# Lab: Iterators and Generators

Please, submit your source code solutions for the described problems to the [Judge System](https://alpha.judge.softuni.org/contests/iterators-and-generators-lab/1944).

## 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** |
| 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 |

## 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 |

## 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** |
| my\_string = vowels('Abcedifuty0o')  for char in my\_string:  print(char) | A  e  i  u  y  o |

## 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] |

## 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** |
| print(list(genrange(1, 10))) | [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] |

## 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** |
| for char in reverse\_text("step"):  print(char, end='') | pets |