Part 1: You are to create a splay tree using a small data set.

For ease it coding, we will do a “top down” splay, in which we start splaying before we ever get to the node we care about. This makes it a bit easier than what the book shows in Chapter 4 as we don’t have to find the node and look up two levels. We make another simplification. Instead of doing the double rotation (for a zigzag case), we just go down ONE level and do a zig. This makes the coding easier as we don’t have to code a double rotation.

Create a splay tree abstract data type, with the following operations:
- void insert(string)
- bool find(string)
- string deleteMax()
- string deleteMin()
- printTree(string msg): prints the msg before printing the tree prettily
- makeEmpty(): remove all nodes from the tree, recursively

Create a splay tree from tiny2.txt and test the various operations. TestSplay.cpp contains code to do this checking. (Leave this code in your program so the grader can verify all operations work.)

Part 2: Create a new splay tree from the input insert2.txt. Read in search2.txt and count the number of words which were not found.

Hints:
In the following example, notice that leftMax (and rightMin) moves to the next place of insertion. In this small example, it is difficult to see the advantage, but in a huge tree, having leftMax means you don’t have to search from the root for an insertion point for the newly cast off part of the tree.

Original Tree:

![Original Tree Diagram]

CS 2420 – Assignment 2
Splay on n (zigzag)

leftExtra

leftMax

rightExtra

rightMin

Tree

Splay on n (reassemble)
Attach n's kids to rightExtra and leftExtra