Part 1: The robot is using a random policy currently. This means that at every time step it takes a random action. Trial and error methods struggle with mountain car for multiple reasons. One reason is because no matter what actions we take we are getting a negative reward. This means that all actions seem equally bad at first and its nearly impossible for a random policy to reach the goal.

Part 2: I was able to get a score of -93. I did this by holding down left and then when I was high enough going right. I also got other scores with other strategies but this was the way to get the lowest score. I don't think it explores the state space well but it was still successful.

Part 3: I was able to make each of the six trials successful after watching my demonstration. It did a really good imitation.

- Max: –112.0
- Min: -122.0
- Average: -116.33

Part 4: It didn't perform better, just slightly worse. This is because when providing five different demonstrations there was more variability in the demonstrations. I tried to make sure they were consistent but some were worse than others.

- Max: -112.0
- Min: -123.0
- Average: -115.33

Part 5: The policy learns that no matter what when you are in the middle of the valley you go right. This is because in both demonstrations this is what happened. Bad demonstrations are a huge problem because all behavioral cloning knows how to do is to imitate. If we only hit certain states in a bad demonstration we will never be able to be successful.

Part 6: The agent learned to oscillate and only got to the flag on a few attempts. I tried to get as close to the flag as possible so even though I trained it not to ever touch the flag it had enough velocity to get up sometimes.

Part 7: The first thing we would have to do is ensure that we are only giving state data to the model rather than action data. To do this we would have to figure out how to observe state without being given the exact state action pairs. We would also need to create an inverse dynamics model that learns through exploration.

Part 8: I actually got it to work surprisingly well. With the hyperparameters I chose I was able to get the car to the flag each time. To do this I made my learning rate a little smaller, increased the number of interactions, gave 5 demonstrations, and did 500 steps of full batch gradient descent.