CSE 2320 Lab Assignment 4

Due April 11

Goals:

1. Understanding of red-black tree structural constraints.

2. Application of a stack.

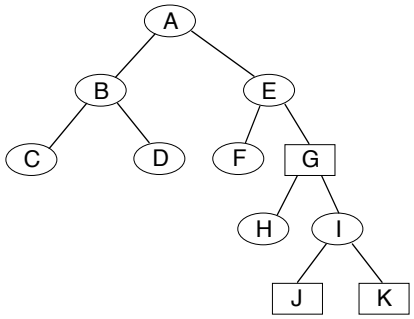
Requirements:

1. Write a linear-time program to test whether a binary tree satisfies the red-black tree properties.

   The simple input format corresponds to postorder traversal. An "`R`" corresponds to a red tree node, a "`B`" to a black tree node, and a "`.`" to the sentinel. If a tree includes *n* data nodes, then the sentinel will be processed *n* + 1 times. The input

   ```
   ..B..BB..B..B..R..RBRBB
     C   DB   F   H   J   KIGEA (This line is not part of the input.)
   ```

   corresponds to the tree:



2. Submit your program on Blackboard by 3:15 pm on April 11. One of the comment lines should indicate the compilation command used on OMEGA (5 point penalty for omitting this).

Getting Started:

1. The following test cases are useful:

   ```
   ..B..BR..R..RBB
   ..B..BB..B..B..R..RBRBB
   ..B..BB..B..B..B..BRRBB
   ..B..BR..R..RBB..B..R..B..BRBBB
   ..R..RB..R..RBR..R..RB..R..RBRB
   ..R.B...RBR...RBB
   ..B..BR
   ..R.B
   ..B.R
   ```

2. Your program should echo the input string. The string will be processed using a stack whose entries contain the color of the root of a subtree and its black height. If an error occurs, then processing stops and a message should be issued (see the sample output). If the string represents a legal tree, then a confirmation message should be printed. At termination, the stack should have one entry for the root of the tree.

3. Your program should not prompt for an input file name. The maximum length for the input string is 200 symbols.

4. Pointers are not necessary for implementing this assignment. Simple tables are sufficient. Recursion should not be used.