1. Q-learning

In this question, we will practice using the Q-learning algorithm to play tic-tac-toe. Tic-tac-toe is a simple two-player game. Each player, either X (cross) or O (circle), takes turns marking a location in a 3x3 grid. The player who first succeeds in placing three of their marks in a column, a row, or a diagonal wins the game.

1	2	3
4	5	6
7	8	9

Table 1: tic-tac-toe board positions

We will model the game as follows: each board location corresponds to an integer between 1 and 9, illustrated in the graph above. Actions are also represented by an integer between 1 and 9. Playing action a results in marking the location a and an action a is only valid if the location a has not been marked by any of the players. We train the model by playing against an expert. The agent only receives a possibly nonzero reward when the game ends. Note a game ends when a player wins or when every location in the grid has been occupied. The reward is +1 if it wins, -1 if it loses and 0 if the game draws.

O	X	
O	O	X
		X

Table 2: State 1 (circle's turn)

To further simplify the question, let's say we are the circle player and it's our turn. Our goal is to try to learn the best end-game strategy given the current state of the game illustrated in table 2. The possible actions we can take are the positions that are unmarked: $\{3, 7, 8\}$. If we select action 7, the game ends and we receive a reward of +1; if we select action 8, the expert will select action 3 to end the game and we'll receive a reward of -1; if we select action 3, the expert will respond by selecting action 7, which results in the state of the game in table 3. In this scenario, our only possible action is 8, which ends the game and we receive a reward of 0.

O	X	O
O	O	X
X		X

Table 3: State 2 (circle's turn)

Suppose we apply a learning rate $\alpha = 0.01$ and discount factor $\gamma = 1$. The Q-values are initialized as:

$$Q(1,3) = 0.5$$
$$Q(1,7) = -0.4$$
$$Q(1,8) = -0.6$$
$$Q(2,8) = 0.7$$

Note: Showing your work in these questions is optional, but it is recommended to help us understand where any misconceptions may occur. Only your answer in the left box will be graded.

(a) (1 point) In the first episode, the agent takes action 7, receives +1 reward, and the episode terminates. Derive the updated Q-value after this episode. Remember that given the sampled experience (s, a, r, s') of (state, action, reward, next state), the update of the Q value is:

$$Q(s,a) = Q(s,a) + \alpha \left(r + \gamma \max_{a' \in A} Q(s', a') - Q(s,a) \right) \tag{1}$$

Note if s' is the terminal state, $Q(s', a') = 0$ for all a'. **Please round to three decimal places**.

Q(1, 7)	Work

(b) (1 point) In the second episode, the agent takes action 8, receives a reward of -1, and the episode terminates. Derive the updated Q-value based on this episode. **Please round to three decimal places**.

Q(1, 8)	Work

(c) (2 points) In the third episode, the agent takes action 3, receives a reward of 0, and arrives at State 2 (3). It then takes action 8, receives a reward of 0, and the episode terminates. Derive the updated Q-values after each of the two experiences in this episode. Suppose we update the corresponding Q-value right after every single step. **Please round to three decimal places**.

Q(1, 3)	Q(2, 8)

Work

(d) (2 points) If we run the three episodes in cycle forever, what will be the final values of the four Q-values. **Please round to three decimal places**.

Q(1, 3)	Q(1, 7)	Q(1, 8)	Q(2, 8)

Work

(e) (2 points) What will happen if the agent adopts the greedy policy (always pick the action that has the highest current Q-value) during training? Calculate the final four Q-values in this case. **Please round to three decimal places**.

Q(1, 3)	Q(1, 7)	Q(1, 8)	Q(2, 8)

Work