3 Name your Jupyter Notebook as:

`TASK3_<your name>_<centre number>_<index number>.ipynb`

A binary search tree is used to store 10 integer values between 0 and 999 (inclusive) in ascending numerical order.

The tree is implemented using Object-Oriented Programming (OOP).

The class `Tree` contains three properties:

- `left_pointer` points to the left subtree
- `right_pointer` points to the right subtree
- `data` is the data in the node.

The class `Tree` contains the following methods:

- a constructor to set the left pointer and right pointer to None, and the data to its parameter
- a recursive method to take the parameter and store it in the correct position in the tree
- a recursive method to use in-order traversal to output the data in the tree
- a recursive method to use post-order traversal to output the data in the tree.

For the sub-task, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
In [1]: #Task 3.1
        Program code
```

```
Output:
```

Task 3.1

Write program code to declare the class `Tree` and its constructor. [4]

Write the recursive method to insert a new node into the tree. [6]

Write the main program to:

- declare a new instance of `Tree`
- generate 10 unique random integer values between 0 and 999 (inclusive)
- store each unique value as a new node in the tree using your method. [5]

Write program code to:

- declare the method to output the in-order traversal of the binary tree
- declare the method to output the post-order traversal of the binary tree.

Call the in-order and post-order methods using your tree structure. [7]

Test your program and show the output from each traversal. [2]

Save your Jupyter Notebook for Task 3.