```swift
import Foundation
/// 剑指 Offer 32 - I. 从上到下打印二叉树
/// 从上到下打印出二叉树的每个节点，同一层的节点按照从左到右的顺序打印。
///
/// 例如:
/// 给定二叉树: [3,9,20,null,null,15,7],
/// 返回：[3,9,20,15,7]
public class TreeNode {
    public var val: Int
    public var left: TreeNode?
    public var right: TreeNode?
    public init(_ val: Int) {
        self.val = val
        self.left = nil
        self.right = nil
    }
}

class Solution {
    /// 执行用时：8 ms, 在所有 Swift 提交中击败了 90.00% 的用户
    /// 内存消耗：13.9 MB, 在所有 Swift 提交中击败了 88.00% 的用户
    /// 通过测试用例：34 / 34
    func levelOrder(_ root: TreeNode?) -> [Int] {
        var results: [Int] = []
        var _root: TreeNode? = root
        if _root == nil { return [] }

        var cur: [TreeNode?] = [_root]
        while !cur.isEmpty {
            var nex: [TreeNode?] = []
            for node in cur {
                if let node = node {
                    results.append(node.val)
                    if let left = node.left {
                        nex.append(left)
                    }
                    if let right = node.right {
                        nex.append(right)
                    }
                }
            }
            cur = nex
        }
        return results
   }
}
```