C. Permutations

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Happy PMP is freshman and he is learning about algorithmic problems. He enjoys playing algorithmic games a lot.

One of the seniors gave Happy PMP a nice game. He is given two permutations of numbers 1 through n and is asked to convert the first one to the second. In one move he can remove the last number from the permutation of numbers and inserts it back in an arbitrary position. He can either insert last number between any two consecutive numbers, or he can place it at the beginning of the permutation.

Happy PMP has an algorithm that solves the problem. But it is not fast enough. He wants to know the minimum number of moves to convert the first permutation to the second.

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the quantity of the numbers in the both given permutations.

Next line contains n space-separated integers — the first permutation. Each number between 1 to n will appear in the permutation exactly once.

Next line describe the second permutation in the same format.

Output

Print a single integer denoting the minimum number of moves required to convert the first permutation to the second.

Examples

input

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

output

```
2
```

input

```
5
1 2 3 4 5
1 5 2 3 4
```

output

```
1
```

input

```
5
1 5 2 3 4
1 2 3 4 5
```

output

```
3
```

Note

In the first sample, he removes number 1 from end of the list and places it at the beginning. After that he takes number 2 and places it between 1 and 3.

In the second sample, he removes number 5 and inserts it after 1.

In the third sample, the sequence of changes are like this:

- 1 5 2 3 4
- 1 4 5 2 3
- 1 3 4 5 2
- 1 2 3 4 5

So he needs three moves.