#include<iostream>

using namespace std;

//void main() {

// int\* a = new int(100);

// int\* b = a;

// int\* c = b;

//

// delete c;

// cout << \*b << endl;

//

//}

//void main() {

// shared\_ptr<int>a(new int(42));

// shared\_ptr<int>b(a);

//}

//template<typename T>

//class SharedPtr {

// T\* address;

// size\_t \* count;

//public:

// SharedPtr():address(nullptr),count(nullptr){}

//

// SharedPtr(T\*address):address(address),count(new size\_t(1)){}

//

// SharedPtr(const SharedPtr& other) :address(other.address), count(other.count) {

// ++(\*count);

// }

//

// SharedPtr& operator=(const SharedPtr& other) {

// if (count != nullptr) {

// (\*count)--;

// if (\*(count) == 0) {

// delete address;

// delete count;

// }

// }

// count = other.count;

// address = other.address;

//

// (\*count)++;

// return \*this;

// }

//

// T\* get() {

// return address;

// }

//

// T\* operator->()const {

// return address;

// }

//

// T& operator\*()const {

// return \*address;

// }

//

// ~SharedPtr() {

// if (count != nullptr) {

// (\*count)--;

// if ((\*count) == 0) {

// delete address;

// delete count;

// }

// }

// }

//};

//

//

//class Student {

//

// string name;

// int age;

//public:

// Student():name("no name"),age(0){}

//

// Student(const string& name, const int& age) {

// SetName(name);

// SetAge(age);

// }

//

// void SetName(const string& name) {

// this->name = name;

// }

// void SetAge(const int& age) {

// this->age = age;

// }

//

// int GetAge()const {

// return age;

// }

// string GetName()const {

// return name;

// }

// void Show()const {

// cout << "Name : " << name << endl;

// cout << "Age : " << age << endl;

// }

//};

//

//void foo(Student\*s) {

// delete s;

//}

//void main() {

// /\*SharedPtr<int>a(new int(100));

// SharedPtr<int>b(new int(200));

// b = a;

// \*/

//

// SharedPtr<Student> s(new Student("Elvin", 30));

// SharedPtr<Student> s2 = s;

// cout << s2->GetName() << endl;

//

//}

void main() {

//unique\_ptr<int>a(new int(100));

//unique\_ptr<int>b(a);

}