Question 1

Given a binary search tree **T** in its parenthesis representation, write a recursive function **isAVL(T)** to check whether **T** satisfies all the properties of an **AVL tree**. Your function should run in **O(n) time complexity**.

Your program should include the following function and should **not** change the given function parameters.

- isAVL(T) : Return the ***height*** of the tree rooted at T if it satisfies all the properties of an **AVL tree**, otherwise return -1.

Here, ***height*** of a tree is defined as the number of nodes in the longest root-to-leaf path.

Note: You are not supposed to use a separate function for calculating the balance factor or the height of a tree.

Use the following node structure for implementing **AVL tree** in your program.

```
Struct node {
        int key;
        struct node *left;
        struct node *right;
        }
```

Input / Output Format:

- The input contains the parenthesis representation of tree **T** in a single line.
- The output should be printed in a new line.

Sample Input 1:
(20 (15 (10 (1 ( ) ( )) (12 ( ) ( ))) (17 ( ) ( ))) (36 (35 ( ) ( )) (60 (42 ( ) ( )) ( ))))

Output 1:
4

Sample Input 2:
(20 (15 (10 (1 ( ) ( )) (12 ( ) ( ))) ( )) (36 (35 ( ) ( )) (60 (45 ( ) ( )) ( ))))

Output 2:
-1