CSE100 Algorithm Design and Analysis
Spring semester 2013

Lab #7
Miguel Á. Carreira-Perpiñán

Binary Search Trees

Description You have to implement standard operations on **binary search trees**, specifically insertion and deletion. See section 12.3 in the textbook. A sample class structure, with empty functions, is given in the `BST.zip` file. You are allowed to use the given class structure or create your own. In this assignment the keys are *strings* that can be stored using the standard C++ string type.

Questions and input structure

1. **Inserting**. Create an empty binary search tree. Then, read the keys from the keyboard and insert them into the tree. The input terminates when the key `ENDINSERT` is read (such a key must not be inserted in the tree). At that point, print the content of the tree to the screen. To print the content of the tree, use a postorder traversal (section 12.1 in the textbook), an example of which is provided in the file `BST.cpp`. It is safe to assume that all the keys are distinct.
   Testcase number to use with `grade_me`: 7.

2. **Deleting**. The second question is about deleting keys from the tree. First, you have to build the tree from scratch as you did for the first problem (reuse the same code). After you encounter the key `ENDINSERT` you stop building the tree (again, `ENDINSERT` must not be inserted). Then, you read keys to be deleted. Read each key and delete it. Stop deleting keys when you read the key `ENDDELETE`. At that point, print the content of the tree to the screen using the postorder traversal. Be aware that it could be the case that some of the keys that you are requested to delete are not in the tree.
   Testcase number to use with `grade_me`: 8.

Examples of input and output

Input for Test Case 7

```
i
t
y
h
j
ENDINSERT
```

Output for Test Case 7

```
h
j
y
t
i
```

Input for Test Case 8

```
i
t
y
h
j
ENDINSERT
t
h
p
ENDDELETE
```

Output for Test Case 8

```
j
y
i
```

Your solutions Before leaving the lab, submit a zipped tar archive of your program through the assignments page of UCMCROPS. Please use your UCMNetID as the filename for the zipped tar archive. Be careful since UCMCROPS strictly enforces the assignment deadlines (deadlines will be every lab date at either 4:20pm or 7:20pm depending on your lab session.).