Karl has an array of integers. He wants to reduce the array until all remaining elements are equal. Determine the minimum number of elements to delete to reach his goal.

For example, if his array is $arr = [1, 2, 2, 3]$, we see that he can delete the 2 elements 1 and 3 leaving $arr = [2, 2]$. He could also delete both twos and either the 1 or the 3, but that would take 3 deletions. The minimum number of deletions is 2.

Function Description

Complete the *equalizeArray* function in the editor below. It must return an integer that denotes the minimum number of deletions required.

equalizeArray has the following parameter(s):

- *arr*: an array of integers

Input Format

The first line contains an integer n, the number of elements in arr.
The next line contains n space-separated integers $arr[i]$.

Constraints

- $1 \leq n \leq 100$
- $1 \leq arr[i] \leq 100$

Output Format

Print a single integer that denotes the minimum number of elements Karl must delete for all elements in the array to be equal.

Sample Input

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

Sample Output

```
2
```

Explanation

Array $arr = [3, 3, 2, 1, 3]$. If we delete $arr[2] = 2$ and $arr[3] = 1$, all of the elements in the resulting array, $A' = [3, 3, 3]$, will be equal. Deleting these 2 elements is minimal. Our only other options would be to delete 4 elements to get an array of either $[1]$ or $[2]$.