XORапельсини

Джанез обожнює апельсини! Тож він створив сканер для апельсинів. З камерами та Raspberry Pi 3b+ комп'ютером, він почав створювати 3D зображення апельсинів. Його процесор зображення не достатньо хороший, тому все, що він отримує - це 32-х бітове число, що підтримує інформацію про дірочки на кірці апельсину. 32-х бітове число D описується як послідовність з 32-х цифр (бітів), кожна з яких дорівнює нулю або одиниці. Якщо ми почнемо з 0, ми можемо отримати D додаванням 2^i для кожного i-го біту, що дорівнює 1. Більш формально, число D описується як послідовність $d_{31}, d_{30}, \ldots d_0$, коли $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \ldots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Наприклад, 13 описується як $0, \ldots, 0, 1, 1, 0, 1$.

Джанез просканував n апельсинів; проте, іноді він вирішує пересканувати один з апельсинів (i-й апельсин) під час виконання вашої програми. Це означає, що після цього пересканування, він використовує змінене значення для i-го апельсину.

Джанез хоче дослідити ті апельсини. Він вважає побітову операцію XOR дуже цікавою, тож він вирішив зробити деякі розрахунки. Він обирає відрізок апельсинів з l до u (де $l \leq u$) та хоче дізнатись значення XOR усіх елементів на цьому відрізку, усіх пар елементів, що йдуть поспіль на цьому відрізку, усіх підпослідовностей з 3 елементів, що йдуть поспіль, і так далі до підпослідовності з $u - l + 1$ елементів, що йдуть поспіль (всі елементи, що входять у відрізок).

Тобто якщо $l = 2$ та $u = 4$ і є масив з відсканованих значень A, програма повинна повернути значення $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, де $\oplus$ позначає XOR та a_i позначає i-й елемент в масиві A.

Зверніть увагу, що XOR операція розглядається так:

Якщо i-й біт першого значення такий же, як і i-й біт другого значення, то i-й біт результату буде `0`; якщо i-й біт першого значення відрізняється від i-го біту другого значення, то i-й біт результату буде `1`.

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

Наприклад, $13 \oplus 23 = 26$.

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

Вхідні дані

В першому рядку вхідних даних дано 2 додатніх цілих числа n та q (загальна кількість пересканувань та запитів).

В наступному рядку дано n невід'ємних цілих чисел, розділених пробілом, що позначають числа в масиві A (результати сканів апельсинів). Елемент a_i містить значення для i-го апельсину. Нумерація i починається з 1.

Операції описуються в наступних q рядках трьома додатніми цілими числами, розділених одним пробілом.

Якщо операція 1-го типу (пересканування), то перше число дорівнює 1 та супроводжується двома числами: i (номер апельсину, який Джанез хоче пересканувати) та j (результат пересканування i-го апельсину).

Якщо операція 2-го типу (запит), то перше число дорівнює 2 і далі дано два числа l та u.

Вихідні дані

Вам потрібно вивести єдине число для кожного запиту. Виведіть кожне значення в окремому рядку. Зверніть увагу, що i-й рядок вихідних даних повинен відповідати результату i-го запиту.

Обмеження

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

Підзадачі

1. **[12 балів]**: $0 < n, q \leq 100$
2. **[18 балів]**: $0 < n, q \leq 500$ і немає операцій пересканування (операцій 1-го типу)
3. **[25 балів]**: $0 < n, q \leq 5000$
4. **[20 балів]**: $0 < n, q \leq 2 \cdot 10^5$ і немає операцій пересканування (операцій 1-го типу)
5. **[25 балів]**: Без додаткових обмежень.

Приклади

Приклад 1

Вхідні дані

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

Вихідні дані

```
2
0
```

Пояснення

На початку, $A = [1, 2, 3]$. Перший запит здійснюється на всьому масиві. Результат розрахунку $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Далі значення для першого апельсину стає рівним 3. Це призводить до змін в такому ж запиті (на відрізку $[1, 3]$) $3 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 0$.

Приклад 2

Вхідні дані

```
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
```

Вихідні дані

```
2
5
4
4
```