XORanges

Janez ima rad pomaranče! Zato si je izdelal skener za pomaranče. S 4 kamerami in Raspberry Pi 3b+ računalnikom, je začel skenirati 3D slike pomaranč. Vendar njegovo procesiranje slik ni prav dobro, edini izhod iz njega je 32-bitno celo število, ki vsebuje informacijo o luknjah v lupini. 32-bitno celo število D je predstavljeno kot zaporedje 32 števk (bitov), ki je vsak lahko 1 ali 0. Če začnemo z 0, dobimo D s prištevanjem 2^i za vsaki i-ti bit, ki ima vrednost 1. Bolj formalno, število D predstavlja zaporedje $d_{31}, d_{30}, \ldots d_0$ ko je $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \ldots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Na primer, število 13 je predstavljeno kot $0, \ldots, 0, 1, 1, 0, 1$.

Janez je poskeniral n pomaranč; vendar, se včasih odloči, da kakšno izmed pomaranč (i-to pomarančo) ponovno poskenira med izvajanjem tvojega programa. To pomeni, da od tega skena naprej, uporablja posodobljeno vrednost za i-to pomarančo.

Janez želi analizirati te pomaranče. Operacija (XOR) se mu zdi zelo zanimiva, zato se odloči, da bo naredil nekaj izračunov. Izbere zaporedje pomaranč od l do u (kjer je $l \leq u$) in želi poiskati vrednost XOR vseh vrednosti v zaporedju, vseh parov zaporednih vrednosti v zaporedju, vseh zaporedij 3 zaporednih vrednosti, ... in zaporedje $u - l + 1$ zaporednih vrednosti (vseh elementov v zaporedju).

Na primer, če je $l = 2$ in $u = 4$ in je A polje skeniranih vrednosti, naj program vrne vrednost $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, kjer $\oplus$ predstavlja opreacijo XOR in a_i predstavlja i-ti element v polju A.

Operacija XOR je definirana kot:

Če je i-ti bit prve vrednosti enak kot i-ti bit druge vrednosti, je i-ti bit rezultata enak `0`; Če je i-ti bit prve vrednosti različen od i-tega bita druge vrednosti, pa je i-ti bit rezultata enak `1`.

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Na primer, $13 \oplus 23 = 26$.

$13 =$	$0 \ldots 001101$
$23 =$	$0 \ldots 010111$

$13 \oplus 23 = 26 =$	$0 \ldots 011010$

Vhod

V prvi vhodni vrstici sta 2 pozitivni celi števili n in q (skupno število ponovnih skeniranj in število analiz - akcij).

V naslednji vrstici je n nenegativnih celih števil ločenih s presledki, ki predstavljajo vrednosti polja A (rezultati skeniranja pomaranč). Element a_i vsebuje vrednost za i-to pomarančo. Indeks i se začne z 1.

Akcije so opisane v naslednjih q vrsticah s tremi pozitivnimi celimi števili ločenimi s presledki.

Če je tip akcije 1 (ponovni scan), je prvo število 1 in mu sledita i (indeks pomaranče, ki ga je Janez želel ponovno skenirati) in j (rezultat ponovnega skeniranja i-te pomaranče).

Če je tip akcije 2 (analiza), je prvo število 2 in mu sledita l in u.

Izhod

Izpiši natanko eno celo število za vsako poizvedbo. Vsaka vrednost naj bo izpisana v novi vrstici. Pozor, i-ta vrstica izhoda naj ustreza rezultatu i-te poizvedbe. Vračajte samo odgovore na akcije tipa 2 (analiza).

Omejitve

- $a_i \leq 10^9$
- $0 < n, q \leq 2 \cdot 10^5$

Podnaloge

1. **[12 točk]**: $0 < n, q \leq 100$
2. **[18 točk]**: $0 < n, q \leq 500$ in ni nobenih rescanov (akcij tipa 1)
3. **[25 točk]**: $0 < n, q \leq 5000$
4. **[20 točk]**: $0 < n, q \leq 2 \cdot 10^5$ in ni nobenih rescanov (akcij tipa 1)
5. **[25 točk]**: Brez dodatnih omejitev.

Primeri

Primer 1

Vhod

```
3 3
1 2 3
2 1 3
1 1 3
2 1 3
```

Izhod

```
2
0
```

Komentar

Na začetku je, $A = [1, 2, 3]$. Prva poizvedba je na celem območju. Rezultat analize je $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Nato je vrednost za prvo pomarančo posodobljena na 3. To spremeni rezultat iste poizvedbe (na območju $[1, 3]$) $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$.

Primer 2

Vhod

```
5 6
1 2 3 4 5
2 1 3
1 1 3
2 1 5
2 4 4
1 1 1
2 4 4
```

Izhod

```
2
5
4
4
```