Tax Fraud Detection (fraud)

Edoardo has been hired by the Inland Revenue Agency with the task of detecting fraudulent tax forms. After months of thorough analysis, he has finally found a measure of *fraudulence*!

Figure 1: Fraudulent tax forms.

You are given a tax form, represented as an array of N elements V_i for $i = 0 \dots N-1$. The *fraudulence* of the sub-array $[V_i, \dots V_j]$ (for $0 \le i \le j \le N-1$) is defined as the product of the frequency of the rarest element (the smallest number of times an element appears in the sub-array) with the frequency of the most common element (the highest number of times an element appears in the sub-array). For example,

$$[1, 3, 2, 3, 1, 2, 3]$$

has fraudulence $6 = 2 \times 3$ since the rarest elements (1 and 2) appear twice, while the most common element (3) appears 3 times. Compute the maximum fraudulence for a sub-array of the given array!

☞ Among the attachments of this task you may find a template file `fraud.*` with a sample incomplete implementation.

Input

The first line contains the only integer N. The second line contains N integers V_i.

Output

You need to write a single line with an integer: the maximum fraudulence for a sub-array of the given array.

Constraints

- $1 \le N \le 100\,000$.
- $1 \le V_i \le 100\,000$ for each $i = 0 \dots N-1$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (15 points) $N \leq 100$.

– **Subtask 3** (20 points) $N \leq 1000$, $V_i \leq 200$ for each $i = 0 \ldots N-1$.

– **Subtask 4** (30 points) $N \leq 2000$.

– **Subtask 5** (35 points) No additional limitations.

Examples

input	output
5 1 2 1 3 2	2
7 7 4 2 2 4 1 4	4

Explanation

In the **first sample case**, you may choose either $[1, 2, 1]$ or $[1, 2, 1, 3, 2]$ or $[2, 1, 3, 2]$: all of them have the lowest frequency equal to one (attained by numbers 2, 3, 1 and 3 respectively), and the highest frequency equal to two (attained by 1, 1 and 2, 2 respectively). Altogether, the fraudulence of any of those sub-arrays is $1 \times 2 = 2$.

In the **second sample case**, the highest fraudulence is attained by sub-array $[4, 2, 2, 4]$ which has both lowest and highest frequencies equal to 2.