



## Objectif

Après avoir correctement repéré la structure du réseau, vous accédez à un salon privé de discussion dont le contenu est chiffré de manière très étrange : chaque message peut être déchiffré en effectuant plusieurs opérations sur une clé initiale de  $N$  octets partagé entre tous les utilisateurs.

Chaque octet du message est calculé avec une opération décrite par deux entiers  $L$  et  $R$  (avec  $L \leq R$ ). Le résultat de l'opération est égal à  $(\text{clé}[L] \text{ XOR } \text{clé}[L+1] \text{ XOR } \text{clé}[L+2] \dots \text{ XOR } \text{clé}[R-1] \text{ XOR } \text{clé}[R])$ , où XOR dénote l'opération Ou Exclusif sur des nombres entiers.

Dans certains cas, le message semble trop lent à reconstituer même avec vos ordinateurs les plus performants, alors que les téléphones des membres du réseau semblent pouvoir faire ces mêmes opérations en une fraction de seconde. Pouvez-vous implémenter cet algorithme de manière efficace afin de déchiffrer les messages échangés à partir de la clé et de la liste des opérations à effectuer ?

**Attention** : les entrées étant d'une taille importante, une solution naïve pourrait fonctionner sur les premiers jeux de test (plus petits) mais ne pas être suffisante pour les plus gros jeux de test. Il faudra redoubler d'ingéniosité pour déchiffrer les messages !

## Données

### Entrée

Ligne 1 : deux entiers **N** et **M**, compris entre 1 et 100 000 séparés par un espace, respectivement la taille de la clé et la taille du message

Ligne 2 : **N** entiers entre 0 et 255 séparés par des espaces, les octets de la clé

Lignes 3 à **M**+2 : A la ligne i, eux entiers **Li** et **Ri** (avec  $0 \leq Li \leq Ri \leq N-1$ ) correspondant à l'opération à effectuer pour retrouver l'octet i du message

## Sortie

1 ligne avec **256** entiers  $N_0$   $N_1$  ...  $N_i$  ...  $N_{255}$  séparés par des espaces,  $N_i$  étant le nombre de  $i$  dans le message déchiffré.

### Example

Pour l'entrée :

5 4

11 22 33 44 55

1 3

0 1

2 2

2 4

La sortie devra être :

[illegible]

En effet :

$$22 \text{ xor } 33 \text{ xor } 44 = 27$$
$$11 \text{ xor } 22 = 29$$
$$33 = 33$$
$$33 \text{ xor } 44 \text{ xor } 55 = 58$$

Le message déchiffré est donc : 27 29 33 58, d'où la sortie attendue