Skills Assessment 2

Description:
In this skills assessment, you will implement two functions that take in a pointer to the root node of a binary tree as the parameter.

starter.zip (Starter Code):
The following is a brief description of the starter code provided to you:

- **ques.cpp**
  - You have been provided with the definition of a Node struct of a Binary Tree. This Node struct contains the following members:
    - **Node* left:** pointer to the left child of this Node
    - **Node* right:** pointer to the right child of this Node
    - **int data:** The integer value stored in the Node

Task
You must implement the **countAscendingPaths** and **sumOfLeftLeaves** functions.

- **Note that the binary tree you will be operating on in both of these functions is not necessarily a binary search tree - i.e. the left child is not necessarily smaller than the parent node, similarly the right child is not necessarily larger than the parent node.**

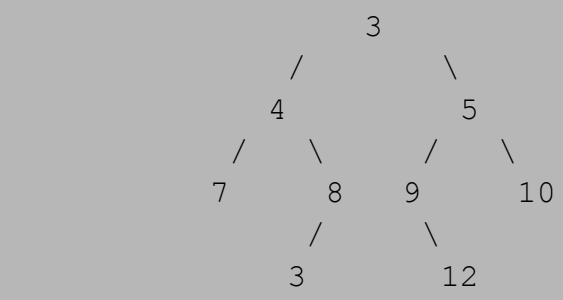
Question 1 (25 points)

countAscendingPaths Function

- The ***int countAscendingPaths(Node* root)*** function takes in a single parameter: a pointer to the root node of a binary tree
- **Your task is to return the number of ascending paths in the binary tree.**
  - **Path:** A path is defined as a sequence of nodes that start from the root node and extend towards any leaf node.
    - If the root does not contain any children, it is considered a leaf.
  - **Ascending Sequence:** An ascending sequence is a sequence of data values where each value is **greater than** the previous value.
- **Counting Ascending Paths:** The objective is to count all the paths within the binary tree where the values along a path form an ascending sequence, and the path starts from the root node and extends to any leaf node.

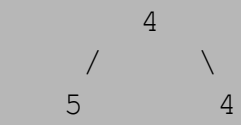
- Example:

Input:
Binary Tree A:



countAscendingPaths(A->root)
Output: **3**
Reason: Ascending Paths are [3 -> 4 -> 7], [3 -> 5 -> 10], [3 -> 5 -> 9 -> 12]

Input:
Binary Tree B:



countAscendingPaths(B->root)
Output: **1**
Reason: The only ascending path is [4 -> 5]. Note that 4 -> 4 is not an ascending path (as we are only considering paths that are strictly increasing).

Question 2 (25 points)
sumOfLeftLeaves Function

- The ***int sumOfLeftLeaves(Node* root)*** function takes in a single parameter: a pointer to the root node of a binary tree
- **Your task is to return the sum of the data of nodes that are left leaves.**
  - A node is considered as a left leaf if it satisfies two conditions:
    - It must not contain any children
    - It must be the left child of its parent node (and it must have a parent node)
      - **This means that the root can never be a left leaf**

- You can return 0 if there are no left leaves in the binary tree.

- **Example:**

```
Input:
Binary Tree A:
                    3
                  /   \
                 4     5
               /   \   /   \
              7     8 9     10
                   /     \
                  3       12

sumOfLeftLeaves(A->root)
Output: 10
Reason: There are two left leaves in the binary tree, and they
have the values 7 and 3. Therefore, 7 + 3 = 10.
```

You have been provided with Binary Tree A (given as an example here) within main() of ques.cpp as a sample test case that you can run on both your functions to verify that this sample test case is working correctly on both of your functions.

You are encouraged to create helper functions to solve these two given functions if you think they are necessary.

- **Note: Please create the helper functions within the respective header guards of each question. This will allow us to isolate the two questions from each other when autograding. Please do not make any changes to the header guards given to you in the starter file.**

Also note that the sample test cases provided are a good indication that your logic is correct but they may not necessarily reflect your final score on the hidden autograded test cases. You are encouraged to create additional sample test cases in main() to test your code.

Deliverables and Submissions:

In a zip folder named using the format **<FirstName>-<LastName>-<UIN>-<SA2>.zip** you must submit the following file to Canvas:

- **ques.cpp** (containing your functional implementation of the countAscendingPaths and sumOfLeftLeaves methods)

Do not use angular brackets in the name of the submission folder.