

```
In [1]: import random
from scipy.stats import norm
import matplotlib.pyplot as plt
from scipy.stats import beta
import numpy as np
```

```
In [2]: def tit_for_tat(num_trials):

    player_a_moves = ["C"]
    player_b_moves = ["C"]

    for _ in range(num_trials - 1):

        if player_b_moves[-1] == "C":
            player_a_moves.append("C")
        else:
            player_a_moves.append("D")

        opponent_response = random.choice(["C", "D"])
        player_b_moves.append(opponent_response)

    return player_a_moves, player_b_moves

def tit_for_tat_with_threshold(num_rounds, threshold):
    actions_a = []
    actions_b = []

    for round_num in range(1, num_rounds+1):
        if round_num <= threshold:
            actions_a.append("C")
            actions_b.append(random.choice(["C", "D"]))
        else:
            if actions_b[-1] == "C":
                actions_a.append("C")
            else:
                actions_a.append("D")
            opponent_response = random.choice(["C", "D"])
            actions_b.append(opponent_response)

    return actions_a, actions_b

def one_cooperating_strategy(num_trials):
    player_d_moves=[]
    player_c_moves=[]
    for _ in range(num_trials):

        player_d_moves.append(random.choice(["C", "D"]))
        player_c_moves.append("C")
    return player_d_moves, player_c_moves

def both_cooperating_strategy(num_trials):
    player_e_moves=[]
    player_f_moves=[]
    for _ in range(num_trials):
```

```

        player_e_moves.append("C")
        player_f_moves.append("C")
    return player_e_moves, player_f_moves

"""payoff_matrix = {
    "CC": (11, 11), # Payoff for mutual cooperation
    "CD": (0, 20), # Payoff for A: Cooperation, B: Defection
    "DC": (20, 0), # Payoff for A: Defection, B: Cooperation
    "DD": (9, 9)   # Payoff for mutual defection
}"""
payoff_matrix = {'CC': (11, 11), 'CD': (0, 20), 'DC': (20, 0), 'DD': (9, 9)

# Simulate trials
num_trials = 100
player_a, player_b = tit_for_tat(num_trials)
player_a1, player_b1 = tit_for_tat_with_threshold(num_trials, threshold=2)
player_a2, player_b2 = one_cooperating_strategy(num_trials)
player_a3, player_b3 = both_cooperating_strategy(num_trials)

mu3 = 4
mu4 = 0
sigma = 1
pdf_values_AB=[]
pdf_values_BA=[]
pdf_values1=[]
pdf_values2=[]
pdf_values3=[]

#print(f"Value of A: {player_a}, Value of y: {player_b}")
print("Strategy: Tit-for-Tat")
for round_num, (move_a, move_b) in enumerate(zip(player_a, player_b), start=1):
    print(f"Round {round_num}: Player A moves {move_a}, Player B moves {move_b}")

print("\nStrategy: Tit-for-Tat with Threshold")
for round_num, (move_a, move_b) in enumerate(zip(player_a1, player_b1), start=1):
    print(f"Round {round_num}: Player A moves {move_a}, Player B moves {move_b}")

print("\nStrategy: One Cooperating Strategy")
for round_num, (move_a, move_b) in enumerate(zip(player_a2, player_b2), start=1):
    print(f"Round {round_num}: Player A moves {move_a}, Player B moves {move_b}")

print("\nStrategy: Both Cooperating Strategy")
for round_num, (move_a, move_b) in enumerate(zip(player_a3, player_b3), start=1):
    print(f"Round {round_num}: Player A moves {move_a}, Player B moves {move_b}")

```

Strategy: Tit-for-Tat

Round 1: Player A moves C, Player B moves C
Round 2: Player A moves C, Player B moves D
Round 3: Player A moves D, Player B moves C
Round 4: Player A moves C, Player B moves C
Round 5: Player A moves C, Player B moves D
Round 6: Player A moves D, Player B moves C
Round 7: Player A moves C, Player B moves D
Round 8: Player A moves D, Player B moves C
Round 9: Player A moves C, Player B moves D
Round 10: Player A moves D, Player B moves D
Round 11: Player A moves D, Player B moves D
Round 12: Player A moves D, Player B moves D
Round 13: Player A moves D, Player B moves C
Round 14: Player A moves C, Player B moves D
Round 15: Player A moves D, Player B moves D
Round 16: Player A moves D, Player B moves C
Round 17: Player A moves C, Player B moves D
Round 18: Player A moves D, Player B moves C
Round 19: Player A moves C, Player B moves C
Round 20: Player A moves C, Player B moves D
Round 21: Player A moves D, Player B moves D
Round 22: Player A moves D, Player B moves C
Round 23: Player A moves C, Player B moves C
Round 24: Player A moves C, Player B moves C
Round 25: Player A moves C, Player B moves D
Round 26: Player A moves D, Player B moves D
Round 27: Player A moves D, Player B moves C
Round 28: Player A moves C, Player B moves C
Round 29: Player A moves C, Player B moves D
Round 30: Player A moves D, Player B moves D
Round 31: Player A moves D, Player B moves C
Round 32: Player A moves C, Player B moves D
Round 33: Player A moves D, Player B moves C
Round 34: Player A moves C, Player B moves D
Round 35: Player A moves D, Player B moves C
Round 36: Player A moves C, Player B moves D
Round 37: Player A moves D, Player B moves D
Round 38: Player A moves D, Player B moves D
Round 39: Player A moves D, Player B moves C
Round 40: Player A moves C, Player B moves D
Round 41: Player A moves D, Player B moves D
Round 42: Player A moves D, Player B moves D
Round 43: Player A moves D, Player B moves D
Round 44: Player A moves D, Player B moves C
Round 45: Player A moves C, Player B moves C
Round 46: Player A moves C, Player B moves C
Round 47: Player A moves C, Player B moves D
Round 48: Player A moves D, Player B moves C
Round 49: Player A moves C, Player B moves C
Round 50: Player A moves C, Player B moves D
Round 51: Player A moves D, Player B moves D
Round 52: Player A moves D, Player B moves D
Round 53: Player A moves D, Player B moves C
Round 54: Player A moves C, Player B moves D
Round 55: Player A moves D, Player B moves D
Round 56: Player A moves D, Player B moves C
Round 57: Player A moves C, Player B moves C
Round 58: Player A moves C, Player B moves C
Round 59: Player A moves C, Player B moves D

Round 60: Player A moves D, Player B moves C
Round 61: Player A moves C, Player B moves D
Round 62: Player A moves D, Player B moves C
Round 63: Player A moves C, Player B moves C
Round 64: Player A moves C, Player B moves C
Round 65: Player A moves C, Player B moves C
Round 66: Player A moves C, Player B moves D
Round 67: Player A moves D, Player B moves C
Round 68: Player A moves C, Player B moves C
Round 69: Player A moves C, Player B moves C
Round 70: Player A moves C, Player B moves C
Round 71: Player A moves C, Player B moves C
Round 72: Player A moves C, Player B moves D
Round 73: Player A moves D, Player B moves C
Round 74: Player A moves C, Player B moves C
Round 75: Player A moves C, Player B moves D
Round 76: Player A moves D, Player B moves D
Round 77: Player A moves D, Player B moves C
Round 78: Player A moves C, Player B moves D
Round 79: Player A moves D, Player B moves C
Round 80: Player A moves C, Player B moves C
Round 81: Player A moves C, Player B moves D
Round 82: Player A moves D, Player B moves C
Round 83: Player A moves C, Player B moves C
Round 84: Player A moves C, Player B moves C
Round 85: Player A moves C, Player B moves D
Round 86: Player A moves D, Player B moves D
Round 87: Player A moves D, Player B moves C
Round 88: Player A moves C, Player B moves C
Round 89: Player A moves C, Player B moves D
Round 90: Player A moves D, Player B moves D
Round 91: Player A moves D, Player B moves D
Round 92: Player A moves D, Player B moves D
Round 93: Player A moves D, Player B moves D
Round 94: Player A moves D, Player B moves C
Round 95: Player A moves C, Player B moves D
Round 96: Player A moves D, Player B moves C
Round 97: Player A moves C, Player B moves D
Round 98: Player A moves D, Player B moves C
Round 99: Player A moves C, Player B moves D
Round 100: Player A moves D, Player B moves D

Strategy: Tit-for-Tat with Threshold

Round 1: Player A moves C, Player B moves D
Round 2: Player A moves C, Player B moves C
Round 3: Player A moves C, Player B moves C
Round 4: Player A moves C, Player B moves C
Round 5: Player A moves C, Player B moves C
Round 6: Player A moves C, Player B moves D
Round 7: Player A moves C, Player B moves D
Round 8: Player A moves C, Player B moves C
Round 9: Player A moves C, Player B moves C
Round 10: Player A moves C, Player B moves C
Round 11: Player A moves C, Player B moves D
Round 12: Player A moves C, Player B moves C
Round 13: Player A moves C, Player B moves D
Round 14: Player A moves C, Player B moves D
Round 15: Player A moves C, Player B moves C
Round 16: Player A moves C, Player B moves C
Round 17: Player A moves C, Player B moves C

[illegible]

Round 78: Player A moves D, Player B moves C
Round 79: Player A moves D, Player B moves D
Round 80: Player A moves C, Player B moves C
Round 81: Player A moves C, Player B moves C
Round 82: Player A moves C, Player B moves D
Round 83: Player A moves C, Player B moves C
Round 84: Player A moves C, Player B moves D
Round 85: Player A moves D, Player B moves C
Round 86: Player A moves D, Player B moves D
Round 87: Player A moves D, Player B moves C
Round 88: Player A moves C, Player B moves C
Round 89: Player A moves C, Player B moves D
Round 90: Player A moves C, Player B moves D
Round 91: Player A moves C, Player B moves C
Round 92: Player A moves C, Player B moves D
Round 93: Player A moves C, Player B moves C
Round 94: Player A moves D, Player B moves D
Round 95: Player A moves D, Player B moves C
Round 96: Player A moves D, Player B moves C
Round 97: Player A moves C, Player B moves D
Round 98: Player A moves D, Player B moves D
Round 99: Player A moves D, Player B moves C
Round 100: Player A moves C, Player B moves C

Strategy: One Cooperating Strategy

Round 1: Player A moves C, Player B moves C
Round 2: Player A moves C, Player B moves C
Round 3: Player A moves C, Player B moves C
Round 4: Player A moves D, Player B moves C
Round 5: Player A moves C, Player B moves C
Round 6: Player A moves C, Player B moves C
Round 7: Player A moves C, Player B moves C
Round 8: Player A moves D, Player B moves C
Round 9: Player A moves C, Player B moves C
Round 10: Player A moves D, Player B moves C
Round 11: Player A moves D, Player B moves C
Round 12: Player A moves C, Player B moves C
Round 13: Player A moves C, Player B moves C
Round 14: Player A moves C, Player B moves C
Round 15: Player A moves D, Player B moves C
Round 16: Player A moves D, Player B moves C
Round 17: Player A moves D, Player B moves C
Round 18: Player A moves C, Player B moves C
Round 19: Player A moves C, Player B moves C
Round 20: Player A moves D, Player B moves C
Round 21: Player A moves C, Player B moves C
Round 22: Player A moves C, Player B moves C
Round 23: Player A moves D, Player B moves C
Round 24: Player A moves C, Player B moves C
Round 25: Player A moves D, Player B moves C
Round 26: Player A moves D, Player B moves C
Round 27: Player A moves C, Player B moves C
Round 28: Player A moves D, Player B moves C
Round 29: Player A moves D, Player B moves C
Round 30: Player A moves C, Player B moves C
Round 31: Player A moves C, Player B moves C
Round 32: Player A moves D, Player B moves C
Round 33: Player A moves D, Player B moves C
Round 34: Player A moves C, Player B moves C
Round 35: Player A moves C, Player B moves C

[illegible]

Round 54: Player A moves C, Player B moves C
 Round 55: Player A moves C, Player B moves C
 Round 56: Player A moves C, Player B moves C
 Round 57: Player A moves C, Player B moves C
 Round 58: Player A moves C, Player B moves C
 Round 59: Player A moves C, Player B moves C
 Round 60: Player A moves C, Player B moves C
 Round 61: Player A moves C, Player B moves C
 Round 62: Player A moves C, Player B moves C
 Round 63: Player A moves C, Player B moves C
 Round 64: Player A moves C, Player B moves C
 Round 65: Player A moves C, Player B moves C
 Round 66: Player A moves C, Player B moves C
 Round 67: Player A moves C, Player B moves C
 Round 68: Player A moves C, Player B moves C
 Round 69: Player A moves C, Player B moves C
 Round 70: Player A moves C, Player B moves C
 Round 71: Player A moves C, Player B moves C
 Round 72: Player A moves C, Player B moves C
 Round 73: Player A moves C, Player B moves C
 Round 74: Player A moves C, Player B moves C
 Round 75: Player A moves C, Player B moves C
 Round 76: Player A moves C, Player B moves C
 Round 77: Player A moves C, Player B moves C
 Round 78: Player A moves C, Player B moves C
 Round 79: Player A moves C, Player B moves C
 Round 80: Player A moves C, Player B moves C
 Round 81: Player A moves C, Player B moves C
 Round 82: Player A moves C, Player B moves C
 Round 83: Player A moves C, Player B moves C
 Round 84: Player A moves C, Player B moves C
 Round 85: Player A moves C, Player B moves C
 Round 86: Player A moves C, Player B moves C
 Round 87: Player A moves C, Player B moves C
 Round 88: Player A moves C, Player B moves C
 Round 89: Player A moves C, Player B moves C
 Round 90: Player A moves C, Player B moves C
 Round 91: Player A moves C, Player B moves C
 Round 92: Player A moves C, Player B moves C
 Round 93: Player A moves C, Player B moves C
 Round 94: Player A moves C, Player B moves C
 Round 95: Player A moves C, Player B moves C
 Round 96: Player A moves C, Player B moves C
 Round 97: Player A moves C, Player B moves C
 Round 98: Player A moves C, Player B moves C
 Round 99: Player A moves C, Player B moves C
 Round 100: Player A moves C, Player B moves C

```
In [3]: import math
import numpy as np
import matplotlib.pyplot as plt

def memory_strength(initial_strength, decay_constant, time):
    """
    Calculate memory strength at a given time using an exponential decay

    Parameters:
        initial_strength (float): Initial strength of memory.
        decay_constant (float): Decay constant determining the rate of de
        time (float): Time elapsed since memory formation.
```

```

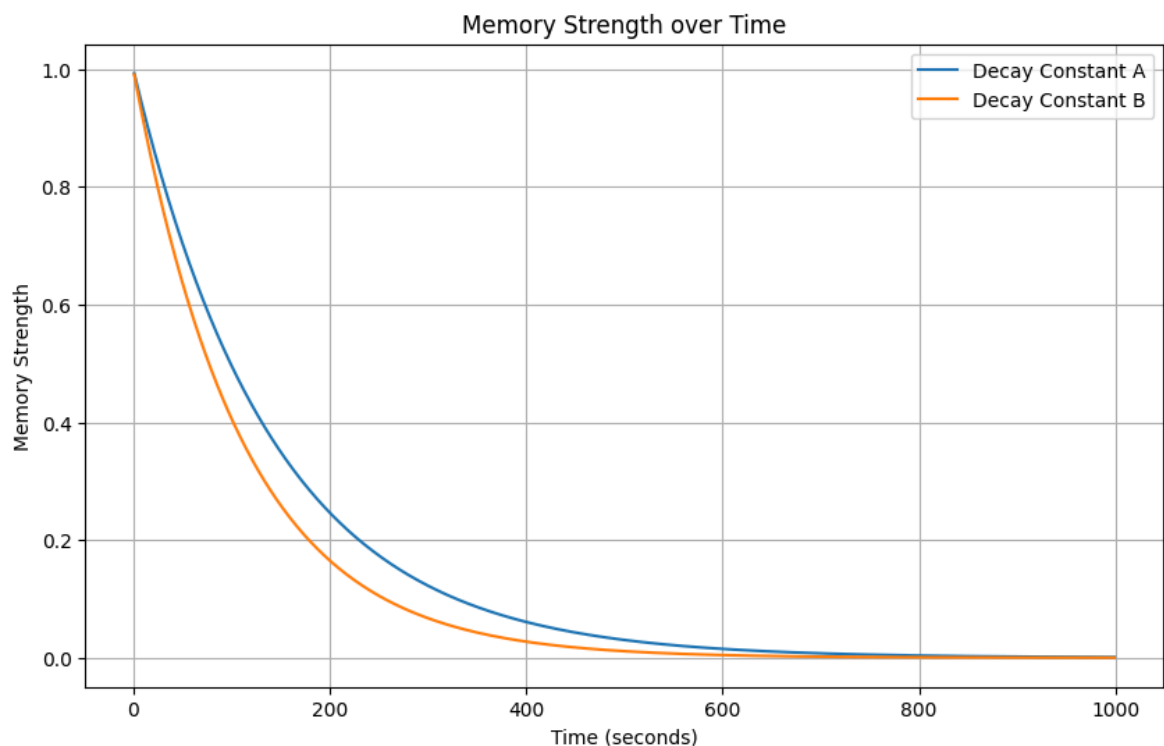
Returns:
    float: Memory strength at the given time.
"""
    return initial_strength * math.exp(-decay_constant * time)

# Define parameters
initial_strength = 1.0 # Initial strength of memory
decay_constant_A = 0.007 # Decay constant
decay_constant_B = 0.009
# Time range from 0 to 100 seconds with 0.1 second intervals
time_range = np.arange(1, 1000, 1)

# Calculate memory strength at each time point
memory_strength_values_A = [memory_strength(initial_strength, decay_constant_A, time) for time in time_range]
# print(memory_strength_values)
# Calculate memory strength at each time point
memory_strength_values_B = [memory_strength(initial_strength, decay_constant_B, time) for time in time_range]

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(time_range, memory_strength_values_A, label='Decay Constant A')
plt.plot(time_range, memory_strength_values_B, label='Decay Constant B')
plt.title('Memory Strength over Time')
plt.xlabel('Time (seconds)')
plt.ylabel('Memory Strength')
plt.legend()
plt.grid(True)
plt.show()

```



```

In [14]: # For the tit-for-tat
payoffs_a = []
w_payoffs_a = []
payoffs_b = []
w_payoffs_b = []
rep_values_AB = []
rep_values_BA = []

```

```

sum_rep_AB=0
sum_rep_BA=0
trust_values_tit_tat_AB = []
trust_values_tit_tat_BA = []

for i in range(len(player_a)):
    action_a = player_a[i]
    action_b = player_b[i]
    key = action_a + action_b
    payoffs_a.append(payload_matrix[key][0])
    payoffs_b.append(payload_matrix[key][1])

for j in range(len(payoffs_a)):
    w_payoffs_a.append(memory_strength_values_A[j] * payoffs_a[len(payload
    w_payoffs_b.append(memory_strength_values_B[j] * payoffs_b[len(payload

print(len(w_payoffs_a))

for a, b in zip(w_payoffs_a, w_payoffs_b):
    if (a + b) != 0:
        rep_values_AB.append(a / (a + b))
        rep_values_BA.append(b / (a + b))
    else:
        rep_values_AB.append(0.4) # Set trust value to a default value
        rep_values_BA.append(0.4)

#print(rep_values_AB)

w_payoffs_a.clear()
w_payoffs_b.clear()

print(len(rep_values_AB))
sum_rep_AB=0
sum_rep_BA=0

for i, k in enumerate(rep_values_AB):
    sum_rep_AB += k
    trust_values_tit_tat_AB.append(sum_rep_AB / (i + 1))

for l, m in enumerate(rep_values_BA):
    sum_rep_BA += m
    trust_values_tit_tat_BA.append(sum_rep_BA / (l + 1))

sum_rep_AB=0
sum_rep_BA=0

#print(trust_values_tit_tat_BA)
#print(trust_values_tit_tat_AB)

# Plot Rep values and Trust values for tit-for-tat FOR AB
plt.plot(range(len(rep_values_AB)), rep_values_AB, label='Reputation_B')
plt.plot(range(len(trust_values_tit_tat_AB)), trust_values_tit_tat_AB, la
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Tit-for-Tat with Threshold)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits

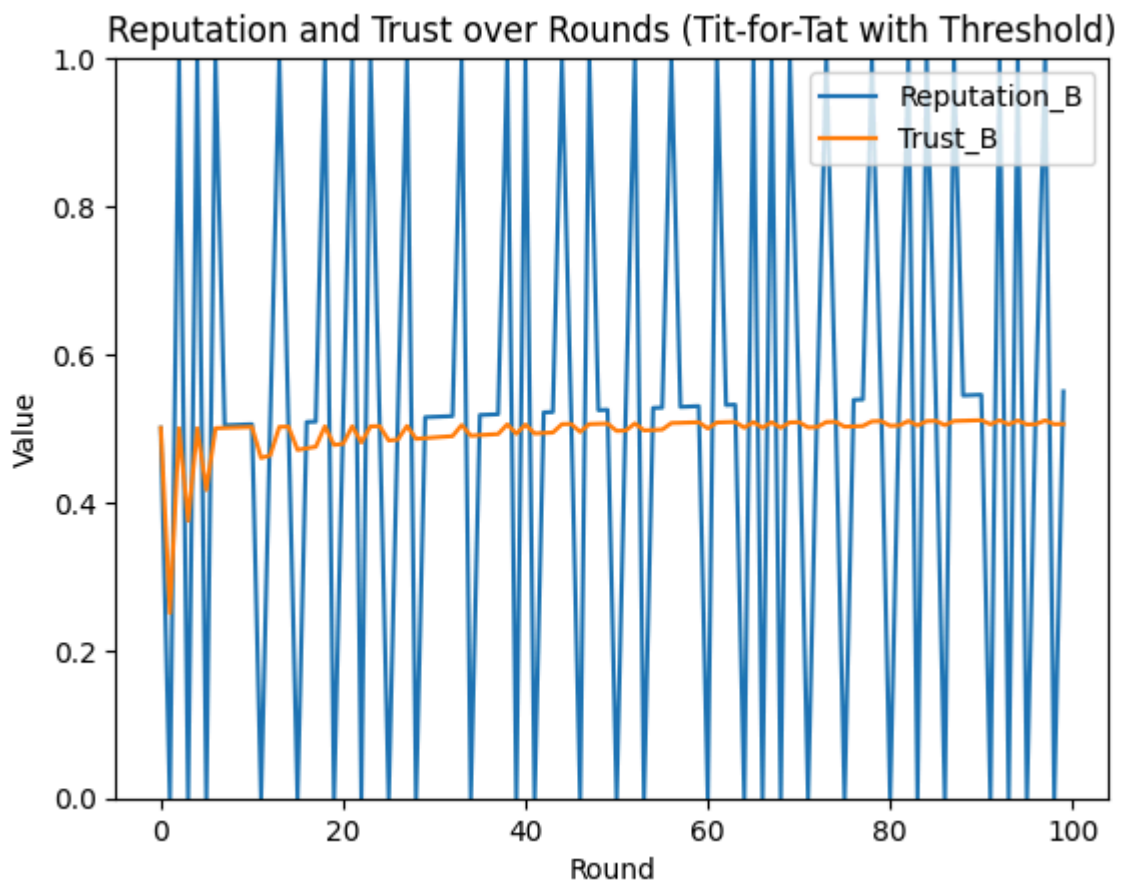
```

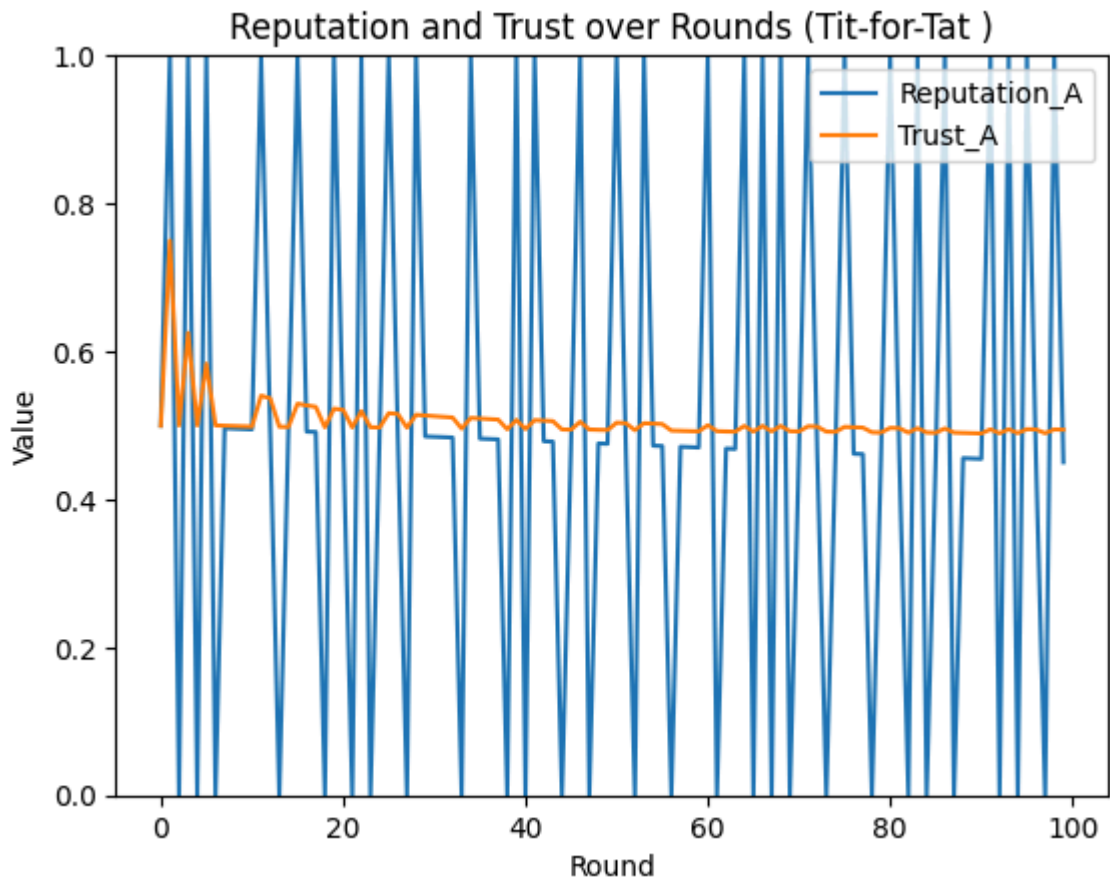
```
plt.show()

# Plot Rep values and Trust values for tit-for-tat FOR BA
plt.plot(range(len(rep_values_BA)), rep_values_BA, label='Reputation_A')
plt.plot(range(len(trust_values_tit_tat_BA)), trust_values_tit_tat_BA, la
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Tit-for-Tat )')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()
```

100

100





```
In [16]: # For the tit-for-tat with threshold strategy
payoffs_a1 = []
w_payoffs_a1 = []
payoffs_b1 = []
w_payoffs_b1 = []
sum_rep_AB1=0
sum_rep_BA1=0
#A has for B=AB
trust_values_tit_tat_AB1 = []
trust_values_tit_tat_BA1= []
rep_values_AB1=[]
rep_values_BA1=[]
for i in range(len(player_a1)):
    action_a1 = player_a1[i]
    action_b1 = player_b1[i]
    key = action_a1 + action_b1
    payoffs_a1.append(payload_matrix[key][0])
    payoffs_b1.append(payload_matrix[key][1])

for j in range(len(payoffs_a1)):
    w_payoffs_a1.append(memory_strength_values_A[j] * payoffs_a1[len(payoffs_a1)-j])
    w_payoffs_b1.append(memory_strength_values_B[j] * payoffs_b1[len(payoffs_b1)-j])

print(len(w_payoffs_a1))

for a, b in zip(w_payoffs_a1, w_payoffs_b1):
    if (a + b) != 0:
        rep_values_AB1.append(a / (a + b))
        rep_values_BA1.append(b / (a + b))
    else:
        rep_values_AB1.append(0.4) # Set trust value to a default value
        rep_values_BA1.append(0.4)
```

```

#print(rep_values_AB1)

w_payoffs_a1.clear()
w_payoffs_b1.clear()

print(len(rep_values_AB1))
sum_rep_AB1=0
sum_rep_BA1=0

for i, k in enumerate(rep_values_AB1):
    sum_rep_AB1 += k
    trust_values_tit_tat_AB1.append(sum_rep_AB1 / (i + 1))

for l, m in enumerate(rep_values_BA1):
    sum_rep_BA1 += m
    trust_values_tit_tat_BA1.append(sum_rep_BA1 / (l + 1))

sum_rep_AB1=0
sum_rep_BA1=0

#print(trust_values_tit_tat_BA1)
#print(trust_values_tit_tat_AB1)

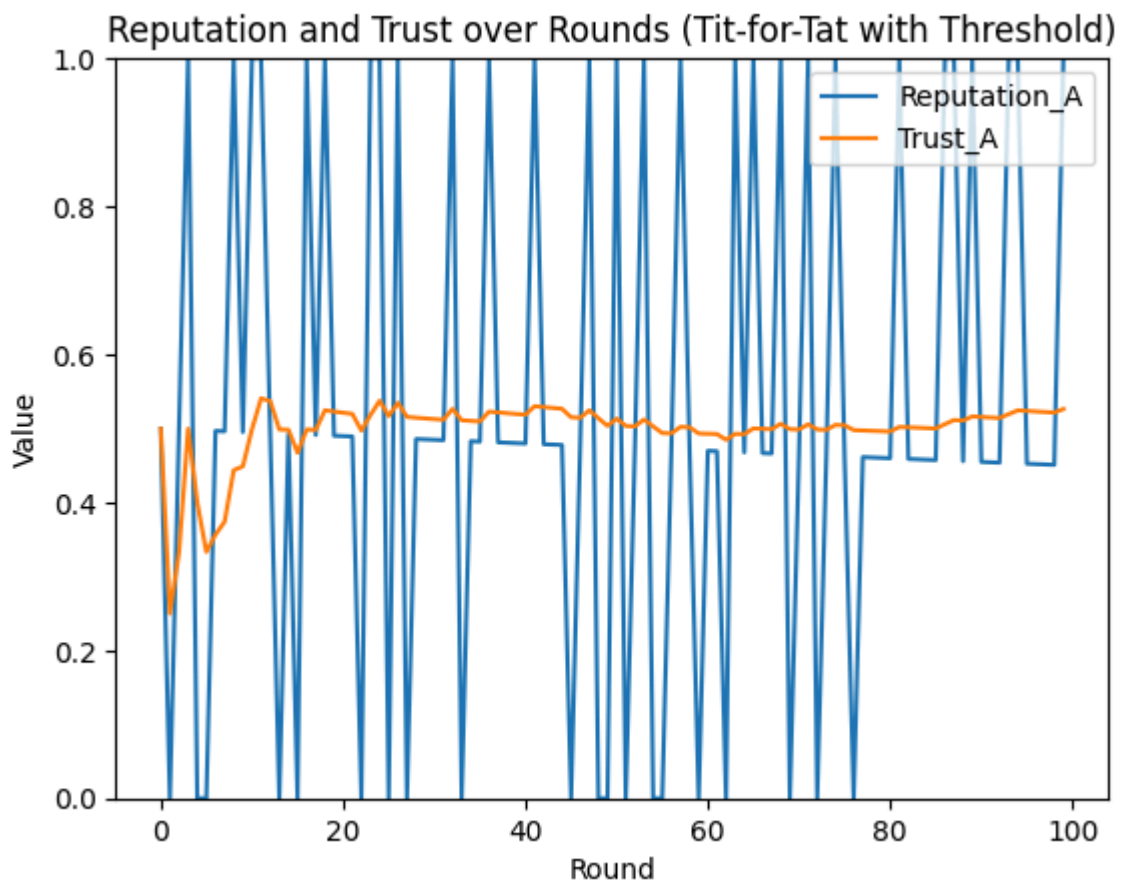
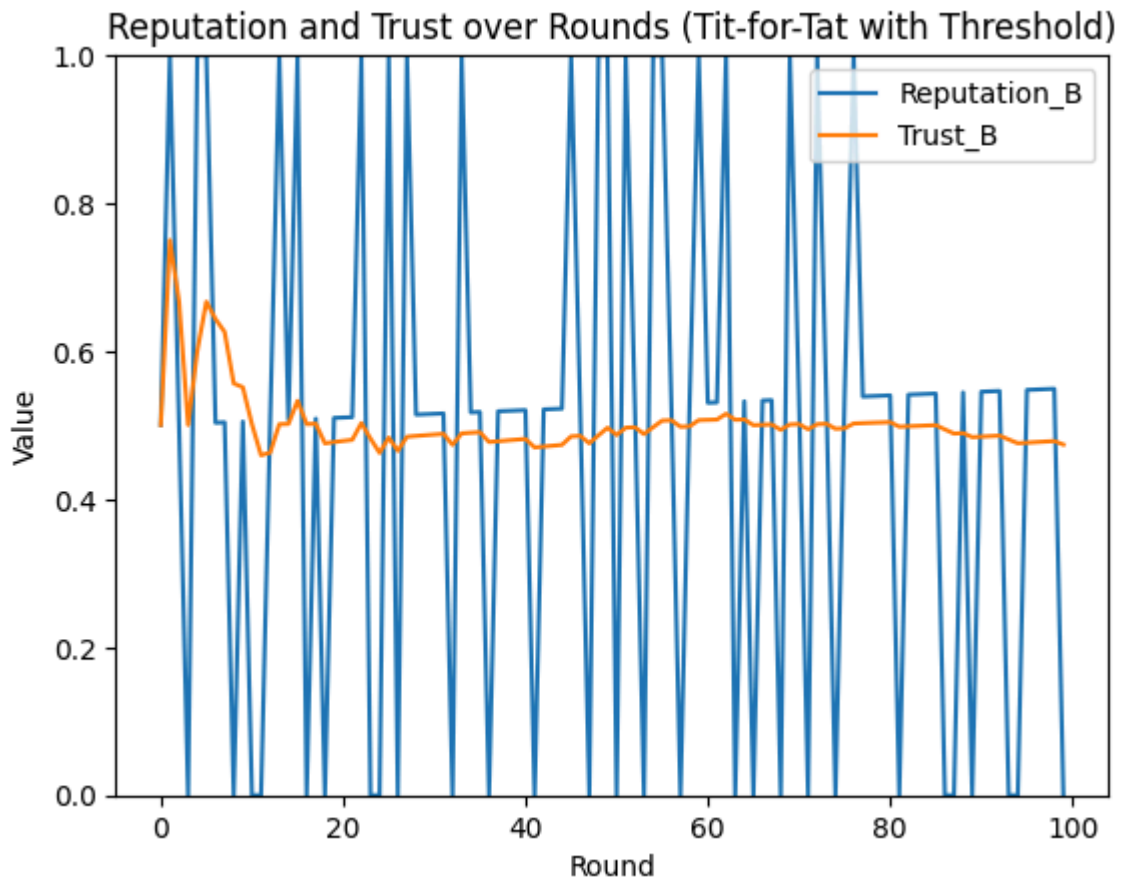
# Plot Rep values and Trust values for tit-for-tat FOR AB
plt.plot(range(len(rep_values_AB1)), rep_values_AB1, label='Reputation_B')
plt.plot(range(len(trust_values_tit_tat_AB1)), trust_values_tit_tat_AB1,
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Tit-for-Tat with Threshold)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

# Plot Rep values and Trust values for tit-for-tat FOR BA
plt.plot(range(len(rep_values_BA1)), rep_values_BA1, label='Reputation_A')
plt.plot(range(len(trust_values_tit_tat_BA1)), trust_values_tit_tat_BA1,
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Tit-for-Tat with Threshold)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

```

100

100



```
In [17]: # For one ever co-operating
payoffs_a2 = []
w_payoffs_a2 = []
payoffs_b2 = []
w_payoffs_b2 = []
```

```

sum_rep_AB2=0
sum_rep_BA2=0
#A has for B=AB
trust_values_tit_tat_AB2 = []
trust_values_tit_tat_BA2= []
rep_values_AB2=[]
rep_values_BA2=[]

for i in range(len(player_a2)):
    action_a2 = player_a2[i]
    action_b2 = player_b2[i]
    key = action_a2 + action_b2
    payoffs_a2.append(payoff_matrix[key][0])
    payoffs_b2.append(payoff_matrix[key][1])

for j in range(len(payoffs_a2)):
    w_payoffs_a2.append(memory_strength_values_A[j] * payoffs_a2[len(payoffs_a2)-j])
    w_payoffs_b2.append(memory_strength_values_B[j] * payoffs_b2[len(payoffs_b2)-j])

print(len(w_payoffs_a2))

for a, b in zip(w_payoffs_a2, w_payoffs_b2):
    if (a + b) != 0:
        rep_values_AB2.append(a / (a + b))
        rep_values_BA2.append(b / (a + b))
    else:
        rep_values_AB2.append(0.4) # Set trust value to a default value
        rep_values_BA2.append(0.4)

#print(rep_values_AB2)

w_payoffs_a2.clear()
w_payoffs_b2.clear()

print(len(rep_values_AB2))
sum_rep_AB2=0
sum_rep_BA2=0

for i, k in enumerate(rep_values_AB2):
    sum_rep_AB2 += k
    trust_values_tit_tat_AB2.append(sum_rep_AB2 / (i + 1))

for l, m in enumerate(rep_values_BA2):
    sum_rep_BA2 += m
    trust_values_tit_tat_BA2.append(sum_rep_BA2 / (l + 1))

sum_rep_AB2=0
sum_rep_BA2=0

#print(trust_values_tit_tat_BA2)
#print(trust_values_tit_tat_AB2)

# Plot Rep values and Trust values for tit-for-tat FOR AB
plt.plot(range(len(rep_values_AB2)), rep_values_AB2, label='Reputation_B')
plt.plot(range(len(trust_values_tit_tat_AB2)), trust_values_tit_tat_AB2,

```



```

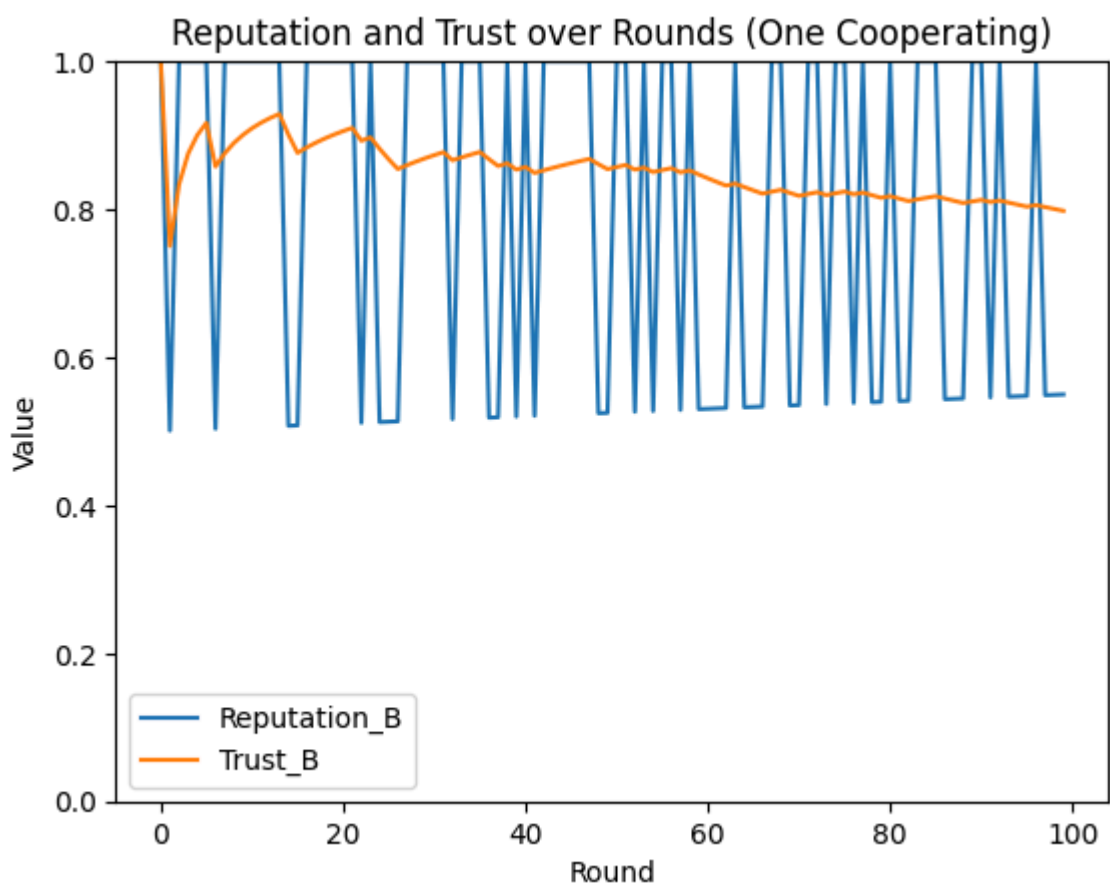
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (One Cooperating)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

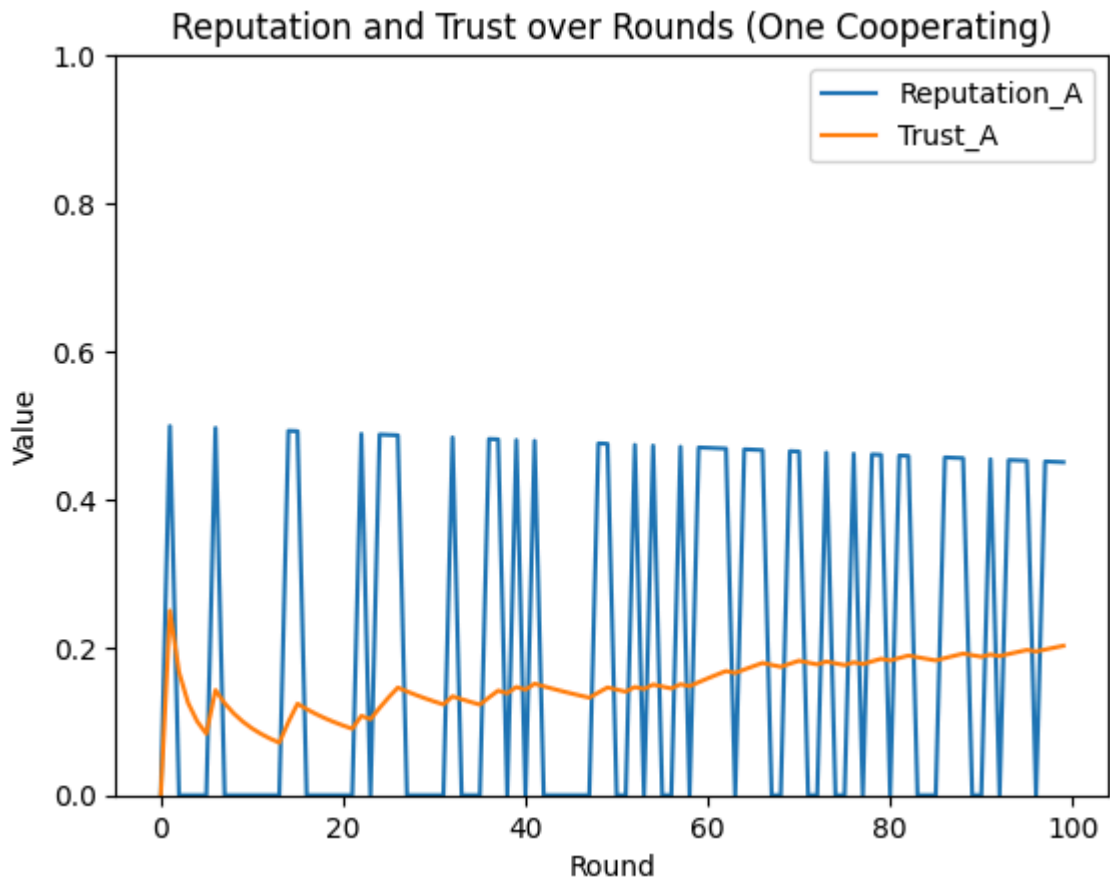
# Plot Rep values and Trust values for tit-for-tat FOR BA
plt.plot(range(len(rep_values_BA2)), rep_values_BA2, label='Reputation_A')
plt.plot(range(len(trust_values_tit_tat_BA2)), trust_values_tit_tat_BA2,
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (One Cooperating)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

```

100

100





```
In [18]: # For both ever co-operating
payoffs_a3= []
w_payoffs_a3 = []
payoffs_b3 = []
w_payoffs_b3 = []
sum_rep_AB3=0
sum_rep_BA3=0
#A has for B=AB
trust_values_tit_tat_AB3 = []
trust_values_tit_tat_BA3= []
rep_values_AB3=[]
rep_values_BA3=[]

for i in range(len(player_a3)):
    action_a3 = player_a3[i]
    action_b3 = player_b3[i]
    key = action_a3 + action_b3
    payoffs_a3.append(payload_matrix[key][0])
    payoffs_b3.append(payload_matrix[key][1])

for j in range(len(payoffs_a3)):
    w_payoffs_a3.append(memory_strength_values_A[j] * payoffs_a3[j])
    w_payoffs_b3.append(memory_strength_values_B[j] * payoffs_b3[j])

print(len(w_payoffs_a3))

for a, b in zip(w_payoffs_a3, w_payoffs_b3):
    if (a + b) != 0:
        rep_values_AB3.append(a / (a + b))
        rep_values_BA3.append(b / (a + b))
    else:
        rep_values_AB3.append(0.4) # Set trust value to a default value
```

```

        rep_values_BA3.append(0.4)

#print(rep_values_AB3)

w_payoffs_a3.clear()
w_payoffs_b3.clear()

print(len(rep_values_AB3))
sum_rep_AB3=0
sum_rep_BA3=0

for i, k in enumerate(rep_values_AB3):
    sum_rep_AB3 += k
    trust_values_tit_tat_AB3.append(sum_rep_AB3 / (i + 1))

for l, m in enumerate(rep_values_BA3):
    sum_rep_BA3 += m
    trust_values_tit_tat_BA3.append(sum_rep_BA3 / (l + 1))

sum_rep_AB3=0
sum_rep_BA3=0

#print(trust_values_tit_tat_BA3)
#print(trust_values_tit_tat_AB3)

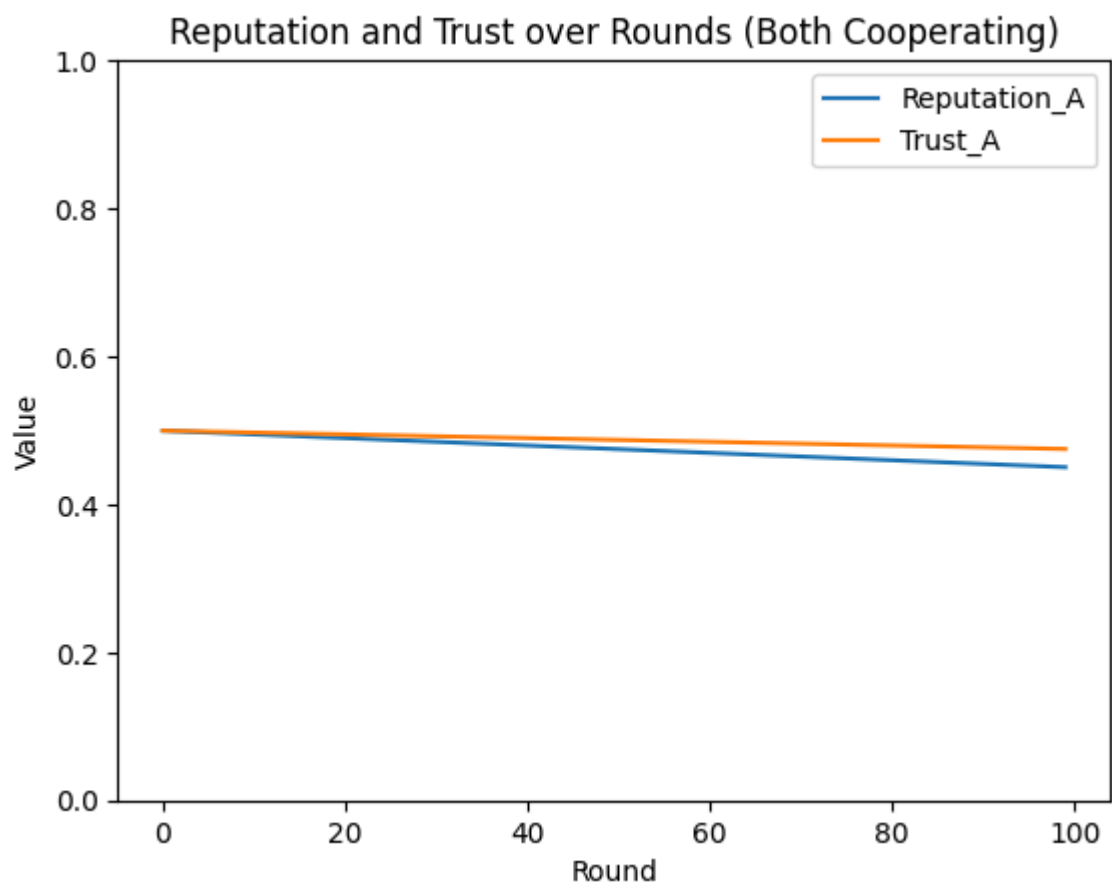
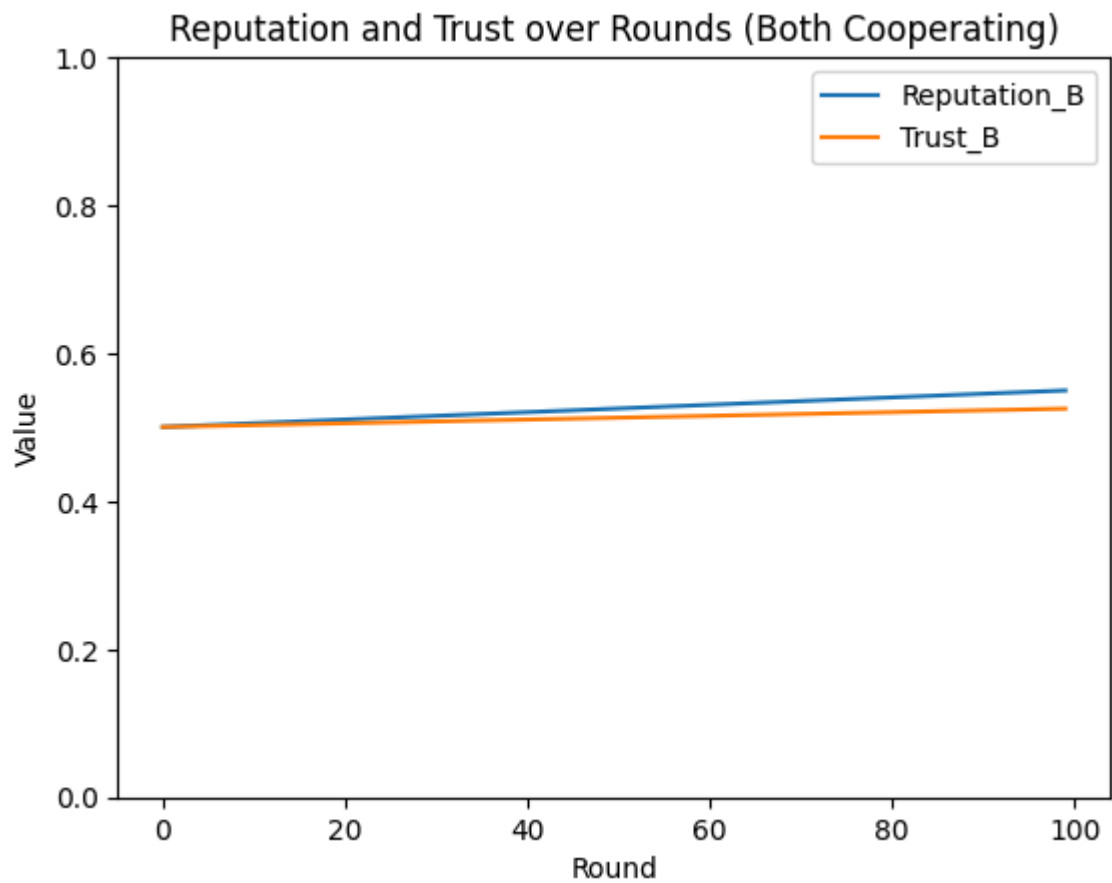
# Plot Rep values and Trust values for tit-for-tat FOR AB
plt.plot(range(len(rep_values_AB3)), rep_values_AB3, label='Reputation_B')
plt.plot(range(len(trust_values_tit_tat_AB3)), trust_values_tit_tat_AB3,
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Both Cooperating)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

# Plot Rep values and Trust values for tit-for-tat FOR BA
plt.plot(range(len(rep_values_BA3)), rep_values_BA3, label='Reputation_A')
plt.plot(range(len(trust_values_tit_tat_BA3)), trust_values_tit_tat_BA3,
plt.xlabel('Round')
plt.ylabel('Value')
plt.title('Reputation and Trust over Rounds (Both Cooperating)')
plt.legend()
plt.ylim(0, 1) # Set y-axis limits
plt.show()

```

100

100



In []: