#include <iostream>

using namespace std;

class Node {

public:

int BookID;

string Title, Author;

Node\* next;

Node\* prev;

Node(int id, string t, string a) {

BookID = id;

Title = t;

Author = a;

next = prev = nullptr;

}

};

class DoublyLinkedList {

public:

Node\* head;

Node\* tail;

DoublyLinkedList() {

head = tail = nullptr;

}

void addAtEnd(int id, string t, string a) {

Node\* newNode = new Node(id, t, a);

if (!head) {

head = tail = newNode;

return;

}

tail->next = newNode;

newNode->prev = tail;

tail = newNode;

}

void addAtBeginning(int id, string t, string a) {

Node\* newNode = new Node(id, t, a);

if (!head) {

head = tail = newNode;

return;

}

newNode->next = head;

head->prev = newNode;

head = newNode;

}

void deleteByID(int id) {

Node\* temp = head;

while (temp != nullptr && temp->BookID != id)

temp = temp->next;

if (!temp) return;

if (temp == head) head = temp->next;

if (temp == tail) tail = temp->prev;

if (temp->prev) temp->prev->next = temp->next;

if (temp->next) temp->next->prev = temp->prev;

delete temp;

}

void displayForward() {

Node\* temp = head;

while (temp) {

cout << temp->BookID << " - " << temp->Title << " by " << temp->Author << endl;

temp = temp->next;

}

}

void displayBackward() {

Node\* temp = tail;

while (temp) {

cout << temp->BookID << " - " << temp->Title << " by " << temp->Author << endl;

temp = temp->prev;

}

}

};

int main() {

DoublyLinkedList library;

library.addAtEnd(1, "METPHORPHASUS, frans kafka");

library.addAtEnd(2, "Hamlet", " Mr Shakespeare");

library.addAtBeginning(3, "Matilda", "Ronald dahl");

cout << "Books (Forward):\n";

library.displayForward();

cout << "\nAfter deleting Book ID 2:\n";

library.deleteByID(2);

library.displayForward();

cout << "\nBooks (Backward):\n";

library.displayBackward();

return 0;

}