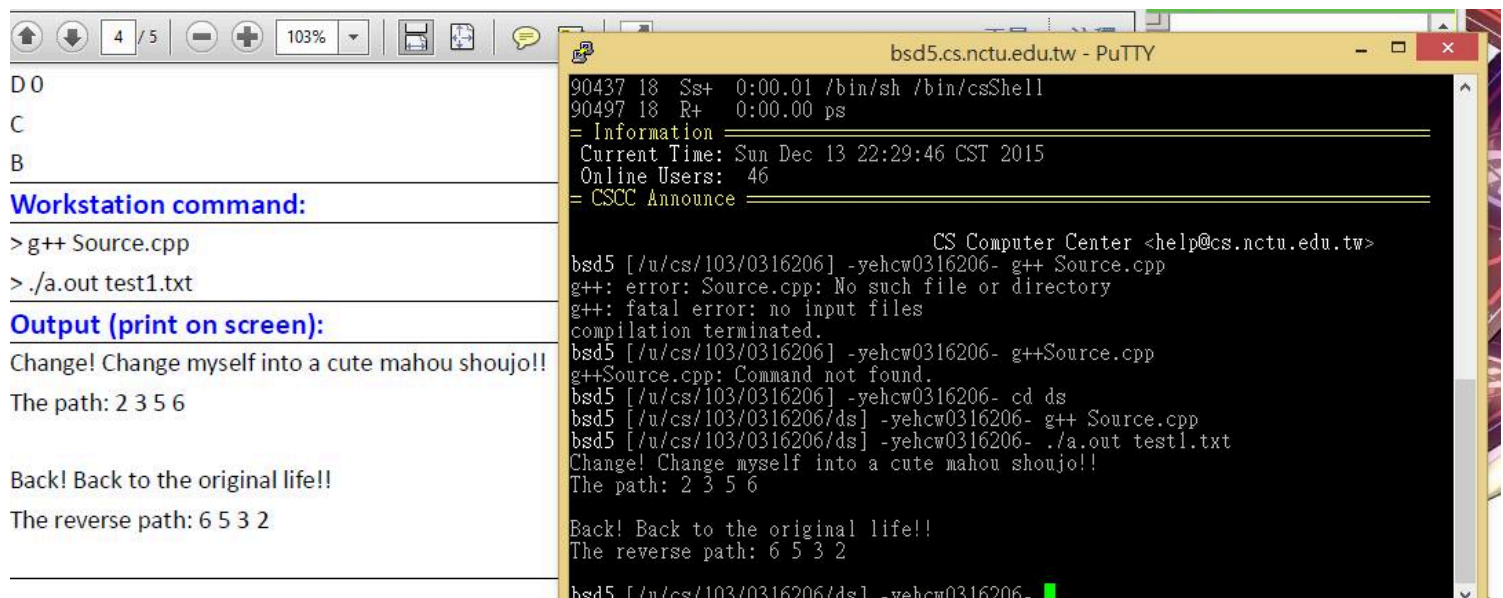


在 insertion 這個 function 我是先創一個新的 node 讓他裡面的 number = s，如果一開始是空的就讓 head 跟 tail 都指向這個新的 node，如果這個 s 比 head 的 number 還小或者比 tail 的 number 還大就代表他們是者科室的最大要不然就是最小直接讓 head 或 tail 指向他們，如果上述條件都不符合，那就開始找他要插入哪一個位子，而 temp 就是他要插入的那個位子的上一個 node，如果 s 比 temp 的 number 大，就差在 temp 的右邊，反之則插在左邊。

在 deletion 這個 function，一開始如果 root 是空的就什麼事都不做，直接 return，再來如果要刪掉的數字是最小的或最大的，就一定是 head 或是 tail 的 number，因為要考慮到 head 或是 tail 的串接，所以我另外分出來先討論，再來就是要找到到底要刪掉哪一個 node，再來就是刪 node，分四種情況，第一種是要刪的那個 node 左右都有 node，第二種是要刪的 node 左右都沒有 node，第三種是左邊有右邊沒有，第四種是左邊沒有右邊有。

在 inorder_run 這個 function 一開始如果 head->right == tail，也就是說這個 tree 是空的，所以就直接 return，再來因為他是 inorder 所以再創一個新的 node 讓他指向 head，從最小的開始走，一直走到 tail，因為我們一開始有用引線把它們串接起來了，所以如果這個 node 的右邊有東西的話我們會先從左邊 node 的最小開始印出來（所以是最左邊的 number），再印這個 node 的右邊，再來才印 node，依照這樣的方式一直印出整棵樹。

在 reverseorder_run 這個 function 因為他是 inorder 整個倒過來印出來，所以他的頭就變成 tail 了，而和 inorder 一樣，如果 head->right == tail 就是代表這棵樹是空的，所以就直接 return，再來也是創一個新的 node 指向 tail，從最大的開始走，一直走到 head，也是因為我們一開始就有用引線把它們串接起來了，所以如果這個 node 的左邊有東西的話我們會先從右邊 node 的最大開始印出來（所以是最右邊的 number），再印這個 node 的左邊，再來才印 node，依照這樣的方式一直印出整棵樹。



```
bsd5.cs.nctu.edu.tw - PuTTY
90437 18 Ss+ 0:00.01 /bin/sh /bin/csShell
90497 18 R+ 0:00.00 ps
= Information
Current Time: Sun Dec 13 22:29:46 CST 2015
Online Users: 46
= CSCC Announce

CS Computer Center <help@cs.nctu.edu.tw>
bsd5 [/u/cs/103/0316206] -yehcw0316206- g++ Source.cpp
g++: error: Source.cpp: No such file or directory
g++: fatal error: no input files
compilation terminated.
bsd5 [/u/cs/103/0316206] -yehcw0316206- g++Source.cpp
g++Source.cpp: Command not found.
bsd5 [/u/cs/103/0316206] -yehcw0316206- cd ds
bsd5 [/u/cs/103/0316206/ds] -yehcw0316206- g++ Source.cpp
bsd5 [/u/cs/103/0316206/ds] -yehcw0316206- ./a.out test1.txt
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

bsd5 [/u/cs/103/0316206/ds] -yehcw0316206-
```