Name : S . SANJAY
Roll no: 241801246
Dept : AI & DS - FD
Date: 26 | 04 | 2025

MINIMAX ALGORITHM

program:

```
PLAYER_X = 1
PLAYER_O = -1
EMPTY = 0

def evaluate(board):
    for row in range(3):
        if board[row][0] == board[row][1] == board[row][2] != EMPTY:
            return board[row][0]
    for col in range(3):
        if board[0][col] == board[1][col] == board[2][col] != EMPTY:
            return board[0][col]
    if board[0][0] == board[1][1] == board[2][2] != EMPTY:
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] != EMPTY:
        return board[0][2]
    return 0

def is_moves_left(board):
    return any(EMPTY in row for row in board)

def minimax(board, depth, is_max):
    score = evaluate(board)
    if score == PLAYER_X:
        return score - depth
    if score == PLAYER_O:
        return score + depth
    if not is_moves_left(board):
        return 0
```

```
    if is_max:
        best = float('-inf')
        for row in range(3):
            for col in range(3):
                if board[row][col] == EMPTY:
                    board[row][col] = PLAYER_X
                    best = max(best, minimax(board, depth + 1, False))
                    board[row][col] = EMPTY
        return best
    else:
        best = float('inf')
        for row in range(3):
            for col in range(3):
                if board[row][col] == EMPTY:
                    board[row][col] = PLAYER_O
                    best = min(best, minimax(board, depth + 1, True))
                    board[row][col] = EMPTY
        return best

def find_best_move(board):
    best_val = float('-inf')
    best_move = (-1, -1)
    for row in range(3):
        for col in range(3):
            if board[row][col] == EMPTY:
                board[row][col] = PLAYER_X
                move_val = minimax(board, 0, False)
                board[row][col] = EMPTY
                if move_val > best_val:
                    best_move = (row, col)
                    best_val = move_val
    return best_move

def print_board(board):
    for row in board:
```

```
        print(" ".join(["X" if x == PLAYER_X else "O" if x == PLAYER_O else "." for x
in row]))

board = [
    [PLAYER_X, PLAYER_O, PLAYER_X],
    [PLAYER_O, PLAYER_X, EMPTY],
    [EMPTY, PLAYER_O, PLAYER_X]
]

print("Current Board:")
print_board(board)

move = find_best_move(board)
print(f"Best Move: {move}")
board[move[0]][move[1]] = PLAYER_X

print("\nBoard after best move:")
print_board(board)
```

Output :

```
Current Board:
X O X
O X .
. O X
Best Move: (1, 2)

Board after best move:
X O X
O X X
. O X


** Process exited - Return Code: 0 **
Press Enter to exit terminal
S . SANJAY - 241801246 - AI & DS - FD - 26.04.2025
```