

```

registerNeeds( T )
Calculates the registers required for expression tree T result,
stores value at T.regCount and returns same.

if ( isLeafNode( T ) ) then (
    if ( T.kind is Identifier ) T.regCount  $\leftarrow$  1
    if ( T.kind is IntegerLiteral ) T.regCount  $\leftarrow$  1
    if ( T.kind is FloatLiteral ) T.regCount  $\leftarrow$  1
    :
) else (
    if ( T.kind is BinaryOp ) then (
        registerNeeds( T.leftTree )
        registerNeeds( T.rightTree )
        if ( T.leftTree.regCount = T.rightTree.regCount ) then (
            T.regCount  $\leftarrow$  T.leftTree.regCount + 1 // Nothin' special about .leftTree
        ) else (
            T.regCount  $\leftarrow$  max{ T.leftTree.regCount, T.rightTree.regCount }
        )
    ) else if ( T.kind is FunctionCall ) then (
        // If the architecture uses registers for functions result passing,
        // they aren't part of the register count for expression trees
        let P be the set of expressions for T's function call arguments
        if ( P =  $\emptyset$  ) then (
            T.regCount  $\leftarrow$  0
        ) else (
            let N  $\leftarrow$  an empty indexed List
            foreach (  $p_i \in P$  ) do (
                append registerNeeds( $p_i$ ) to N
            )
            sort N in descending order
            regneeds  $\leftarrow$  N[0]
            // need enough registers for the ``largest'' parameter
            let R  $\leftarrow$  regneeds
            // storing the N[0] result while other
            // params are calculated requires one register
            let rem  $\leftarrow$  R - 1 // remaining registers for use
            for (  $j \in [1, \dots, |P| - 1]$  ) do (
                regneeds  $\leftarrow$  N[j]
                if ( rem < regneeds ) then (
                    R  $\leftarrow$  R + (regneeds - rem)
                    rem  $\leftarrow$  regneeds
                )
                // storing the current result requires one register
                rem  $\leftarrow$  rem - 1
            )
            T.regCount  $\leftarrow$  R
        )
    )
)
return T.regCount

```