**AI FOR LLM- CSA1704**

**14. Python implementation of Alpha–Beta pruning**

**CODE:**

# Alpha-Beta Pruning Example

def alphabeta(node, depth, alpha, beta, maximizingPlayer, values, index=0):

# If leaf node

if depth == 0 or index >= len(values):

return values[index]

if maximizingPlayer:

best = float('-inf')

for i in range(2): # Left and Right child

val = alphabeta(node\*2+i, depth-1, alpha, beta, False, values, index\*2+i)

best = max(best, val)

alpha = max(alpha, best)

if beta <= alpha: # Beta cut-off

break

return best

else:

best = float('inf')

for i in range(2):

val = alphabeta(node\*2+i, depth-1, alpha, beta, True, values, index\*2+i)

best = min(best, val)

beta = min(beta, best)

if beta <= alpha: # Alpha cut-off

break

return best

# Example Game Tree

# Depth = 3, so 2^3 = 8 leaf values

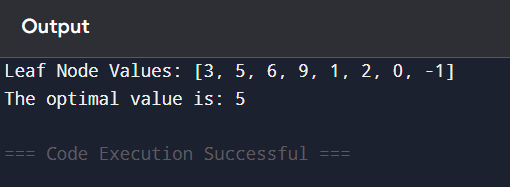
values = [3, 5, 6, 9, 1, 2, 0, -1]

print("Leaf Node Values:", values)

best\_score = alphabeta(0, 3, float('-inf'), float('inf'), True, values)

print("The optimal value is:", best\_score)

**OUTPUT:**

****