

L'Étoile Noire

L'Alliance pour la Restauration de la République intercepte une transmission chiffrée à haute-fréquence provenant d'un établissement secret. Elle contient des schémas décrivant un nouveau système de blindage modulaire pour **l'Étoile Noire**.

Le blindage est représenté par une grille de $(2N + 1) \times (2M + 1)$ secteurs. Chaque secteur $C_{i,j}$ est occupé soit par du **blindage renforcé** (blanc) ou par un **ventilateur d'échappement** (noir). La doctrine des ingénieurs impériaux stipule que l'intégrité de la structure du blindage n'est assurée que lorsque les secteurs du blindage renforcé sont **convexes**, c'est à dire que dans chaque ligne et colonne, les secteurs blancs forment un segment continu, et que chaque paire de secteurs blancs sont reliés par un chemin composé uniquement lui-même de secteurs blancs.

Votre Mission

Votre robot, R2-D2, a su pirater avec réussite l'unité centrale, mais la connexion est instable. Il ne peut pas télécharger toutes les données de la carte. Il peut cependant demander des infos sur une paire de coordonnées (x, y) .

Le système de sécurité impérial répond aux requêtes en retournant un contrôle de parité soi-disant "mélangé":

1. Si le secteur en coordonnées (x, y) est **blindage renforcé** (blanc), le résultat de la requête est le produit du nombre de cellules blanches dans sa ligne et le nombre de cellules noires dans sa colonne.
2. Sinon, le secteur (x, y) est un **ventilateur d'échappement** (noir), et le système retourne -1 .

Le Rebéllion sait d'emblée que la secteur en (N, M) est toujours renforcé ($C_{N,M}$ est un secteur blanc).

En faisant moins de $(2N + 1) \times (2M + 1)$ requêtes, déterminez le contenu de la grille pour que l'Alliance puisse identifier les points faibles de l'Étoile Noire.

Spécifications d'implémentation

Vous devez implémenter la fonction suivante:

```
std::vector<std::vector<int>> compute_grid(int N, int M)
```

- N, M : Nombres entiers positifs donnant la taille de la grille.
- Cette fonction doit retourner une grille G de taille $(2N + 1) \times (2M + 1)$ qui correspond à la grille initiale, telle que $G[i][j] = 0$ signifie que le secteur $C_{i,j}$ est noir, et $G[i][j] = 1$ signifie qu'il est blanc.
- Cette fonction est appelée une fois par cas de test.

Cette fonction doit appeler la procédure suivante pour effectuer des requêtes:

```
int ask(int x, int y)
```

- x, y : deux entiers tels que $0 \leq x < 2N + 1, 0 \leq y < 2M + 1$
- Cette procédure donne le nombre de secteurs blancs en ligne x multiplié par le nombre de secteurs noirs en colonne y si le secteur en (x, y) est blanc, sinon ça donne -1 .
- La procédure peut être appelée un maximum de $(2N + 1) \times (2M + 1)$ fois par exécution.

Contraintes

- $1 \leq N \leq 500$
- $1 \leq M \leq 1000$

Sous-tâches et Barême

Soit C le nombre d'appels à la procédure `ask` dans un cas de test; on a:

$$X = \min \left(1, \frac{100e^{\frac{C-5716}{25000}} + 913}{500 \left(e^{\frac{C-5716}{25000}} + 1 \right)} \right)$$

Sous-tâche	Score	Contraintes
1	15	$N, M \leq 10$
2	$85X$	Aucune contrainte

Votre score final est le score minimal atteint sur l'ensemble des cas de test de la sous-tâche donnée. Si votre fonction donne un résultat erroné pour au moins un des cas de test d'une sous-tâches, vous aurez une note de 0 pour cette sous-tâche.

Exemple

Grille Exemple

```
5 6
00000000000000
00000000000000
00000000000000
00001100000000
00001100000000
00011110000000
00011000000000
00000000000000
00000000000000
00000000000000
00000000000000
```

Exemple de requêtes

- Puisque la cellule en coordonnées $(5,6)$ est une cellule blanche, un appel à `ask(5, 6)` va donner le produit du nombre de cellules blanches en ligne 5 et le nombre de cellules noires en colonne 6: Il y a 4 cellules blanches en ligne 5 et 10 cellules noires en colonne 6, le produit de ces deux valeurs est 40, ce qui est le résultat donné par la fonction.
- Un appel à `ask(6, 6)` va retourner -1 puisque la cellule en coordonnées $(6,6)$ est noire.