#Name: Manasi Kale

#Roll.No: B 13

import math

def alpha\_beta\_pruning(depth, node\_index, maximizing\_player, values, alpha, beta):

    if depth == 0 or node\_index >= len(values):

        return values[node\_index]

    if maximizing\_player:

        max\_eval = -math.inf

        for i in range(2):

            eval = alpha\_beta\_pruning(depth - 1, node\_index \* 2 + i, False, values, alpha, beta)

            max\_eval = max(max\_eval, eval)

            alpha = max(alpha, eval)

            if beta <= alpha:

                break

        return max\_eval

    else:

        min\_eval = math.inf

        for i in range(2):

            eval = alpha\_beta\_pruning(depth - 1, node\_index \* 2 + i, True, values, alpha, beta)

            min\_eval = min(min\_eval, eval)

            beta = min(beta, eval)

            if beta <= alpha:

                break

        return min\_eval

if \_\_name\_\_ == "\_\_main\_\_":

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

    depth = 3

    optimal\_value = alpha\_beta\_pruning(depth, 0, True, values, -math.inf, math.inf)

    print(f"The optimal value is {optimal\_value}")

