Problem J: Kernel Knights

Time limit: 2 s
Memory limit: 512 MiB

Jousting is a medieval contest that involves people on horseback trying to strike each other with wooden lances while riding at high speed. A total of $2n$ knights have entered a jousting tournament – n knights from each of the two great rival houses. Upon arrival, each knight has challenged a single knight from the other house to a duel.

A *kernel* is defined as some subset S of knights with the following two properties:

- No knight in S was challenged by another knight in S.
- Every knight not in S was challenged by some knight in S.

Given the set of the challenges issued, find one kernel. It is guaranteed that a kernel always exists.

Input

The first line contains an integer n ($1 \leq n \leq 100\,000$) – the number of knights of each house. The knights from the first house are denoted with integers 1 through n, knights from the second house with integers $n+1$ through $2n$.

The following line contains integers $f_1, f_2, \ldots, f_n$ – the k-th integer f_k is the index of the knight challenged by knight k ($n+1 \leq f_k \leq 2n$).

The following line contains integers $s_1, s_2, \ldots, s_n$ – the k-th integer s_k is the index of the knight challenged by knight $n+k$ ($1 \leq s_k \leq n$).

Output

Output the indices of the knights in the kernel on a single line. If there is more than one solution, you may output any one.

Example

input

```
4
5 6 7 7
1 3 2 3
```

output

```
1 2 4 8
```