

Problem A. Paint The Fence

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Pasha and Polina decided to paint the fence. The fence is made of n boards (enumerated from 1 to n). Pasha painted all the boards from a -th to b -th (inclusively), and Polina painted all the boards from c -th to d -th (inclusively).

Unfortunately, they did not talk over with each other, therefore some boards were painted twice, and some boards were not painted at all.

Your task is to determine how many boards **were not painted at all**.

Input

The first line contains one integer n ($1 \leq n \leq 10^9$) — the number of boards.

The second line contains two integers a and b ($1 \leq a \leq b \leq n$) — Pasha's painting range.

The third line contains two integers c and d ($1 \leq c \leq d \leq n$) — Polina's painting range.

Output

Print the number of boards that **were not painted at all**.

Scoring

Subtask	Points	Constraints	Comment
0	0	—	Sample tests.
1	40	$1 \leq n \leq 1000$	Points are awarded if all tests of this and previous subtasks are passed.
2	60	Main constraints	Points are awarded if all tests of this and previous subtasks are passed.

Examples

standard input	standard output
10 1 2 5 7	5
10 2 5 4 7	4
10 10 10 8 9	7
10 5 8 6 7	6

Problem B. Alice's Transformation

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Alice have an array a consisting of n integers. She loves to make different array transformations, and today she come up with the following algorithm:

1. She searches for three identical consecutive integers;
2. If there is any tuple of three identical consecutive integers, then she deletes one of these integers, and go back to the step 1, otherwise she finished the algorithm.

Your task is to determine the resulting array after applying the algorithm described previously.

Input

The first line contains one integer n ($1 \leq n \leq 2 \cdot 10^5$) — the size of an array a .

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — the elements of an array a .

Output

Print the resulting array.

Scoring

Subtask	Points	Constraints	Comment
0	0	—	Sample tests.
1	40	$1 \leq n \leq 100$	Points are awarded if all tests of this and previous subtasks are passed.
2	60	Main constraints	Points are awarded if all tests of this and previous subtasks are passed.

Examples

standard input	standard output
6 4 3 10 10 10 1	4 3 10 10 1
5 1 1 1 1 1	1 1
5 1 2 3 4 9	1 2 3 4 9

Problem C. Interesting lecture

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Today Vasya visited a lecture on the topic of arrays. Lecture was really interesting, but given homework is also fascinating. Here it is:

*You are given an array of size n . Your task is to get a subarray of maximum possible length, where all elements are equal, using **at most** one reverse of some subarray.*

Vasya loved this task, but he is too busy tonight, so he asked you to help him.

Just to remind, reverse of a subarray from position l to position r (inclusively) transforms an array of elements $a_1, \dots, a_{l-1}, a_l, \dots, a_r, a_{r+1}, \dots, a_n$ to an array of elements $a_1, \dots, a_{l-1}, a_r, a_{r-1}, \dots, a_{l+1}, a_l, a_{r+1}, \dots, a_n$

Input

The first line contains one integer n ($1 \leq n \leq 2 \cdot 10^5$) — the size of an array a .

The second line contains n integers a_i ($1 \leq a_i \leq 10^6$) — the elements of an array a .

Output

Print the maximum possible length of subarray of equal elements, that can be received by using **at most** one reverse of some subarray.

Scoring

Subtask	Points	Constraints	Comment
0	0	—	Sample tests.
1	40	$1 \leq n \leq 100$	Points are awarded if all tests of this and previous subtasks are passed.
2	60	Main constraints	Points are awarded if all tests of this and previous subtasks are passed.

Examples

standard input	standard output
5 1 3 1 3 1	2
6 1 1 3 2 2 2	3
6 1 1 3 4 1 1	4
5 1 2 3 4 5	1

Note

In the first sample the best answer can be received by reversing subarray of positions [2; 3].

In the second sample you do not need to reverse any subarray to get the best answer.

In the third sample the best answer can be received by reversing subarray of positions [1; 4].

Problem D. The Bottles

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

Masha has n empty bottles (enumerated from 1 to n). Each bottle can store unlimited amount of water. One operation is defined as the following:

1. She selects two bottles: l -th bottle and r -th bottle ($l < r$). Also, she guarantees that there are **even** number of bottles between the selected ones, so the bottle in the middle has the number $m = \frac{l+r}{2}$;
2. For each position i from l to m (inclusively): she pours $i - l + 1$ litres of water to bottle number i ;
3. For each position j from $m + 1$ to r (inclusively): she pours $r - j + 1$ litres of water to bottle number j ;

What is the maximum amount of water (in litres) will have some bottle after k such operations?

Input

The first line contains two integers n and k ($1 \leq n \leq 10^5, 1 \leq k \leq 10^5$) — the number of bottles and the number of operations respectively.

Each of the following k lines contain two integers l and r ($1 \leq l < r \leq n$) — operations' description.

Output

Print the single integer — the maximum amount of water (in litres) after processing all the operations.

Scoring

Subtask	Points	Constraints	Comment
0	0	—	Sample tests.
1	40	$1 \leq n \leq 1000, 1 \leq k \leq 1000$	Points are awarded if all tests of this and previous subtasks are passed.
2	60	Main constraints	Points are awarded if all tests of this and previous subtasks are passed.

Examples

standard input	standard output
7 1 3 7	3
7 2 1 3 5 7	2
7 2 2 6 1 5	5

Note

In the first example, the bottles after all the operations: $[0, 0, 1, 2, 3, 2, 1]$

In the second example, the bottles after all the operations: $[1, 2, 1, 0, 1, 2, 1]$

In the third example, the bottles after all the operations: $[1, 3, 5, 5, 3, 1, 0]$