Q1c) The accuracy for the policy was low and would range somewhere from the low single digits to around 20 percent randomly.

Q1d) The environment is stochastic and the agent will first visit on times where you took the action that was optimal but might have landed you on a sub optimal square. Without the ability to continuously evaluate each state/action pair multiple times in the algorithm, this increased noise makes it difficult for a first-visit MC to converge.

Q2a) The agent observes the action and reward at each step, however in this case the Q values are only updated for the previous step when a reward is received. In the current reward structure, the only time this algorithm will update the value function is in the last step before the goal if the agent even manages to reach there in the first place using exploration. This causes problems as there is no feedback in the earlier states that would update the policy to avoid the holes and consistently reach the end.

Q2b) The same reasons from the previous question hold true here. The main similarity in these methods is that they learn in one step and the same problems apply which is that the agent has no information to consistently reach the end and will update only for the steps right before reaching the goal.

Q2c) I chose to change the rewards with +1 given for reaching the goal and -1 given for entering one of the hole states. I also performed a hyperparameter search for alpha and gamma to determine the best results with the ones I ended up choosing being [09. 065] for TD-SARSA and [09. 09] for Q-learning. Running the file now gives me the mean and variance [99.80, 0.29] for TD-SARSA and [99.72, 0.71] for Q-learning. Increasing the value for alpha increase the learning rate so that the agent was able to quickly update the policy using less iterations. Additionally, giving the agent a negative reward for terminating the episode in a hole gives the agent intermediate feedback to update the policy in the earlier stages to avoid the holes. Especially in the case of the stochastic lake it can take many iterations to reach the goal if the agent does not know to avoid the holes, and if your policy is not being updated frequently because of this the effect of noise becomes greater.

