#include<iostream>

using namespace std;

//void initArray(int\*& arr, const int& size) {

// int min = 1;

// int max = 100;

// int random = 0;

// for (size\_t i = 0; i < size; i++)

// {

// random = min + rand() % (max - min);

// arr[i] = random;

// }

//}

//

//void printArray(int\*& arr, const int& size) {

// for (size\_t i = 0; i < size; i++)

// {

// cout << arr[i] << " ";

// } cout << endl;

//}

//void swapArray(int\*& arr, const int& size) {

// for (size\_t i = 0; i < size/2; i++)

// {

// int temp = arr[i];

// arr[i] = arr[size-i - 1];

// arr[size - i - 1] = temp;

// }

//}

//void main() {

//

// /\*

// Pointerlərdən istifadə edərək, ölçüsü 10 olan massivdə elementlərin

//sırasını tərsinə çevirən program yazın.

// \*/

//

// int size = 0;

// cin >> size;

// auto arr = new int[size] {};

// initArray(arr, size);

// printArray(arr, size);

// swapArray(arr, size);

// printArray(arr, size);

//}

//void initArray(int\*& arr, const int& size) {

// int min = 1;

// int max = 10;

// int random = 0;

// for (size\_t i = 0; i < size; i++)

// {

// random = min + rand() % (max - min);

// arr[i] = random;

// }

//}

//

//void printArray(int\*& arr, const int& size) {

// for (size\_t i = 0; i < size; i++)

// {

// cout << arr[i] << " ";

// } cout << endl;

//}

//

//void SetNotIntersectArray(int\*& arr1, int\*& arr2,int& size1, const int& size2) {

// auto newarray = new int[size1 + size2]{};

// int index = 0;

// for (size\_t i = 0; i < size1; i++)

// {

// bool hasFound = false;

// for (size\_t k = 0; k < size2; k++)

// {

// if (arr1[i] == arr2[k]) {

// hasFound = true;

// break;

// }

// }

// if (!hasFound) {

// newarray[index] = arr1[i];

// ++index;

// }

// }

//

// for (size\_t i = 0; i < size2; i++)

// {

// bool hasFound = false;

// for (size\_t k = 0; k < size1; k++)

// {

// if (arr2[i] == arr1[k]) {

// hasFound = true;

// break;

// }

// }

// if (!hasFound) {

// newarray[index] = arr2[i];

// ++index;

// }

// }

//

//

// delete[]arr1;

// arr1 = newarray;

// newarray = nullptr;

// size1 = index;

//

//}

//

//

//void main() {

//

//

// int size1 = 0;

// cin >> size1;

// auto arr1 = new int[size1] {};

// initArray(arr1, size1);

// printArray(arr1, size1);

//

//

// int size2 = 0;

// cin >> size2;

// auto arr2 = new int[size2] {};

// initArray(arr2, size2);

// printArray(arr2, size2);

// SetNotIntersectArray(arr1, arr2, size1, size2);

// printArray(arr1, size1);

//

//

//}

//Task 1

int getSecondSmallestElement(int\*& arr, const int& size);

//Task 2 massivde en chox tekrarlanan element ve neche defe tekrarlandigini

//massive yazib qaytarin

//meselen 2 3 4 4 4 4 4 5 5 5 6 7 7 => 4 ededinden 5 ededdir

//massivde olmalidir [4,5]