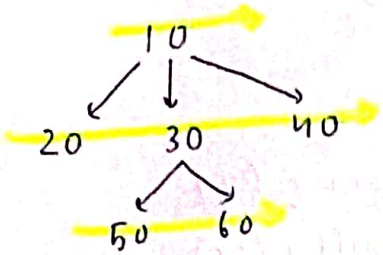
LEVEL ORDER TRAVERSAL IN A GENERIC TREE



Output: $10^1\ 20^2\ 30^2\ 40^2\ 50^3\ 60^3$ → levels

- Hume print krna hai ek tree in levels left to right fashion mein, nodes separated by spaces.
- Hum Queue Data Structure use krengy!
- Phele root wenesi tree ka add krengy Queue me!
- Uske baad, hum yeh Rule use krengy → R → REMOVE, P → PRINT, A → ADD
- Remove element from Queue
  ↓
  Print that element
  ↓
  and then add it's children to the queue.

START → | 10 |

| 10 | 20 | 30 | 40 |

10 removed and printed and it's child added — 20, 30, 40
Output: 10

| ~~10~~ | ~~20~~ | 30 | 40 |

20 removed and printed (No child)
Output: 10 20

| ~~30~~ | ~~20~~ | ~~50~~ | 40 | 50 | 60 |

30 removed and printed and it's child added — 50, 60
Output: 10 20 30

| ~~10~~ | ~~20~~ | ~~30~~ | ~~40~~ | 50 | 60 |

40 removed and printed (No child)
Output: 10 20 30 40

| ~~10~~ | ~~20~~ | ~~30~~ | ~~40~~ | ~~50~~ | 60 |

50 removed and printed (No child)
Output: 10 20 30 40 50

| ~~10~~ | ~~20~~ | ~~30~~ | ~~40~~ | ~~50~~ | ~~60~~ |

60 removed and printed (No child)
Output: 10 20 30 40 50 60

Code

```
public class Main{
  public static class Node{
    int data;
    ArrayList<Node> childsen = new ArrayList<>();

    Node(int data){
    this.data = data;
    }
  }
}
public static void LevelOrder(Node root){
Queue<Node> q = new ArrayDeque<Node>();
q.add(root);

while(q.size()>0){
Node temp = q.remove(); // remove
Syso(temp.data + " "); //print
for(Node child : temp.children){ //add
    q.add(child);
  }
}
Syso(".");
}

public static void main(static [] args){
Node root = new Node(10);
Node twenty = new Node(20);
root.children.add(twenty);
Node thirty = new Node(30);
```

Output: 10 20 30 40 50 60

END