Sumlist in C++

```
#include <iostream>
using namespace std;

// Node class for the linked list
class Node {
public:
    int data;
    Node* next;

    // Default constructor
    Node() {
        data = 0;
        next = nullptr;
    }

    // Constructor with data parameter
    Node(int data) {
        this->data = data;
        next = nullptr;
    }

    void setNext(Node* next) {
        this->next = next;
    }
};

// Function to print the linked list
void printList(Node* head) {
    Node* current = head;
    while (current != nullptr) {
        cout << current->data << " -> ";
        current = current->next;
    }
    cout << "null" << endl;
}

// Function to add two linked lists
representing numbers
Node* add(Node* l1, Node* l2, int carry) {
    if (l1 == nullptr && l2 == nullptr &&
carry == 0) {
        return nullptr;
    }

    Node* result = new Node();
    int value = carry;
    if (l1 != nullptr) {
        value += l1->data;
    }
    if (l2 != nullptr) {
        value += l2->data;
    }
    result->data = value % 10;

    if (l1 != nullptr || l2 != nullptr) {
        Node* more = add(l1 == nullptr ?
nullptr : l1->next, l2 == nullptr ? nullptr : l2-
>next, value >= 10 ? 1 : 0);
        result->setNext(more);
    }
    return result;
```

What the Code Does

- Adds two numbers represented by linked lists in **reverse order** (just like how we add numbers manually from right to left).
- Example:
    - List 1: 7 -> 1 -> 6 = 617
    - List 2: 5 -> 9 -> 2 = 295
    - Sum: **617 + 295 = 912**
    - Result list: 2 -> 1 -> 9

📦 Input Linked Lists

List	Nodes	Represents
l1	7 → 1 → 6	617
l2	5 → 9 → 2	295

🔄 add(l1, l2, carry) Dry Run

Step	l1->data	l2->data	Carry In	Sum	Digit Stored	Carry Out	Notes
1	7	5	0	12	2	1	result->data = 2
2	1	9	1	11	1	1	result->next->data = 1
3	6	2	1	9	9	0	result->next->next->data = 9
4	null	null	0	-	-	-	Recursion stops

🔁 Result Linked List After Addition

2 -> 1 -> 9 -> null

```
}

int main() {
    // Creating two linked lists representing
numbers
    Node* head1 = new Node(7);
    head1->next = new Node(1);
    head1->next->next = new Node(6);

    Node* head2 = new Node(5);
    head2->next = new Node(9);
    head2->next->next = new Node(2);

    // Adding the two linked lists
    Node* result = add(head1, head2, 0);

    // Printing the result linked list
    cout << "Result of addition:" << endl;
    printList(result);

    return 0;
}
```

```
Result of addition:
2 -> 1 -> 9 -> null
```