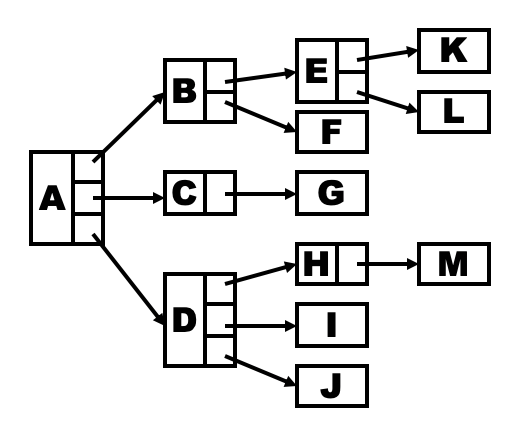
Lab 9 Report Template

**1.Given a tree, say in linked list, construct another tree using** **FirstChild-NextSibling representation.**

1.1 linked list tree:

The linked list tree is organized in this way:



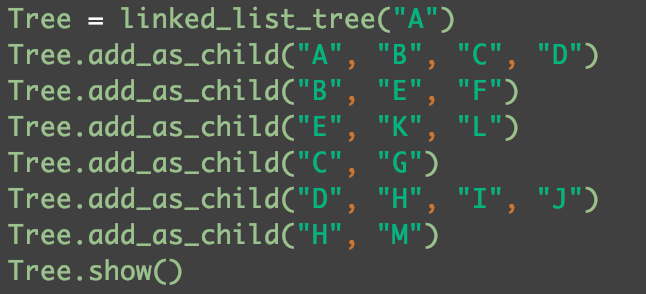
1.1.1 coding in Python:



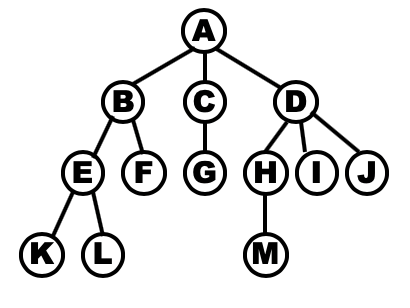
1.1.2 details:

Every node in the tree contain a value and a list called next, which contain the next nodes of this node. When I try to add some value into the tree, I find a special value and generate a node as one of the next nodes of this node. Use the method show, to get the preorder traversal of this tree.

1.1.3 test:



1.1.4 The tree:

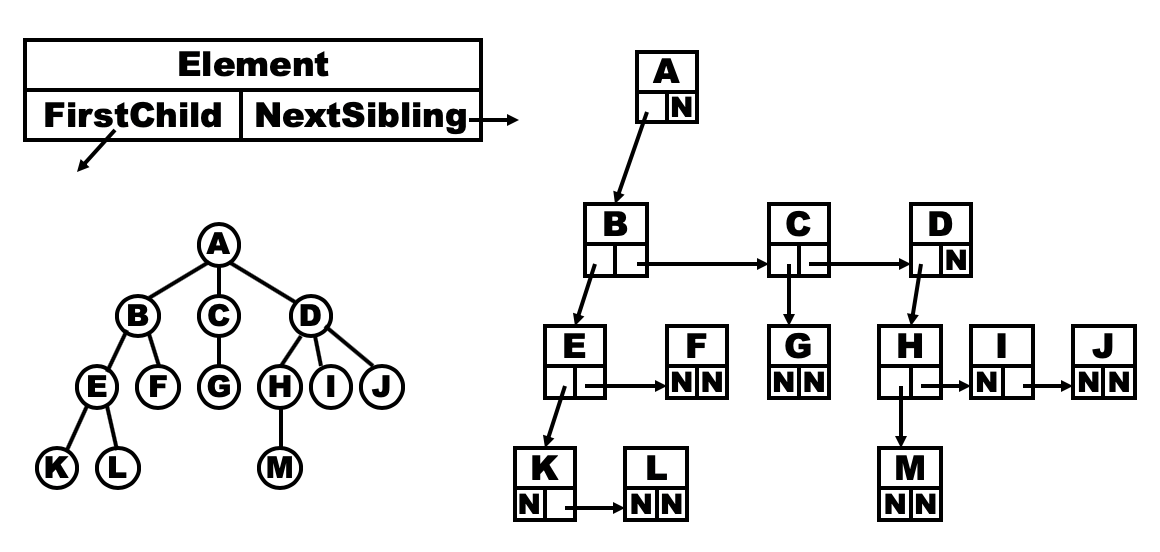


1.1.5 Output:

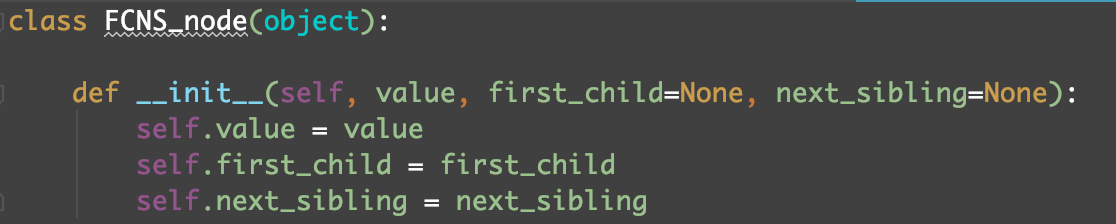


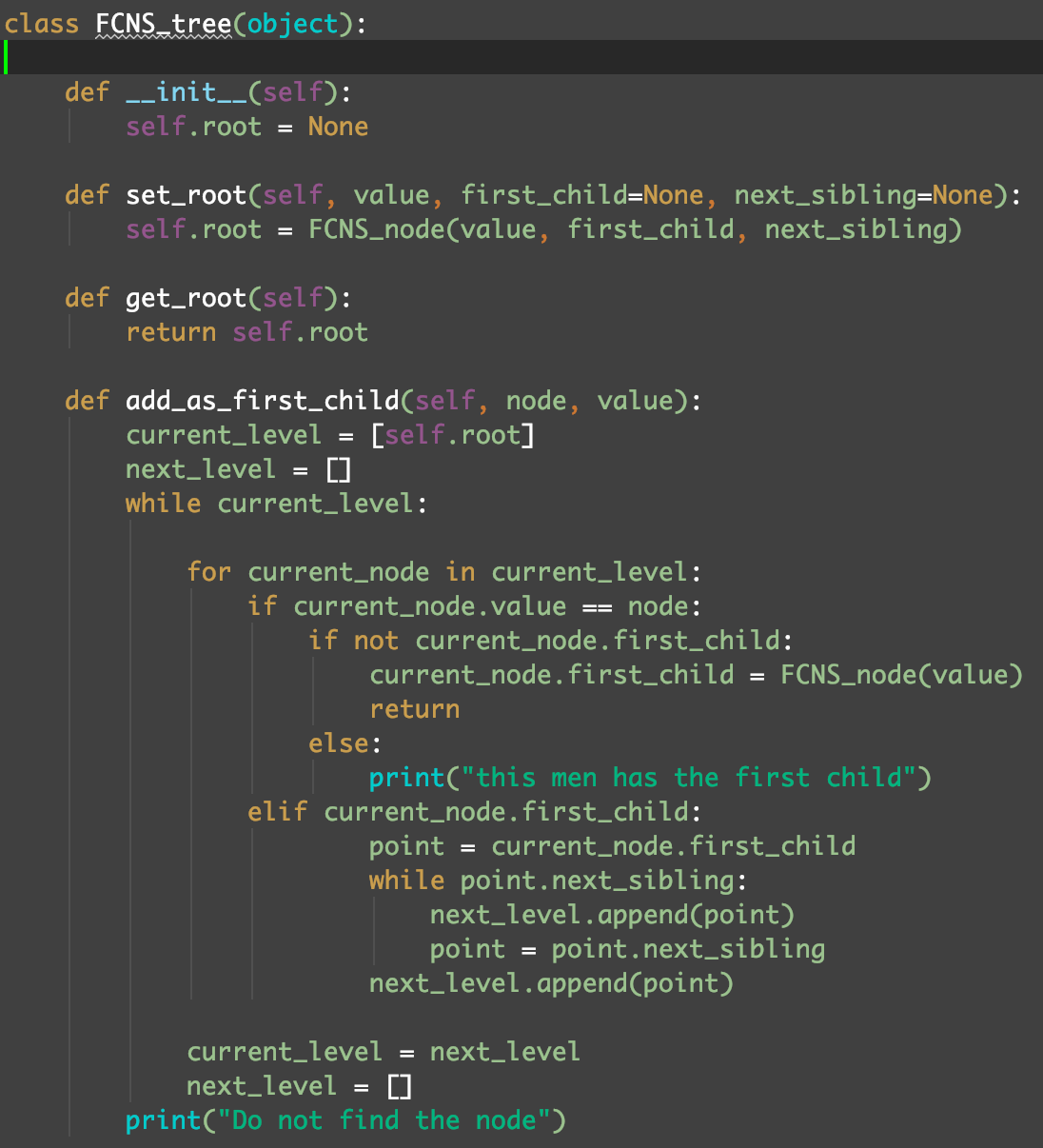
In fact, this is the preorder traversal of the tree.

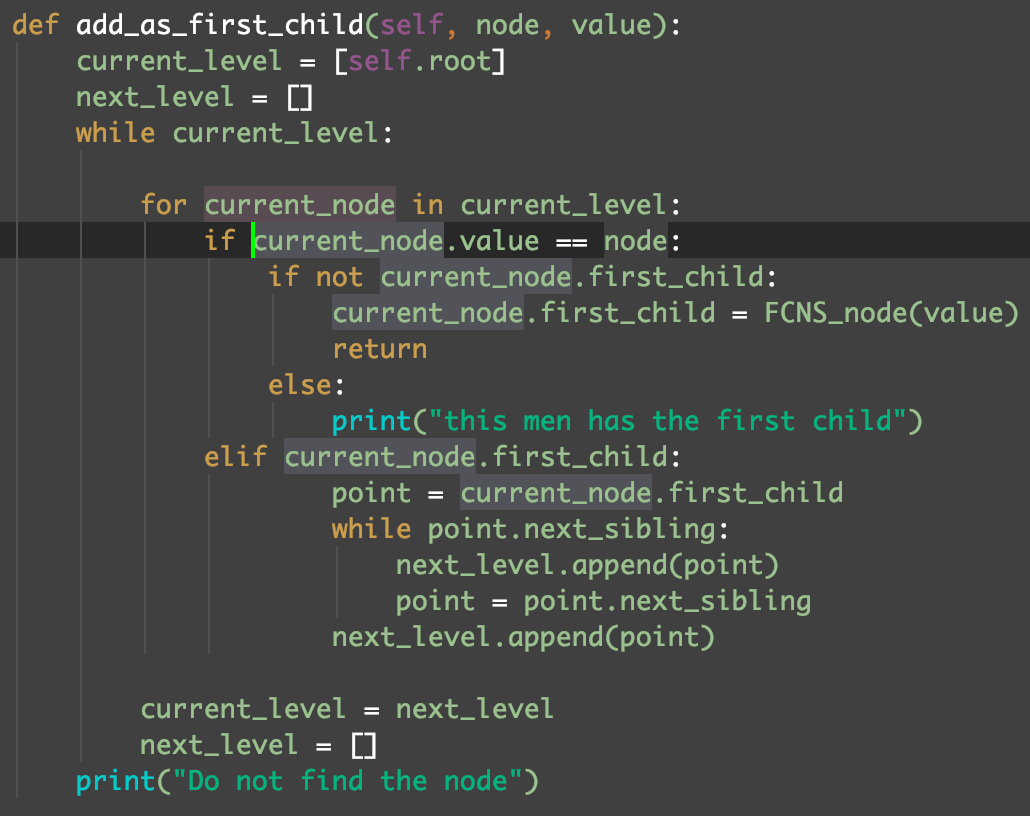
1.2 FirstChild-NextSibling tree:

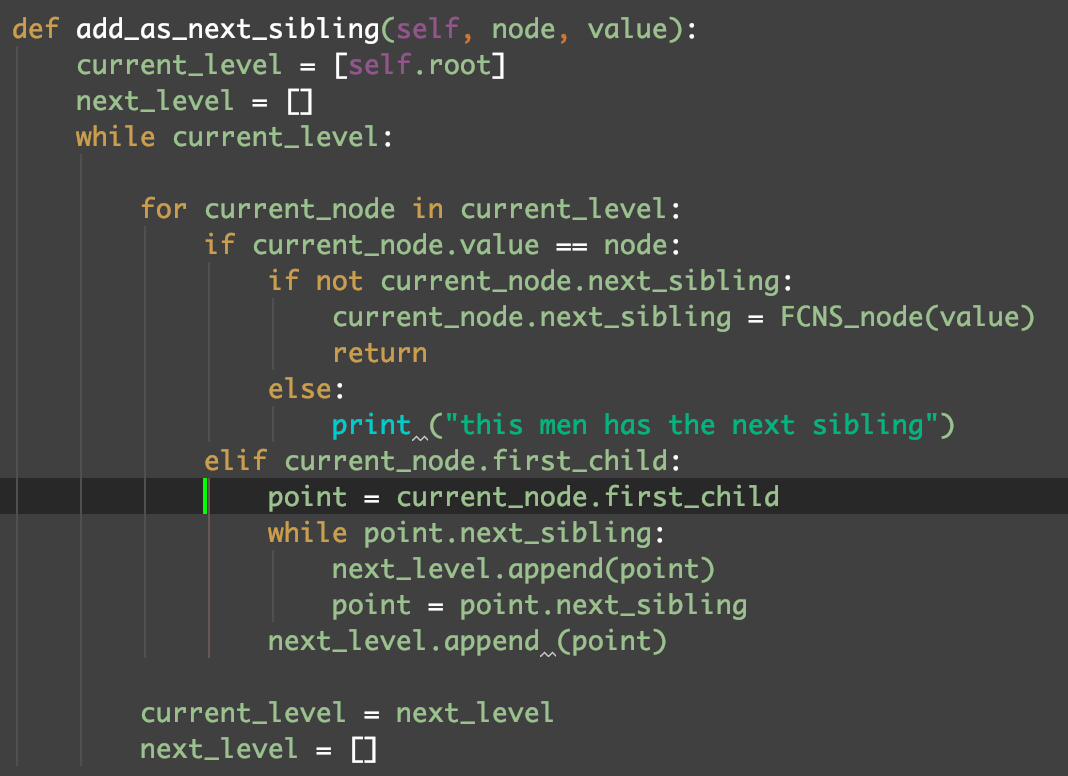


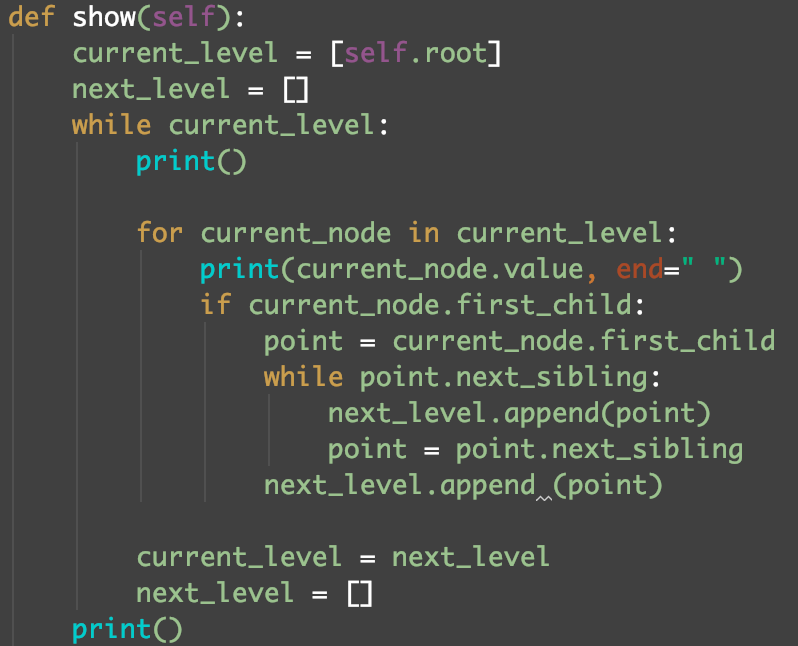
1.2.1: code in Python:



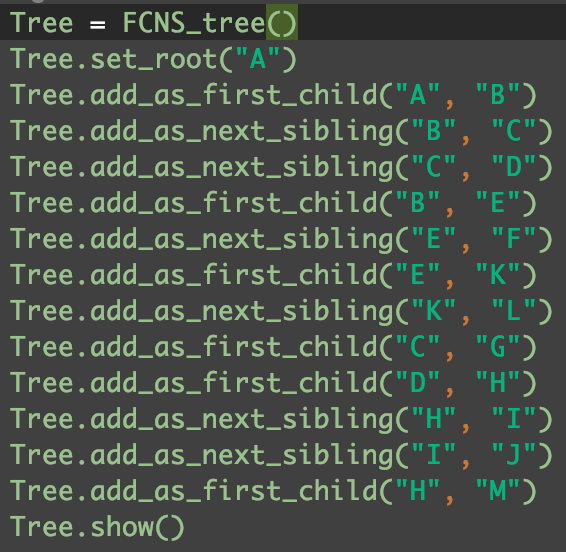






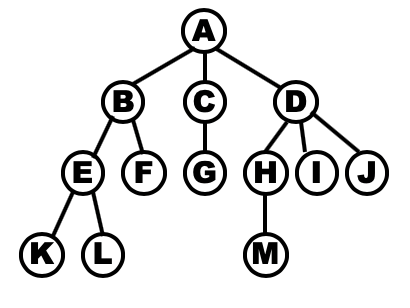


1.2.2

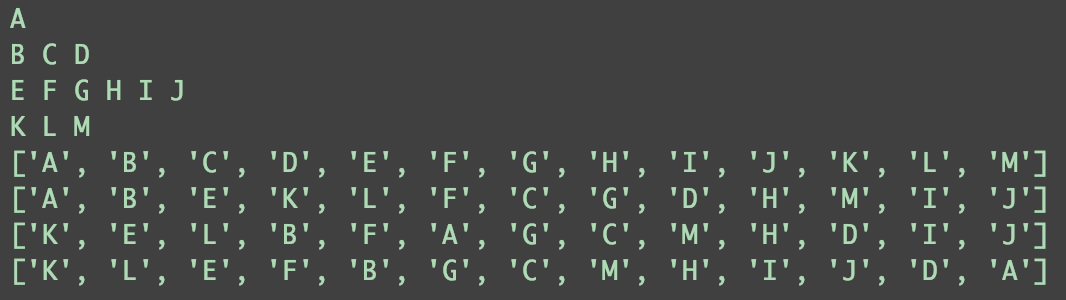
Every node in this tree has a first child and a next sibling, which all can be none. Every tree has a specific root node. There are two methods can be used to build a tree.

1.2.3 Test:

1.2.4 the tree:



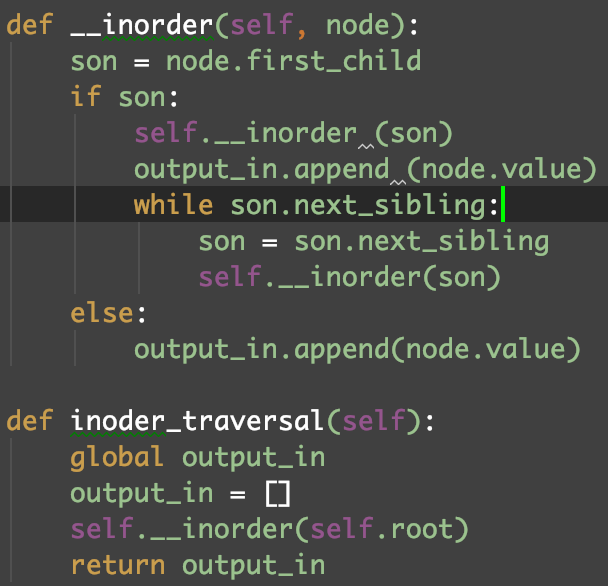
1.2.4 Output:

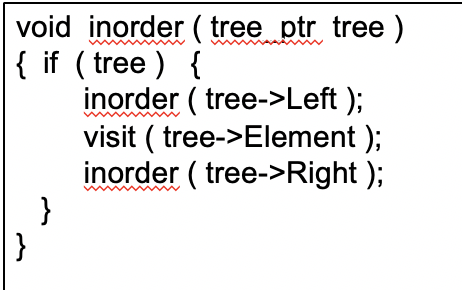


**2.Implement** **inorder, preoder, postorder and levelorder tree traversal algorithms.**

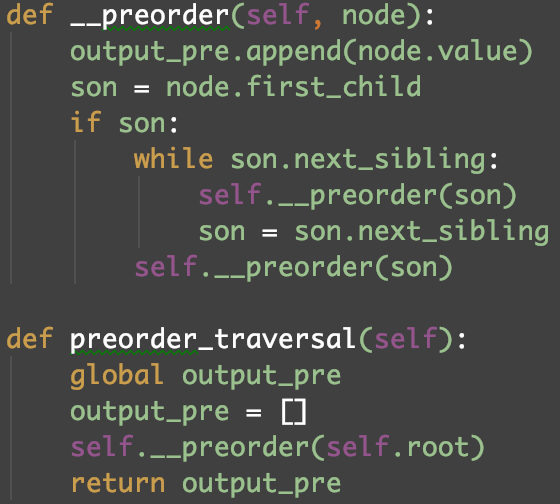
2.1: background:

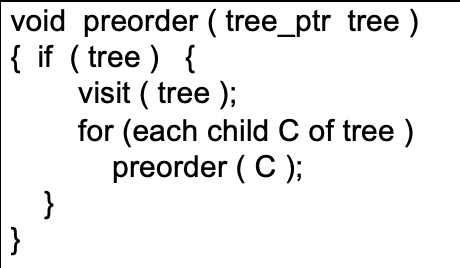
I used the FirstChild-NextSibling tree to finish the inorder, preoder, postorder and levelorder tree traversal algorithms.

2.2: inorder:

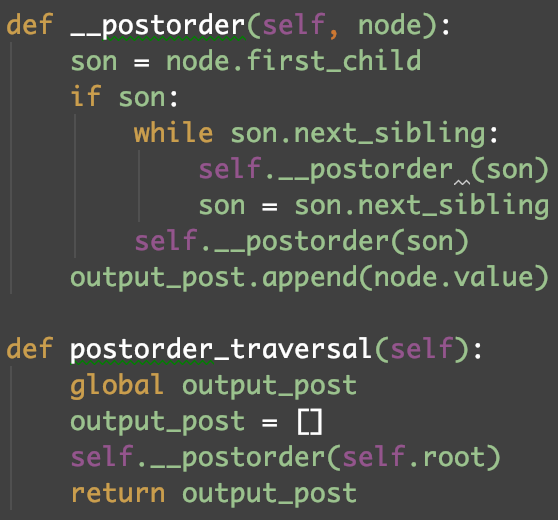


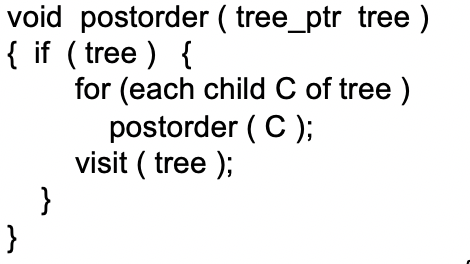
2.2.1: coding in Python:

2.3: preorder:

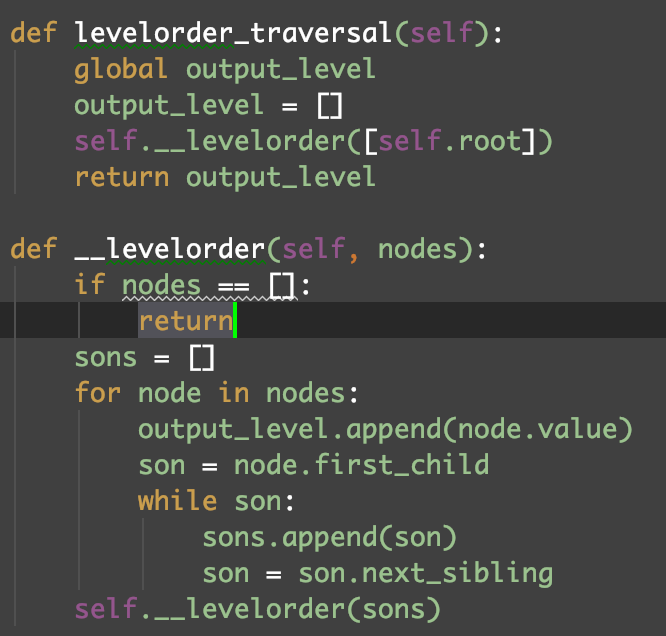


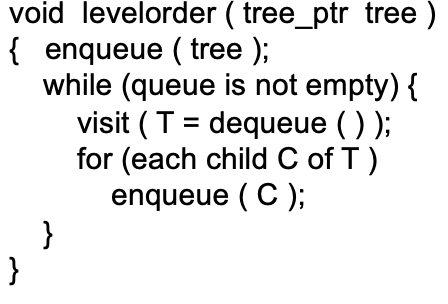
2.3.1 coding in Python:

2.4: postorder:



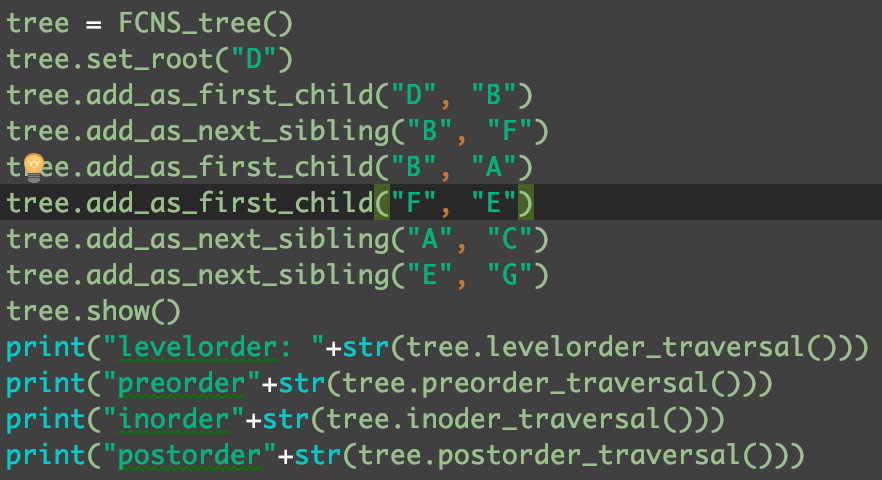
2.4.1 coding in Python:

2.5: levelorder:



2.5.1 coding in Python:

2.6: Test:



2.7: The tree:

2.8:Output:

