and Generators

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/1944

1. Custom Range

Create a **class** called **`custom_range`** that receives **start** and **end** upon initialization. Implement the **`__iter__`** and **`__next__`** methods, so the iterator returns the numbers from the start to the end (inclusive).

Note: Submit only the class in the judge system

Examples

Test Code	Output

`one_to_ten = custom_range(1, 10)`
`for num in one_to_ten:`
`    print(num)`
	1
	2
	3
	4
	5
	6
	7
	8
	9
	10

2. Reverse Iter

Create a class called **`reverse_iter`** which should receive an iterable upon initialization. Implement the **`__iter__`** and **`__next__`** methods, so the iterator returns the items of the iterable in **reversed** order.

Note: Submit only the class in the judge system

Examples

Test Code	Output

`reversed_list = reverse_iter([1, 2, 3, 4])`
`for item in reversed_list:`
`    print(item)`
	4
	3
	2
	1

3. Squares

Create a generator function called **squares** that should receive a number **n**. It should generate the squares of all numbers from **1 to n** (inclusive).

Note: Submit only the function in the judge system

Examples

Test Code	Output
`print(list(squares(5)))`	`[1, 4, 9, 16, 25]`

4. Generator Range

Create a generator function called **genrante** that receives a **start** and an **end**. It should generate all the numbers from the **start** to the **end** (inclusive).

Note: Submit only the function in the judge system

Examples

Test Code	Output
`print(list(genrange(1, 10)))`	`[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]`