1 Frozen Lake Q-learning (Not graded)

This problem is meant to give some (optional) practice with simple Q-learning, before proceeding to full fledged DQN.
Implement tabular Q-learning for the `Stochastic-4x4-FrozenLake-v0` environment from Assignment 1. You should build on the provided starter code.
This environment is considered solved when 100 consecutive trials achieve an average score of 0.78, so you can use this to check the performance of your agent. (Consider averaging over larger number of trials if you observe lots of stochasticity)

(a) Implement Q-learning with ϵ-greedy exploration in `tabQ-learning.py`. Return an array of the Q values for each state-action pair. Have a look at the starter code for more details. You are free to play around with the values of the hyper-parameters.

(b) Plot the running average score of the Q-learning agent for the first 1000 training episodes. The x-axis should be the episode number, and the y-axis should be the average score over all training episodes so far.

You can roughly compare the obtained graph with ours to verify correctness (It will vary from run to run).