#include <iostream>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

using namespace std;

/\*

\* 序列 A 和序列 B都是升序序列，A的长度和 B一致，求两个序列联合在一起并排序后的中位数

\*/

//获取以\*a指向的数组 下标 p到 r之间的中数

int getMiddle(int \*a, int p, int r, int &x){

x = a[(p+r)/2];

return x;

}

int getMiddleBetwAB(int \*a, int \*b, int length) {

int p1 = 0, p2 = 0, r1 = length-1, r2 = length-1, x1 = -1, x2 = -1;

while(getMiddle(a, p1, r1, x1) != getMiddle(b, p2, r2, x2) && (r1!=p1) && (r2!=p2)) {

if((r1-p1)%2 == 0) { //奇数个元素

if(x1 < x2) {

p1 = (p1+r1)/2+1;

r2 = (p2+r2)/2-1;

} else {

r1 = (p1+r1)/2-1;

p2 = (p2+r2)/2+1;

}

} else if((r1-p1)%2 == 1) { //偶数个元素

if(x1 < x2) {

p1 = (p1+r1)/2+1;

r2 = (p2+r2)/2;

} else {

r1 = (p1+r1)/2;

p2 = (p2+r2)/2+1;

}

}

}

return x1<x2?x1:x2;

}

int main(int argc, char\*\* argv) {

int A[11] = {1,4,6,8,12,14,15,20,45,60,70};

int B[11] = {0,2,5,6,17,19,21,25,38,47,58};

int \*a = A;

int \*b = B;

int length = sizeof(A)/sizeof(A[0]);

int result = getMiddleBetwAB(a,b,length);

cout << result << endl;

return 0;

}

