Not-And

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

`NAND`(Not-And) is a bitwise operation defined as follows: a NAND $b :=$ NOT (a AND b) .

Zag is given an array a which consists of n 32-bit unsigned integers. He asks you to perform two types of operations for him, which are described below in detail.

Input

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

The second line contains n integers $a_i (0 \le a_i < 2^{32})$.

Then q lines follow, each of which contains a request of one of the two types:

1. `1 l r x`$(1 \le l \le r \le n, 0 \le x < 2^{32})$, denoting that you need to calculate and output the answer of x NAND a_l NAND a_{l+1} NAND ... NAND a_r.

2. `2 p x`$(1 \le p \le n, 0 \le x < 2^{32})$, denoting that you need to replace the number at position p with the given x.

Output

For every request of the first type, output a line containing the answer.

Example

standard input	standard output
5 5 571 342 228 152 192 1 1 5 409 2 1 414 1 1 2 100 2 4 341 1 2 5 315	4294967103 4294966957 4294967103