XORanges

Janez adore les oranges! Tellement qu'il a construit un scanner pour oranges. Avec une caméra et un ordinateur Raspberry Pi 3b+, il s'est lancé dans la création d'images 3D d'oranges. Son processeur graphique n'étant pas très bon, tout ce qu'il obtient d'un scan est un entier 32 bits, qui contient de l'information sur les trous dans la peau. Un entier 32 bits D est représenté par une suite de 32 chiffres (bits) d'une valeur de 1 ou de 0. En partant de 0, on peut obtenir D en sommant 2^i pour chaque i-ème bit égal à 1. Formellement, le nombre D est représenté par la suite $d_{31}, d_{30}, \ldots d_0$ avec $D = d_{31} \cdot 2^{31} + d_{30} \cdot 2^{30} + \ldots + d_1 \cdot 2^1 + d_0 \cdot 2^0$. Par exemple, 13 est représenté par la suite $0, \ldots, 0, 1, 1, 0, 1$.

Janez a scanné n oranges. Cela dit, parfois il décide de rescanner une des oranges (la i-ème) pendant l'exécution de votre programme. Ceci veut dire qu'à partir de ce nouveau scan, il utilise la nouvelle valeur pour la i-ème orange.

Janez veut analyser ces oranges. Il trouve l'opérateur *ou exclusif* (XOR) très intéressant et décide d'effectuer quelques calculs. Il choisit un intervalle d'oranges de l à u (avec $l \leq u$) et décide de trouver la valeur du XOR de tous les éléments de l'intervalle, ainsi que de toutes les paires d'éléments consécutifs de l'intervalle, ainsi que de tous les triplets d'éléments consécutifs de l'intervalle, et ainsi de suite jusqu'à la séquence des $u - l + 1$ éléments consécutifs (tous les éléments dans l'intervalle).

Ainsi, si $l = 2$ et $u = 4$, et que les valeurs des oranges sont stockées dans le tableau A, le programme doit renvoyer la valeur de $a_2 \oplus a_3 \oplus a_4 \oplus (a_2 \oplus a_3) \oplus (a_3 \oplus a_4) \oplus (a_2 \oplus a_3 \oplus a_4)$, où $\oplus$ représente le XOR et a_i représente le i-ème élément du tableau A.

L'opérateur XOR est défini ainsi:

Si le i-ème bit du premier opérande est le même que le i-ème bit du second opérande, le i-ème bit du résultat est `0`. Sinon, si le i-ème bit du premier opérande est différent du i-ème bit du second opérande, le i-ème bit du résultat est `1`.

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

Par exemple, $13 \oplus 23 = 26$.

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

Entrée

Il y a 2 entiers positifs n et q (le nombre total d'actions de nouveau scan et de requête) sur la première ligne.

Sur la ligne suivante, il y a n entiers positifs ou nuls séparés par des espaces, qui représentent les valeurs du tableau A (les résultats des scans des oranges). Le i-ème de ces entiers contient la valeur pour la i-ème orange, avec les indices commençant à 1.

Ensuite, chacune des q actions est décrite par une ligne contenant trois entiers positifs séparés par des espaces.

Si le type de l'action est 1 (nouveau scan), le premier entier vaut 1 et est suivi par i (l'indice de l'orange que Janez veut rescanner) et j (le résultat du nouveau scan de la i-ème orange).

Si le type de l'action est 2 (requête), le premier entier vaut 2 et est suivi par l et u.

Sortie

Affichez exactement un entier pour chaque requête avec le résultat correspondant pour la requête. Affichez chaque valeur sur une nouvelle ligne. Notez que l'entier affiché à la i-ème ligne de la sortie doit correspondre au résultat de la i-ème requête.

Contraintes

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

Sous-tâches

1. **[12 points]**: $0 < n, q \leq 100$
2. **[18 points]**: $0 < n, q \leq 500$ et il n'y a pas de nouveau scan (action de type 1)
3. **[25 points]**: $0 < n, q \leq 5000$
4. **[20 points]**: $0 < n, q \leq 2 \cdot 10^5$ et il n'y a pas de nouveau scan (action de type 1)
5. **[25 points]**: Pas de contraintes supplémentaires.

Exemples

Exemple 1

Entrée

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

Sortie

```
2
0
```

Commentaire

Au début, $A = [1, 2, 3]$. La première requête est sur tout l'intervalle. Le résultat de cette analyse est $1 \oplus 2 \oplus 3 \oplus (1 \oplus 2) \oplus (2 \oplus 3) \oplus (1 \oplus 2 \oplus 3) = 2$.

Ensuite, la valeur de la première orange est remplacée par 3. Cela conduit à un changement pour la même analyse (sur l'intervalle $[1, 3]$): $3 \oplus 2 \oplus 3 \oplus (3 \oplus 2) \oplus (2 \oplus 3) \oplus (3 \oplus 2 \oplus 3) = 0$.

Exemple 2

Entrée

```
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
```

Sortie

```
2
5
4
4
```