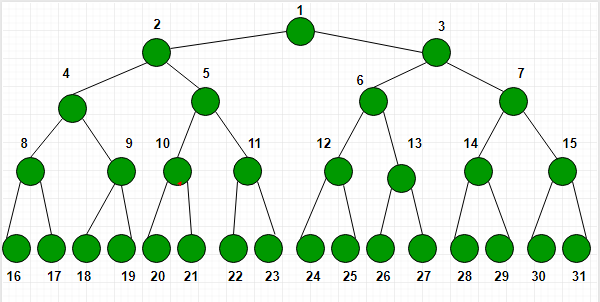
**Extreme nodes in alternate order:-**

Given a binary tree, print nodes of extreme corners of each level but in alternate order.

Example:  


For above tree, the output is

**1 2 7 8 31**  
– print rightmost node of 1st level  
– print leftmost node of 2nd level  
– print rightmost node of 3rd level  
– print leftmost node of 4th level  
– print rightmost node of 5th level

**Example 1:**

**Input:**

1

  / \

  2 3

**Output:** 1 2

**â€‹Explanation:** This represents a tree

with 3 nodes and 2 edges where root

is 1, left child of 1 is 3 and

right child of 1 is 2.

**Your Task:**  
You don't have to take any input. Just complete the function**ExtremeNodes()**that takes root **node**as **paramter**and return the answer (as vector<int> in **cpp,**as ArrayList<Integer> in **Java,**as list in python)  
  
**Expected Time Complexity:**O(N).  
**Expected Auxiliary Space:**O(N).  
  
**Constraints:**  
1 <=Number of nodes<= 100  
1 <=Data of a node<= 1000