Sublist in C++

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

class Node {
public:
    int data;
    Node* next;

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

void printList(Node* head) {
    Node* current = head;
    while (current != nullptr) {
        cout << current->data << " -> ";
        current = current->next;
    }
    cout << "null" << endl;
}

void sublists(Node* head) {
    Node* i = head;
    while (i != nullptr) {
        Node* j = i;
        while (j != nullptr) {
            cout << j->data << " -> ";
            j = j->next;
        }
        cout << "null" << endl;
        i = i->next;
    }
}

int main() {
    // Create 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);

    // Print the linked list
    printList(head);

    // Print all sublists
    sublists(head);

    // Clean up memory
    Node* current = head;
    while (current != nullptr) {
        Node* next = current->next;
        delete current;
        current = next;
    }
```

Linked List Creation

Step	Node Created	data	next Points To
1	head	1	Node with 2
2	head->next	2	Node with 2
3	...	2	Node with 3
4	...	3	Node with 4
5	...	4	Node with 3
6	...	3	Node with 5
7	...	5	nullptr

⏫ printList(head) Output

1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null

🔁 sublists(head) Dry Run Table

Outer Loop (i->data)	Inner Loop Iteration (→ values printed)
1	1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null
2 (1st)	2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null
2 (2nd)	2 -> 3 -> 4 -> 3 -> 5 -> null
3	3 -> 4 -> 3 -> 5 -> null
4	4 -> 3 -> 5 -> null
3 (last)	3 -> 5 -> null
5	5 -> null

🧹 Cleanup (Memory Deallocation)

Step	Node Deleted	data
1	head	1
2		2
3		2
4		3
5		4

```
    return 0;
}
```

Step	Node Deleted	data
6		3
7		5

1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null
1 -> 2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null
2 -> 2 -> 3 -> 4 -> 3 -> 5 -> null
2 -> 3 -> 4 -> 3 -> 5 -> null
3 -> 4 -> 3 -> 5 -> null
4 -> 3 -> 5 -> null
3 -> 5 -> null
5 -> null