

XORanges

Janez imádja a narancsot, ezért készített egy narancs szkennert. A narancsokról 3D képeket készít 4 kamerával és egy Raspberry Pi 3b+ számítógéppel. A képfeldolgozója még nem az igazi, egy-egy narancsról csak egy 32-bites egész számot ad ki, amely a narancs héján található lyukakról tartalmaz információt.

Egy D 32-bites egész szám egy 32 elemű bitsorozattal ábrázolható, minden elem egyes vagy nullás lehet. A D értékét megkaphatjuk, ha 0-ról indulva hozzáadunk 2^i -t minden olyan i -edik bitre, melynek értéke egy. Formálisan a D szám a $d_{31}, d_{30}, \dots, d_0$ sorozattal ábrázolható, ahol $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Például a 13-nak a $0, \dots, 0, 1, 1, 0, 1$ bit sorozat felel meg.

Janez összesen n narancsot szkennelt be, azonban néha úgy döntött, hogy egy-egy narancsot **újraszkenel**. Ha az i -edik narancsot újraszkeneli, akkor onnantól kezdve az új szkennelés értékét fogja használni az i -edik narancshoz.

A szkennelések mellett Janez **elemzi** is a narancsokat. Kiválasztja a narancsok egy intervallumát az l -edikől az u -adikig ($l \leq u$), és a következő számítást végzi a kedvenc műveletével, a bitenkénti kizáró vaggal (XOR): Össze XOR-olja az intervallumban szereplő összes értéket, az összes szomszédos pár XOR-ját, az összes 3 hosszú folytonos sorozat XOR-ját, ... és így tovább egészen az $u - l + 1$ hosszú folytonos sorozat (az intervallum összes eleme) XOR-jáig.

Például ha $l = 2$, $u = 4$, és az A tömb tartalmazza a beolvasott értékeket, a programodnak az $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$ értéket kell kiszámolnia, ahol \oplus jelöli a XOR-t és a_i az A tömb i -edik elemét.

A XOR műveletet két érték között a következőképp definiáljuk:

Ha az első és a második érték i -edik bitje megegyezik, akkor az eredmény i -edik bitje 0, hogyha pedig a két érték i -edik bitje különböző, akkor az eredmény i -edik bitje 1.

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

Például, $13 \oplus 23 = 26$.

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

Bemenet

A bemenet első sorában található két pozitív egész szám, n (a narancsok száma) és q (az újraszkenelések és elemzések száma).

A következő sorban n darab szóközzel elválasztott nemnegatív egész szám írja le az A tömb elemeit (a narancs szkennelés eredményét). Az a_i elem az i -edik narancs beszkenelt értéke, 1-től indexelve.

A következő q sorban soronként 3 szám írja le az akciókat.

Újraszkenelés akció esetében az első szám 1-es, és következik i (az újraszkenelt narancs indexe) és j (az újraszkenelés eredménye). Elemzés akció esetében az első szám 2-es, és ezt követi az l és az u érték.

Kimenet

Pontosan annyi sort kell kiírnod, ahány elemzés történt. Az i -edik sorba egyetlen számot, az i -edik elemzés eredményét kell írni.

Korlátok

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

Pontozás

1. **[12 pont]:** $0 < n, q \leq 100$
2. **[18 pont]:** $0 < n, q \leq 500$ és nincs újraszkenelés
3. **[25 pont]:** $0 < n, q \leq 5000$
4. **[20 pont]:** $0 < n, q \leq 2 \cdot 10^5$ és nincs újraszkenelés
5. **[25 pont]:** Nincs további kikötés.

Példák

1. példa

Bemenet

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

Kimenet

```
2
0
```

Magyarázat

Kezdetben $A = [1, 2, 3]$. Az első elemzés a teljes tömbön történik. Az eredménye $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Ez után az első narancs értéke 3-ra változik. Ennek következtében ugyanez az elemzés (az $[1, 3]$ intervallumon) $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$ lesz.

2. példa

Bemenet

```
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
```

Kimenet

```
2
5
4
4
```