**Question:**

You are given two binary trees, tree1 and tree2, with the same structure (same shape, possibly different node values). You are also given an integer divisor k.

Your task is to write a recursive function that traverses both trees simultaneously and updates the first tree (tree1) according to the following rules:

**If the values at the same node position are equal:**

* If divisible by k, replace the value in tree1 with 100.
* If not divisible, keep it unchanged.

**If the values are not equal:**

* Replace the value in tree1 with the sum of both node values.

Assume that the Node class, having data, left, and right attributes, is already given.

| **Sample Input** | **Sample Output** |
| --- | --- |
| **K =** 3  **Tree1:** 6  / \  5 9  / \  2 3  **Tree2:**  6  / \  1 4  / \  2 7 | 100 ← 6 == 6 and 6 % 3 == 0 → 100  / \  6 13 ← 5 + 1 = 6 and 9 + 4 = 13  / \  2 10 ← 2 == 2 and 2 % 3 != 0 →unchanged  ; 3 + 7 = 10 |

**Question:**

You are given two binary trees, tree1 and tree2, with the same structure (same shape, possibly different node values). You are also given an integer divisor k.

Your task is to write a recursive function that traverses both trees simultaneously and updates the second tree (tree2) according to the following rules:

**If the values at the same node position are not equal:**

* If the tree2 node value is divisible by k, replace the value in tree2 with 100.
* If not divisible, keep it unchanged.

**If the values are equal:**

* Replace the value in tree2 with the sum of both node values.

Assume that the Node class, having data, left, and right attributes, is already given

| **Sample Input** | **Sample Output** |
| --- | --- |
| **K =** 3  **Tree1:** 6  / \  5 9  / \  2 3  **Tree2:**  6  / \  1 6  / \  2 9 | 12 **← 6 == 6 → 6 + 6 = 12**  / \  1 100 **← 6 != 9 and 6 % 3 == 0 → 100**  / \  4 100  **← 2 == 2 → 2 + 2 = 4 and 9 != 3**  **and 9 % 3 == 0 → 100** |