CS2204 Homework: Estimating Euler's number (e)

Background

During the lectures in Week 1 we developed alternative algorithms for approximating the value of π. We briefly used another contant, also, called Euler's number (e), which is the base of the *natural logarithm* and comes up very often in scientific formulas (e.g. *Optimal Stopping*, our classwork example).

Just like π, the constant e is *transcendental*, but its numerical value can be estimated by various methods. In this homework assignment we are going to use two alternative approximations.

Series Approximation

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{1} + \frac{1}{1} + \frac{1}{1 \cdot 2} + \frac{1}{1 \cdot 2 \cdot 3} + \ldots$$

As you can see, you need to iteratively calculate a sum up until a given number of steps (n).

Limit (Bernoulli)

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$$

With this method, you need to evalute a mathematical expression with a given (n) paramter.

Tasks

You are given a skeleton program (`euler.py`). You need to add the following parts:

1. Set the `__author__` variable to your VUnetID (as a string)
2. Implement the `series` function, based on the formula above. First, remove the `pass` statement, and replace it with your algorithm. Do not forget to `return` the result. (40 pts)
3. Implement the `limit` function, based on the formula above. First, remove the `pass` statement, and replace it with your algorithm. Do not forget to `return` the result. (40 pts)
4. Add a small test program to the end of the file, but **make sure your test code runs only if the script is executed at the top level** (hint: `__name__` programming idiom, as discussed in the Lecture). Your test code should print out the following output (use your own functions to calculate the results) (20 pts):

```
series(10) = 2.7182815255731922
limit(1000000) = 2.7182804690957534
```

Grading

You can use the attached `validator.py` program to check your work. It will also estimate your final score for the homework.

Penalties

Points will be deducted if you fail to set `__author__` variable (-10 pts) and for **each PEP 8 style errors** (-1 pt for each) in your program. Also, we will remove points if your test code is not protected by the `__main__` idiom (i.e. it prints the test results when the module is *imported*, not *executed directly*).

Submission

Please, upload the final version of the following file (**and only that file**) to Brightspace:

- `euler.py`