**Remove duplicate in LL in C++**

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

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

    Node(int data) {
        this->data = data;
        this->next = nullptr;
    }
};

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

// Function to remove duplicates from the linked list
void deleteDups(Node* head) {
    if (head == nullptr || head->next == nullptr)
return;

    Node* current = head;
    while (current != nullptr) {
        Node* runner = current;
        while (runner->next != nullptr) {
            if (runner->next->data == current->data) {
                runner->next = runner->next->next;
            } else {
                runner = runner->next;
            }
        }
        current = current->next;
    }
}

int main() {
    // Creating a linked list with 5 hard-coded nodes
    Node* head = new Node(1);
    head->next = new Node(2);
    head->next->next = new Node(2);
    head->next->next->next = new Node(3);
    head->next->next->next->next = new Node(4);
    head->next->next->next->next->next = new
Node(3);
    head->next->next->next->next->next->next = new
Node(5);
```

**Creates a linked list: 1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null**

**Initial Linked List Creation**

| Node | Value | Next Points To |
|---|---|---|
| head | 1 | Node 2 |
| head->next | 2 | Node 2 |
| ... | 2 | Node 3 |
| ... | 3 | Node 4 |
| ... | 4 | Node 3 |
| ... | 3 | Node 5 |
| ... | 5 | nullptr |

**🖷 Initial Output from printList(head)**

Original Linked List:
1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null

**🔂 deleteDups(head) Dry Run**

**Loop Over current Node**

| current->data | Duplicate(s) Found and Removed | Resulting List |
|---|---|---|
| 1 | None | 1 → 2 → 2 → 3 → 4 → 3 → 5 |
| 2 | Second 2 removed | 1 → 2 → 3 → 4 → 3 → 5 |
| 3 | Second 3 removed | 1 → 2 → 3 → 4 → 5 |
| 4 | None | 1 → 2 → 3 → 4 → 5 |
| 5 | None | 1 → 2 → 3 → 4 → 5 |

**✅ Final Linked List After deleteDups(head)**

Linked List after removing duplicates:

| | |
|---|---|
| // Print the original linked list<br>cout << "Original Linked List:" << endl;<br>printList(head);<br><br>// Remove duplicates<br>deleteDups(head);<br><br>// Print the linked list after removing duplicates<br>cout << "Linked List after removing duplicates:" << endl;<br>printList(head);<br><br>return 0;<br>} | 1 -> 2 -> 3 -> 4 -> 5 -> null |
| Original Linked List:<br>1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null<br>Linked List after removing duplicates:<br>1 -> 2 -> 3 -> 4 -> 5 -> null | |