Dive Jagdishchandra Anantrao

220960920025

Q1. Sort elements using insertion sort.

**package** com.dsLabExam;

**import** java.util.Arrays;

**public** **class** InsertionSort {

**void** insertionSort(**int** array[]) {

**int** size = array.length;

**for** (**int** step = 1; step < size; step++) {

**int** key = array[step];

**int** j = step - 1;

**while** (j >= 0 && key < array[j]) {

array[j + 1] = array[j];

--j;

}

array[j + 1] = key;

}

}

**public** **static** **void** main(String args[]) {

**int**[] data = { 9, 5, 1, 4, 3 };

InsertionSort is = **new** InsertionSort();

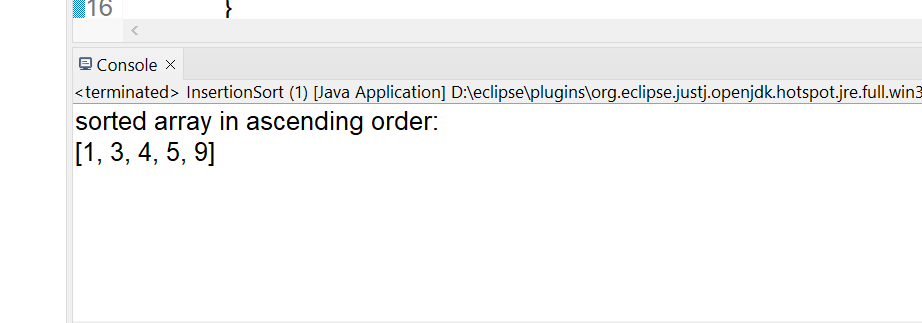
is.insertionSort(data);

System.***out***.println("array in ascending order: ");

System.***out***.println(Arrays.*toString*(data));

}

}



2.Implement breadth first tree traversal

**package** com.dsLabExam;

**class** Node {

**int** data;

Node left, right;

**public** Node(**int** item)

{

data = item;

left = right = **null**;

}

}

**package** com.dsLabExam;

**class** Main {

Node root;

**public** Main() { root = **null**; }

**void** LevelOrder()

{

**int** h = height(root);

**for** (**int** i=1; i<=h; i++)

CurrentLevel(root, i);

}

**int** height(Node root) {

**if** (root == **null**)

**return** 0;

**else** {

**int** lheight = height(root.left);

**int** rheight = height(root.right);

**if** (lheight > rheight)

**return**(lheight+1);

**else** **return**(rheight+1);

}

}

**void** CurrentLevel (Node root ,**int** level) {

**if** (root == **null**){

**return**;

}

**if** (level == 1){

System.***out***.print(root.data + " ");

}

**else** **if** (level > 1) {

CurrentLevel(root.left, level-1);

CurrentLevel(root.right, level-1);

}

}

**public** **static** **void** main(String args[])

{

Main tree = **new** Main();

tree.root = **new** Node(1);

tree.root.left = **new** Node(2);

tree.root.right = **new** Node(3);

tree.root.left.left = **new** Node(4);

tree.root.left.right = **new** Node(5);

tree.LevelOrder();

}

}

