

1. Code a `TreeNode` Class which contains two `TreeNode` pointers and an integer.
2. Code a `BinaryTree` class for integers, implement the following public interface:

```
BinaryTree(); // constructor
void add(toadd int); // both the add and height methods are
int height() // implemented by a private recursive method
```

3. Code a main program which adds several integers to a binary tree and prints the height of the tree.

Note: you will have two add and two height methods, one public and one private. So for example . You need an extra private method such as `add(TreeNode *toAdd, TreeNode *addHere)` to call recursively.

Note: For a tree with just one node, the root node, the height is defined to be 0, if there are 2 levels of nodes the height is 1 and so on. A null tree (no nodes except the null node) is defined to have a height of -1.

4. Starting a new Visual Studio solution, write the code for a binary search tree of chars, and use a recursive method to search the tree for a particular character and return true if it is present, false otherwise.
5. Add the code to print all the values in the char binary search tree in ascending order
6. Will this use a Pre-order traversal? A Post-order traversal? Or some other type of traversal?
7. Starting a new Visual Studio solution, write the code for a template binary search tree class.