**package** Question12;

**public** **class** CountSmaller {

**private** **static** Node *root* = **null**;

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

**int** [] A = {8,7,6,5,4,3,2,1};

*root* = *buildBST*(*root*, A);

System.***out***.println("Printing inorder - data and count of smaller elements to the right: ");

*printInorder*(*root*);

System.***out***.println();

*root*=**null**;

**int** [] B = {10, 6, 15, 20, 30, 5, 7};

*root* = *buildBST*(*root*, B);

System.***out***.println("Printing inorder - data and count of smaller elements to the right: ");

*printInorder*(*root*);

}

**public** **static** **void** printInorder(Node node) {

**if** (**null** == node) { **return** ; }

**if** (**null** != node.left) {

*printInorder*(node.left);

}

System.***out***.println(node.data + "::" + node.counter);

**if** (**null** != node.right) {

*printInorder*(node.right);

}

}

**public** **static** Node buildBST(Node node, **int** [] A) {

**if** (**null** == A) { **return** **null**; }

**int** len = A.length;

**for** (**int** i=len-1; i>=0; i--) {

node = *insertIntoBST*(node, A[i],0);

}

**return** node;

}

**public** **static** Node insertIntoBST(Node node, **int** value, **int** leftSum) {

**if** (**null** == node) { **return** **new** Node(value); }

**if** (value <= node.data) {

++node.left\_count;

node.left = *insertIntoBST*(node.left, value,leftSum);

**if** (node.left.left == **null** && node.left.right == **null**)

node.left.counter = leftSum;

} **else** {

node.right = *insertIntoBST*(node.right, value, node.left\_count+leftSum);

**if** (node.right.right == **null** && node.right.left == **null**)

node.right.counter = node.left\_count+node.counter;

}

**return** node;

}

}

**class** Node {

Integer data;

Integer left\_count;

Integer counter;

Node left;

Node right;

**public** Node(Integer data) {

**this**.data = data;

**this**.left\_count = 1;

**this**.counter = 0;

**this**.left = **null**; **this**.right = **null**;

}

}