[B - XOR Game](https://vjudge.net/problem/CodeChef-XORGM" \t "_blank)

 You are given an odd integer *N* and two integer sequences *A*1​,*A*2​,…,*AN*​ and *B*1​,*B*2​,…,*BN*​.

Your task is to reorder the elements of *B*, forming a new sequence  *C*1​,*C*2​,…,*CN*​ (i.e. choose a permutation *P*1​,*P*2​,…,*PN*​ of the integers 1 through *N*, where  *Ci*​=*BPi*​​ for each valid i), in such a way that the following condition holds: (*A*1​⊕*C*1​)=(*A*2​⊕*C*2​)=…=(*AN*​⊕*CN*​), where ⊕ denotes bitwise XOR. Find one such reordered sequence or determine that it is impossible.

### Input

The first line of the input contains a single integer *T* denoting the number of test cases. The description of *T* test cases follows.

The first line of each test case contains a single integer *N*.

The second line contains *N* space-separated integers *A*1​,*A*2​,…,*AN*​.

The third line contains N*N* space-separated integers *B*1​,*B*2​,…,*BN*​.

### Output

For each test case:

* If there is no valid way to reorder the sequence *B*, print a single line containing the integer −1.
* Otherwise, print a single line containing N*N* space-separated integers *C*1​,*C*2​,…,*CN*​. If there are multiple solutions, you may find any one.

### Constraints

* 1≤*T*≤10
* 1≤*N*≤105
* *N* is odd
* 0≤*Ai*​≤106 for each valid i
* 0≤*Bi*​≤106 for each valid i

### Example Input

1

5

3 1 2 4 5

2 4 5 1 3

### Example Output

3 1 2 4 5

### **XOR Game**

Problem statement has been made confusing intentionally.  Simplest problem statement is

Given two arrays A and B, can we rearrange element of B in such a way that after re-arrangement of B elements,

Each element of A XORED with each corresponding element of re-arranged B becomes same value.

Ai ^ Bi is same for i=[1, N] assuming 1-based array indexing.

**CATCH**:  N is given to be ODD.

Let's assume that XOR value is **X**.

A1 ^ B1 = A2 ^ B2 = ..... = AN ^ BN = **X**

What will we get if we XOR each of above PART together?  Since N is ODD, it should be easy to find it out.

=> (A1 ^ B1) XOR (A2 ^ B2) .... XOR (AN ^ BN)

=> X ^ X ....... ^ X = X (because there are odd terms and each term is X,  X xored with itself even times = 0, X xored with itself odd times = X)

CONCLUSION

(A1 ^ B1) XOR (A2 ^ B2) .... XOR (AN ^ BN) = X

Re-Arranging above equation, A's together and B's together (because XOR is ASSOCIATIVE AND COMMUTATIVE),

(A1 ^ A2^ A3 ....... ^ AN) XOR (B1 ^ B2^ B3 ....... ^ BN) = X

So finding X became easy.  We can XOR all value of A with all values of B together to get X.

Once we got X, using A and X, we can easily construct array C.

How we've got X value as well as we know A.

Let's try to construct C, using array A and X value.

if A XOR C = X, Doing XOR on both sides with A will make it

A XOR C XOR A= X XOR A

C = X XOR A

So all that is needed now is to create C and then just figure out if C is a permutation of B.  Just SORT C and SORT B and compare them if they are same.