Minimum Swaps 2

You are given an unordered array consisting of consecutive integers ∈ [1, 2, 3, ..., n] without any duplicates. You are allowed to swap any two elements. You need to find the minimum number of swaps required to sort the array in ascending order.

For example, given the array $arr = [7, 1, 3, 2, 4, 5, 6]$ we perform the following steps:

```
i   arr                         swap (indices)
0   [7, 1, 3, 2, 4, 5, 6]   swap (0,3)
1   [2, 1, 3, 7, 4, 5, 6]   swap (0,1)
2   [1, 2, 3, 7, 4, 5, 6]   swap (3,4)
3   [1, 2, 3, 4, 7, 5, 6]   swap (4,5)
4   [1, 2, 3, 4, 5, 7, 6]   swap (5,6)
5   [1, 2, 3, 4, 5, 6, 7]
```

It took 5 swaps to sort the array.

Function Description

Complete the function *minimumSwaps* in the editor below. It must return an integer representing the minimum number of swaps to sort the array.

minimumSwaps has the following parameter(s):

- *arr*: an unordered array of integers

Input Format

The first line contains an integer, n, the size of arr.
The second line contains n space-separated integers $arr[i]$.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq arr[i] \leq n$

Output Format

Return the minimum number of swaps to sort the given array.

Sample Input 0

```
4
4 3 1 2
```

Sample Output 0

```
3
```

Explanation 0

Given array $arr : [4, 3, 1, 2]$
After swapping $(0, 2)$ we get $arr : [1, 3, 4, 2]$
After swapping $(1, 2)$ we get $arr : [1, 4, 3, 2]$
After swapping $(1, 3)$ we get $arr : [1, 2, 3, 4]$
So, we need a minimum of 3 swaps to sort the array in ascending order.

Sample Input 1

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

Sample Output 1

```
3
```

Explanation 1

Given array $arr : [2, 3, 4, 1, 5]$
After swapping $(2, 3)$ we get $arr : [2, 3, 1, 4, 5]$
After swapping $(0, 1)$ we get $arr : [3, 2, 1, 4, 5]$
After swapping $(0, 2)$ we get $arr : [1, 2, 3, 4, 5]$
So, we need a minimum of 3 swaps to sort the array in ascending order.

Sample Input 2

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

Sample Output 2

```
3
```

Explanation 2

Given array $arr : [1, 3, 5, 2, 4, 6, 7]$
After swapping $(1, 3)$ we get $arr : [1, 2, 5, 3, 4, 6, 7]$
After swapping $(2, 3)$ we get $arr : [1, 2, 3, 5, 4, 6, 7]$
After swapping $(3, 4)$ we get $arr : [1, 2, 3, 4, 5, 6, 7]$
So, we need a minimum of 3 swaps to sort the array in ascending order.

CODE:

```
n=int(input())
a=list(map(int,input().split()))
c=0
d={}

for i in range(n):
    d[a[i]-1]=i

for i in range(1,n+1):
    temp=d[i-1]
    if(temp!=i-1):
        d[i-1],d[a[i-1]-1]=d[a[i-1]-1],d[i-1]
        a[temp],a[i-1]=a[i-1],a[temp]
        c+=1

print(c)
```