Homework 4:
Policy-based DRL

In this problem set you will study Deep RL for algorithms based on policy function.

Instructions. You are to submit a PDF for your answers and Python notebooks for your codes. For the PDF, you can type your answers for example using latex or use handwritten notes as long as they are **clearly and easily readable**. Do not forget to **fully justify** your answers.

Exercise 1. Consider the code provided for PPO on the pendulum example.

1. What is the neural network architecture used for the policy?
2. What are the main hyperparameters of the PPO algorithm and what are their values in the code provided on Brightspace?
3. **(Bonus question)** Can you find a different choice of hyperparameters which lead to a similar performance but with only half the number of episodes? To show that your choice of hyperparameters answers the question, you need to run the PPO algorithm twice in a single notebook and show the two reward curves in a single plot. Provide the code and the results in a notebook called:

   `RL-Fall23-HW4_LastName_FirstName_PPO-Pendulum-improved` (with your name).

Exercise 2.
1. How would you describe the main differences between PPO and DDPG (conceptually)?
2. What are the main hyperparameters in DDPG?
3. For this question, create a new notebook and call it:

   `RL-Fall23-HW4_LastName_FirstName_DDPG-Pendulum` (with your name).

   Modify the PPO code to use instead the DDPG algorithm (still on the pendulum example). Tune the hyperparameters and run the code to make it converge. Plot the training reward. Show a video illustrating the performance of the policy learnt by DDPG.

Exercise 3.
1. How would you describe the main differences between DDPG and SAC (conceptually)?
2. What are the main hyperparameters in SAC?
3. For this question, create a new notebook and call it:

   `RL-Fall23-HW4_LastName_FirstName_SAC-Pendulum` (with your name).

   Modify the PPO code (or your DDPG code) to use instead the SAC algorithm (still on the pendulum example). Tune the hyperparameters and run the code to make it converge. Plot the training reward. Show a video illustrating the performance of the policy learnt by DDPG.