```
import gym
import numpy as np

env =
        gym.make("Taxi-v3")

num_states =
            num_
actions =
Q_table
                    env.observation_space.n
                      env.action_space.n
                np.zeros((num_states, num_actions))


num_episode
s
gamma = 0.9
epsilon  = 0.1
                 = 5000



def choose_action(state):
    if np.random.uniform(0, 1) < epsilon:
        return
        env.action_space.sample()
    else:
        return np.argmax(Q_table[state,
        :])

def update_q_values(episode_memory):
    G = 0
    for i in reversed (range(len(episode_memory))):
        state, action, reward =
        episode_memory[i]
        G = gamma * G + reward # Update the
        return
        Q_table[state] [action] += 0.1 * (G - Q_table[state][action])

for episode in
range(num_episodes):
    state =
    env.reset()
    episode_memor
    y
    while True:
        action =
                      [
                      ]

                choose_action(state
                )
        next_state, reward, done,
                                    env.step(action)
        episode_memory.append((state, action, reward))
```

```
            state = next_state
            if done:
                update_q_values(episode_memory
                )
                break


    total_reward =  0
    num_test_episodes = 10

    for in
    range(num_test_episodes):
        state =
                env.reset()
        while True:
            action =
                    np.argmax(Q_table[state,
                    :])
            next_state, reward, done, =
            total_reward += reward
            state = next_state
            if done:
                break


                                  env.step(action)




    average_reward total_reward / num_test_episodes
    print (f"Average reward over {num_test_episodes} test episodes: {average_reward}")
```

```
        Average reward over 10 test episodes: -200.0
```