LAB # 12

Game Theory In Artificial Intelligence

Lab Tasks:-

- Finding optimal move in Tic-Tac Toe using Minimax Algorithm in Game Theory.

Code:-

```
import math

# Initialize the board
board = [' ' for _ in range(9)]

def print_board(board):
    for row in [board[i*3:(i+1)*3] for i in range(3)]:
        print('| ' + ' | '.join(row) + ' |')

def check_winner(b, p):
    # Winning combinations
    win_states = [[0, 1, 2], [3, 4, 5], [6, 7, 8], [0, 3, 6], [1, 4, 7], [2, 5, 8], [0, 4, 8], [2, 4, 6]]
    return any(all(b[i] == p for i in state) for state in win_states)

def is_board_full(b):
    return ' ' not in b

def minimax(board, depth, is_maximizing):
    # Base cases: evaluate terminal states
    if check_winner(board, 'X'): return 10 - depth
    if check_winner(board, 'O'): return depth - 10
    if is_board_full(board): return 0

    if is_maximizing:
        best_score = -math.inf
        for i in range(9):
            if board[i] == ' ':
                board[i] = 'X'
                score = minimax(board, depth + 1, False)
                board[i] = ' '
                best_score = max(score, best_score)
        return best_score
    else:
        best_score = math.inf
        for i in range(9):
            if board[i] == ' ':
                board[i] = 'O'
                score = minimax(board, depth + 1, True)
                board[i] = ' '
                best_score = min(score, best_score)
        return best_score

def find_best_move(board):
    best_score = -math.inf
    move = -1
    for i in range(9):
        if board[i] == ' ':
            board[i] = 'X'
            score = minimax(board, 0, False)
            board[i] = ' '
            if score > best_score:
                best_score = score
                move = i
    return move

# Main Game Loop
print("AI (X) vs You (O). AI moves first.")
while not is_board_full(board):
    # AI Turn
    ai_move = find_best_move(board)
    board[ai_move] = 'X'
    print_board(board)

    if check_winner(board, 'X'):
        print("\nAI Wins!")
        break
    if is_board_full(board):
        print("\nIt's a Draw!")
        break

    # Player Turn
    player_move = int(input("\nEnter your move (0-8): "))
    if board[player_move] != ' ':
        print("Invalid move! Try again.")
        continue
    board[player_move] = 'O'

    if check_winner(board, 'O'):
        print_board(board)
        print("\nYou Win!")
        break
```

Output:-

```
AI (X) vs You (O). AI moves first.
| X |   |   |
|   |   |   |
|   |   |   |

Enter your move (0-8): 4
| X | X |   |
|   | O |   |
|   |   |   |

Enter your move (0-8): 2
| X | X | O |
|   | O |   |
| X |   |   |

Enter your move (0-8): 3
| X | X | O |
| O | O | X |
| X |   |   |

Enter your move (0-8): 8
| X | X | O |
| O | O | X |
| X | X | O |

It's a Draw!
```