# Explanation of Minimax Algorithm Code

* **This Python program demonstrates the Minimax Algorithm, used in game theory to find the optimal move for a player assuming the opponent also plays optimally.**
* **The function minimax() takes five parameters: current depth, node index, whose turn it is, the list of scores, and the target depth.**
* **If the current depth equals the target depth, it returns the score at that index, meaning it's a leaf node.**
* **If it’s the maximizing player’s turn, it picks the maximum value from its two child nodes.\**
* **If it’s the minimizing player’s turn, it picks the minimum value.  
  The list scores = [3, 5, 2, 9, 3, 5, 2, 9] represents the possible outcomes of the game.**
* **The tree depth is calculated using math.log(len(scores), 2).  
  The function is called initially with parameters (0, 0, True, scores, treeDepth) meaning the game starts at depth 0 and with the maximizing player.**
* **It explores all possible moves recursively to find the best result.  
  Finally, it prints the optimal value, which represents the best possible outcome for the maximizing player.**
* **For this example, the output is 9, which is the best achievable score if both players play perfectly.**