




## Tree: Inorder Traversal (LNR) - Duyệt cây BST theo LNR

 [Download the code template](#)

Sample source: <https://www.hackerrank.com>

### Tree: Inorder Traversal (Duyệt cây theo thứ tự LNR)

Implement the **function void postOrder(Node\* root)** where root is a pointer to the root of the binary tree; this function will print entire the binary tree in the NLR order traversal as a single line of the value of tree with a space separated.

Sinh viên cài đặt hàm **void postOrder(Node\* root)** với root là con trỏ chỉ đến gốc của cây nhị phân ; hàm này sẽ in toàn bộ cây nhị phân thành một dòng các giá trị cách nhau với 1 khoảng trắng theo thứ tự duyệt NLR

### Sample (Ví dụ):

| Input                                       | Output      |
|---|-------------|
| 1<br>\<br>2<br>\<br>5<br>/<br>3 6<br>\<br>4 | 1 2 3 4 5 6 |

[IT003.N212.1] Assignment 4 

10 problems with a total score of 1000



## IT003 - CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

|    |  |     |
|----|--|-----|
| 1  | <a href="#">Tree: Hieght of Tree</a>   | 100 |
| 2  | <a href="#">Binary Search Tree: Insert (không dùng đệ quy)</a>                           | 100 |
| 3  | <a href="#">Binary Search Tree: Insert</a>   | 100 |
| 4  | <a href="#">Tree: levelOrder Traversal - Duyệt cây BST theo chiều rộng</a>               | 100 |
| 5  | <a href="#">Tree: Inorder Traversal (LNR) II - Duyệt cây BST theo LNR không đệ quy</a>   | 100 |
| 6  | <a href="#">Tree: Inorder Traversal (LNR) - Duyệt cây BST theo LNR</a>                   | 100 |
| 7  | <a href="#">Tree: Postorder Traversal (LRN) II - Duyệt cây BST theo LRN không đệ quy</a> | 100 |
| 8  | <a href="#">Tree: Postorder Traversal (LRN) - Duyệt cây BST theo LRN</a>                 | 100 |
| 9  | <a href="#">Tree: Preorder Traversal (NLR) II - Duyệt cây BST theo NLR không đệ quy</a>  | 100 |
| 10 | <a href="#">Tree: Preorder Traversal (NLR) - Duyệt cây BST theo NLR</a>                  | 100 |