**Assignment No: 5**

**Problem Statement:-**

Implement the Minimax Algorithm for game playing.

**Theory:-**

The **Minimax Algorithm** is a recursive algorithm used to make decisions in turn-based, two-player games such as tic-tac-toe, chess, or checkers. The goal of the algorithm is to maximize the score for one player while minimizing it for the opponent, assuming both players play optimally.

* **Minimax Decision**: The player maximizes their minimum guaranteed score.
* **Game Tree**: A tree of possible moves where each node represents a game state and edges represent actions that lead to the next state.

**Methodology:-**

1. **Game Representation**:
   * Represent the game state as a tree of possible moves.
   * Each node represents the current game configuration, and each edge represents a possible move.
2. **Minimax Function**:
   * Recursively evaluate the game tree. At each step:
     + **Maximizing Player**: Chooses the move with the maximum score.
     + **Minimizing Player**: Chooses the move with the minimum score.
3. **Game Evaluation**:
   * Assign scores to terminal states (e.g., a win, draw, or loss).
   * The algorithm explores all possible game outcomes to determine the best move.
4. **Alpha-Beta Pruning** (Optional):
   * Prune branches of the game tree that won't affect the final decision, reducing the number of nodes evaluated.
5. **Application**:
   * **Tic-Tac-Toe**: The algorithm can ensure that the game either ends in a win or a draw for the maximizing player.
   * **Chess or Checkers**: Minimax can be used, though more sophisticated algorithms are used in practice due to the complexity of these games.

**Conclusion:-**

We implemented the Minimax algorithm for game playing, enabling the computer to make optimal decisions by evaluating all possible moves.