

```

import math

def print_board(board):
    print("\n")
    for i in range(3):
        print(" " + " | ".join(board[i]))
        if i < 2:
            print("----+----+---")
    print("\n")

def check_winner(board, player):
    for row in board:
        if all(cell == player for cell in row):
            return True
    for col in range(3):
        if all(board[row][col] == player for row in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)):
        return True
    if all(board[i][2 - i] == player for i in range(3)):
        return True
    return False

def is_full(board):
    return all(cell != " " for row in board for cell in row)

def minimax(board, depth, is_maximizing):
    if check_winner(board, "O"):
        return 1
    if check_winner(board, "X"):
        return -1
    if is_full(board):
        return 0

    if is_maximizing:
        best_score = -math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "O"
                    score = minimax(board, depth + 1, False)
                    board[i][j] = " "
                    best_score = max(best_score, score)
        return best_score
    else:
        best_score = math.inf
        for i in range(3):
            for j in range(3):
                if board[i][j] == " ":
                    board[i][j] = "X"
                    score = minimax(board, depth + 1, True)

```

```

        board[i][j] = " "
        best_score = min(best_score, score)
    return best_score

def best_move(board):
    best_score = -math.inf
    move = (-1, -1)
    for i in range(3):
        for j in range(3):
            if board[i][j] == " ":
                board[i][j] = "O"
                score = minimax(board, 0, False)
                board[i][j] = " "
                if score > best_score:
                    best_score = score
                    move = (i, j)
    return move

def play_round():
    board = [[" " for _ in range(3)] for _ in range(3)]
    print_board(board)

    while True:
        try:
            pos = int(input("Enter position (1-9, left to right, top to bottom): "))
            if pos < 1 or pos > 9:
                print("Invalid! Choose a number between 1 and 9.")
                continue

            row, col = (pos - 1) // 3, (pos - 1) % 3
            if board[row][col] != " ":
                print("That spot is already taken, try again.")
                continue

            # Player move
            board[row][col] = "X"
            print_board(board)

            if check_winner(board, "X"):
                print(" You win this round!")
                return "player"
            if is_full(board):
                print("It's a draw!")
                return "draw"

            # Computer move
            print("Computer is thinking...")
            row, col = best_move(board)
            board[row][col] = "O"
            print_board(board)

```

```

        if check_winner(board, "O"):
            print(" Computer wins this round!")
            return "computer"
        if is_full(board):
            print("It's a draw!")
            return "draw"

    except ValueError:
        print("Please enter a valid number.")

def tic_tac_toe():
    scores = {"player": 0, "computer": 0, "draw": 0}
    print("Welcome to Tic Tac Toe! You are X, Computer is O.\n")

    while True:
        result = play_round()
        scores[result] += 1

        print(f" Score -> You: {scores['player']} | Computer: {scores['computer']} | Draws: {scores['draw']}")

        again = input("Do you want to play again? (y/n): ").lower()
        if again != "y":
            print("Thanks for playing! Final Scores:")
            print(f"You: {scores['player']} | Computer: {scores['computer']} | Draws: {scores['draw']}")
            break

if __name__ == "__main__":
    tic_tac_toe()

```

output:

Welcome to Tic Tac Toe! You are X, Computer is O.

```
  |  |
--+--+
  |  |
--+--+
  |  |
```

Enter position (1-9, left to right, top to bottom): 1

```
X |  |
--+--+
  |  |
--+--+
  |  |
```

Computer is thinking...

```
X |  |
--+--+
  | O |
--+--+
  |  |
```

Enter position (1-9, left to right, top to bottom): 3

```
X |  | X
--+--+
  | O |
--+--+
  |  |
```

Computer is thinking...

```
X | O | X
--+--+
  | O |
--+--+
  |  |
```

Enter position (1-9, left to right, top to bottom): 8

```
X | O | X
```

Enter position (1-9, left to right, top to bottom): 8

```
X | O | X
---+---+---
| O |
---+---+---
| X |
```

Computer is thinking...





```
X | O | X
---+---+---
O | O |
---+---+---
| X |
```

Enter position (1-9, left to right, top to bottom): 9

```
X | O | X
---+---+---
O | O |
---+---+---
| X | X
```

Computer is thinking...

```
X | O | X
---+---+---
O | O | O
---+---+---
| X | X
```

 Computer wins this round!
 Score -> You: 0 | Computer: 1 | Draws: 0
Do you want to play again? (y/n): n
Thanks for playing! Final Scores:
☒ You: 0 |  Computer: 1 |  Draws: 0

-