**Project Title: Minimax Algorithm Implementation (Game Theory)**

**Purpose (Why this code is used):**

* The **Minimax Algorithm** is a decision-making algorithm used in **two-player games** like Chess, Tic-Tac-Toe, or Checkers.
* Its purpose is to help a player choose the **best possible move**, assuming the opponent also plays optimally.
* The **Maximizing Player** tries to get the **highest score**, while the **Minimizing Player** tries to reduce it.
* It ensures the **optimal decision** in competitive game scenarios.

**Working (How it works):**

1. **Game Tree:**
   * The algorithm assumes all possible moves can be represented as a **binary tree**.
   * Each **leaf node** represents a possible game outcome (a score).
   * Each **internal node** represents a decision point for one of the players.
2. **Recursion:**
   * The algorithm works recursively by exploring all possible outcomes.
   * If it’s the **Maximizer’s turn**, the algorithm chooses the **maximum** value from the next level.
   * If it’s the **Minimizer’s turn**, it chooses the **minimum** value.
3. **Base Case:**
   * When the algorithm reaches the **target depth** (end of the tree), it returns the value of that leaf node (final score)

