

# A binary tree node

```
class Node:
```

```
    # Constructor to create a new node
```

```
    def __init__(self, data):
```

```
        self.data = data
```

```
        self.left = None
```

```
        self.right = None
```

# A simple function to print leaf nodes of a Binary Tree

```
def printLeaves(root):
```

```
    if(root):
```

```
        printLeaves(root.left)
```

```
        # Print it if it is a leaf node
```

```
        if root.left is None and root.right is None:
```

```
            print(root.data),
```

```
        printLeaves(root.right)
```

# A function to print all left boundary nodes, except a

# leaf node. Print the nodes in TOP DOWN manner

```
def printBoundaryLeft(root):
```

```
    if(root):
```

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

```
            # to ensure top down order, print the node
```

```
            # before calling itself for left subtree
```

```
            print(root.data)
```

```
            printBoundaryLeft(root.left)
```

```
        elif(root.right):
```

```
            print (root.data)
```

```
            printBoundaryLeft(root.right)
```

```
        # do nothing if it is a leaf node, this way we
```

```
        # avoid duplicates in output
```

```
# A function to print all right boundary nodes, except
# a leaf node. Print the nodes in BOTTOM UP manner
def printBoundaryRight(root):
    if(root):
        if (root.right):
            # to ensure bottom up order, first call for
            # right subtree, then print this node
            printBoundaryRight(root.right)
            print(root.data)
        elif(root.left):
            printBoundaryRight(root.left)
            print(root.data)
        # do nothing if it is a leaf node, this way we
        # avoid duplicates in output

# A function to do boundary traversal of a given binary tree
def printBoundary(root):
    if (root):
        print(root.data)
        # Print the left boundary in top-down manner
        printBoundaryLeft(root.left)
        # Print all leaf nodes
        printLeaves(root.left)
        printLeaves(root.right)
        # Print the right boundary in bottom-up manner
        printBoundaryRight(root.right)
```