Lab 7 Min max algorithm with application

Name- Priyanshu Rishi
Rge No- RA1911042010041
Batch- CSBS (R1)

CODE:

```
# Python3 program to demonstrate
# working of Alpha-Beta Pruning

# Initial values of Alpha and Beta
MAX, MIN = 1000, -1000

# Returns optimal value for current player
#(Initially called for root and maximizer)
def minimax(depth, nodeIndex, maximizingPlayer,
            values, alpha, beta):

    # Terminating condition. i.e
    # leaf node is reached
    if depth == 3:
        return values[nodeIndex]

    if maximizingPlayer:

        best = MIN

        # Recur for left and right children
        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)

            # Alpha Beta Pruning
            if beta <= alpha:
                break

        return best

    else:
        best = MAX

        # Recur for left and
        # right children
```

```
        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)

            # Alpha Beta Pruning
            if beta <= alpha:
                break

        return best

# Driver Code
if __name__ == "__main__":

    values = []
    # number of elements as input
    n = int(input("Enter number of elements : "))

    # iterating till the range
    for i in range(0, n):
        ele = int(input())

        values.append(ele) # adding the element
    print("The optimal value is :", minimax(0, 0, True, values, MIN, MA
X))
```

OUTPUT:

```
Enter number of elements : 8
0
-3
3
4
5
6
-5
0
The optimal value is : 0
```