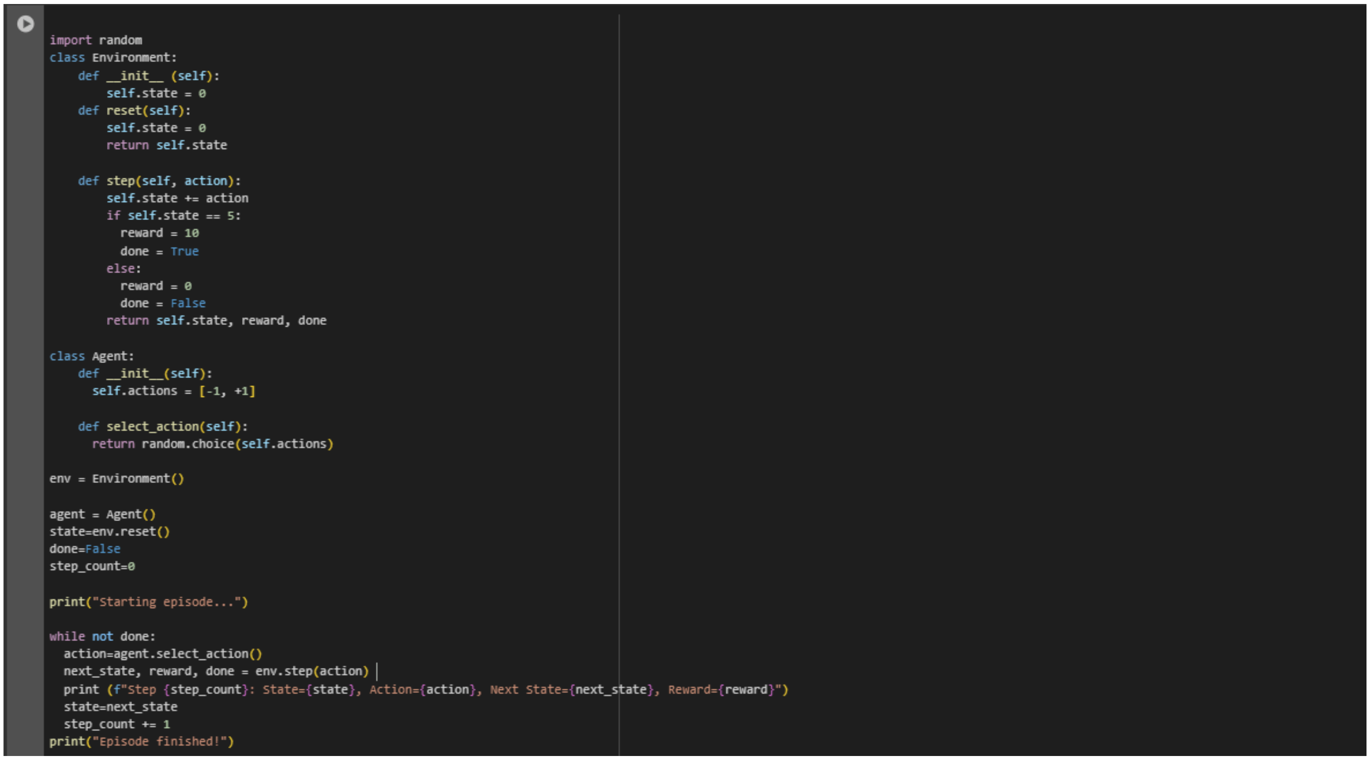
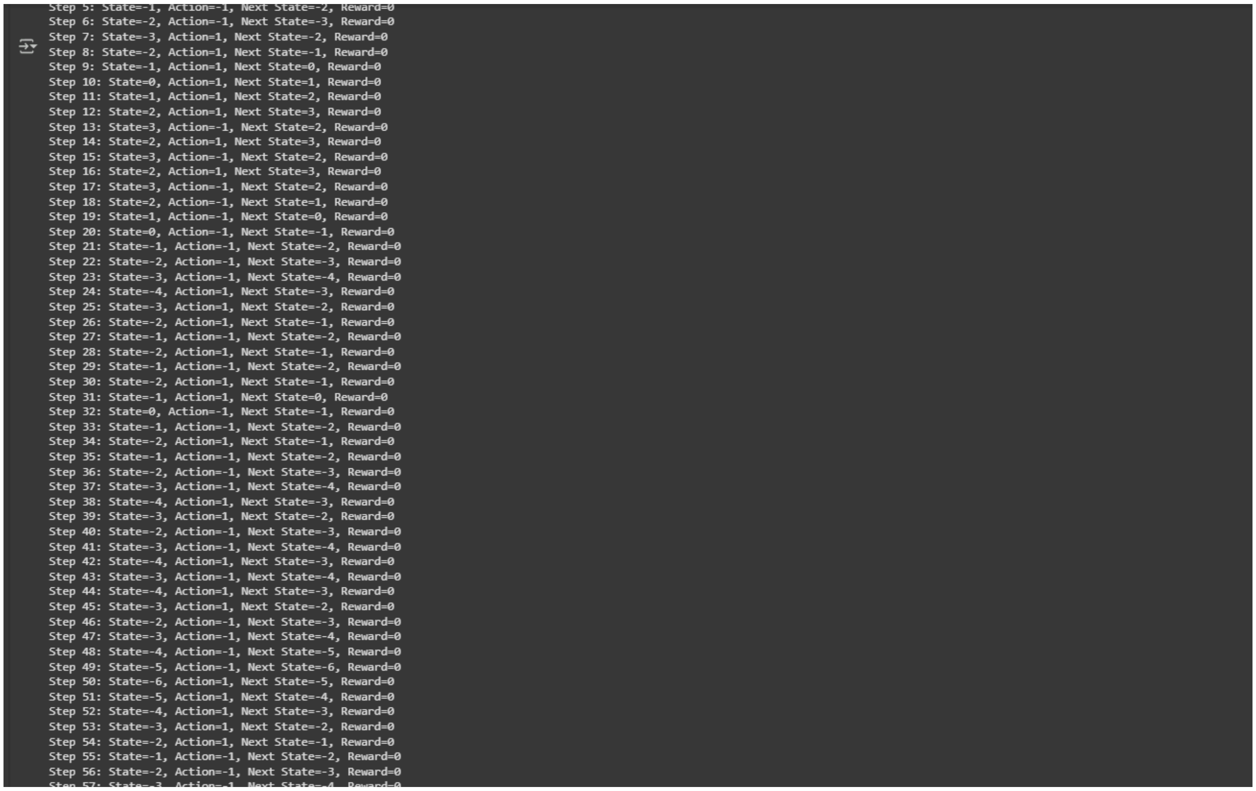
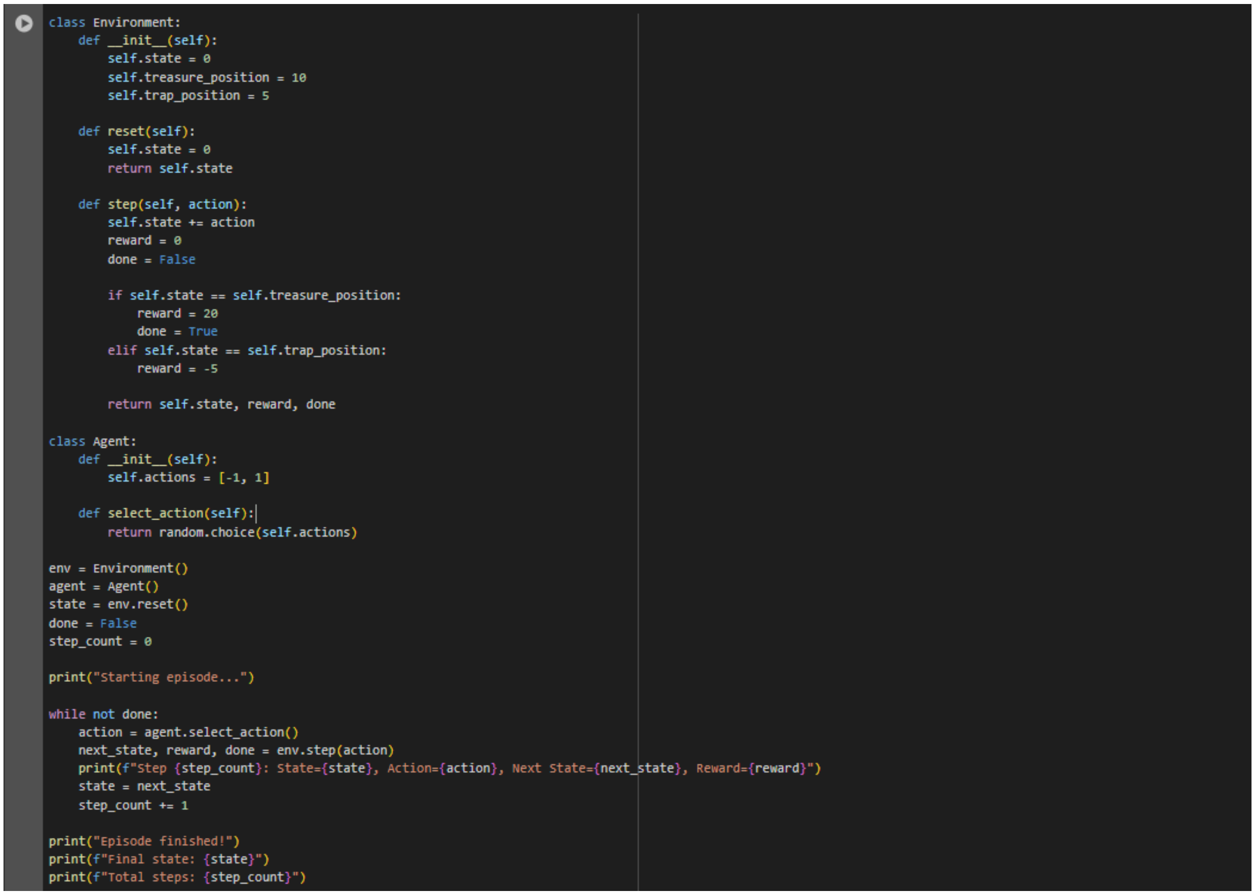


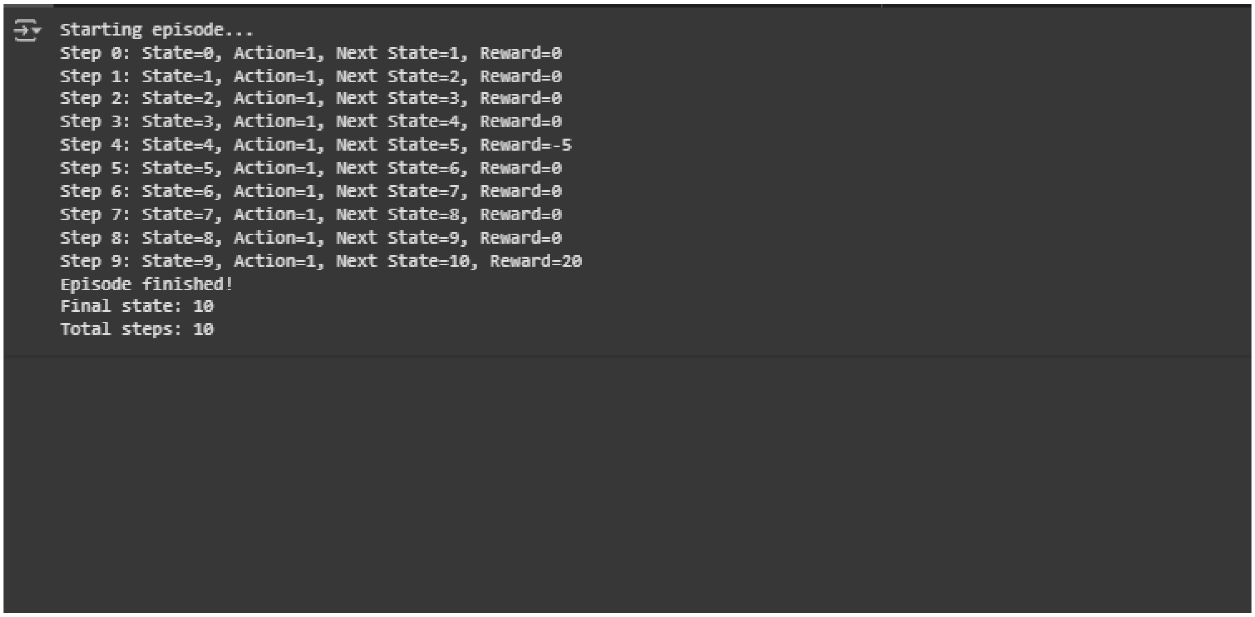
```python
import random
class Environment:
    def __init__ (self):
        self.state = 0
    def reset(self):
        self.state = 0
        return self.state

    def step(self, action):
        self.state += action
        if self.state == 5:
          reward = 10
          done = True
        else:
          reward = 0
          done = False
        return self.state, reward, done

class Agent:
    def __init__(self):
      self.actions = [-1, +1]

    def select_action(self):
      return random.choice(self.actions)

env = Environment()

agent = Agent()
state=env.reset()
done=False
step_count=0

print("Starting episode...")

while not done:
  action=agent.select_action()
  next_state, reward, done = env.step(action)
  print (f"Step {step_count}: State={state}, Action={action}, Next State={next_state}, Reward={reward}")
  state=next_state
  step_count += 1
print("Episode finished!")
```



```
Step 5: State=-1, Action=-1, Next State=-2, Reward=0
Step 6: State=-2, Action=-1, Next State=-3, Reward=0
Step 7: State=-3, Action=1, Next State=-2, Reward=0
Step 8: State=-2, Action=1, Next State=-1, Reward=0
Step 9: State=-1, Action=1, Next State=0, Reward=0
Step 10: State=0, Action=1, Next State=1, Reward=0
Step 11: State=1, Action=1, Next State=2, Reward=0
Step 12: State=2, Action=1, Next State=3, Reward=0
Step 13: State=3, Action=-1, Next State=2, Reward=0
Step 14: State=2, Action=1, Next State=3, Reward=0
Step 15: State=3, Action=-1, Next State=2, Reward=0
Step 16: State=2, Action=1, Next State=3, Reward=0
Step 17: State=3, Action=-1, Next State=2, Reward=0
Step 18: State=2, Action=-1, Next State=1, Reward=0
Step 19: State=1, Action=-1, Next State=0, Reward=0
Step 20: State=0, Action=-1, Next State=-1, Reward=0
Step 21: State=-1, Action=-1, Next State=-2, Reward=0
Step 22: State=-2, Action=-1, Next State=-3, Reward=0
Step 23: State=-3, Action=-1, Next State=-4, Reward=0
Step 24: State=-4, Action=1, Next State=-3, Reward=0
Step 25: State=-3, Action=1, Next State=-2, Reward=0
Step 26: State=-2, Action=1, Next State=-1, Reward=0
Step 27: State=-1, Action=-1, Next State=-2, Reward=0
Step 28: State=-2, Action=1, Next State=-1, Reward=0
Step 29: State=-1, Action=-1, Next State=-2, Reward=0
Step 30: State=-2, Action=1, Next State=-1, Reward=0
Step 31: State=-1, Action=1, Next State=0, Reward=0
Step 32: State=0, Action=-1, Next State=-1, Reward=0
Step 33: State=-1, Action=-1, Next State=-2, Reward=0
Step 34: State=-2, Action=1, Next State=-1, Reward=0
Step 35: State=-1, Action=-1, Next State=-2, Reward=0
Step 36: State=-2, Action=-1, Next State=-3, Reward=0
Step 37: State=-3, Action=-1, Next State=-4, Reward=0
Step 38: State=-4, Action=1, Next State=-3, Reward=0
Step 39: State=-3, Action=1, Next State=-2, Reward=0
Step 40: State=-2, Action=-1, Next State=-3, Reward=0
Step 41: State=-3, Action=-1, Next State=-4, Reward=0
Step 42: State=-4, Action=1, Next State=-3, Reward=0
Step 43: State=-3, Action=-1, Next State=-4, Reward=0
Step 44: State=-4, Action=1, Next State=-3, Reward=0
Step 45: State=-3, Action=1, Next State=-2, Reward=0
Step 46: State=-2, Action=-1, Next State=-3, Reward=0
Step 47: State=-3, Action=-1, Next State=-4, Reward=0
Step 48: State=-4, Action=-1, Next State=-5, Reward=0
Step 49: State=-5, Action=-1, Next State=-6, Reward=0
Step 50: State=-6, Action=1, Next State=-5, Reward=0
Step 51: State=-5, Action=1, Next State=-4, Reward=0
Step 52: State=-4, Action=1, Next State=-3, Reward=0
Step 53: State=-3, Action=1, Next State=-2, Reward=0
Step 54: State=-2, Action=1, Next State=-1, Reward=0
Step 55: State=-1, Action=-1, Next State=-2, Reward=0
Step 56: State=-2, Action=-1, Next State=-3, Reward=0
Step 57: State=-3, Action=-1, Next State=-4, Reward=0
```



```python
class Environment:
    def __init__(self):
        self.state = 0
        self.treasure_position = 10
        self.trap_position = 5

    def reset(self):
        self.state = 0
        return self.state

    def step(self, action):
        self.state += action
        reward = 0
        done = False

        if self.state == self.treasure_position:
            reward = 20
            done = True
        elif self.state == self.trap_position:
            reward = -5

        return self.state, reward, done

class Agent:
    def __init__(self):
        self.actions = [-1, 1]

    def select_action(self):
        return random.choice(self.actions)

env = Environment()
agent = Agent()
state = env.reset()
done = False
step_count = 0

print("Starting episode...")

while not done:
    action = agent.select_action()
    next_state, reward, done = env.step(action)
    print(f"Step {step_count}: State={state}, Action={action}, Next State={next_state}, Reward={reward}")
    state = next_state
    step_count += 1

print("Episode finished!")
print(f"Final state: {state}")
print(f"Total steps: {step_count}")
```



```
Starting episode...
Step 0: State=0, Action=1, Next State=1, Reward=0
Step 1: State=1, Action=1, Next State=2, Reward=0
Step 2: State=2, Action=1, Next State=3, Reward=0
Step 3: State=3, Action=1, Next State=4, Reward=0
Step 4: State=4, Action=1, Next State=5, Reward=-5
Step 5: State=5, Action=1, Next State=6, Reward=0
Step 6: State=6, Action=1, Next State=7, Reward=0
Step 7: State=7, Action=1, Next State=8, Reward=0
Step 8: State=8, Action=1, Next State=9, Reward=0
Step 9: State=9, Action=1, Next State=10, Reward=20
Episode finished!
Final state: 10
Total steps: 10
```