

**Name: Muhammad-Jawad-Haider**

**Roll No.: SU92-BSAIM-F24-056**

**Section: BSAI-3A**

**Subject: Artificial Intelligence (Lab)**

**Task: 8**

At the beginning, the program imports the **math** library because it needs the logarithm function to calculate the depth of the game tree. Then, the **minimax ()** function is defined. This function takes five parameters — the current depth (curDepth), the node index (nodeIndex), a boolean to check whose turn it is (maxTurn), the list of scores (scores), and the target depth of the tree (targetDepth). Inside the function, there’s a condition that checks if the current depth has reached the target depth. If it has, the function returns the score at that node. If it’s the maximizer’s turn, it picks the maximum value between two child nodes. If it’s the minimizer’s turn, it picks the minimum value. This process repeats recursively until the target depth is reached.

The **scores** list in this program is [3, 5, 2, 9, 3, 5, 2, 9]. The tree depth is calculated using math.log(len(scores), 2), which gives 3.0. However, in computers, decimal numbers are not always stored perfectly — 3.0 is actually stored as something like 2.9999999999999996. Because of this tiny difference, the condition curDepth == targetDepth does not become true when curDepth reaches 3. As a result, the recursion stops one level earlier than it should. This means the program doesn’t reach the actual leaf scores, and it makes its final decision using the level above them.

Due to this early stopping, the program gives an **output of 5** instead of 9.