#include <iostream>

using namespace std;

class Feature {

public:

string Name;

Feature(string n) { Name = n; }

virtual void analyze() = 0;

virtual ~Feature() {}

};

class LandFeature : public Feature {

public:

LandFeature(string n) : Feature(n) {}

void analyze() override {

cout << "Land Feature Detected: " << Name << endl;

}

};

class WaterFeature : public Feature {

public:

WaterFeature(string n) : Feature(n) {}

void analyze() override {

cout << "Water Feature Detected: " << Name << endl;

}

};

class Node {

public:

int FeatureID;

Feature\* feature;

Node\* next;

Node(int id, Feature\* f) {

FeatureID = id;

feature = f;

next = nullptr;

}

};

class SinglyLinkedList {

public:

Node\* head;

SinglyLinkedList() {

head = nullptr;

}

void insertAtEnd(int id, Feature\* f) {

Node\* newNode = new Node(id, f);

if (head == nullptr) {

head = newNode;

return;

}

Node\* temp = head;

while (temp->next != nullptr)

temp = temp->next;

temp->next = newNode;

}

void deleteByID(int id) {

if (head == nullptr) return;

if (head->FeatureID == id) {

Node\* temp = head;

head = head->next;

delete temp;

return;

}

Node\* curr = head;

Node\* prev = nullptr;

while (curr != nullptr && curr->FeatureID != id) {

prev = curr;

curr = curr->next;

}

if (curr == nullptr) return;

prev->next = curr->next;

delete curr;

}

void displayAll() {

Node\* temp = head;

while (temp != nullptr) {

cout << "Feature ID: " << temp->FeatureID << " → ";

temp->feature->analyze();

temp = temp->next;

}

}

void reverseList() {

Node\* prev = nullptr;

Node\* curr = head;

Node\* next = nullptr;

while (curr != nullptr) {

next = curr->next;

curr->next = prev;

prev = curr;

curr = next;

}

head = prev;

}

~SinglyLinkedList() {

Node\* temp = head;

while (temp != nullptr) {

Node\* del = temp;

temp = temp->next;

delete del;

}

}

};

int main() {

SinglyLinkedList list;

list.insertAtEnd(1, new LandFeature("Forest"));

list.insertAtEnd(2, new WaterFeature("River"));

list.insertAtEnd(3, new LandFeature("Desert"));

list.insertAtEnd(4, new WaterFeature("Lake"));

cout << "Initial List:\n";

list.displayAll();

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

list.deleteByID(2);

list.displayAll();

cout << "\nAfter Reversing List:\n";

list.reverseList();

list.displayAll();

return 0;

}