Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department

Subject/Odd Sem 2023-23/Experiment 1

Program :

1.

```
def prisoners_dilemma(player_a_choice, player_b_choice):
    if player_a_choice == 'Cooperate' and player_b_choice == 'Cooperate':
        return "Both players cooperate. Each gets 3 points."
    elif player_a_choice == 'Cooperate' and player_b_choice == 'Betray':
        return "Player A cooperates, but Player B betrays. Player A gets 0
points, Player B gets 5 points."
    elif player_a_choice == 'Betray' and player_b_choice == 'Cooperate':
        return "Player A betrays, but Player B cooperates. Player A gets 5
points, Player B gets 0 points."
    elif player_a_choice == 'Betray' and player_b_choice == 'Betray':
        return "Both players betray. Each gets 1 point."

player_a_choice = input("Player A, choose 'Cooperate' or 'Betray': ")
player_b_choice = input("Player B, choose 'Cooperate' or 'Betray': ")

result = prisoners_dilemma(player_a_choice, player_b_choice)
print(result)
```

Output Screenshots :

```
PS C:\Users\Admin1\Desktop\Game Theory Lab> & "C:/Program Files/Python310/python.exe" "c:/Users/Admin1/Desktop/Game Theory Lab/dilemma.py"
Player A, choose 'Cooperate' or 'Betray': Cooperate
Player B, choose 'Cooperate' or 'Betray': Betray
Player A cooperates, but Player B betrays. Player A gets 0 points, Player B gets 5 points.
PS C:\Users\Admin1\Desktop\Game Theory Lab> & "C:/Program Files/Python310/python.exe" "c:/Users/Admin1/Desktop/Game Theory Lab/dilemma.py"
Player A, choose 'Cooperate' or 'Betray': Betray
Player B, choose 'Cooperate' or 'Betray': Cooperate
Player A betrays, but Player B cooperates. Player A gets 5 points, Player B gets 0 points.
PS C:\Users\Admin1\Desktop\Game Theory Lab> & "C:/Program Files/Python310/python.exe" "c:/Users/Admin1/Desktop/Game Theory Lab/dilemma.py"
Player A, choose 'Cooperate' or 'Betray': Cooperate
Player B, choose 'Cooperate' or 'Betray': Cooperate
Both players cooperate. Each gets 3 points.
PS C:\Users\Admin1\Desktop\Game Theory Lab> & "C:/Program Files/Python310/python.exe" "c:/Users/Admin1/Desktop/Game Theory Lab/dilemma.py"
Player A, choose 'Cooperate' or 'Betray': Betray
Player B, choose 'Cooperate' or 'Betray': Betray
Both players betray. Each gets 1 point.
PS C:\Users\Admin1\Desktop\Game Theory Lab>
```

Results and Discussions : The Prisoner's Dilemma illustrates the tension between individual self-interest and collective cooperation. Despite the rational choice for both players being to betray, the optimal outcome for both is achieved through cooperation. This implementation demonstrates how real-world scenarios can be modeled using game theory concepts and how outcomes can be affected by different choices, leading to insights into strategic decision-making and the interplay between competing interests.

Program :

1.

```python
up_left = []
up_right = []
down_left = []
down_right = []

print('Please Enter the Values for your Matrix')
up_left.append(float(input('Please Enter A Value for A: ')))
up_left.append(float(input('Please Enter A Value for B: ')))
up_right.append(float(input('Please Enter A Value for C: ')))
up_right.append(float(input('Please Enter A Value for D: ')))
down_left.append(float(input('Please Enter A Value for E: ')))
down_left.append(float(input('Please Enter A Value for F: ')))
down_right.append(float(input('Please Enter A Value for G: ')))
down_right.append(float(input('Please Enter A Value for H: ')))

if up_left[0] >= down_left[0] and up_left[1] >= up_right[1]:
    up_left_bool = 1
else:
    up_left_bool = 0
```

```
if down_left[0] >= up_left[0] and down_left[1] >= down_right[1]:
    down_left_bool = 1
else:
    down_left_bool = 0

if up_right[0] >= down_right[0] and up_right[1] >= up_left[1]:
    up_right_bool = 1
else:
    up_right_bool = 0

if down_right[0] >= up_right[0] and down_right[1] >= down_left[1]:
    down_right_bool = 1
else:
    down_right_bool = 0

bool_values = [up_left_bool, up_right_bool, down_left_bool, down_right_bool]

if up_left_bool == 1:
    print('Up, Left is a Nash Equilibrium')
else:
```

```
    pass


if up_right_bool == 1:

    print('Up, Right is a Nash Equilibrium')

else:

    pass


if down_left_bool == 1:

    print('Down, Left is a Nash Equilibrium')

else:

    pass


if down_right_bool == 1:

    print('Down, Right is a Nash Equilibrium')

else:

    pass


if 1 not in bool_values:

    print('There are no Nash Equilibria')

else:

    pass
```

Output Screenshots :

```
PS C:\Users\Admin1\Desktop\Game Theory Lab> & "C:/Program Files/Python310/python.exe" "c:/Users/Admin1/Desktop/Game Theory Lab/nash.py"
Please Enter the Values for your Matrix
Please Enter A Value for A: 3
Please Enter A Value for B: 3
Please Enter A Value for C: -1
Please Enter A Value for D: 5
Please Enter A Value for E: 5
Please Enter A Value for F: -1
Please Enter A Value for G: 0
Please Enter A Value for H: 0
Down, Right is a Nash Equilibrium
PS C:\Users\Admin1\Desktop\Game Theory Lab>
```

Results and Discussions : Nash Equilibrium in the context of the Prisoner's Dilemma demonstrates how rational decision-making can lead to stable outcomes even when individual choices conflict with optimal collective results.