Beautiful Pairs

You are given two arrays, A and B, both containing N integers.

A pair of indices (i, j) is *beautiful* if the i^{th} element of array A is equal to the j^{th} element of array B. In other words, pair (i, j) is *beautiful* if and only if $A[i] = B[j]$. A set containing beautiful pairs is called a *beautiful set*.

A beautiful set is called *pairwise disjoint* if for every pair $(l[i], r[i])$ belonging to the set there is no repetition of either $l[i]$ or $r[i]$ values. For instance, if $A = [10, 11, 12, 5, 14]$ and $B = [8, 9, 11, 11, 5]$ the beautiful set $[(1, 2), (1, 3), (3, 4)]$ is not pairwise disjoint as there is a repetition of 1, that is $l[0][0] = l[1][0]$.

Your task is to change **exactly** 1 element in B so that the size of the pairwise disjoint beautiful set is maximum.

Function Description

Complete the *beautifulPairs* function in the editor below. It should return an integer that represents the maximum number of pairwise disjoint paris that can be formed.

beautifulPairs has the following parameters:

- *A*: an array of integers
- *B*: an array of integers

Input Format

The first line contains a single integer n, the number of elements in A and B.
The second line contains n space-separated integers $A[i]$.
The third line contains n space-separated integers $B[i]$.

Constraints

- $1 \leq n \leq 10^3$
- $1 \leq A[i], B[i] \leq 10^3$

Output Format

Determine and print the maximum possible number of pairwise disjoint beautiful pairs.

Note: You must first change 1 element in B, and your choice of element must be optimal.

Sample Input 0

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

Sample Output 0

```
4
```

Explanation 0

You are given $A = [1, 2, 3, 4]$ and $B = [1, 2, 3, 3]$.
The beautiful set is $[(0, 0), (1, 1), (2, 2), (2, 3)]$ and maximum sized pairwise disjoint beautiful set is either $[(0, 0), (1, 1), (2, 2)]$ or $[(0, 0), (1, 1), (2, 3)]$.
We can do better. We change the 3^{rd} element of array B from 3 to 4. Now new B array is: $B = [1, 2, 4, 3]$ and the pairwise disjoint beautiful set is $[(0, 0), (1, 1), (2, 3), (3, 2)]$. So, the answer is 4.

Note that we could have also selected index 3 instead of index 2 but it would have yeilded the same result. Any other choice of index is not optimal.

Sample Input 1

```
6
3 5 7 11 5 8
5 7 11 10 5 8
```

Sample Output 1

```
6
```