# PHOT 110: Introduction to programming

**Midterm exam (2nd retake) questions**

Michaël Barbier, Spring semester (2023-2024)

## Before you start

The midterm exam counts for 10% of your total grade of the course. The exam is performed on the computer, and you do not need to provide any answers on paper. There is one question (the third question) which asks to correct a script which contains various errors, other questions require you to write Python scripts.

Take the following point into account before you start the exam.

- For those who use their own computer/laptop: your exam is placed as an assignment on MS-Teams. Download all files attached before you start:
  - The questions: `midterm_3_questions.pdf`
  - The script with errors for question 3: `script_with_errors.py`
  - Three cheat-sheets: for Python, Numpy, and Matplotlib.
- For those who work on the desktop computers in the lab: all the required files should be found on the Desktop.
- Make sure that the `Numpy` and `Matplotlib` libraries are installed already (we will test this together before the exam starts).
- At the end of the exam we will go around with USB drives to collect the exams: put your solutions (and output files) in a folder which has both your name and your student number in the folder name, for example "Michael_Barbier_30029034".
- Let us know if during the exam there is any issue with the computer, PyCharm, or libraries. We will try to verify this up front, but please inform us in case of any issues.

You will be asked to save plots to your folder, this can be done using the `savefig` method, see the following example (the output of this example is a simple line plot):

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
x = [1, 3]
y = [2, 5]
ax.plot(x, y)
fig.savefig("output_plot_example.png")
```

## Questions

Please solve all the following questions.

### Question 1: Multiplication table

Write a script that prints the multiplication table of N. Use a loop (e.g. `for` or `while`) structure to automate the process. Your solution script should print each product on a separate line. If you choose `N = 2` the output should look similar to:

```
0 x 2 = 0
1 x 2 = 2
2 x 2 = 4
3 x 2 = 6
4 x 2 = 8
5 x 2 = 10
6 x 2 = 12
7 x 2 = 14
8 x 2 = 16
9 x 2 = 18
10 x 2 = 20
```

Save your solution as a script with file name: `solution_1.py`.

### Question 2: Ask for a multiple of 3

Make a script that prompts a user repeatedly for input of a number until the number is a multiple of 3. To test whether a number is a multiple of 3 you can use the fact that the rest after integer division by 3 should be 0:

```python
# Example: the modulo operator % gives the rest after division
rest = 5 % 3
print("The rest after integer division of 5 by 3 = " + str(rest))
```

```
The rest after integer division of 5 by 3 = 2
```

If the number provided by the user is not a multiple of 3, prompt the user again. You can assume that the user provides a valid number as input (and not some random word/characters). This is example output of a correct script:

```
Please enter a multiple of 3: 5
The number that you provided is not a multiple of 3, please try again.

Please enter a multiple of 3: 22
The number that you provided is not a multiple of 3, please try again.

Please enter a multiple of 3: 12
Thank you, the number you provided is a multiple of 3.
```

Save your solution as a script with file name: `solution_2.py`.

## Question 3: Correct a Python script

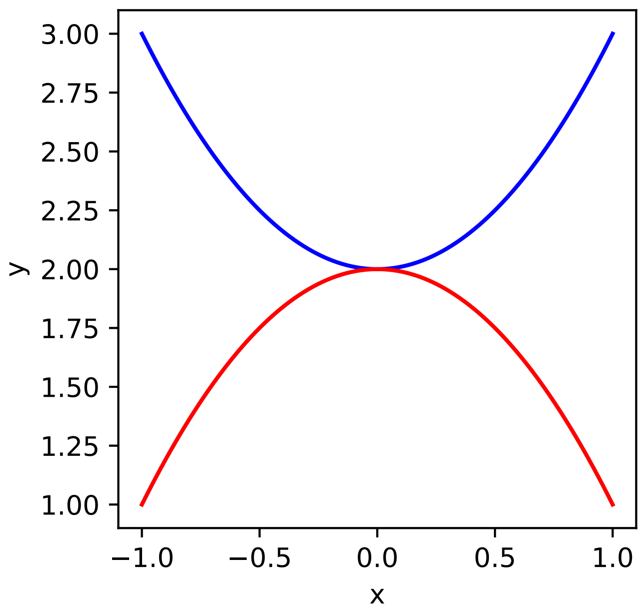
Open the script with name: `script_with_errors.py` and correct the errors. Save the corrected script with file name `solution_3.py`.

The corrected script should plot two parabolic curves with equations:

$$\begin{cases} y_1(x) = x^2 + 2 \\ y_2(x) = -x^2 + 2 \end{cases}$$

This graph is then saved as a png-file with file name `output_script_with_errors.png` to the hard disk.

The output graph should look as the plot below:



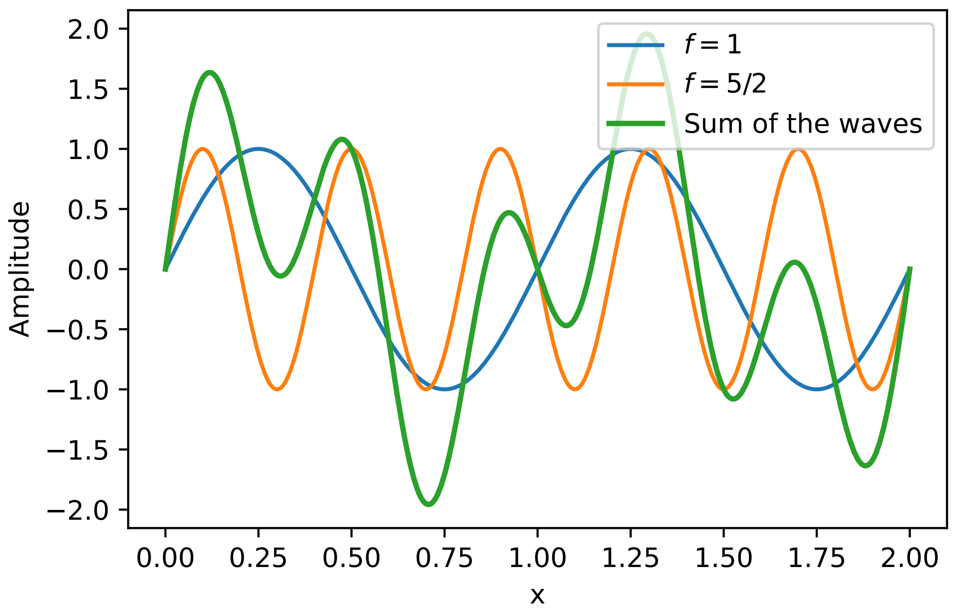
## Question 4: Plot superposition of waves

Plot two sine waves with frequencies $f_1 = 1$ and $f_2 = 5/2$:

$$\begin{cases} y_1(x) = \sin(2\pi f_1 x) \\ y_2(x) = \sin(2\pi f_2 x) \end{cases}$$

Plot both these sinusoidal functions as smooth curves. Then add a curve representing the sum of the two function $y_3(x) = y_1(x) + y_2(x)$. Plot both the separate functions and the summed function using a line plot.

For each of the curves you can use the same $x$-values within an interval $[0, 2]$ (take a sufficiently high number of $x$ values to obtain smooth curves). **Save the plot** under the file name: `output_plot_wave.png`. Save your solution as a script with file name: `solution_4.py`. The output of the script should look as in the plot below:



## Question 5: Creating and using modules

Create a **module** (a separate script file) with file name `module_validate.py` which helps converting "invalid" text or characters into valid ones. Here we assume that text is invalid if it contains spaces or commas. These symbols will be replaced by an underscore character "_". The module should contain two functions:

- `convert_text(input_text)` which accepts a string `input_text` as argument and returns the same text but with spaces and commas replaced by underscores (_ symbols)
- `check_validity(input_text)` which accepts a string `input_text` as argument and returns `True` if the text does not contain any spaces or commas, otherwise returns `False`.

Afterwards, import and use this module in a script (with file name: `solution_5.py`) that prompts a user to enter a sentence and converts it to a "valid" sentence using the `convert_text(input_text)` of above. See the following example output of a correct script:

```
Please enter a sentence: Bread, butter, and eggs
The validated text is: Bread__butter__and_eggs
```

See the hints below:

- See the following example to understand how to replace characters in a string:

```python
# In this example we replace all "e" characters by "Q"
my_text = "A green house"
adapted_text = my_text.replace("e", "Q")
print(adapted_text)
```

```
A grQQn housQ
```

- To verify whether a string contains a certain character look at following example:

```python
# In this example we verify whether the text contains character "s"
my_text = "A green house"
contains_s = "s" in my_text
print(contains_s)
```

```
True
```