Free Response Question 2

Pause and write the pseudocode for deleting a node in a BST given that we know this node is either a leaf (it has zero kids) or that it has only one child. After deletion, return the root of the new tree.

Node DeleteSingle(Node root, int val).

Assume that the node with this value exists, and that all values are unique.

PAUSE and write pseudocode.

One possible code solution will be in next section.

ANSWER:

There should be some **Pseudocode** below. If you don't see it, then please refresh.

Please note, that even though it says below that the language is Java, it's actually **Pseudocode**. Java is selected only to provide better syntax highlighting.

```
JAVA
Node DeleteSingleMain( Node root, int val)

    (parent, curr) = FindWithParent(root, val)
     Return DeleteSingle(root, val, parent, curr)


Node DeleteSingle(Node root, int val, Node parent, Node curr)

    // Let us find the one kid of this curr node, if any
    kid = NULL
    if (! curr.left)
        kid = curr.left
    if (! curr.right)
        kid = curr.right

    // We have the parent, we have the new kid. Release the memory of the node
    DeallocateNode(curr)

    // Extreme case: the root is the node to be deleted.
    if (parent == NULL)
        return kid

    // We still do not know if the kid should be parent’s left child or right.
    if (parent.val > val)
        parent.left = kid
    else
        parent.right = kid
    return root


(Node, Node) FindWithParent (Node root, int val)
    curr = root, parent = NULL

    // Loop Invariant: curr is the next val to consider and parent is curr’s parent
    while (curr.val != val)
        parent = curr
        if (curr.val > val)
            curr = curr.left
        else
            curr = curr.right
    Return (parent, curr)
```