1.C.)Implement MINIMAX Algorithm

AIM:

To implement the Minimax algorithm that chooses the optimal move for a player, assuming the opponent also plays optimally.

CODE:

```
import random

def minimax(depth, nodeIndex, maximizingPlayer, values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]

    if maximizingPlayer:
        best = -1000
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, False, values,
alpha, beta)
            best = max(best, val)
            alpha = max(alpha, best)
            if beta <= alpha:
                break
        return best
    else:
        best = 1000
        for i in range(0, 2):
            val = minimax(depth + 1, nodeIndex * 2 + i, True, values,
alpha, beta)
            best = min(best, val)
            beta = min(beta, best)
            if beta <= alpha:
                break
        return best
values = [3, 5, 6, 9, 1, 2, 0, -1]
print("The optimal value is :", minimax(0, 0, True, values, -1000, 1000))
```

OUTPUT:

```
The optimal value is : 5
```

RESULT:

The code is executed as expected and the output have been verified successfully.