C. Pavel and barbecue

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Pavel cooks barbecue. There are n skewers, they lay on a brazier in a row, each on one of n positions. Pavel wants each skewer to be cooked some time in every of n positions in two directions: in the one it was directed originally and in the reversed direction.

Pavel has a plan: a permutation p and a sequence $b_1, b_2, ..., b_n$, consisting of zeros and ones. Each second Pavel move skewer on position i to position p_i, and if b_i equals 1 then he reverses it. So he hope that every skewer will visit every position in both directions.

Unfortunately, not every pair of permutation p and sequence b suits Pavel. What is the minimum total number of elements in the given permutation p and the given sequence b he needs to change so that every skewer will visit each of $2n$ placements? Note that after changing the permutation should remain a permutation as well.

There is no problem for Pavel, if some skewer visits some of the placements several times before he ends to cook. In other words, a permutation p and a sequence b suit him if there is an integer k ($k \geq 2n$), so that after k seconds each skewer visits each of the $2n$ placements.

It can be shown that some suitable pair of permutation p and sequence b exists for any n.

Input

The first line contain the integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of skewers.

The second line contains a sequence of integers $p_1, p_2, ..., p_n$ ($1 \leq p_i \leq n$) — the permutation, according to which Pavel wants to move the skewers.

The third line contains a sequence $b_1, b_2, ..., b_n$ consisting of zeros and ones, according to which Pavel wants to reverse the skewers.

Output

Print single integer — the minimum total number of elements in the given permutation p and the given sequence b he needs to change so that every skewer will visit each of $2n$ placements.

Examples

input

```
4
4 3 2 1
0 1 1 1
```

output

```
2
```

input

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

output

```
1
```

Note

In the first example Pavel can change the permutation to $4, 3, 1, 2$.

In the second example Pavel can change any element of b to 1.