Minimax Algorithm with Alpha-Beta Pruning:

```
function minimax(node, depth, isMaximizingPlayer, alpha, beta):

    if node is a leaf node
        return value of the node

    if isMaximizingPlayer
        bestVal = -INFINITY
        for each child node
            value = minimax(node, depth+1, false, alpha, beta)
            bestVal = max( bestVal, value)
            alpha = max( alpha, bestVal)
            if beta <= alpha
                break
        return bestVal
   else :
        bestVal = +INFINITY
        for each child node
            value = minimax(node, depth+1, true, alpha, beta)
            bestVal = min( bestVal, value)
            beta = min( beta, bestVal)
            if beta <= alpha
                break
        return bestVal


// Calling the function for the first time.
minimax(0, 0, true/false, -INFINITY, +INFINITY)
```

Instruction:

1. Asks the first node if maximizer/minimizer
2. Asks number of child, n: >=3
3. Asks the depth of the tree, d
4. In the last depth child will get value. The number of child/leaf node will be: n^(d-1)