Data Structures and Algorithms Lab

07. Trees Part II

Subject Code: 17ECSP201 **Lab No:** 07 **Semester:** III

Date: 07 Oct, 2017 **Batch:** C1&C2

Question: Computer Representation of a Binary Search Tree

Objective: Usage of list representation to implement a BST and its operations

----

Implement and add the following functions to the BST code that was implemented in the lab number 07. You are supposed to implement as many functions as you can in the given time slot.

1. Print the root of the tree

2. Generate a random number and insert it into the tree

3. Count and print the total number of interior nodes (nodes other than root and leaf)

4. Print the tree items in decreasing order

5. Print all the even data items from the tree

6. Print all the odd data items from the tree

7. Implement a function to print the in-order predecessor for the given node

8. Implement the following Tree search algorithm

```
TREE-SEARCH (x, k)
If x = NULL or k = key[x]
    then return x
If k < key[x]
   then return TREE-SEARCH(left[x], k)
   else return TREE-SEARCH(right[x], k)
```

9. The following procedure returns a minimum element in the subtree rooted at a given node x. Implement the function.

```
TREE-MINIMUM(x)
while left[x] != NULL
   do x ← left[x]
return x
```

10. The following procedure returns a maximum element in the subtree rooted at a given node x. Implement the function.

```
TREE-MAXIMUM(x)
while right[x] != NULL
   do x ← right[x]
return x
```

11. Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

12. User will enter two keys which are present in the tree. Find the distance between the given two keys. How many edges needs to be traversed to start from one node and reach another? Print the number of edges as output.

13. Print all the numbers present in-between in the tree from the user supplied two ranges K1 and K2.

14. Increment every node data in the tree by 1

15. Check if the tree has duplicate nodes. User will supply the key for which the duplicates will be searched. Print the number of duplicates present in the tree.

**** Happy Coding ****