**Report on Task 8: Minimax Algorithm Implementation for Tic-Tac-Toe**

1. Introduction

This task involves implementing the Minimax Algorithm to develop an AI for playing Tic-Tac-Toe. The AI determines the best possible move to maintain an optimal strategy against a human opponent.

2. Implementation Details

2.1 Minimax Algorithm Overview

The Minimax algorithm is a recursive approach used in decision-making and game theory. It assesses potential future game states and assigns a score to each possible move.

* The AI (X) aims to maximize its score.
* The human player (O) seeks to minimize the AI’s score.
* The AI selects the move with the highest Minimax value.
* 2.2 Key Functions
* print\_board(board) → Displays the Tic-Tac-Toe board.
* check\_winner(board) → Checks for a winner.
* is\_board\_full(board) → Determines if the game ends in a draw.
* minimax(board, depth, is\_maximizing) → Evaluates board positions recursively.
* find\_best\_move(board) → Identifies the optimal move for the AI.

2**.3 Game Loop**

* The AI (X) makes its best move using the Minimax strategy.
* The player (O) inputs their move manually.
* The game continues until a player wins or the board is full.

**3. AI Response and Game Continues**

The AI continues to make moves, and the game progresses until one of the following occurs:

* AI Wins
* Human Player Wins
* It's a Draw

**4 output**

