

Login

Username

Password

☐ Remember Me[Password forgotten?](#)

Navigation

[Register](#)[Learn C++](#)[Algorithms](#)[Training](#)[Competitions](#)[Forums](#)[Wiki](#)[Help](#)[Report Bug](#)

Swaps

As a programmer you are familiar that for every integer there is a binary representation of the number. For example, 4 is 100 in binary, 5 is 101, 15 is 1111... Every number can be represented with leading zeros, so 5 can be represented also as 0101, 000000101, ...

In this task, you are given a non-sorted list of N integers, each between 1 and 4095, inclusive. You should make changes in the list, so the result will be a new list of integers that are sorted in non-decreasing order.

The only operation permitted for making changes is swapping two bits in the binary representation of a specific number.

Output the minimum number of swaps necessary to reach the goal – a new sorted list. It will always be possible to reach the goal.

Note: all the numbers are considered to have exactly 12 bits for swap purposes. So, 1 would be represented as 000000000001 and with one swap we could get 512 = 001000000000, or even 2048 = 1000000000000; but not 3 = 000000000011 (because you're only allowed to swap two bits in the specific number), or 4096 (because 4096 can't be represented with 12 bits).

Input

The input is in the following format: line 1 contains N , the size of the list ($1 \leq N \leq 2000$). In the next N lines, you have the numbers A_i , the i -th number in the list for $0 \leq i < N$.

Output

The output consists of a single integer - the count of minimum swaps.

Constraints

Time limit: 1 second

Memory limit: 64 megabytes

Examples

input	output
5	2
2	
9	
8	
12	
11	

input	output
3	1
3	
2	
7	

Explanation for the first test case: We can swap the 2nd and 4th bit of the binary representation of 9 (000000001001) to get 3 (000000000011). Similarly with one swap in the number 11 (000000001011) we can get 14 (000000001110). In total, we need 2 swaps to get the sorted list: 2,3,8,12,14. In this example 2 is the minimal number of swaps that have to be done to reach the goal.

Explanation for the second test case: 2 = 000000000010, and we can swap the 2-nd with the 3-rd bit to get 4=000000000100.

Note: You are now solving the task on the MENDO grading system. During the actual JBOI 2016 contest, for this task, there were a number of subtasks earning different amount of points. They were as follows:

- For 13 points, $N \leq 10$, and each of the numbers in the list, when represented as a binary number, will have at most 4 bits equal to 1.
- For 11 points, $N \leq 100$, and each of the numbers in the list, when represented as a binary number, will have at most 8 bits equal to 1. The minimal swaps can be accomplished by making at most 1 swap

in each number in the list.

- For 25 points, $N \leq 100$, and each of the numbers in the list, when represented as a binary number, will have at most 8 bits equal to 1.
- For 12 points, $N \leq 2000$, and the minimal swaps can be accomplished by making at most 1 swap in each number in the list.
- For 39 points, $N \leq 2000$.

Submit **Test**

Submit your code

Solution language:

Automatic Detection...

Upload solution (option 1):

Select

Paste your code (option 2):



 CodeMirror

Submit!