## Part 1

From the [Toy Text](https://example.com/toy_text) collection of problems available, choose one. Adapt the sample Q-Learning code to learn a good policy for your problem. Choose one of “Blackjack-v0” or “Taxi-v2”. As discussed in class, report the expected rewards per epoch, by using your learned policy on a statistically significant number of epochs and averaging the sum of rewards.

## Part 2

From the [Classic Control](https://example.com/classic_control) collection of problems available, choose one. Adapt the sample Q-Learning code to learn a good policy for your problem. It is recommended that you choose “Acrobat-v1”, “CartPole-v1”, or “MountainCar-v0”. The other problems have continuous (non-discrete) action spaces. All of these problems have continuous observation space, so you’ll need to adapt to use a finite sized Q-table for Q-Learning.

### Notes

- If a variable is continuous, you may divide its values up into sections in order to make it pseudo-discrete. For example, if the legal values are -5.0 to 10.0, and you want 100 discrete values, divide the range into chunks of (10.0 - (-5.0)) / 100. = 0.15. The first bin is from -5.0 to -4.85, the next is -4.85 to -4.70, etc.
- If a variable is infinite, set a cap for the largest reasonable value in either direction, and put all values that are outside that range into the most extreme bin. For example, if you decide that -100 and +100 are the maximum reasonable values, any value less than -100 will be in the same bin as -100, and any value greater than 100 will be in the same bin as 100.
- If there are multiple variables, you can combine them into one state number. See the “encode” method of the [Taxi environment](https://example.com/taxi_environment).