

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.org/Contests/1944>.

1. Custom Range

Create a **class** called **custom_range** that receives a **start** (int) and an **end** (int) 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
<pre>one_to_ten = custom_range(1, 10) for num in one_to_ten: print(num)</pre>	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
<pre>reversed_list = reverse_iter([1, 2, 3, 4]) for item in reversed_list: print(item)</pre>	4 3 2 1

3. Vowels

Create a class called **vowels**, which should receive a string. Implement the **__iter__** and **__next__** methods, so the iterator returns only the vowels from the string.

Note: Submit only the class in the judge system

Examples

Test Code	Output
<pre>my_string = vowels('Abcedifuty0o') for char in my_string: print(char)</pre>	A e i u y o

4. 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
<pre>print(list(squares(5)))</pre>	[1, 4, 9, 16, 25]

5. Generator Range

Create a generator function called **genrange** that receives a **start** (int) and an **end** (int) upon initialization. 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
<pre>print(list(genrange(1, 10)))</pre>	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

6. Reverse String

Create a generator function called **reverse_text** that receives a **string** and yields all string characters on one line in reversed order.

Note: Submit only the function in the judge system

Examples

Test Code	Output
<pre>for char in reverse_text("step"): print(char, end='')</pre>	pets