DS Project 2 - Crazy wizard’s magic totem

1. Description (very important):

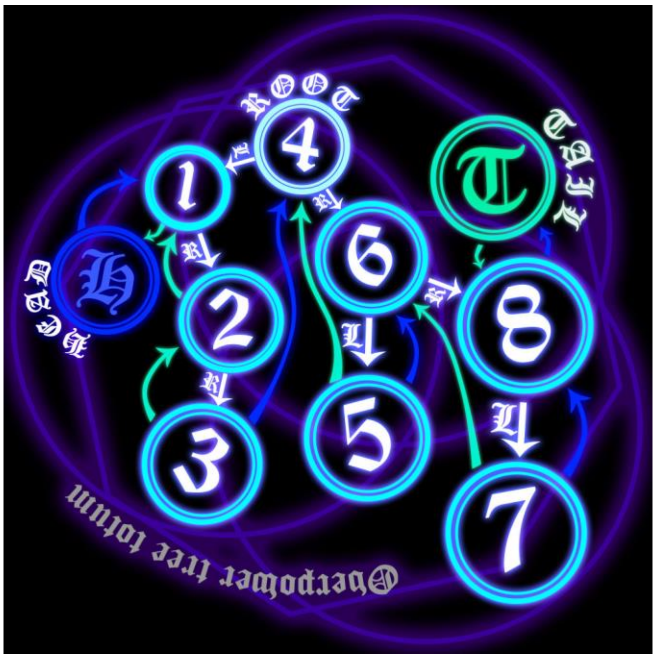
Gandalf was once a hardworking programmer. However, after he failed to confess his love to the girl he loved, his temper had changed drastically. He became a cynical and wicked person. He locked himself in the lab and conducted dark sorcery every day and refused to communicate with anyone outside. People said he has become a “Crazy wizard”.

TAs are very cute and nice.
I love to study DS happily.
Circular totum, circular myth
Circular totum

One day, Gandalf found a book that is about magic totem in a forbidden shelf in the lab. He found out that the magic totem has many different types and each type has its own unique features. For example, the circular totem has the power to create strong plasma shield that can protect the target from any type of damage. It is said that all of the Egyptian Pyramids can be preserved is because there are circular totem set in the heart of these Pyramids. On the other hand, the hexagonal totem has overwhelming power to spread deadly, incurable disease. They said that the Black Death in Europe was actually accidently caused by a negligent alchemist who underestimated the power of the hexagonal totem. After pages of pages, a beautiful magic totem finally caught Gandalf’s eyes.

Can girls go to the movies with me? You can get five points for bonus. That's true! You can contact me on my facebook.
Death will rule the world!!!
This totem is dangerous!!!!!!
Hexagonal totem

This beautiful magic totem is called “**The overpower tree totem**”. The totem itself has the power to transform a man into an adorable **mahou shoujo**, which means the magical girl in Japanese. Since Gandalf never had a girlfriend, he decided to do a total transformation on himself to become his own girlfriend (Yes, I know this sounds very sick. But everyone knows Gandalf is a weird person, right? Try to be nice with your classmates to prevent them from using this sick totem).



So, the next problem is, how to build **the overpower tree totem**? First, it has many nodes, and each node has its own number. These nodes will arrange in a specific order into a **binary search tree**. In the other words, the first entered node will become the root node, and the later will compare its number to decide its position. The smaller one will go to the left subtree; and the bigger one will go to the right subtree. And then, each node with less than two children will make its empty edge into a “thread”, if the left edge has no child, it would connect to the previous node in the in-order traversal order; if the right edge has no child, it would connect to the next node in the in-order traversal order. Finally, there are two special nodes, named “the head node” and “the tail node”. The head node would connect to the node which has the smallest number; and the tail node would connect to the biggest one.

If Gandalf walked through the totem from the head node to the tail node in in-order traversal order, he would get a sequence which is sorted from the smallest number to the biggest number, and he would become a cute **mahou shoujo**. If he walked through from the other side, he would change himself back to the original.

Now here’s the problem: Gandalf has many nodes with numbers, and the method to build **the overpower tree totem**, which has the special ability to add node and delete node. Can you help him build the totem correctly? He will be very appreciating if you help him!!

2. Input/Output:

In this project you have to implement how to build the overpower tree totem (threaded binary search tree) with n nodes (n<100). There are two files attached in this project you can use:

1) **Source.cpp**: The file to handle test data's input, you SHOULD NOT change anything in the file.
2) **op_tree_totum.h**: The file to implement the structure of the overpower tree totem, you have to fill in four functions in this file:

a. insertion: If the input file command is "I [number]", it would call this function. Insert the node with the number into the tree. It have to be the binary search tree order.

b. deletion: If the input file command is "D [number]", it would call this function. Delete the node with the same number. If you don't found any node with the same number, just ignore it.

c. in-order run: If the input file command is "C", it would call this function. Walk through the tree from the head node to the tail node in in-order traversal and output the nodes. If you succeed, you will get a sorted sequence from small to large. Each numbers are interval with a space and end with a line break. You should not use any stack or recursion method to implement it.

d. reverse-order run: If the input file command is "B", it would call this function. Same as in-order run, but walking from the tail node back to the head node.

You have to run the project in workstation. Here's the testing command:

```
> g++ Source.cpp
> ./a.out [input filename]
```

3.Sample testing data:

Input file(test1.txt):

```
I 1
I 0
I 5
I 4
D 4
D 1
I 6
```

```
I 3
I 2
D 0
C
B
```

Workstation command:

```
> g++ Source.cpp
> ./a.out test1.txt
```

Output (print on screen):

```
Change! Change myself into a cute mahou shoujo!!
The path: 2 3 5 6

Back! Back to the original life!!
The reverse path: 6 5 3 2
```

4. Requirements

Program

I. You need to turn in the codes. “Source.cpp” and “op_tree_totum.h”

II. Using standard output to print out your results on the screen and record them in your report.

III. Your program must be readable (Ex. Comments, variable names, function names)

IV. **Do not use any STL function to build the structure.**

V. In function in-order run and reverse-order run, do not use stack or recursion to implement them.

Report

I. Name the file”hw2_StudentID.pdf”, Ex: hw2_0316000.pdf)

II. Describe your implementation. (Ex: algorithm, program executing process)

III. Sample testing Results: the execution result for sample testing (show it by taking screenshots)

IV. No more than 2 pages.

5. Grading policy

I. Programming (80%)
Your code can run without crashing and giving correct results to the sample files in your report.
II. Report (20%)
III. Bonus (0%)
If you turn yourself into a cute mahou shoujo and come to DEMO, you'll get the bonus.

If there have any undesirable mistake to the requirement. Each mistake minus 10 points.

6. Submit (e3 will be closed on time)

Compress all your files (including your code(.h/.cpp) and report(.pdf)) Name your compressed file "**hw2_studentID.zip**". Upload your compressed file to e3.

Deadline: 2015.12.13, 23:59

No late upload.