```python
# AlphaBetaPruning
MIN, MAX = -1000, 1000

def minmax(depth, nodeIndex, maximizing_player, values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]
    if maximizing_player:
        best = MIN
        for i in range(0, 2):
            val = minmax(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta)
            best = max(best, val)
            alpha = max(best, alpha)
            if beta <= alpha:
                break
        return best
    else:
        best = MAX
        for i in range(0, 2):
            val = minmax(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta)
            best = min(best, val)
            beta = min(best, beta)
            if beta <= alpha:
                break
        return best

if __name__ == '__main__':
    values = [3, 5, 6, 9, 1, 2, 0, -1]
    print(minmax(0, 0, True, values, MIN, MAX))
```

⇥ 5

```python
def minmax(depth, nodeIndex, maximizing_player, values, alpha, beta):
    if depth == 3:
        return values[nodeIndex]
    if maximizing_player:
        best = MIN
        for i in range(0, 2):
            val = minmax(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta)
            best = max(best, val)
            alpha = max(best, alpha)
            if beta <= alpha:
                break
        return best
    else:
        best = MAX
        for i in range(0, 2):
            val = minmax(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta)
            best = min(best, val)
            beta = min(best, beta)
            if beta <= alpha:
                break
        return best
```