Q) 3. Write a java program to determine whether a given binary tree is a BSTor not.

```
package com.java.Trees;

public class BST {
        static class node {
          int data;
          node left, right;
         }
         static node newNode(int data)
         {
          node Node = new node();
          Node.data = data;
          Node.left = Node.right = null;

          return Node;
         }

         static int maxValue(node Node)
         {
          if (Node == null) {
            return Integer.MIN_VALUE;
          }
          int value = Node.data;
          int leftMax = maxValue(Node.left);
          int rightMax = maxValue(Node.right);

          return Math.max(value, Math.max(leftMax, rightMax));
         }

         static int minValue(node Node)
         {
          if (Node == null) {
            return Integer.MAX_VALUE;
          }
          int value = Node.data;
          int leftMax = minValue (Node.left);
          int rightMax = minValue (Node.right);

          return Math.min(value, Math.min(leftMax, rightMax));
         }
```

```
static int isBST(node Node)
{
  if (Node == null) {
    return 1;
  }

  /* false if the max of the left is > than us */
  if (Node.left != null
      && maxValue(Node.left) > Node.data) {
    return 0;
  }


  if (Node.right != null
      && minValue (Node.right) < Node.data) {
    return 0;
  }


  if (isBST(Node.left) != 1
      || isBST(Node.right) != 1) {
    return 0;
  }


  return 1;
}

public static void main(String[] args)
{
  node root = newNode(1);
  root.left = newNode(2);
  root.right = newNode(3);

  root.left.left = newNode(4);
  root.left.right = newNode(5);

  // Function call
  if (isBST(root) == 1) {
    System.out.print("Is BST");
  }
  else {
    System.out.print("Not a BST");
  }
```

```
        }
      }
```

Output :

```
Not a BST
```