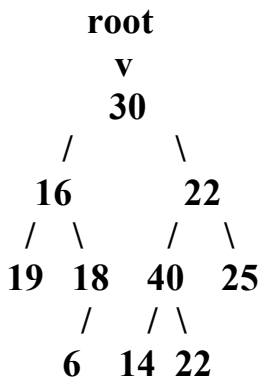
1) (10 pts) ALG (Binary Trees)

The following code, when passed the root of a binary tree and returns a result calculated by adding the values of some nodes and subtracting the values of others. If the function whatDoesItDo is called on the root of the binary tree shown below, for each value in the tree, indicate whether it gets added (A) or subtracted (S), **with respect to the original call on the root of the tree below.** No need to state the final return value. (**Grading Note:** +1 for each correct slot, +0 for each slot left blank, -1 for each incorrect slot, minimum score is 0.)

```
#include <stdio.h>
#include <stdlib.h>
typedef struct bintreenode {
    int data;
    struct bintreenode* left;
    struct bintreenode* right;
} bintreenode;

int whatDoesItDo(bintreenode* root) {

    if (root == NULL) return 0;
    if (root->left == NULL && root->right == NULL) return root->data;

    if (root->left == NULL) return root->data + whatDoesItDo(root->right);
    if (root->right == NULL) return root->data + whatDoesItDo(root->left);

    if (root->left->data > root->right->data)
        return root->data + whatDoesItDo(root->left) - whatDoesItDo(root->right);

    return root->data + whatDoesItDo(root->right) - whatDoesItDo(root->left);
}
```



For each open slot, either write the letter 'A' for added, or 'S' for subtracted.

30	**A**	16	**S**	22	**A**
19	**S**	18	**A**	40	**A**
25	**S**	6	**A**	14	**S**

23 **A (Grading is described in the question +1 correct, 0 blank, -1 incorrect, cap at 0)**