#include<bits/stdc++.h>

using namespace std;

class node{

public:

int data;

node\* next;

};

class linked\_list{

private:

node\* head;

node\* tail;

int size;

public:

linked\_list(){

head = NULL;

tail = NULL;

size = 0;

}

node\* gethead(){

return head;

}

node\* gettail(){

return tail;

}

int getsize(){

return size;

}

void addlast(int value){

node\* temp = new node();

temp->data = value;

temp->next = NULL;

if(head==NULL){

head = temp;

tail = temp;

}

else{

tail->next = temp;

tail = temp;

}

size++;

}

void display(){

node\* temp = head;

while(temp!=0){

cout<<temp->data<<" ";

temp = temp->next;

}

cout<<endl;

}

static int intersection(linked\_list one, linked\_list two){

node\* h1 = one.gethead();

node\* h2 = two.gethead();

int s1 = one.getsize();

int s2 = two.getsize();

int diff = abs(s1-s2);

if(s1>s2){

for(int i=0;i<diff;i++){

h1 = h1->next;

}

}

else{

for(int i=0;i<diff;i++){

h2 = h2->next;

}

}

while(h1->data!=h2->data){

h1 = h1->next;

h2 = h2->next;

}

return h1->data;

}

static int inter(node\* h1, node\* h2){

node\* p1 = h1;

node\* p2 = h2;

while(p1!=NULL && p2!=NULL && p1!=p2){

p1 = p1->next;

p2 = p2->next;

if(p1->data = p2->data){

return p1->data;

}

if(p1==NULL){

p1 = h2;

}

if(p2==NULL){

p2 = h1;

}

}

return p1->data;

}

};

int main(){

linked\_list l1;

linked\_list l2;

int n,m;

cin>>n;

for(int i=0;i<n;i++){

int num;

cin>>num;

l1.addlast(num);

}

cin>>m;

for(int i=0;i<m;i++){

int num;

cin>>num;

l2.addlast(num);

}

int res = linked\_list::intersection(l1,l2);

int ans = linked\_list::inter(l1.gethead(),l2.gethead());

cout<<res<<endl;

cout<<ans<<endl;

}