A3 (16%) Submission due on 14 April 2023 at 11.59pm (Week 13 Friday, no extensions will be entertained)

Write a single python file to perform the following tasks:

(a) Perform gradient descent to minimize the cost function $f_1(a) = a^6$ with an initialization of $a = 1.0$.
(b) Perform gradient descent to minimize the cost function $f_2(b) = \cos^2(b)$ with an initialization of $b = 1.2$ (where b is assumed to be in radians).
(c) Perform gradient descent to minimize the cost function $f_3(c,d) = (c + 2d - 7)^2 + (2c + d - 5)^2$ with an initialization of $(c,d) = (0.5, 2.5)$.

Submit a single python file with filename "A3_StudentMatriculationNumber.py" (like "A3_A1234567R.py"). It should contain a function `A3` **(do not rename function name in this assigment)** that takes the following inputs and returns the following outputs in the following order:

Python function inputs:

- `learning_rate`: A fraction that falls between 0 and 0.2 (inclusive of 0 and 0.2), e.g., 0.1, etc.
- `num_iters`: A positive integer specifying the number of gradient descent iterations for each minimization task. Note that for the purpose of this exercise, you do not need to check for convergence. Just simply run the gradient descent algorithm for `num_iters` iterations.

Python function outputs in the following order:

- `a_out`: numpy array of length `num_iters` containing the updated values of a. For example, `a_out[0]` is the value of a after the first round of gradient descent (NOT the initialized value of a). (2%)
- `f1_out`: numpy array of length `num_iters` containing the updated values of f_1. For example, `f1_out[0]` is the value of f_1 after the first round of gradient descent, i.e., the value of f_1 given `a_out[0]`. (2%)
- `b_out`: numpy array of length `num_iters` containing the updated values of b. For example, `b_out[0]` is the value of b after the first round of gradient descent (NOT the initialized value of b). (2%)
- `f2_out`: numpy array of length `num_iters` containing the updated values of f_2. For example, `f2_out[0]` is the value of f_2 after the first round of gradient descent, i.e., the value of f_2 given `b_out[0]`. (2%)
- `c_out`: numpy array of length `num_iters` containing the updated values of c. For example, `c_out[0]` is the value of c after the first round of gradient descent (NOT the initialized value of c). (2%)
- `d_out`: numpy array of length `num_iters` containing the updated values of d. For example, `d_out[0]` is the value of d after the first round of gradient descent (NOT the initialized value of d). (2%)
- `f3_out`: numpy array of length `num_iters` containing the updated values of f_3. For example, `f3_out[0]` is the value of f_3 after the first round of gradient descent, i.e., the value of f_3 given `c_out[0]`and `d_out[0]` (4%)

Please use the python template provided to you. **Remember to rename** "A3_StudentMatriculationNumber.py" using your student matriculation number (**but in this assignement do not rename** "`A3`" function). For example, if your matriculation ID is A1234567R, then you should submit "A3_A1234567R.py" that contains the function "A3". Please do NOT zip/compress your file. Because of the large class size, **points will be deducted if instructions are not followed**. The way we would run your code might be something like this:

```
>> import A3_A1234567R as grading
>> learning_rate = 0.1
>> num_iters = 10
>> a_out, f1_out, b_out, f2_out, c_out, d_out, f3_out \
= grading.A3(learning_rate, num_iters)
```

Submission folder: Canvas/EE2211/Assignments/Assignment 3