Problem 1:

Using good coding practices, make your own binary search tree data structure of integers that will contain classes `BinarySearchTree` and `TreeNode`. You can then use your tree to define a `TreeIterator` class that contains a pointer to a `TreeNode`. Create a `BinarySearchTree` class's `begin` member function that returns a `TreeIterator` object that points to the node with the smallest value while the tree's `end` member function returns a `TreeIterator` object that points past the node with the largest value.

Write all functions needed such that the main function below compiles and runs showing the sample output in Figure 1.

```
BinarySearchTree bst;
bst.insert(3); bst.insert(2); bst.insert(11);
bst.insert(13); bst.insert(5); bst.insert(17); bst.insert(17);
std::cout << "Your tree consists of : ";
for (auto x : bst) std::cout << x << ",";
std::cout << std::endl;

bst.erase(6);
bst.erase(11);
std::cout << "Tree now  consists of : ";
for (auto x : bst) std::cout << x << ",";
std::cout << std::endl;

TreeIterator start = bst.begin();
TreeIterator stop = bst.end();
std::cout << "Checking initial value: ";
if (start != stop) std::cout << *start++ << std::endl;
```

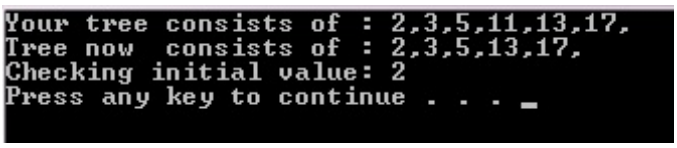

Figure 1: Sample output.

Good Coding Practices:

- think about cross-platform. Don't use Windows or Mac only commands. For example, pause == cin.get() twice, write many \n vs. system(clear) or system('cls).
- passing objects by reference & or const & when possible
- using field initializer list when possible in all constructors

Instructions for submission:

- Name your main file exactly `hw6.cpp`, and the rest being `TreeNode.h`, `TreeNode.cpp`, `BinarySearchTree.h`, `BinarySearchTree.cpp`, `TreeIterator.h`, and `TreeIterator.cpp`.
- You may not use `#include "stdafx.h"`.
- Add code description in the comment at the beginning of the file.
  A sample description may look like:

```
/*
    PIC 10B 2A, Homework 1
    Purpose: Tic-tac-toe game
    Author: Hanqin Cai
    Date: 10/10/2019
*/
```

- Submit your header files and source codes to CCLE in separate files. Only .h and .cpp files should be uploaded.