#include <iostream>

struct Node {

int data;

Node\* next;

Node\* prev;

Node(int data) : data(data), next(nullptr), prev(nullptr) {}

};

class DoublyLinkedList {

private:

Node\* head;

public:

DoublyLinkedList() : head(nullptr) {}

void append(int data) {

Node\* newNode = new Node(data);

if (!head) {

head = newNode;

return;

}

Node\* temp = head;

while (temp->next) {

temp = temp->next;

}

temp->next = newNode;

newNode->prev = temp;

}

void display() const {

Node\* temp = head;

while (temp) {

std::cout << temp->data << " ";

temp = temp->next;

}

std::cout << std::endl;

}

void reverse() {

Node\* temp = nullptr;

Node\* current = head;

while (current) {

temp = current->prev;

current->prev = current->next;

current->next = temp;

current = current->prev;

}

if (temp) {

head = temp->prev;

}

}

};