⌐Allowed to change anything in test.py / environment.py.
  Won't score if you hardcode⌐

How to know if code is implemented correctly?
  ⇒ Make sure no bug
  → Look at code and check logic if implemented correctly.

✓ How to submit final Q table when each run is different?
    → Q table should be relatively stable at the end. If policy is good.
    → There will be uncertainty so fluctuation should be fine but overall
       trend should be the same

✓ If you implement algo correctly, all Q table should not be too different.

✓ If Q table value is different from model answer?
    → Lab supervisor will be grading the lab.
    → Grade posted 2 weeks after.
✓ Do we have to explain selection of value iteration / policy interaction / Q-learning?
    → Not needed. All can be used to solve the problem. Final solution should be the same.

✓ Q-learning has epilson value but very minimal. Normally very small but usually $\varepsilon < 0.1$. No common value.

✓ If Assignment 1 grade not posted 3 weeks after submission deadline, send email to lab supervisor.

✓ Can choose value of episode.

✓ Must use new value formula in the lecture? No.

✓ Function `train` is our own implementation of value iteration / policy iteration.

✓ Given transition model. But if use Q-learning then don't need to use the model ?
   Q learning does not use this model but need to show what is your next state.

✓ Can change anything you want in the code. Just need to make sure we are not hard-coding