**Problem Statement: Given the heads of two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.**

**Examples**:

### **Input Format:**

### **(Pointer/Access to the head of the two linked lists)**

### **num1 = 243, num2 = 564**

### **l1 = [2,4,3]**

### **l2 = [5,6,4]**

### **Result: sum = 807; L = [7,0,8]**

### **Explanation: Since the digits are stored in reverse order, reverse the numbers first to get the or original number and then add them**

### **as → 342 + 465 = 807.**

### **Refer to the image below.**

***Disclaimer***: *Don't jump directly to the solution, try it out yourself first.*

**Solution 1:**

**Approach**: Try to write the approach, don’t write the pseudocode, try writing the thought process, and how you are solving the problem.   
  
In case this is an algorithm and follows a certain intuition, please mention it.   
  
Follow this article for reference on how to write the approach: <https://takeuforward.org/data-structure/4-sum-find-quads-that-add-up-to-a-target-value/>

**Tip**: *Use images if required in approach.*

**[Please add running code with int main or psvm in java]  
[follow 4 sum article above to have an idea]   
[use <https://codebeautify.org/cpp-formatter-beautifier> to format your c++ code]**

**[use <https://codebeautify.org/javaviewer> to format your java code]   
Code**:

[tabby title = "C++ Code"]

#include<iostream>

using namespace std;

int main()

{

return 0;

}

[tabby title = "Java Code"]

public class tUf {

public static void main(String args[])

{

System.out.println("Hello World");

}

}

[tabby title = "Python Code"]

# Python code for "Hello World"

# nothing else to type...see how simple is the syntax.

print("Hello World")

[tabbyending]

**Time Complexity:**

**Space Complexity:**

**Author Name:**

**Linkedin Profile:**