XOR Boba Shop

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	512 megabytes

Ian is very sleepy and requires boba immediately. However, the XOR boba shop wants to do something different and will only sell bobas if their types follow certain properties.

In particular, the XOR boba shop has an array $a_1, a_2, \ldots, a_n$ of non-negative integers, representing the different types of boba.

For a segment $[L, R]$ define $b_k = a_L \oplus a_{L+1} \oplus \cdots \oplus a_k$ for $L \le k \le R$, where $\oplus$ is bitwise XOR. The segment is called *tasty* if

$$b_L \le b_{L+1} \le \cdots \le b_R.$$

A segment of length 1 is always *tasty*.

The boba shop only sells segments of boba that are *tasty*. Thus, Ian wants to compute the largest *tasty* segments for certain starting points. However, Alperen is evil and wants to mess up Ian's boba computations by XORing the array by random numbers at random times. You are tasked with helping Ian navigate the following two queries:

- `1 L` — print the largest $R \ge L$ such that the segment $[L, R]$ is *tasty* (with respect to the current array);
- `2 x` — replace every element by its XOR with x, i.e. $a_i := a_i \oplus x$ for all $1 \le i \le n$. This change is persistent across queries.

Input

The first line contains two integers n and q ($1 \le n \le 10^5$, $1 \le q \le 2 \cdot 10^5$) — the length of the array and the number of queries.

The second line contains n integers $a_1, \ldots, a_n$ ($0 \le a_i < 2^{60}$).

Each of the next q lines describes a query: two integers t and v.

- If $t = 1$, then $v = L$ and $1 \le L \le n$.
- If $t = 2$, then $v = x$ and $0 \le x < 2^{60}$.

Output

For every query of type 1 print one integer — the largest R such that $[L, R]$ is tasty. Print the answers in the order of the queries, one per line.

Examples

standard input	standard output
5 6 3 1 2 0 5 1 1 1 2 2 1 1 1 1 3 1 4	1 5 2 3 5
6 7 4 4 1 1 2 2 1 1 1 2 1 5 2 4 1 1 1 2 1 4	1 3 5 3 3 4