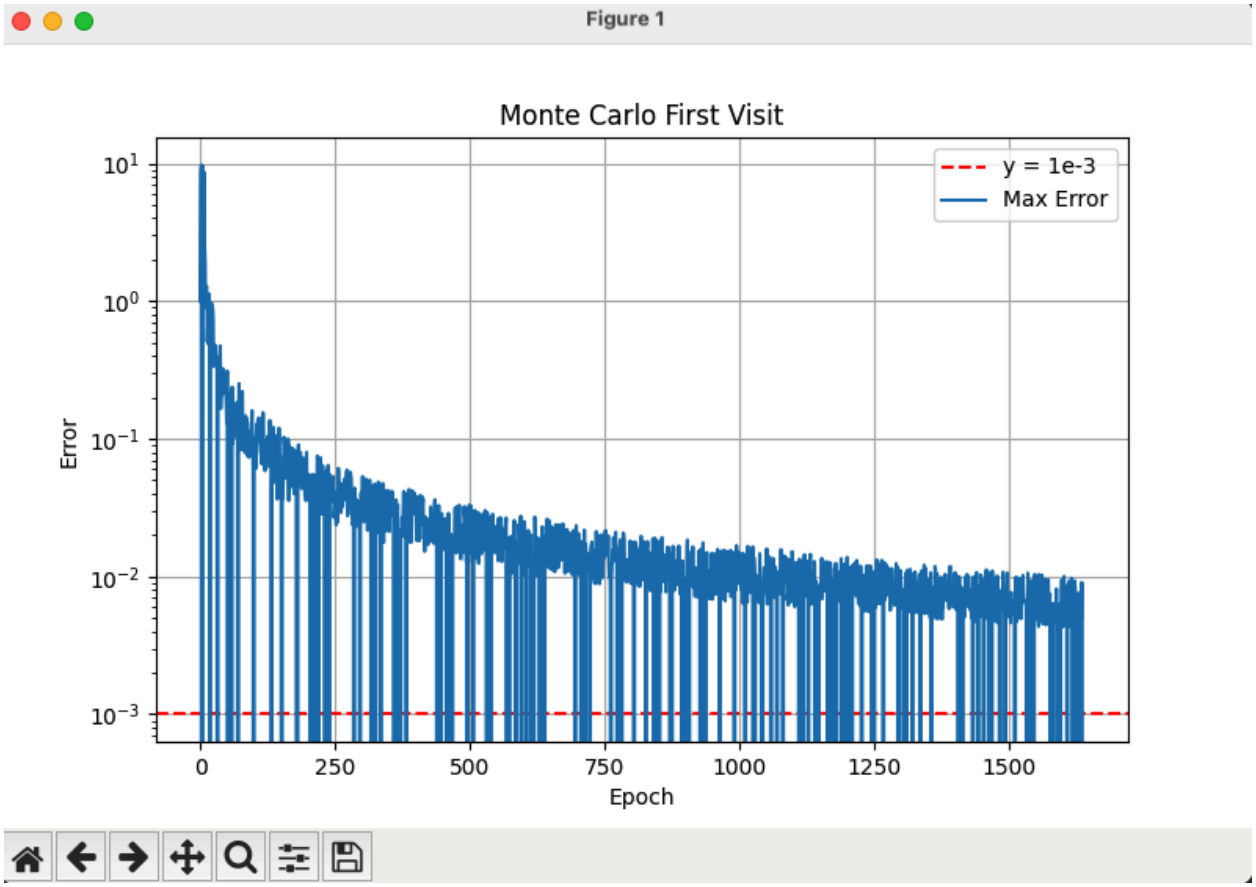
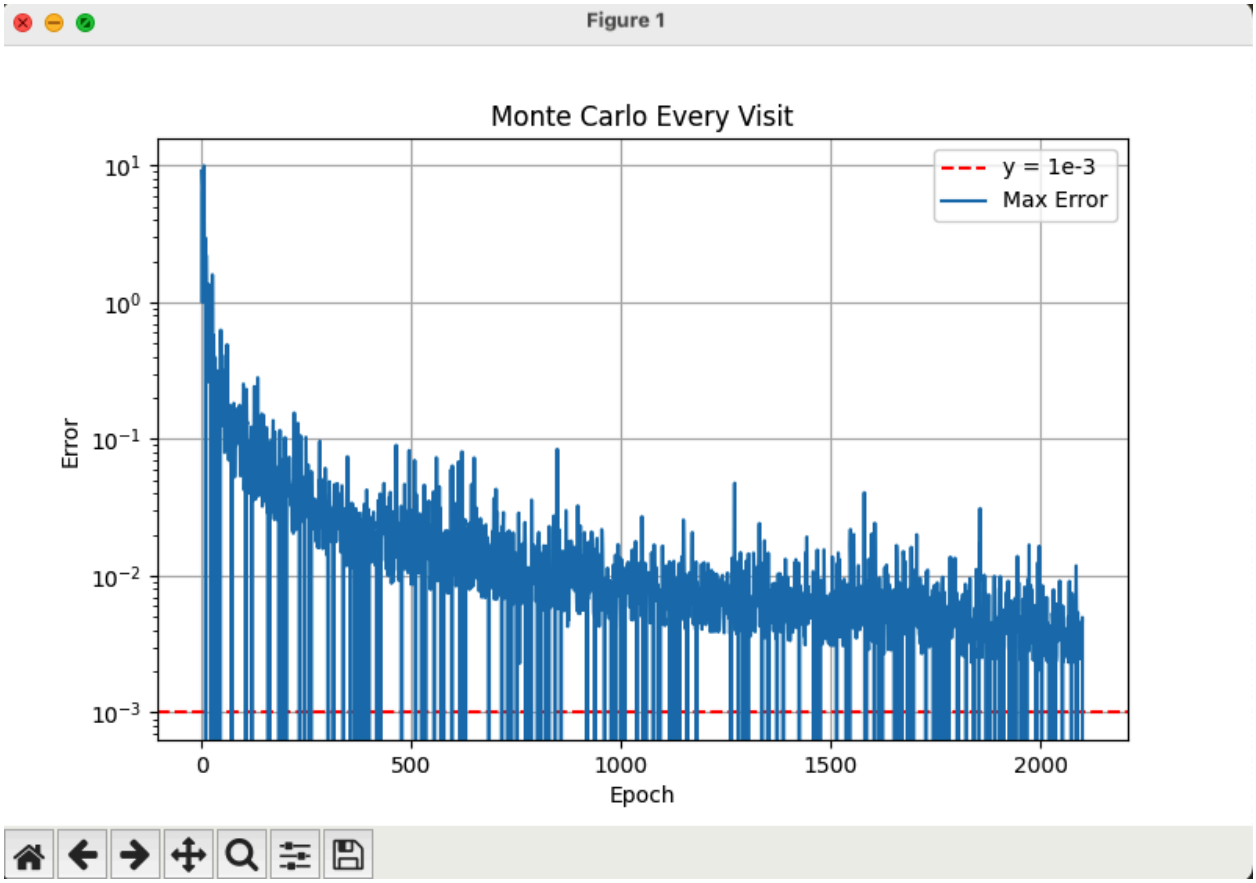
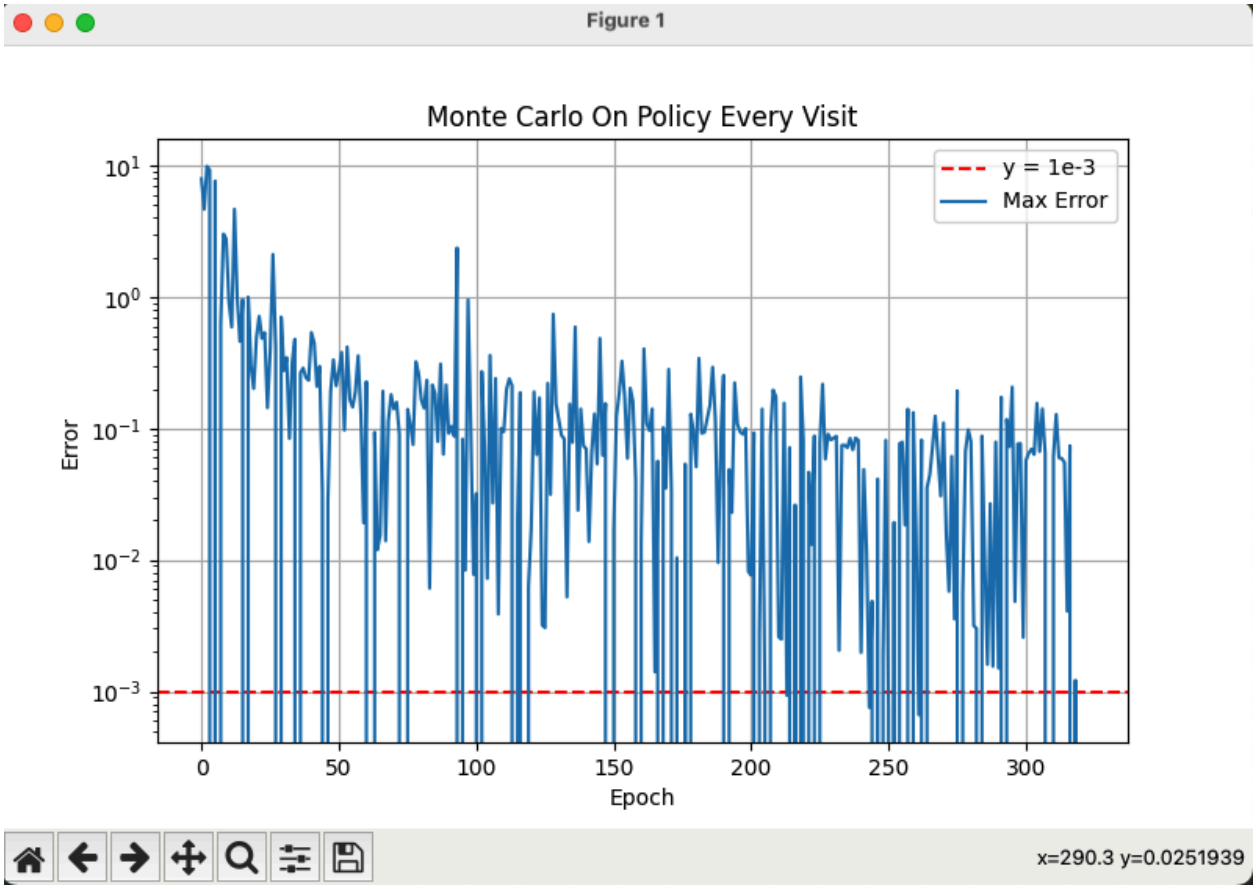
Plot of error value for part 1-



Plot of error value for part 2



Plot of error value for part 3-

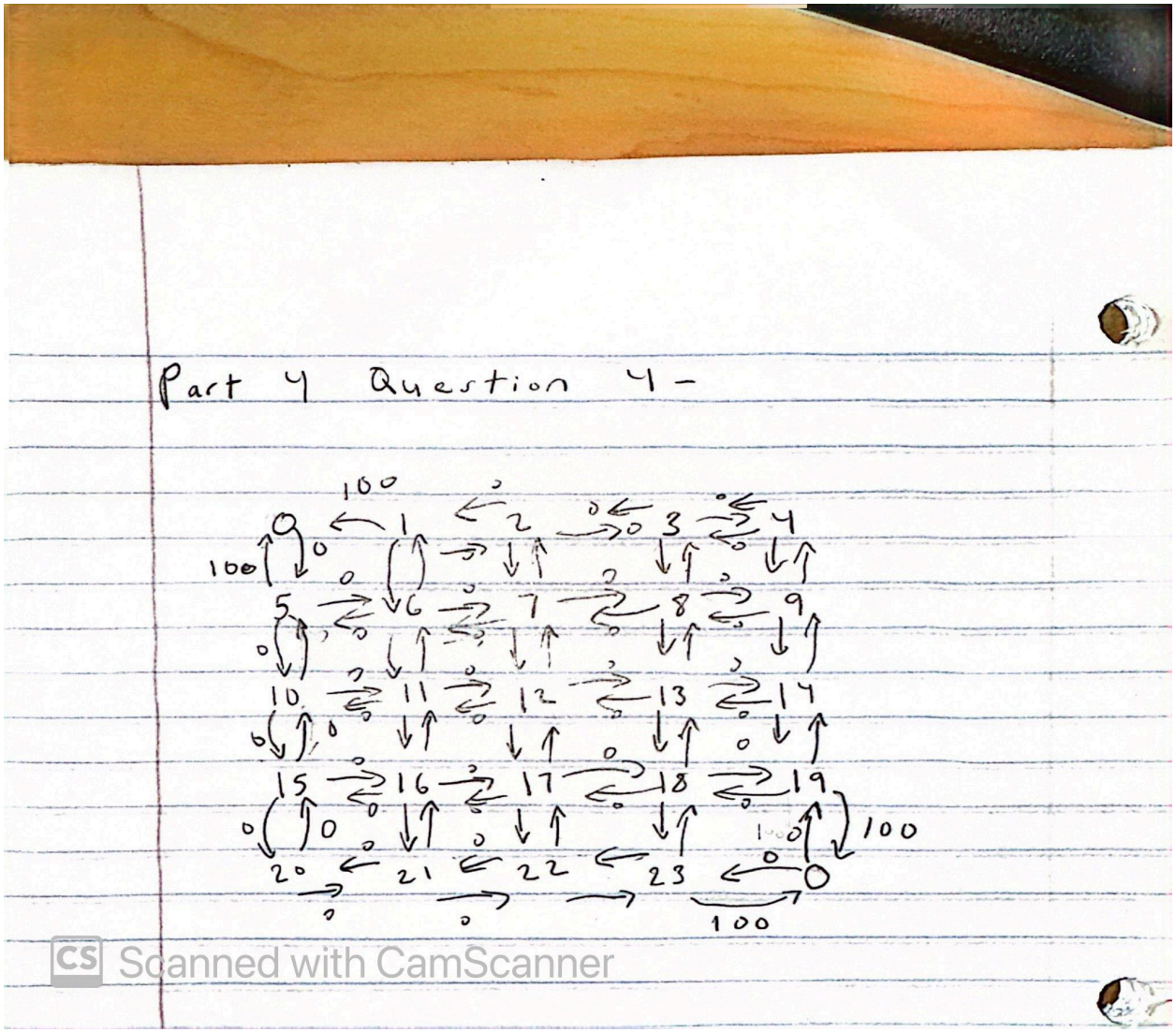



Part 1 Question 1- The convergence method that I picked was based off of the lecture slides which was to see how much the current iteration differed from the previous iterations. I added an error window of 3 to account for the randomness factor because this is an exploratory algorithm which means that the previous iteration could have been very similar to the current iteration due to chance. The way I measured the difference was to store the max difference from each value of matrix for each previous iteration and measured it against the threshold of 1e-3. This method was easy to implement as all I had to do was find the max absolute value difference between the matrices and it converges to a relatively good point depending on the threshold and error window.

Part 2 Question 2- No, they did not. This is because these algorithms are exploratory and they choose a random action every time so of course the number of epochs won't be the same. However, they should eventually converge to the same policy matrix ideally.

Part 3 Question 3- Monte Carlo on policy has aspects similar to both q-learning and sarsa because it starts off by choosing a randomly selected state that has not been visited which is more similar to q-learning because it is off policy. Then, once it reaches a state that it has already visited, it starts choosing the best action based off of the policy, which is more similar to sarsa because it is on policy. However, I would say overall it is more similar to sarsa, because once all the cells are visited, then it defaults to an on policy algorithm which is what sarsa is.

Part 4 Question 4-




Part 4 Question 5- The method that I chose was again similar to the methods used before from the lecture slides which is to compare the values from the current q matrix to the q matrix from the previous iteration and make sure none of the differences in the matrix exceed the threshold value that I set. The method was simple to implement and it converges to a good enough point based off the threshold value

Part 4 Question 6- did not finish, but should be up left left

Part 5 Question 7- Same method that used for q-learning because of the fact these algorithms are quite similar and doing the same thing except for the on policy/off policy part

Part 5 Question 8- did not finish, but should be up left left

Part 5 Question 9 - i would assume the epochs would not be the exact same because of the randomness factor introduced by the off - policy and randomness from the initial state. However, they should ideally converge to same policy eventually.

Part 6 Question 10- same method used for both q-learning and sarsa as these are all very similar algorithms with slight tweaks.

Part 6 Question 11- did not finish, but should be up left left

Part 6 Question 12- No the number of episodes was different because of the randomness factor. Everytime i ran it, it would be different.

Part 7 Question 13- The cumulative average reward were different for each part because of the following reasons:
Sarsa is a greedy algorithm so it is possible for it to converge to a suboptimal point.
Both q-learning and epsilon-greedy will converge to the optimal point but epsilon-greedy will be slightly below due to it’s exploratory nature and so it will keep slight veering off the optimal path even if it is already on the optimal path

Part 7 Question 14- this could be due to the fact that the tasks are not even the same. One is a gridworld task and the other a cliff walking example.