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 2

Name : Prasad Jawale	**Class/Roll No. :** D16AD 20	**Grade :**

Title of Experiment : Nash Equilibrium of Prisoner's Dilemma

Objective of Experiment : The objective is to understand and explain the concept of Nash Equilibrium in the context of the Prisoner's Dilemma, a classic example in game theory. The aim is to highlight the decision-making dynamics that lead to a stable equilibrium point even when individual choices might not result in the best overall outcome**.**

Outcome of Experiment : The outcome of this explanation will be a clear understanding of how Nash Equilibrium applies to the Prisoner's Dilemma. Readers will grasp the concept of rational decision-making, wherein both players make choices that maximize their individual payoffs, resulting in an equilibrium where no player has an incentive to unilaterally change their decision.

Problem Statement : To find the Nash equilibrium of Prisoner's dilemma

Description / Theory :

	Cooperate	Defect
Cooperate	3, 3	−1, 5
Defect	5, −1	0, 0

Payoff Matrix for Prisoner's Dilemma

The Nash Equilibrium is a foundational concept in game theory where players, acting rationally, reach a point where no one can improve their outcome by unilaterally changing their decision. In situations like the Prisoner's Dilemma, where individual interests clash with cooperative gains, the Nash Equilibrium captures the stable outcome where players make choices that maximize their individual payoffs, leading to a balance between self-interest and collective well-being. This equilibrium sheds light on the complexities of strategic decision-making, offering insights into scenarios where rational choices can result in unexpected outcomes.

Algorithm/ Pseudo Code / Flowchart (whichever is applicable)

Input: Payoff matrix M

For each row r in M (Player A's strategies):
    Find the column c that maximizes the payoff in cell (r, c) for Player A
    Store c as Player A's best response to strategy r

For each column c in M (Player B's strategies):
    Find the row r that maximizes the payoff in cell (r, c) for Player B
    Store r as Player B's best response to strategy c

For each (r, c) cell in M:
    If r is Player A's best response to strategy c, and c is Player B's best response to strategy r:
        (r, c) is a Nash equilibrium

Program :

1.

```
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.