Cooperation:

This assignment is to be done individually. Do not share code with others.

1. Complete the `MyTree01.java` class:

- Construct MyTree01(String preorder)
    - Given string of preorder, construct binary tree
        - Operations are *, /, +, and -
        - All other characters are variable (ex. x, y, s, t)
        - ex.) --xy*+st/xs
- String postorder()
    - Report tree postorder
        - ex.)xy-st+xs/*-
    - Do not include any space or \n
- String inorder()
    - Report tree inorder
    - Seal each operation with parenthesis
        - ex.) ((x-y)-((s+t)*(x/s)))
    - Do not include any space or \n
- String levelorder()
    - Report each level starting from root from left to write
      ex.) --*xy+/stxs
    - Do not include any space or \n

2. Print the path common to the two paths from the root to the two given nodes.

Given a binary tree with distinct nodes (no two nodes have the same have data values), your task is to complete `LCA.java` that contains the method `PrintCommonPath(node root, node n1, node n2)` which prints the path common to the two paths from the root to the two given nodes **n1** and **n2**. If either of the nodes are not present then print "No Common Path".

Example :
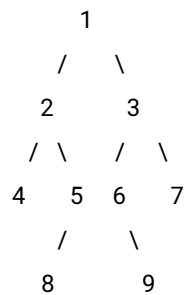
```
        1
      /   \
     2     3
    / \   / \
   4   5 6   7
      /   \
     8     9


       n1 = 4, n2 = 8
```

```
Output : 1->2
Path from root to n1:
1->2->4

Path from root to n2:
1->2->5->8

Common Path:
1->2
```

Note:
1. Your Node Class should contain (int data, Node left and Node right).
2. You can use your own Node class but consider the variables name.
3. The example is not actual input. The main method for evaluating is like below:

```
Node root = new Node(1);
root.left = new Node(2);
```

```
        root.right = new Node(3);
        root.left.left = new Node(4);
        root.left.right = new Node(5);
        root.right.right = new Node(6);
        root.right.right.left = new Node(7);
        root.left.right.left = new Node(8);
        root.left.right.right = new Node(9);

    int n1 = 4, n2 = 8;
    printCommonPath(root, n1, n2);
```

Assessment:
- Copying others' work will result zero.
- If your program doesn't compile, then you will get a zero score.
- Make sure your program runs on Linux. If you comment using any Korean character will cause error during process of compiling.

Submission:
- By eTL. Make sure you write your name and student number as a comment on top of each java.
- Once you have submitted assignment, pressing 'mark for grading' will prevent from making any change. Do NOT email for changing in assignment after you have clicked 'mark for grading'.

Late submissions will not be accepted.