CSCI2100B: Data Structures (Spring 2011)

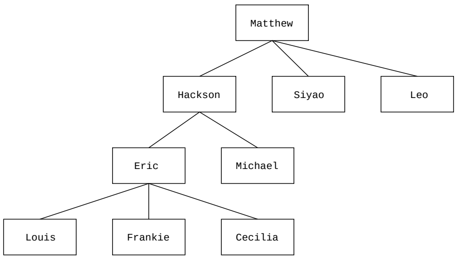
Assigned: *22 Mar 2011* Version 1.1 (Updated: 21/03 16:55) Due: *8 Apr 2011 23:59*

Programming Assignment 3: Who is the Boss?

Limits: Runtime - 1 secs • Memory - 64 MB • Submission - 20 times

Background

DataTech is a software development company that keeps its staff in a tree hierarchy. The biggest boss in the company is sitting at the root of the hierarchy and may hire any number of employees to work under his/her team. Any member of the company is then allowed to hire others to work under his/her own group. As a result, a hierarchy tree can be drawn to represent the relationship between all members of the company:

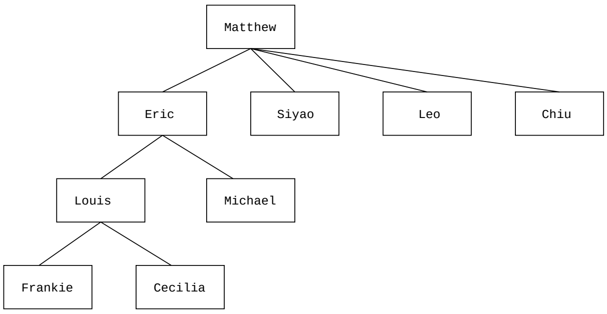


Matthew is the big boss of the company, and he has three team members: Hackson, Siyao and Leo. The tree is ordered and the member has an decreasing seniority when it goes from left to right. Therefore Hackson is the most senior member of the team.

Please note that some members can have *no* hiring like Michael or Leo.

When a member M hires a new staff S, S will join the team of M as the least senior member. For example, if Matthew hires Chiu, then Matthew's team has four members Hackson, Siyao, Leo and Chiu, in order of decreasing seniority.

At times, a member X may get fired by his/her boss. If X has hired no one, then X will be dropped from the company directly; otherwise, the most senior member Y from X's team will be promoted to fill the vacancy. The promoted member Y also inherits X's seniority. The promotion process will cascade down the tree until a member working a 1-man team gets promoted. For example, if Hackson is fired, then Eric will replace Hackson and Louis will replace Eric. The following figure shows the hierarchy after Chiu is hired by Matthew and Hackson is fired by Matthew:



Problem

Write a program to keep track of the changes in the company hierarchy and print the hierarchy tree.

Input

The input begins with the name of the biggest boss. All names in the input contain of 2 and 25 characters (lower and upper letters only, no spaces). The names are case-sensitive (e.g. Matthew and matthew are different). You can assume there are at most 1000 different names in a input file.

It is then followed by a sequence of commands, representing the orders to your program:

- *existing member* `hires` *new member*
- `fire` *existing member*
- `print`

Existing member is the name of a certain member that has been entered to the system (that is already in the tree) while *new member* is the name of a person who is not a member yet (that does not appear in the tree). Please note that the commands can appear in any order and in any number of times.

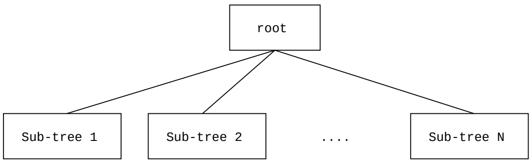
Sample Input:

```
Matthew
Matthew␣hires␣Hackson
Matthew␣hires␣Siyao
Matthew␣hires␣Leo
Hackson␣hires␣Eric
Hackson␣hires␣Michael
Eric␣hires␣Louis
Eric␣hires␣Frankie
Eric␣hires␣Cecilia
print
Matthew␣hires␣Chiu
fire␣Hackson
print
fire␣Leo
fire␣Louis
print
```

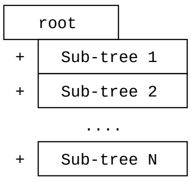
Output

Print the current hierarchy of the company whenever you receive a `print` command.
The tree should be printed exactly as the following format:

- A line contains exactly one name.
- The first line contains the biggest boss's name.
- Then output the tree with N sub-trees which has the form



in the textual form as:



Print a line of exactly 60 hyphens after the tree.

Sample output:

```
Matthew
+Hackson
++Eric
+++Louis
+++Frankie
+++Cecilia
++Michael
+Siyao
+Leo
------------------------------------------------------------
Matthew
+Eric
++Louis
+++Frankie
+++Cecilia
++Michael
+Siyao
+Leo
+Chiu
------------------------------------------------------------
Matthew
+Eric
++Frankie
+++Cecilia
++Michael
+Siyao
+Chiu
------------------------------------------------------------
```

Test Cases & Scoring

Any reasonable (non-trivial) code that can be compiled successfully scores 2 marks.

Your program will be tested against a set of 4 test cases, namely:
1. The sample input.
2. `hires` and `print` only. Maximum 10 different names.
3. All commands. Maximum 100 different names.
4. All commands. Maximum 1000 different name.

Every test case scores 2 marks.

Therefore your program can score 10 marks in total.

Submission

You should write your program in a single C source file. Submit your program using your UNIX account, following the instructions below:

1. Copy/upload it to your CSE UNIX account.
2. SSH to any CSE UNIX workstation through SSH, compress the source code by
   ```
   gtar zcvf <sid>.tar.gz <file_name>.c
   ```
   where `<sid>` is your 10-digit student ID and `<file_name>` is the name of the source code you wish to submit. Do **NOT** include other files.
3. Submit it to our Judge System by
   ```
   uuencode <sid>.tar.gz <sid>.tar.gz | mailx -s "ASG<code> <sid>" csci2100b
   ```
   where `<sid>` is your 10-digit student ID and `<code>` is the assignment code you are trying to solve.
4. Upon successful submission, you will receive a submission receipt and the judge reply very soon. You should **KEEP** your submission receipt for future references.

- End -