## Problem 1.

## Find the Maximum Depth of given Binary Tree

**TestCase1:**

Input: 1, 2, 3, 4, 5, 6, 7, 8, 9

```
           1

          /   \

        2       3

      /    \   /   \

    4       5 6    7

      /    \

     8      9
```

Output: Maximum depth of tree is 4

**TestCase2:**

Input: 1, 2, 3, 4, 5

```
           1

          /   \

        2       3

         /    \

        4       5
```

Output: Maximum depth of tree is 3

**Problem 2.**
**Write a function to determine if two trees are identical or not:**
**Two trees are identical when they have the same data and the arrangement of data is also the same**

**Examples:**

Input: 1, 2, 3, 4 ; 1, 2, 3, 4

```
          1               1
        /   \           /   \
       2     3         2     3
     /               /
    4               4
```

Output: Both trees are identical

Input: 1, 2, 3, 4 ; 1, 2, 3, 4

```
          1               1
        /   \           /   \
       2     3         5     3
           /         /
          4         4
```

Output: Trees are not identical

**Problem 3.**
**You are given a binary tree and a given sum. The task is to check if there exists a subtree whose sum of all nodes is equal to the given sum.**

***Input :***

```
               1

             /   \

            3     6

          /   \  /

         5     9 8
```

1, 3, 6, 5, 9, 8

*sum = 17*

**Output:** *"Yes"*

*// sum of all nodes of subtree {3, 5, 9} = 17*

```
 Input :   1
          /   \
         3      6
       /    \   /
      5      9 8
```

1, 3, 6, 5, 9, 8

*sum = 11*

**Output:** *"No"*

*// no subtree with given sum exist*

**Problem 4.**

**Check if a given Binary Tree is a SumTree**

**Write a function that returns true if the given Binary Tree is a SumTree, else false. A SumTree is a Binary Tree where the value of a node is equal to the sum of the nodes present in its left subtree and right subtree. An empty tree is a SumTree, and the sum of an empty tree can be considered as 0. A leaf node is also considered a SumTree.**

**Following is an example of SumTree.**

```
      26

     /  \

    10    3

  /   \    \

 4     6     3
```

**TESTCASE 1:**

Input:

```
      26

     /  \

    10    3

  /   \    \

 4     6     3
```

26, 10, 3, 4, 6, 3

Output: The given tree is a SumTree

**TESTCASE 2:**

```
Input: 1

           /   \

          3      6

        /    \   /

       5      9 8
```

1, 3, 6, 5, 9, 8

Output:  The given tree is not a SumTree

**Problem 5.**
**Insert a node before the middle of the linked list**
**Find the middle of the linked list and insert a given value before it.**

**Description:**

**When the number of nodes in the linked list is odd, the middle node is straightforward to determine; it's the node at position (n+1)/2, where 'n' is the total number of nodes. However, when the number of nodes is even, there are two middle nodes. In such cases, we consider the one closer to the head as the middle node.**

**Input:**

List = 1->2->3->4->5,

Insert a node before the middle of the linked list with value 9

**Output**

1->2->9->3->4->5

**Input:**

List = 11->10->9->7->6->5->4->3->2->1,

Insert a node before the middle of the linked list with value 8

**Output**

12->11->10->9->8->7->6->5->4->3->2->1

## Problem 6.
**Delete the last occurrence of an item from a linked list:**
**Delete the very last occurrence of an item in a linked list.**

**TESTCASE1:**

Created Linked list: 1  --> 2  --> 3  --> 4  --> 5  --> 4  --> 4  --> NULL

List after deletion of 4: 1  --> 2  --> 3  --> 4  --> 5  --> 4  --> NULL

**TESTCASE2:**

Created Linked list:      11  --> 32  --> 123  --> 344  --> 445  --> 484  --> 534  --> NULL

List after deletion of 445: 11  --> 32  --> 123  --> 344  --> 484  --> 534  --> NULL

## Problem 7.
**Rotate a Linked List**
**Given a singly linked list and a positive integer 'k', rotate the linked list to the right by 'k' places.**

**Examples**:

Input: linked list = 10->20->30->40->50->60, k = 4

Output: 30->40->50->60->10->20.

Input: linked list = 30->40->50->60, k = 2

Output: 50->60->30->40.