Assignment 2
CP8319: Reinforcement Learning
Student Name: Albina Cako

Question 1.

a. Coding
b. Coding
c. The accuracy of the policy using 1000 iterations across 500 episodes was 100.00% on the Deterministic environment.
d. The accuracy of the policy using 1000 iterations across 500 episodes was 0% on the Stochastic environment.
   The accuracy of the policy using 10000 iterations across 500 episodes was 4.4% on the Stochastic environment.

   The policy fails due to the environment being Stochastic. In a Stochastic environment, the actions that the agent chooses are not dependent on previous state, rather random. With a few iterations such as 1000, the agent might not be able to explore all the states and it might not be able to learn to reach the optimal policy, and therefore, the goal. By increasing the number of iterations to 10000, the agent is able to explore more states, learn more and achieve a higher accuracy. Due to the randomness of the environment, it would be easier for the policy to fail and for the agent to get stuck in a hole or go backwards, instead of forward.

   As it is seen, the accuracy even at 10000 iterations is still low. To further elaborate on this point, the agent can learn the better actions to reach the goal, however, due to the stochastic environment (randomness), some of the actions might not be executed. This can also lead for the policy to fail.

   Finally, another reason contributing to the policy to perform very poorly (0% and 4.2%) might be the alpha and gamma values. A value of 0.1 means that the agent is learning slowly, therefore, it might not be ideal for this environment, especially with 1000 iterations. Increasing the alpha value could help the agent learn faster and perform better. In addition, a high gamma value of 0.9, means that the agent is strongly considering future rewards. This might not be ideal in this environment and lowering the gamma could help the agent consider recent rewards more and navigate the environment better.

Question 2.

a. Accuracy of the policy on the Deterministic environment across 500 episodes using 1000 iterations was 100%.

Accuracy of the policy on the Stochastic environment across 500 episodes is shown below:

Number of iterations	Accuracy
1000	5.8%
10000	11.2 %

The explanation for this section is same as 1d. The policy fails due to the environment being Stochastic. In a stochastic environment, the actions that the agent chooses are not dependent on previous state, rather random. With a few iterations such as 1000, the agent might not be able to explore all the states and it might not be able to learn to reach the optimal policy, and therefore, the goal. By increasing the number of iterations to 10000, the agent is able to explore more states, learn more and achieve a higher accuracy. Due to the randomness of the environment, it would be easier for the policy to fail and for the agent to get stuck in a hole or go backwards, instead of forward.

As it is seen, the accuracy even at 10000 iterations is still low. To further elaborate on this point, the agent can learn the actions to reach the goal, however, due to the stochastic environment (randomness), some of the actions might not be executed. This can also lead for the policy to fail.

Finally, another reason contributing to the policy to perform poorly might be the alpha and gamma values. A value of 0.1 means that the agent is learning slowly, therefore, it might not be ideal for this environment, especially with 1000 iterations. Increasing the alpha value could help the agent learn faster and perform better. In addition, a high gamma value of 0.9, means that the agent is strongly considering future rewards. This might not be ideal in this environment and lowering the gamma could help the agent consider recent rewards more and navigate the environment better.

b. Accuracy of the policy on the deterministic environment across 500 episodes using 1000 iterations was 100%.

Accuracy of the policy on the stochastic environment across 500 episodes is shown below:

Number of iterations	**Accuracy**
1000	15.2 %
10000	21.2 %

The explanation for this section is same as 2a. The policy fails due to the environment being Stochastic. In a stochastic environment, the actions that the agent chooses are not dependent on previous state, rather random. With a few iterations such as 1000, the agent might not be able to explore all the states and it might not be able to learn to reach the optimal policy, and therefore, the goal. By increasing the number of iterations to 10000, the agent is able to explore more states, learn more and achieve a higher accuracy. Due to the randomness of the environment, it would be easier for the policy to fail and for the agent to get stuck in a hole or go backwards, instead of forward.

As it is seen, the accuracy even at 10000 iterations is still low. To further elaborate on this point, the agent can learn the better actions to reach the goal, however, due to the stochastic environment (randomness), some of the actions might not be executed. This can also lead for the policy to fail.

Finally, another reason contributing to the policy to perform poorly might be the alpha and gamma values. A value of 0.1 means that the agent is learning slowly, therefore, it might not be ideal for this environment, especially with 1000 iterations. Increasing the alpha value could help the agent learn faster and perform better. In

addition, a high gamma value of 0.9, means that the agent is strongly considering future rewards. This might not be ideal in this environment and lowering the gamma could help the agent consider recent rewards more and navigate the environment better.

c. Please see the results on temporal-difference SARSA algorithm and Q-learning using 100000 iterations in the Stochastic environment using the changes I made:

Run #	**Temporal-difference SARSA Accuracy**	**Q-learning Accuracy**
1	100%	100%
2	97.2%	100 %
3	100 %	96.6%
4	100 %	100%
5	100 %	100%
6	100 %	98.2%
7	98.2%	100%
8	100 %	100%
9	100 %	100%
10	100 %	100%
Average	99.54%	99.48%
Variance	0.996	1.344

The first change that I made was to change the alpha value to 0.4. When alpha value is close to 1, the agent will focus on learning faster, while when it is near 0 value, Q-values are never updated, thus the agent does not learn. I decided to increase the value of alpha to 0.4, so the agent is learning at a faster rate. This can help the agent retain more information that it learned recently and learn faster. This could potentially speed up the learning process and get the agent to succeed better than before in the same number of iterations.

In addition, I changed gamma value to 0.5. Gamma is responsible for controlling whether the agent considers recent (close to 0) or future rewards (close to 1). The goal was to lower gamma, so the agent could consider recent rewards more. I experimented with this section and I noticed that choosing a value of 0.5 gamma which allows the agent to equally consider future and recent rewards, gave the best results in its performance.

Finally, I updated the rewards for the agent. The goal for updating the rewards was to guide the agent to reach the optimal policy. These are the rewards updated:

State 1: reward = 8
State 2: reward = 9
State 6: reward = 10
State 10: reward = 11
State 14: reward = 12
State 7: reward = -100
State 9: reward = -100

State 12: reward = -100
State 15: reward = 200

The reasoning behind my change is to guide the agent to achieve the optimal policy and reach the goal. I increased by 1 reward for state 2, in order to encourage the agent to move forward. Then, I increased the reward by 1 from state 2 to state 6. This was done so the agent is encouraged to go to state 6 (which is right before the hole). The idea here is to motivate the agent to go to this state so it moves forward, despite it being besides a hole. The same was done for states 10 and 14. For the states 7, 9, 12 where the holes are, a very low reward of -100 was given, in order to discourage the agent to go to these states. Finally, a high reward of 200 was given to state 15, in order to guide the agent to reach the goal.

As it is seen with all the changes the average accuracy for the Temporal-difference SARSA and Q-Learning was 99.54% and 99.48 %, respectively. This shows that due to all the changes, and reasons mentioned above, the agent's ability to navigate the environment improved and it was able to achieve very high accuracy.