

Matrix 0

Il y a longtemps, bien longtemps avant que Keanu Reeves ne vienne y tourner des films de Kung Fu, la vie dans la matrice était beaucoup plus simple. Le travail de l'agent Smith était simple et routinier : reconnaître les bugs dans la matrice, et les écraser tous.

Les bugs et la matrice elle-même sont représentés chacun par une grille de bits valant 0 ou 1. Un bug est présent dans la matrice s'il est possible de trouver un groupe de bits **valant 1** de même forme que ce bug, aux rotations et symétries près. Les bits à 0 d'un bug ne sont pris en compte que dans sa description, **jamais** dans sa reconnaissance.

Exemple : le bug

```
1 0 1 0
0 1 0 1
```

sera reconnu tel quel, mais aussi sous les formes suivantes :

	1 0	0 1
0 1 0 1	0 1	1 0
1 0 1 0	1 0	0 1
	0 1	1 0

Données :

Sur une ligne, un entier, nombre de bugs à détecter. Puis, pour chaque bug :

- sur une ligne, deux entiers N et M donnant respectivement le nombre de lignes et de colonnes du bug.
- puis N lignes de M valeurs 0 ou 1, séparées par des espaces.

Et enfin, pour la matrice elle-même, deux entiers donnant ses nombres de lignes et de colonnes, puis ses bits (0 ou 1) en lignes et séparés par des espaces.

Résultat :

Un grille de mêmes dimensions que la matrice d'entrée, donnée elle aussi par ses bits (0 ou 1) en lignes et séparés par des espaces (attention, pas d'espace en fin de ligne!). Dans cette grille de sortie, les bits à 1 sont ceux (et seulement ceux) qui faisaient partie d'au moins un des bugs recherchés.

Les dimensions de la matrice ne doivent pas être répétées dans la sortie.

Exemple :

Entrée

```
2
3 3
0 1 0
1 1 1
0 1 0
2 4
1 0 1 0
0 1 0 1
7 6
0 1 0 0 0 1
1 1 1 1 1 0
0 1 0 0 1 1
0 1 1 1 1 0
0 0 1 0 0 0
1 0 1 0 1 1
1 1 1 1 1 0
```

Sortie

```
0 1 0 0 0 1
1 1 1 0 1 0
0 1 0 0 0 1
0 0 1 0 1 0
0 0 0 0 0 0
1 0 1 0 1 0
0 1 0 1 0 0
```

Explicitons ci-dessous les bugs reconnus : le premier est coloré en rouge, le second en bleu. Les bits colorés en vert ont été utilisés dans deux bugs distincts. Enfin, remarquons que, sur les deux dernières lignes, deux exemplaires du même bug se chevauchent.

```
0  1  0  0  0  1
1  1  1  0  1  0
0  1  0  0  0  1
0  0  1  0  1  0
0  0  0  0  0  0
1  0  1  0  1  0
0  1  0  1  0  0
```