**Assignment 02**

**Time: 90 minutes**

In this question you should complete some methods in **BSTree.java** file.

The class **User** with 3 data members: name, age, and id are given and you do not need to edit it. The **BSTree** class is a binary search tree of **User** objects. The variable **id is the key of the tree (thus it must be unique)**. The following methods should be completed:

* **(2 marks)** Void **insert(xName, xAge,xId)**- insert new User with name=xName, age=xAge, and id=xId to the tree **(age>0)**
* Void **f1()**- You **do not** need to edit this function. Your task is to complete the insert(…) function above only. Output in the file **f1.txt** must be the following:

Content of the file f1.txt:

(A,50,8) (B,10,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,25,4) (G,7,6) (J,19,3) (I,20,7)

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

* **(2 marks)** Void **f2()**- Perform the post-order traversal from the root but display to file f2.txt nodes with age<25 only. Output in the file **f2.txt** must be the following:

Content of the file f2.txt:

(A,50,8) (B,10,2) (C,4,5) (D,25,4) (J,19,3) (G,7,6) (I,20,7) (E,30,10) (F,14,9) (H,34,11)

(J,19,3) (I,20,7) (G,7,6) (C,4,5) (B,10,2) (F,14,9)

* **(2 marks)** Void **f3()** – add age of all nodes have only one child to 3 (age=age+3). Output in the file **f3.txt** must be the following:

Content of the file f3.txt:

(A,50,8) (B,10,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,25,4) (G,7,6) (J,19,3) (I,20,7)

(A,50,8) (B,13,2) (E,30,10) (C,4,5) (F,14,9) (H,34,11) (D,28,4) (G,10,6) (J,19,3) (I,20,7)

* **(2 marks)** Void **f4()**- Calculate height of last node in postOrder traversal from the root. Output in the file **f4.txt** must be the following:

Content of the file f4.txt:

(A,50,8) (B,10,2) (C,4,5) (D,25,4) (J,19,3) (G,7,6) (I,20,7) (E,30,10) (F,14,9) (H,34,11)

5

* **(1 marks)** Void **f5()**- Reset age of all nodes locate in left branch of 1st node in preOrder traversal from root to zero(age=0). Content of the file f5.txt:

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

(B,0,2) (J,0,3) (D,0,4) (C,0,5) (G,0,6) (I,0,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

* **(1 marks)** Void **f6()**- Find the node which has biggest id (right most) node in the tree. The output of **f6.txt** is as follows:

(B,10,2) (J,19,3) (D,25,4) (C,4,5) (G,7,6) (I,20,7) (A,50,8) (F,14,9) (E,30,10) (H,34,11)

(H,34,11)

Before submission:

* **Clean and Build** Project (Shift+F11)
* Rename the folder **dist** to **run**. (If the folder **run** already exists, delete it before renaming).

Submission:

* **Compress folder Q2 to SE123455\_HoVaTen.rar or SE123455\_HoVaTen.zip**
* **Submit SE123455\_HoVaTen.rar or SE123455\_HoVaTen.zip to lms**

----- End -----