

### Question 2: (5 marks)

The given file Q2.py already contains statements to implement a simple program to monitor Phone objects using Binary Search Tree structure.

You must read carefully the function **insertNode()** to identify the **key** of BST (key is the information to identify whether a node is bigger or smaller than the others).

You should write statements to the following functions:

- a. **f1()**: Find the height of the given Binary Search Tree (BST).  
**Expected result:**  
2
- b. **f2()**: Perform the Post-Order traverse on the BST, but ONLY visit nodes that contains Phone's amount > 50.  
**Expected result:**  
100, Apple, 72, 8.540  
156, Samsung, 100, 3.555  
123, Apple, 60, 3.762
- c. **f3()**: Insert into the current tree a new Phone which code = 111, make = 'FPTPhone', amount = 10, price = k, where k is height of the current tree before insertion.  
**Expected result:**  
123, Apple, 60, 3.762  
110, Vivo, 10, 7.590  
100, Apple, 72, 8.540  
111, FPTPhone, 10, 2.000  
156, Samsung, 100, 3.555  
234, BPhone, 4, 3.690
- d. **f4()**: Decrease the Watch's price of root by 20%.  
**Expected result:**  
123, Apple, 60, 3.010  
110, Vivo, 10, 7.590  
156, Samsung, 100, 3.555  
100, Apple, 72, 8.540  
234, BPhone, 4, 3.690
- e. **f5()**: Remove all leaf nodes from the given BST.  
**Expected result:**  
123, Apple, 60, 3.762  
110, Vivo, 10, 7.590  
156, Samsung, 100, 3.555

### Notes:

- Do not edit given statements in the **main** function
- You can create new functions if you see they are necessary.
- Carefully read the instructions in each question to complete the practical exam.

===== End =====