#include <iostream>

//using namespace std;

//

////int main()

////{

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

//// int\* b = a;

//// int\* c = b;

////

////

//// delete c;

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

////

////

//// return 0;

////}

//

////

//// // Haziri

//// int main()

////{

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

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

////}

//

//template<typename T>

//class SharedPtr

//{

//private:

// T\* adress;

// size\_t\* count;

//public:

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

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

//

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

// ++(\*count);

// }

//

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

// {

// // if obj had other adress and count we delete them

// if (count != nullptr)

// {

// (\*count)--;

// // // if another pointer does not show this address and number

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

// {

// delete adress;

// delete count;

// }

// }

// count = other.count;

// adress = other.adress;

//

// (\*count)++;

// return \*this;

// }

//

// ~SharedPtr()

// {

// if (count != nullptr)

// {

// (\*count)--;

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

// {

// delete adress;

// delete count;

// }

// }

// }

//

//};

//

//int main()

//{

// SharedPtr<int>obj(new int(100));

// SharedPtr<int>obj2;

// obj2 = obj;

//

// return 0;

//}