

XORanges

Джеймс настолько любит апельсины, что он сделал для них сканер, используя 4 камеры и компьютер Raspberry Pi 3b+, и начал создавать 3D изображения апельсинов. Его процессор изображений не очень хорош, поэтому в результате сканирования он получает только 32-битное целое число, которое содержит информацию о повреждениях на коже. 32-битное число D представляется последовательностью из 32 битов, каждый из которых может быть нулем или единицей. Если занумеровать биты с 0, то можно получить число D , сложив 2^i для каждого i -го бита, равного единице. Более формально, число D задается последовательностью $d_{31}, d_{30}, \dots, d_0$, если $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \dots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Например, число 13 представляется как $0, \dots, 0, 1, 1, 0, 1$.

Джеймс отсканировал n апельсинов; тем не менее иногда он решает пересканировать один из апельсинов (i -й апельсин) во время выполнения вашей программы. Это значит, что после пересканирования нужно использовать новое значение для i -го апельсина.

Джеймс хочет анализировать полученные данные. Он считает операцию "исключающего ИЛИ" (XOR) очень интересной, поэтому решает использовать ее в вычислениях. Он выбирает диапазон апельсинов с l до u (где $l \leq u$) и хочет вычислить результат операции XOR, примененной ко всем числам диапазона, ко всем парам соседних элементов диапазона, всем последовательностям из 3 соседних элементов, и т. д. вплоть до последовательности из $u - l + 1$ соседних элементов (всех элементов диапазона).

То есть, если $l = 2$, $u = 4$ и A — массив полученных в результате сканирования значений, программа должна вернуть результат следующего выражения $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, где \oplus обозначает XOR и a_i обозначает i -й элемент в массиве A .

Операция XOR над двумя числовыми значениями определяется так:

Если i -й бит первого числа такой же, как i -й бит второго, то i -й бит результата равен 0; если i -й бит первого числа отличается от i -го бита второго числа, то i -й бит результата равен 1.

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

Например, $13 \oplus 23 = 26$.

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

Входные данные

В первой строке входных данных расположены 2 целых положительных числа n и q (общее число операций пересканирования и анализа данных).

В следующей строке расположены n разделенных пробелом целых неотрицательных чисел, которые представляют значения массива A (результаты сканирования апельсинов). Элемент a_i содержит описание i -го апельсина. Индексы нумеруются начиная с 1.

Операции описываются в следующих q строках, с помощью трех разделенных пробелами целых положительных чисел.

Если операция имеет тип 1 (пересканирование), то первое число равно 1, за ним следует i (индекс апельсина, который хочется пересканировать) и j (результат пересканирования i -го апельсина).

Если тип операции 2 (анализ), первое число равно 2, за ним следуют l и u .

Выходные данные

Вы должны вывести в точности одно целое число для каждой операции типа 2 (анализ данных). Вы должны выводить каждое значение в отдельной строке. Обратите внимание, что i -я строка выходных данных должна соответствовать i -й операции типа 2.

Ограничения

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

Подзадачи

1. **[12 баллов]**: $0 < n, q \leq 100$
2. **[18 баллов]**: $0 < n, q \leq 500$ и нет операций изменения значений
3. **[25 баллов]**: $0 < n, q \leq 5000$
4. **[20 баллов]**: $0 < n, q \leq 2 \cdot 10^5$ и нет операций изменения значений
5. **[25 баллов]**: Нет дополнительных ограничений

Примеры

Пример 1

Входные данные

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

Выходные данные

```
2
0
```

Комментарий

В начале $A = [1, 2, 3]$. Первая операция производится над полным диапазоном значений. Результат анализа есть $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Затем значение первого апельсина меняется на 3. Это приводит к изменению результата в операции анализа всех данных (в диапазоне $[1, 3]$): $3 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 0$.

Пример 2

Входные данные

```
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
```

Выходные данные

2
5
4
4