Hora Unirii

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

> *Where there's only one, there's no strength*
> *In need or in pain*
> *Where there are two, strength grows*
> *And the enemy does not gain ground*
>
> Vasile Alecsandri - "Hora Unirii"

We all know those lines. But have you ever wondered what happens when there are three? Or more? Stephen the Great, a very clever ruler, seems to have already known the answer. He understood that where there are three, strength wanes. For the upcoming Battle of Vaslui, he needed a solid army. That's why he decided to organize his troops into pairs (wherever possible), so they could pool their strength and successfully face Suleiman Pasha's forces.

You have n ($1 \leq n \leq 200\ 000$) soldiers standing in a fixed line (you can't swap their order), labeled from 1 to n. Each soldier has a natural-number power value. Two soldiers make the best team if and only if they have exactly the same value.

The ruler's plan is simple: he wants to pick a contiguous subarray of soldiers such that it can be partitioned perfectly into pairs, each pair consisting of two soldiers with equal power.

Finding such a subarray quickly can be challenging, so he's turned to you. You must answer q ($1 \leq q \leq 200\ 000$) queries — each query gives you the endpoints L, R of a subarray, and you must respond with exactly one of these four messages:

- `!1` – all soldiers in the subarray have distinct power values;
- `!2` – the subarray can be perfectly partitioned into pairs of equal power (so its length is even and each value appears exactly twice);
- `!1!2` – there are some soldiers whose power is unique within the subarray, and the rest form pairs of equal power; both the number of unique-powered soldiers and the number of pairs must be nonzero;
- `x` – none of the above: there's at least one power value that appears three or more times in the subarray.

Input

The first line contains two integers n (the number of soldiers) and q (the number of queries), and on the second line you read n integers between -10^9 and 10^9 — the sequence of soldier strengths.

Each of the next q lines contains two integers L and R, the left and right endpoints of a query's subarray.

Output

Print one of the four specified messages for each of the q queries, in order.

Example

standard input	standard output

10 4
1 2 3 1 2 2 3 3 3 1
3 5
1 5
1 6
5 8
	!1
	!1!2
	x
	!2

Note

For the first query, the subarray from indices 3 to 5 is $[3, 1, 2]$, so you print `!1`.

For the second query, the subarray from indices 1 to 5 is $[1, 2, 3, 1, 2]$, so you print `!1!2`.