

Problem 1: Populate Next Right pointers of Tree class TreeNode:

```
def __init__(self, val=0, left=None, right=None, next=None):
```

```
    self.val = val
```

```
    self.left = left
```

```
    self.right = right
```

```
    self.next = next
```

```
def connect(root):
```

```
    if not root:
```

```
        return None
```

```
    level_start = root
```

```
    while level_start:
```

```
        current = level_start
```

```
        while current:
```

```
            if current.left:
```

```
                current.left.next = current.right
```

```
            if current.right and current.next:
```

```
                current.right.next = current.next.left
```

```
            current = current.next
```

```
        level_start = level_start.left
```

```
    return root
```

```
root = TreeNode(1)
```

```
root.left = TreeNode(2)
```

```
root.right = TreeNode(3)
```

```
root.left.left = TreeNode(4)
```

```
root.left.right = TreeNode(5)
```

```
root.right.left = TreeNode(6)
```

```
root.right.right = TreeNode(7)
```

```
connect(root)
```

```
print(root.val, "->", root.next)
```

```
print(root.left.val, "->", root.left.next.val)
```

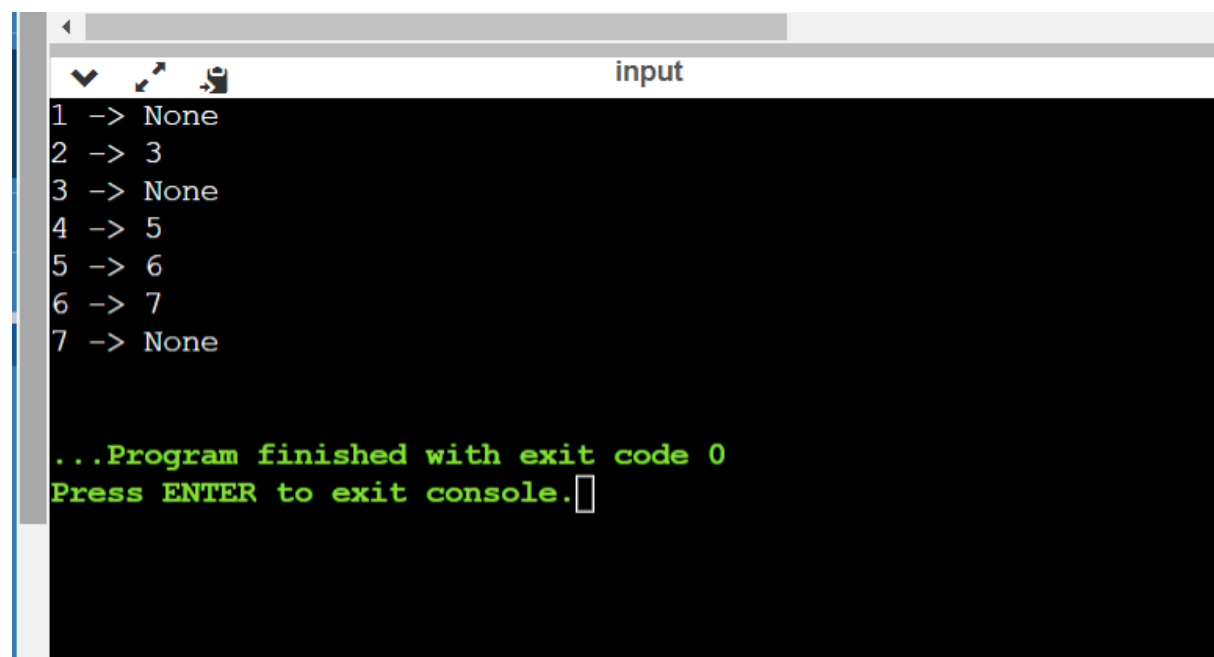
```
print(root.right.val, "->", root.right.next)
```

```
print(root.left.left.val, "->", root.left.left.next.val)
```

```
print(root.left.right.val, "->", root.left.right.next.val)
```

```
print(root.right.left.val, "->", root.right.left.next.val)
```

```
print(root.right.right.val, "->", root.right.right.next)
```



```
input
1 -> None
2 -> 3
3 -> None
4 -> 5
5 -> 6
6 -> 7
7 -> None

...Program finished with exit code 0
Press ENTER to exit console.
```