**Name: Muhammad Huzaifa**

**Roll No: SU92-BSAIM-S24-010**

**Section: 3A**

**Subject: AI LAB**

**DOCUMENTATIONS:**

**Code Documentation:**

**1. mini\_max(cur\_Depth, node\_Index, max\_Turn, scores, target\_Depth):**

This is a **recursive function** that simulates the **Min-Max decision process**.

**Parameters:**

* cur\_Depth: The current depth in the game tree.
* node\_Index: The index of the current node in the scores list.
* max\_Turn: A boolean indicating whether it's the **maximizing player's turn** (True) or the **minimizing player's turn** (False).
* scores: A list of terminal node values (leaf values in the game tree).
* target\_Depth: The maximum depth of the game tree.

**Base Case:**

* If cur\_Depth == target\_Depth, return the score of the current node.

**Recursive Case:**

* If it's the **maximizing player's turn (max\_Turn = True)**, return the **maximum** of the two child nodes.
* If it's the **minimizing player's turn (max\_Turn = False)**, return the **minimum** of the two child nodes.

**2. scores List:**

****

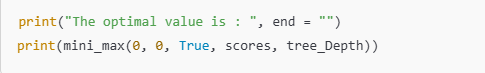
* Represents the possible outcomes (leaf nodes) of the game.

**3. tree\_Depth Calculation:**

****

* Computes the depth of the binary tree using **log₂(n)**, where n is the number of leaf nodes.

**4. Running the Algorithm:**

****

 **Starts from** depth 0, node 0, and the **maximizing player's turn**.

 **Finds the optimal move** using the **Min-Max algorithm**.

**5. Expected Output:**

****

**­­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**