8.3 Again, start with the tree.java program and make a tree from characters typed

by the user. This time, make a complete tree—one that is completely full

except possibly on the right end of the bottom row. The characters should be

ordered from the top down and from left to right along each row, as if writing

a letter on a pyramid. (This arrangement does not correspond to any of the

three traversals we discussed in this chapter.) Thus, the string ABCDEFGHIJ

would be arranged as

A

B C

D E F G

H I J

One way to create this tree is from the top down, rather than the bottom up as

in the previous two Programming Projects. Start by creating a node which will

be the root of the final tree. If you think of the nodes as being numbered in

the same order the letters are arranged, with 1 at the root, then any node

numbered n has a left child numbered 2\*n and a right child numbered 2\*n+1.

You might use a recursive routine that makes two children and then calls itself

for each child. The nodes don’t need to be created in the same order they are

arranged on the tree. As in the previous Programming Projects, you can jettison

the search-tree routines from the Tree class.