XORgulje

Klepec voli orgulje! Lažemo, zapravo voli skenirati naranče. Zašto?... Prevoditeljima u 23:56 nije pao dobar razlog na pamet. Rezultat skeniranja pohranjuje u 32-bitni broj koji sadrži informacije o skreniranoj kori naranče. 32-bitni broj D predstavlja niz od 32 znamenke (bita) od kojih svaki može imati vrijednost nula ili jedan. Ako počnemo od 0 možemo dobiti broj D dodavajući 2^i za svaki i-ti bit koji je jednak jedan. Odnosno, broj D može se predstaviti na sljedeći način: $d_{31}, d_{30}, \ldots d_0$ što je jednako $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \ldots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Primjerice, broj 13 prikazuje se na sljedeći način: $0, \ldots, 0, 1, 1, 0, 1$.

Klepec je skenirao n naranči. Međutim, on ponekad odluči ponovno skenirati jednu od naranči (i-ta naranča) tijekom izvršavanja vašeg programa. To znači da od tog skeniranja na dalje on koristi novu vrijednost za i-tu naranču.

Klepec želi analizirati podatke o skeniranim narančama. Funkcija ekskluzivno ili (XOR) mu se čini jako zanimljiva te je odlučio napraviti neke izračune korištenjem te funkcije. Klepec bira interval skeniranih naranči od l do u (gdje je $l \leq u$) i želi pronaći vrijednost funkcije XOR svih vrijednosti iz tog intervala, svih parova susjednih vrijednosti iz tog intervala, svih trojki susjednih vrijednosti iz tog intervala, ... i svih $u - l + 1$-torki susjednih vrijednosti iz tog intervala (svi elementi iz intervala).

Primjerice, ako je $l = 2$ i $u = 4$ i postoji niz skeniranih vrijednosti A, program treba izračunati vrijednost $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, gdje $\oplus$ predstavlja operaciju XOR i a_i predstavlja i-ti element niza A.

Neka je XOR operacija definirana na sljedeći način:

Ako je i-ti bit prve vrijednosti isti kao i-ti bit druge vrijednosti, tada je vrijednost i-tog bita u rezultatu jednaka `0`; Ako je i-ti bit prve vrijednosti različit od i-tog bita druge vrijednosti, tada je i-ti bit rezultata jednak `1`.

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

Primjerice, $13 \oplus 23 = 26$.

$13 =$	$0 \ldots 001101$
$23 =$	$0 \ldots 010111$
$13 \oplus 23 = 26 =$	$0 \ldots 011010$

Ulazni podaci

U prvom retku nalaze se 2 pozitivna broja n i q (ukupan broj ponovnih skeniranja i analiza - akcija).

U drugom retku, nalazi se n nenegativnih brojeva odvojenih razmakom, koji predstavljaju vrijednosti niza A (rezultati skeniranja naranči). Element a_i predstavlja vrijednost za i-tu naranču. Indeks i počinje od 1.

Akcije su opisane u sljedećih q redaka s tri nenegativna broja odvojena razmacima.

Ako je tip akcije 1 (ponovno skeniranje), prvi broj je 1, drugi broj je i (indeks naranče koju je Klepec ponovno skenirao), treći broj je j (rezultat ponovnog skeniranja i-te naranče).

Ako je tip akcije 2 (analiza), prvi broj je 2, drugi broj je l (iz teksta), a treći broj je u (iz teksta).

Izlazni podaci

Trebate ispisati točno jedan broj za svaki upit s odgovarajućim rješenjem za taj upit. Rezultat svakog upita potrebno je ispisati u novi redak. Primijetite da bi i-ta linija izlaza trebala odgovarati i-tom upitu. Trebate ispisati odgovore samo za upite tipa 2 (analiza).

Ograničenja

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

Podzadaci

1. **[12 bodova]**: $0 < n, q \leq 100$
2. **[18 bodova]**: $0 < n, q \leq 500$ i ne postoje ponovna skeniranja (akcije tipa 1)
3. **[25 bodova]**: $0 < n, q \leq 5000$
4. **[20 bodova]**: $0 < n, q \leq 2 \cdot 10^5$ i ne postoje ponovna skeniranja (akcije tipa 1)
5. **[25 bodova]**: Nema dodatnih ograničenja.

Primjeri testnih podataka

Primjer 1

Ulaz

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

Izlaz

```
2
0
```

Napomena

Na početku je $A = [1, 2, 3]$. Prvi upit računa se na cijelom intervalu. Rezultat analize je $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Nakon toga prva naranča je ponovno skenirana te se njena vrijednost mijenja na 3. To dovodi do promjene izlaza za isti upit (na cijelom intervalu $[1, 3]$) $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$.

Primjer 2

Ulaz

```
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
```

Izlaz

```
2
5
4
4
```