

Range XOR

time limit per test: 1 second (2 seconds for Python)
memory limit per test: 976.6 megabytes
input: standard input
output: standard output
points: 100 · problem code: rangexor

You are given an array of n integers and must answer q range queries.

Each query gives indices l and r and asks for the bitwise XOR of all values in the subarray $a_l, a_{l+1}, \dots, a_r$.

Input

The first line contains the integer n .

The second line contains n integers, the values in the array.

The third line contains the integer q , the number of queries.

Each of the next q lines contains two integers l and r , asking for the XOR of all values in the range $[l, r]$.

Output

For each query, output a line containing the XOR of the requested range.

Constraints

- $1 \leq l \leq r \leq n \leq 2 \cdot 10^5$
- $1 \leq q \leq 2 \cdot 10^5$
- $1 \leq a_i \leq 10^9$

Example 1

Input

```
10
1 5 10 5 3 2 5 10 5 9
4
1 4
7 8
1 3
5 9
```

Output

```
11
15
14
11
```

Example 2

Input

```
10
2 6 7 7 6 10 4 6 8 9
4
5 10
1 6
6 8
1 9
```

Output

```
15
8
8
2
```

Hints

Hint 1

With respect to XOR, every value is its own inverse: $x \oplus a \oplus a = x$.

Hint 2

Think about how prefix sums answer range-sum queries, then replace addition with XOR.