

ХпОRтокали

Тео обожава портокали! Па затоа, тој си направил скенер за портокали. Користејќи камери и Raspberry Pi 3b+ компјутер, тој започнал да создава 3D слики од портокали. Неговиот процесор на слики не е многу добар, па единствениот излез што го добива тој е 32-битен цел број, кој што ја чува информацијата во врска со дупчињата на кората. 32-битниот цел број D се претставува како секвенца од 32 бита, при што секој од нив е 1 или 0. Ако започнеме од 0 можеме да го добиеме D додавајќи 2^i за секој i -ти бит што е еднаков на 1. Поформално, бројот D се претставува преку секвенцата $d_{31}, d_{30}, \dots, d_0$ каде $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. На пример, 13 се претставува како $0, \dots, 0, 1, 1, 0, 1$.

Тео скенирал n портокали; но, понекогаш тој одлучува повторно да скенира еден од портокалите (i -от портокал) за време на извршувањето на вашата програма. Ова значи дека од тоа скенирање, тој ја користи ажурираната вредност за i -от портокал.

Тео сака да ги анализира овие портокали. Нему многу му е интересна операцијата исклучиво или (XOR), па одлучил да направи некои пресметки. Тој избрал еден ранг од портокали, од l до u (каде што $l \leq u$) и сака да ја утврди вредноста на XOR од сите елементи во тој ранг, сите парови од последователни елементи во тој ранг, сите секвенци од по 3 последователни елементи итн., се' до секвенцата од $u - l + 1$ последователни елементи (сите елементи во рангот).

На пример, ако $l = 2$ и $u = 4$ и имаме низа A од скенирани вредности, програмата треба да ја врати вредноста на $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, каде \oplus ја претставува операцијата XOR, а a_i го претставува i -от елемент на низата A .

Нека операцијата XOR е дефинирана со:

Ако i -от бит од првата вредност е еднаков со i -от бит од втората вредност, тогаш i -от бит од резултатот е 0; ако пак i -от бит од првата вредност е различен од i -от бит од втората вредност, тогаш i -от бит од резултатот е 1.

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

На пример, $13 \oplus 23 = 26$.

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

Влез

Во првата линија од влезот има 2 позитивни цели броја n и q (вкупниот број на повторни скенирања и прашанки (анг. queries) - акции).

Во следната линија се наоѓаат n ненегативни цели броеви разделени со по едно празно место, кои што ги претставуваат вредностите на низата A (резултати од скенирањата на портокалите). Елементот a_i ја содржи вредноста за i -от портокал. Редниот број i започнува од 1.

Акциите се опишани во следните q линии, со по три цели броеви разделени со по едно празно место.

Ако типот на акцијата е 1 (повторно скенирај - rescan), тогаш првиот цел број е еднаков на 1 и по него следуваат i (реден број на портокал кој што Тео сака повторно да го скенира) и j (резултатот од повторното скенирање на i -от портокал).

Ако типот на акцијата е 2 (прашанка - query), тогаш првиот цел број е еднаков на 2 и по него следуваат l и u .

Излез

Треба да отпечатите точно по еден цел број за секоја прашанка (анг. query) со соодветниот резултат за неа. Секоја вредност треба да ја отпечатите во посебна линија. Да забележиме дека i -тата линија од излезот треба да соодветствува на резултатот од i -тата прашанка.

Ограничувања

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

Подзадачи

1. **[12 поени]:** $0 < n, q \leq 100$
2. **[18 поени]:** $0 < n, q \leq 500$ и нема повторни скенирања (акции од тип 1)
3. **[25 поени]:** $0 < n, q \leq 5000$
4. **[20 поени]:** $0 < n, q \leq 2 \cdot 10^5$ и нема повторни скенирања (акции од тип 1)

5. [25 поени]: Нема дополнителни ограничувања.

Примери

Пример 1

Влез

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

Излез

```
2
0
```

Коментар

На почетокот, $A = [1, 2, 3]$. Првата прашанка се однесува на целиот ранг. Резултатот од анализата е $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Потоа вредноста на првиот портокал се ажурира во 3. Ова води до промена на истата прашанка (над рангот $[1, 3]$) $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$.

Пример 2

Влез

```
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
```

Излез

2
5
4
4