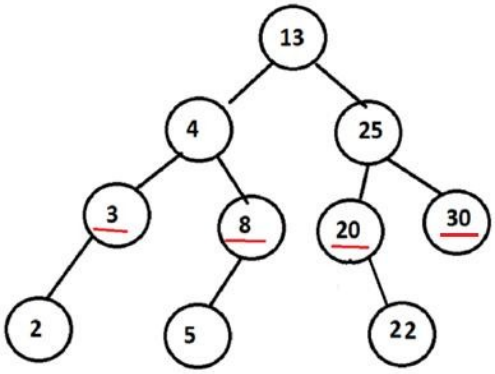
1) (10 pts) DSN (Binary Trees)

Write a **recursive** function named `sumAtDepth` that takes a pointer to the root of a binary tree, `root`, and non-negative integer, `depth`, and returns the sum of all the values in the nodes that are at a level `depth` below the root. For example, if you pass the root of the following binary tree and `depth` = 2, the function should return 61 (= 3 + 8 + 20 + 30) since the each of the nodes storing 3, 8, 20 and 30 are 2 levels below the root node of the tree. You may assume that depth is a non-negative integer.



You must write your solution in a **single** function. You cannot write any helper functions.

The function signature and node struct are given below.

```
typedef struct node
{
   int data;
   struct node *left;
   struct node *right;
} node;

int sumAtDepth(node *root, int depth) {

  if (root == NULL)              // 1 pt
    return 0;                    // 1 pt

  if (depth == 0)                // 1 pt
    return root->data;           // 1 pt

  // 1 pt return, 2 pts each recursive call, 1 pt adding
  return sumAtDepth(root->left, depth-1) + sumAtDepth(root->right, depth-1);
}
```