#include<iostream>

using namespace std;

class ListNode {

public:

int data;

ListNode\* next;

ListNode\* random;

ListNode(int x) :data(x), next(nullptr), random(nullptr) {}

};

ListNode\* copy(ListNode\* phead) {

if (phead == nullptr)

return nullptr;

//复制每一个节点

ListNode\* pnode = phead;

while (pnode != nullptr) {

ListNode\* temp = new ListNode(pnode->data);

temp->next = pnode->next;

temp->random = nullptr;

pnode->next = temp;

pnode = temp->next;

}

//复制random指针

pnode = phead;

while (pnode != nullptr) {

ListNode\* p = pnode->next;

if (pnode->random != nullptr) {

p->random = pnode->random->next;

}

pnode = p->next;

}

ListNode\* temp = nullptr;

ListNode\* ptemp = nullptr;

pnode = phead;

temp = ptemp = phead->next;

pnode->next = temp->next;

pnode = pnode->next;

while (pnode != nullptr) {

temp->next = pnode->next;

temp = temp->next;

pnode->next = temp->next;

pnode = pnode->next;

}

return ptemp;

}

int main() {

ListNode\* node1 = new ListNode(1);

ListNode\* node2 = new ListNode(2);

ListNode\* node3 = new ListNode(3);

ListNode\* node4 = new ListNode(4);

node1->next = node2;

node2->next = node3;

node3->next = node4;

node1->random = node4;

node2->random = node3;

node3->random = node2;

node4->random = node1;

ListNode\* head = node1;

ListNode\* q1 = copy(head);

while (q1) {

if (q1 != nullptr) {

cout << "(" << q1->data << "," << q1->random->data << ")" << "->";

q1 = q1->next;

}

}

return 0;

}