Part 4 - Manual VS STL List

A **doubly linked list** is a data structure where each node contains a pointer to the **next** and **previous** nodes, allowing traversal in both directions. In C++, you can implement it manually or use the built-in STL `std::list`.

Differences between Manual Doubly Linked List and STL `std::list`:

Feature	Manual Doubly Linked List	STL `std::list`
Implementation	You define the `Node` structure and pointers manually.	Already implemented as a doubly linked list in the STL.
Memory Management	Manual allocation and deallocation using `new` and `delete`.	Automatic memory management.
Operations	Insert, delete, traversal, and search must be implemented manually.	Provides built-in functions like `push_back`, `push_front`, `erase`, and `find`.
Complexity	More control but more prone to errors like memory leaks.	Safer and easier to use, less prone to bugs.

1. Manual Doubly Linked List Example

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

struct Node {
    int data;
    Node* next;
    Node* prev;
};

// Add a node at the end
void pushBack(Node*& head, int value) {
    Node* newNode = new Node{value, nullptr, nullptr};
    if (!head) {
        head = newNode;
        return;
    }
    Node* temp = head;
    while (temp->next) temp = temp->next;
```

```
    temp->next = newNode;
    newNode->prev = temp;
}

// Print the list
void printList(Node* head) {
    Node* temp = head;
    while (temp) {
        cout << temp->data << " ";
        temp = temp->next;
    }
    cout << endl;
}

int main() {
    Node* head = nullptr;
    pushBack(head, 10);
    pushBack(head, 20);
    pushBack(head, 30);

    printList(head); // Output: 10 20 30
}
```

2. STL `std::list` Example

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

int main() {
    list<int> l;

    l.push_back(10);
    l.push_back(20);
    l.push_back(30);

    for (int x : l)
        cout << x << " "; // Output: 10 20 30
    cout << endl;
}
```

Explanation:

- Manual doubly linked list requires defining nodes and managing pointers yourself.
- `std::list` simplifies everything: memory management and operations like insertion, deletion, and traversal are built-in.

Revision #1
Created 2025-09-02 08:51:58 UTC by BH
Updated 2025-09-02 08:54:22 UTC by BH