```
In [1]:
            # Recursive Python program for level
          1
             # order traversal of Binary Tree
          3
             class Node:
          4
          5
                 def __init__(self, key):
          6
                     self.data = key
          7
                     self.left = None
          8
                     self.right = None
          9
         10
         11
             # Function to print level order traversal of tree
            def printLevelOrder(root):
         12
                 h = height(root)
         13
         14
                 for i in range(1, h+1):
                     printCurrentLevel(root, i)
         15
         16
         17
             # Print nodes at a current level
         18
         19
             def printCurrentLevel(root, level):
                 if root is None:
         20
         21
                     return
                 if level == 1:
         22
                     print(root.data, end=" ")
         23
         24
                 elif level > 1:
         25
                     printCurrentLevel(root.left, level-1)
         26
                     printCurrentLevel(root.right, level-1)
         27
         28
         29
             # Compute the height of a tree-
         30
             def height(node):
         31
                 if node is None:
         32
                     return 0
         33
                 else:
                     lheight = height(node.left)
         34
         35
                     rheight = height(node.right)
         36
                     if lheight > rheight:
         37
                         return lheight+1
         38
                     else:
         39
                         return rheight+1
         40
         41
             if __name__ == '__main__':
         42
         43
                 root = Node(1)
         44
                 root.left = Node(2)
                 root.right = Node(3)
         45
         46
                 root.left.left = Node(4)
         47
                 root.left.right = Node(5)
         48
                 printLevelOrder(root)
```

1 2 3 4 5

```
In [5]:
             # Recursive Python program for Level
          1
             # order traversal of Binary Tree
          2
          3
          4
             class Node:
          5
                 def __init__(self, key):
          6
                     self.data = key
          7
                     self.left = None
          8
                     self.right = None
          9
         10
         11 # Function to print level order traversal of tree
             def printLevelOrder(root):
         12
         13
                 h = height(root)
         14
                 for i in range(1, h+1):
         15
                     printCurrentLevel(root, i)
         16
         17
         18 # Print nodes at a current level
             def printCurrentLevel(root, level):
         19
                 if root is None:
         20
         21
                     return
         22
                 if level == 1:
                     print(root.data, end=" ")
         23
         24
                 elif level > 1:
         25
                     printCurrentLevel(root.left, level-1)
         26
                     printCurrentLevel(root.right, level-1)
         27
         28
         29
             # Compute the height of a tree-
            def height(node):
         30
         31
                 if node is None:
         32
                     return 0
         33
                 else:
                     lheight = height(node.left)
         34
         35
                     rheight = height(node.right)
         36
                     if lheight > rheight:
         37
                         return lheight+1
         38
                     else:
         39
                         return rheight+1
         40
         41
             if __name__ == '__main__':
         42
         43
                 root = Node(input("Enter root node "))
         44
                 root.left = Node(input("Enter left to root node "))
         45
                 root.right = Node(input("Enter Right to root node "))
         46
                 root.right.left = Node(input("Enter left to Right root node "))
         47
                 root.left.left = Node(input("Enter left to left root node "))
                 root.left.right = Node(input("Enter Right to left root node "))
         48
         49
                 root.left.left.left = Node(input("Enter left to left to left root n
         50
                 root.left.left.right = Node(input("Enter right to left to left root
         51
                 printLevelOrder(root)
```

Enter root nodeA

Enter left to root nodeB
Enter Right to root nodeC

Enter left to Right to root nodeF

Enter left to left root nodeD Enter Right to left root nodeE

Enter left to left to left root nodeG Enter right to left to left root nodeH

ABCDEFGH

In [ ]: 1