|  |  |
| --- | --- |
| EX.NO:-8 | **Min max algorithm** |
| DATE: |

**Program:**

import math

def minimax(curDepth, nodeIndex, maxTurn, scores, targetDepth):

if curDepth == targetDepth:

return scores[nodeIndex]

if maxTurn:

return max(minimax(curDepth + 1, nodeIndex \* 2, False, scores, targetDepth),

minimax(curDepth + 1, nodeIndex \* 2 + 1, False, scores, targetDepth))

else:

return min(minimax(curDepth + 1, nodeIndex \* 2, True, scores, targetDepth),

minimax(curDepth + 1, nodeIndex \* 2 + 1, True, scores, targetDepth))

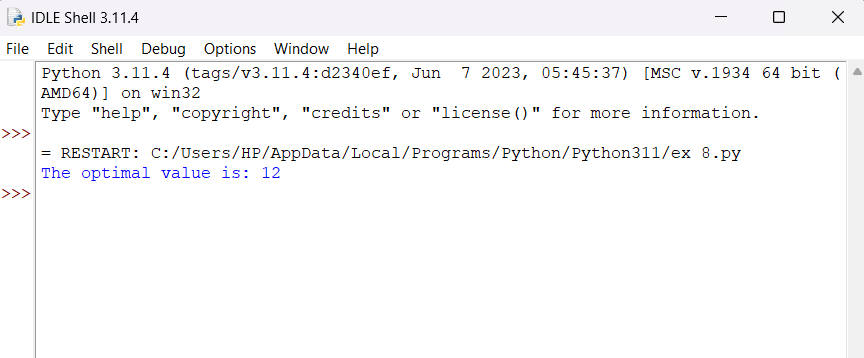
scores = [3, 5, 2, 9, 12, 5, 23, 23]

treeDepth = math.log(len(scores), 2)

print("The optimal value is:", end=" ")

print(minimax(0, 0, True, scores, treeDepth))

**Output:**

****

**Result:**

The given Min-Max is compiled and implemented.