Assignment Objective: Build understanding of balanced Binary Search Trees by implementing an AVL ADT.

Requirements:

* Create a class called sAVL that implements an AVL BST. Do this by copying p4.h to p5.h, p4.cpp to p5.cpp, and changing the class sBST to sAVL in the p5.h and p5.cpp files; don’t forget to include p5.h, not p4.h, inside the p5.cpp. The new members sAVL are:
  + Private members:
    - void rotateLeft(sNode \* &p1) ; p1 is a node that is right heavy imbalanced.
    - void rotateRight(sNode \* &p1) ; p1 is a node that is left heavy imbalanced.
    - void bal(sNode \* &p) to balance p
    - int height(sNode \*p) const; // returns the height of p if it is real; otherwise 0 if it is NULL
    - int calcHeight(sNode \*p) to return the calculation of p’s height based on the height of its children; **this is NOT a recursive call**
  + Public members:
    - Modify printIt(sNode \*p, int &index) to print the height of each node and its “index”; see the supplied output files for the format.
    - Otherwise, there are no new members.
* Implement, if not already there, a non-member function “int max(int a, int b)” that returns the maximum value of ***a*** and ***b***.
* You must not use any other data structure, whether built-in or otherwise.
* Compile your program: g++ p5.cpp p5m.cpp -o p5
* First submission: The highlighted portions shall be submitted as described below. Note, to do any testing, there should be a “stub” for each of the functions that are not yet completed.
* Final submission: The complete assignment shall be submitted as described below.
* Demonstrate that the sAVL data structure works:
  + Run your program as follows:

P5 p5In\_1.txt p5Remove\_1.txt p5IsIn\_1.txt > p5output1.txt

P5 p5In\_2.txt p5Remove\_2.txt p5IsIn\_2.txt > p5output2.txt

* + Compare your output files to the corresponding p5CorrectOutput1.txt and p5CorrectOutput2.txt files.
  + For the p5b submission, also do the following:
  + Recompile your program as follows:
  + g++ p5.cpp p52m.cpp -o p52
  + Capture the output of the following run:

P52 > p52Output1.txt

* + Modify your p5.cpp by commenting out the calls to bal().
  + Recompile your program as follows:
  + g++ p5.cpp p52m.cpp -o p52
  + Capture the output of the following run:

P52 > p52Output2.txt

* Deliverables:
  + Turned into D2L: put a zip file containing:
    - Your p5.h file
    - Your p5.cpp file
    - Your p5output1.txt file
    - Your p5output2.txt file
    - For the final submission
      * Your p52Output1.txt file
      * Your p52Output2.txt file
    - DO NOT put a project into D2L
  + Turned into class: a hardcopy of the above files in the order given.