Assignment #4 Binary Search Trees Updated Version To Follow Later Today

Dear Class. I am not above stealing. This assignment looks muuuch easier than my original assignment. This is an updated version of the assignment which includes some skeleton files that should point you in the right direction (i.e. the node class I used and the main file). If you are working through it and you see a mistake, please tell me about it.
-Ashok

In this assignment, you are required to implement the binary search tree (BST) as a abstract data type that can hold any TYPE as the data field (think template class). You should support the following methods in your implementation:

find(k): return an entry with key k if it exists, null otherwise.
insert (k): insert an entry with key k.
remove(k): remove all entries with key k.
size(): return the number of elements in the tree.
traverse(): get the string of the in-order traversal of the tree. YOUR FORMAT SHOULD LOOK EXACTLY LIKE THE OUTPUT GIVEN BELOW!

The files you will provide for this are:
bst.h where you will put your class definition AND all of your functions (you can put your functions after the class definition or use the backdoor trick we learned in class.
You are also required to provide a test program to test the required methods. Finally a README must be provided to explain the files and how your program works.

The actions to your program will be provided in a line by line and will be terminated by a line containing “end”. Each line of input starts with a character C, and may be followed by an integer I. C can be any of the following:

• "i" which is followed by I. It requires the program to insert I to the tree;
• "r" which is followed by I. It requires the program to remove I from the tree;
• “f” which is followed by I. It requires the program to find if the tree contains I or not;
• “s” which is not followed by I. It requires the program to return the tree size;
• “t” which is not followed by I. It requires the program to traverse the tree using in-order traversal.
• “q” which is not followed by I will quit the program.

After each input, your program should print an input block. The block should start by printing the command surrounded by three asterisks from left and right. There should also by a single space between the command and either set of asterisks.
If the input is removal or insertion, then from the next line the program should print each level of the tree from head to bottom consecutively. Printing row j of the tree

starts with "Rj" and then each element from left to right should be printed on a separate line. Each row should be followed with 9 dashes, i.e. "---------". There should be a blank line after each block. If the input is searching for a key, it should print “Found” if it was contained in the tree or “Not found” otherwise. If the input is to print the size, then simply print the size as an integer. If the input is to print the in-order traversal, then print it on one line with the keys separated by one space.Sample run (input in green):

```
i 20
*** i 20 ***
R0
20
---------
i 50
*** i 50 ***
R0
20
---------
R1
50
---------
i 30
*** i 30 ***
R0
20
---------
R1
50
---------
R2
30
---------
f 50
*** f 50 ***
Found
i 70
*** i 70 ***
R0
20
---------
R1
50
---------
R2
30
70
---------
f 100
```

*** f 100 ***
Not foundi 5
*** i 5 ***
R0
20

R1
5
50

R2
30
70
---
i 100
*** i 100 ***
R0
20

R1
5
50

R2
30
70

R3
100
---
r 50
*** r 50 ***
R0
20

R1
5
70

R2
30
100
---
s
*** s ***
5

i 110*** i 110 ***
R0
20

R1
5
70

R2
30
100

R3
110
---
t
*** t ***
5 20 30 70 100 110
r 30
*** r 30 ***
R0
20

R1
5
70

R2
100

R3
110
---
end