Problem: Equality in an Array

Karl has an array of  integers defined as . In one operation, he can delete any element from the array.

Karl wants all the elements of the array to be equal to one another. To do this, he must delete zero or more elements from the array. Find and print the *minimum* number of deletion operations Karl must perform so that all the array's elements are equal.

**Input Format**

The first line contains an integer, , denoting the number of elements in array .   
The next line contains  space-separated integers where element  corresponds to array element  (.

**Constraints**

**Output Format**

Print a single integer denoting 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 . If we delete  and , all of the elements in the resulting array, , will be equal. Deleting these  elements is minimal because our only other options would be to delete  elements to get an array of either  or . Thus, we print  on a new line, as that is the minimum number of deletions resulting in an array where all elements are equal.

Solution

int main()

{

int size;

cin>>size;

int array[size];

int separate[100]={0};

int temp=0;

for(int i=0; i<size; i++)

{

cin>>temp;

array[i]=temp;

separate[temp]++;

}

int max=0;

for(int i=1; i<100; i++)

{

(separate[i]>separate[max] ? max=i : max+=0);

}

cout<<size-separate[max];

return 0;

}

* Anshul Aggarwal