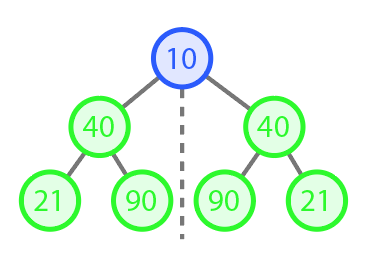
**Q1 - Given the root of a binary tree, check whether it is symmetrical within itself.**

**Example 1 -**

Input - [10,40,40,21,90,90,21]

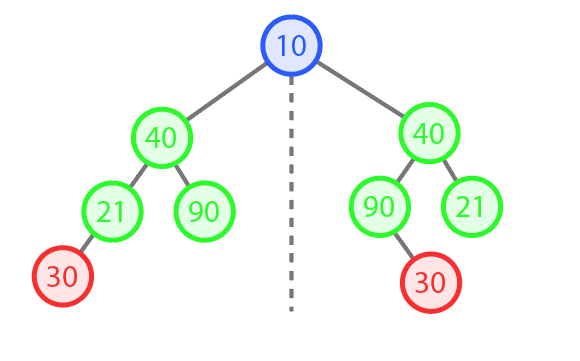
Output - true



**Example 2 -**

Input - [10,40,40,21,90,90,21,30,null,null,null,null,30]

Output - false



Sample definition for the Binary tree in c++ :   
 Struct BinaryTreeNode{

int data;

BinaryTreeNode \*left;

BinaryTreeNode \*right;

BinaryTreeNode() {

data= 0;

left = nullptr;

right = nullptr;

}

BinaryTreeNode(int x) {

data= x;

left = nullptr;

right = nullptr;

}

}

**Q2 - Given an unsorted array, print Kth Largest an unsorted array.(Do not use sorting)**

**Example 1-**

Input: Array = [1,2,6,4,5,3] ,

K = 3

Output: kth largest element = 4

**Example 2-**

Input: Array = [1,2,6,4,5] ,

k = 3

Output : kth largest element = 4

**Q3 - Given the head of a singly linked list, return true if it is a palindrome or false otherwise.**

**Example 1-**

Input: head = [1,2,2,1]

Output: true

**Example 2-**

Input: head = [1,2]

Output: false

Constraints:

The number of nodes in the list is in the range [1, 105].

0 <= Node.val <= 9

**Ques 4. Given an undirected graph, you have to find the number of connected components in that graph. If the graph is connected, then print -1.**

**E.g. -**

**Input -** Nodes: [1, 2, 3], Edges: [(1, 2), (1, 3), (2, 3)]

**Output**  = -1, the graph is connected

E.g.-

**Input -** Nodes: [1, 2, 3,4,5], Edges: [(1, 2), (1, 3), (2, 3)]

**Output =** 3 - As {1,2,3} {4} and {5} are not connected to anyone