CSC 220
Data Structures

Assignment 8
Tree Traversal

Due by midnight, 26 February 2014

NOTE: Assignments are to be done individually. Do not look at another student's results and do not allow another student to look at your results. Doing so will result in charges of academic misconduct.

For this program, you will create ordered binary trees from this input:

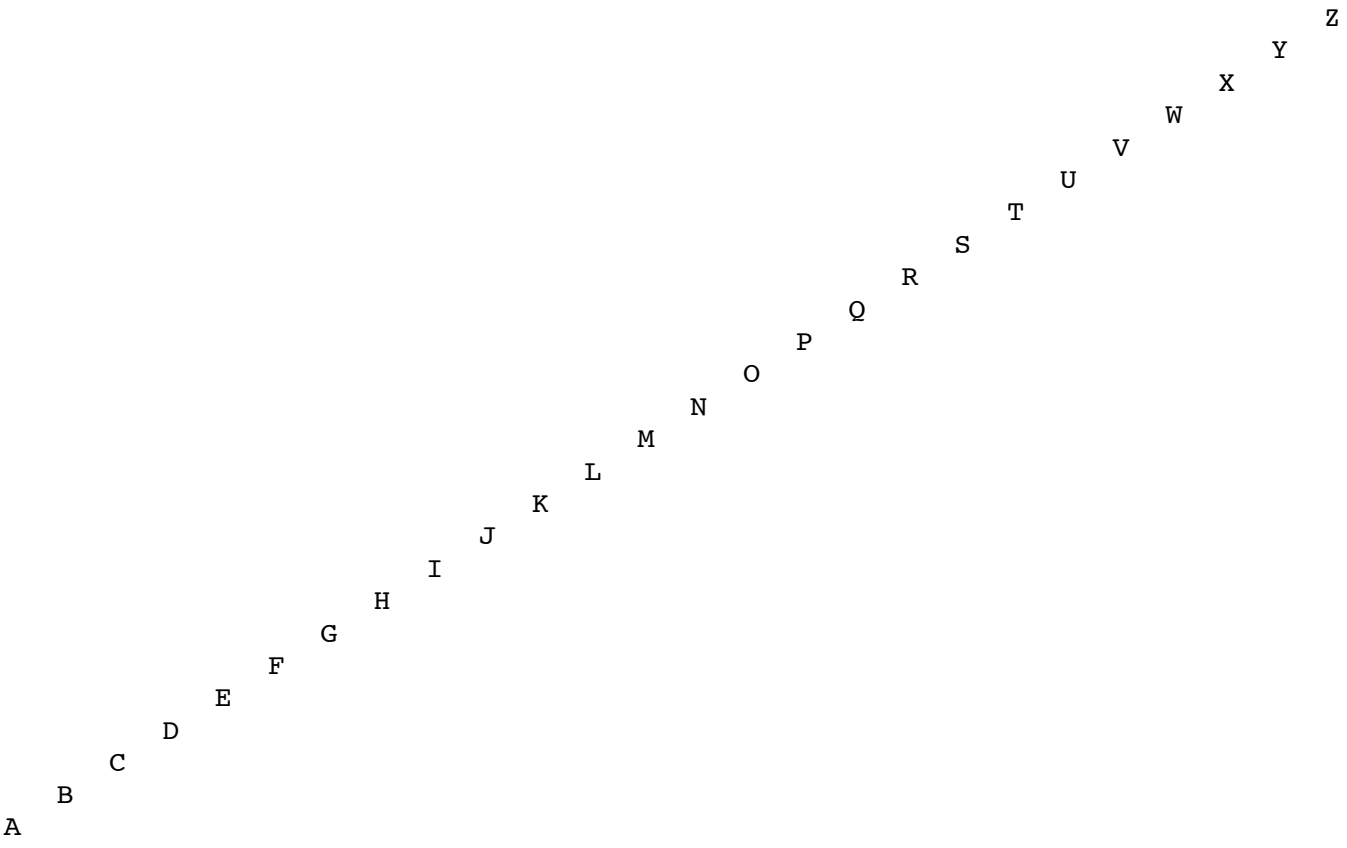
```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
9876543210
MFCJABDEHGIKLTPNORSQWUVYXZ
SPHINCTERSAYSWHAT
524137968
MATLSO
FTERFO
EYBLEIF
LYBWIOL
SYGTHO
FPNOEDESO
LLTDREOI
```

Each line contains the input for a single tree. The input for a tree is read from left-to-right and is inserted into the tree one character at a time. After you have created the tree for an input, print out its *preorder*, *inorder*, *postorder*, and *reverse inorder* traversals. A reverse inorder traversal is simply going to the right, visiting the node and then going to the left (i.e., RVL). Finally, print the tree out sideways as illustrated below. Indent three (3) spaces in between levels. Make sure to read the input from `stdin` so that I can execute your program as follows:

```
./treefun < input
```

Here is my output for the first two lines of input:

```
Preorder: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Inorder: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Postorder: ZYXWVUTSRQPONMLKJIHGFEDCBA
Reverse Inorder: ZYXWVUTSRQPONMLKJIHGFEDCBA
```




```
Preorder: 9876543210
Inorder: 0123456789
Postorder: 0123456789
Reverse Inorder: 9876543210
9
   8
      7
         6
            5
               4
                  3
                     2
                        1
                           0
```