```
class Node:

  def __init__(self, data):
    self.data = data
    self.left = None
    self.right = None

def printLeaves(root):
  if(root):
    printLeaves(root.left)

    if root.left is None and root.right is None:
      print(root.data),

    printLeaves(root.right)

def printBoundaryLeft(root):

  if(root):
    if (root.left):

      print(root.data)
      printBoundaryLeft(root.left)

    elif(root.right):
      print (root.data)
      printBoundaryLeft(root.right)

def printBoundaryRight(root):

  if(root):
    if (root.right):

      printBoundaryRight(root.right)
      print(root.data)

    elif(root.left):
      printBoundaryRight(root.left)
      print(root.data)

def printBoundary(root):
  if (root):
    print(root.data)

    printBoundaryLeft(root.left)

    printLeaves(root.left)
    printLeaves(root.right)

    printBoundaryRight(root.right)



root = Node(1)
root.left = Node(2)
root.left.left = Node(4)
root.left.right = Node(5)
root.left.right.left = Node(8)
root.left.right.right = Node(9)
root.right = Node(3)
root.right.left = Node(6)
root.right.right = Node(7)
printBoundary(root)
```

```
1
2
4
8
9
6
7
3
```

✓ 0s completed at 9:10 AM