a. The code trains three Q-Learning models, each with a unique epsilon value: 0.008, 0.08, and 0.8. Afterwards, it generates a plot to illustrate the avg. rewards earned by each model in relation to the number of steps taken.

b. Epsilon value 0.08 performs the best. Since from the first steps, the model with the epsilon value 0.08 performs better. However, the model's performance begins to show a significant advantage from the step 30 k. Moving from 30 k steps to 50 k steps it could further be observed that the model’s rate of change also tends to increase more rapidly.

c. If we think that the number of steps goes to infinity
   i. 0.008 => Will end up performing the best because the each expected value of each action will converge to q*(the actual avg. expected reward) and since 0.008 will choose the greedy/exploiting approach more often it will end up with a better overall outcome.
   ii. 0.08 => Will end up performing better than 0.8 but worse than 0.008, simply because it will tend to explore more often than 0.008 but less than 0.8.
   iii. 0.8 => Will perform the worst because it will tend to explore much more often than 0.08 and 0.008

d.
   i. When testing many different epsilon values (exploration rates), there is a risk of overfitting to the specific environment or task the agent was trained on. The optimal epsilon value found during training may not generalize well to new domains.
   ii. With a large number of hyperparameter combinations tested, the best performing set may exploit peculiarities of the training environment that don't hold in other domains. This could lead to poor performance or unexpected behavior when deployed in a new setting.