Permutation Exponentiation

Alice, a permutation aficionado, has thought up a permutation of **N** (1 <= **N** <= 100000) integers in the range [0, **N**-1], **P**. So impressed with herself she has told her friend Bob about **P**.

Normally Alice would call it a day after creating such an impressive permutation but today she decided that she wanted to raise **P** to the power **k** (a positive integer) as well! Unfortunately, after working on the problem for a while she gave up because it was taking too long. Not wanting her efforts to go to waste she once again tells Bob about all the elements she has determined so far. Unfortunately, she neglected to tell Bob the value **k**.

Bob, very interested in Alice's work, needs your help to try and determine any additional elements of **P^k**. Bob is suspicious of Alice's work so he also asks you to check it for errors.

Notes

For two permutations **P**, **Q**, the **i**th element of the product, **(P * Q)[i]**, can be computed as **Q[P[i]]** where arrays are all 0-indexed.

Then **P^k** is simply **P** * **P** * ... **k** times ... * **P**.

Input

The first line of input contains **N** (1 <= **N** <= 100000), the number of elements in the permutation. The next line contains a permutation of the integers 0 through **N**-1 each separated by a space. The following line will contain the result of applying the permutation **k** times with the exception that elements that are not known will be -1 instead.

Output

Print **P^k** as a space separated list on its own line with as many elements as possible determined. If an element can't be determined leave it as -1. If there is no **k** such that **P^k** has the values given in the input print "Inconsistent" (quotes for clarity) on its own line instead.

Example

Input:
4
1 2 3 0
3 -1 -1 -1

Output:
3 0 1 2

Note: The first four permutations generated are

1 2 3 0
2 3 0 1
3 0 1 2
0 1 2 3

Example 2

Input:
4
1 2 3 0
3 -1 2 -1

Output:
Inconsistent