

Problème XCopy

Fichier d'entrée `stdin`
Fichier de sortie `stdout`

Aujourd'hui, à la fin du cours de programmation, l'enseignant a donné un devoir très difficile, et les élèves ont donc décidé de tricher en copiant les uns sur les autres. Cependant, ils doivent le faire intelligemment pour ne pas se faire repérer.

La classe est constituée de $N \times M$ élèves, assis sur $N \times M$ bancs répartis en N rangées de M colonnes. Deux enfants sont considérés comme étant voisins si l'un est assis sur un banc adjacent à gauche, à droite, au dessus ou en dessous du banc sur lequel l'autre est assis. Le devoir consiste à trouver un certain entier positif ou nul. Pour tricher sans se faire repérer, tous les entiers choisis par les enfants doivent être différents. Par ailleurs, les enfants sont très fainéants, donc ils vont à peine modifier leur réponse lorsqu'ils copient sur leurs voisins. Plus précisément, la réponse de chaque enfant doit différer d'exactly un bit en base 2, comparée à la réponse de chacun de ses voisins. Par exemple 3 et 2 diffèrent d'exactly un bit, mais ce n'est pas le cas de 2 et 4.

Les enfants ne souhaitent pas éveiller de soupçons, donc ils souhaitent que la plus grande de leurs réponses soit la plus petite possible. Étant donnés N et M , créez une configuration de réponses telle que l'enseignant ne remarquera pas que les enfants trichent.

Données d'entrée

L'entrée consiste en une seule ligne, contenant N et M , séparés par une espace.

Données de sortie

La sortie doit contenir les réponses optimales des enfants. La sortie doit contenir N lignes, chacune contenant M entiers positifs séparés par des espaces. Ces entiers représentent les réponses des enfants selon leur disposition dans la salle.

Contraintes

- $1 \leq N, M \leq 2000$

#	Points	Contraintes
1	7	$N = 1$.
2	9	N, M sont des puissances de 2.
3	14	N est une puissance de 2.
4	70	Pas de contrainte particulière.

Score

Ce sujet accepte les réponses non optimales. Le score partiel d'une réponse dépendra de la distance entre la réponse fournie et la réponse optimale, selon la formule de score suivante :

$$S \cdot \max \left(1 - \sqrt{\frac{G - O}{3}}, 0 \right)$$

Où :

- S est le score maximal attribué au fichier test,
- G est la réponse donnée,
- O est la réponse optimale.

Attention ! Une solution qui ne respecte pas le format de sortie (tous les nombres sont distincts et deux nombres adjacents diffèrent d'exactly 1 bit dans leur représentation en base 2), obtiendra un score de 0 pour le fichier test correspondant.

Exemples

Fichier d'entrée	Fichier de sortie
3 3	5 4 6 1 0 2 9 8 10

Explications

Dans cette section, un nombre en indice après un autre nombre, représente la base dans laquelle ce nombre est écrit. Par exemple, le nombre huit peut être écrit comme $8_{10} = 1000_2$.

Le tableau suivant décrit une réponse optimale pour les élèves :

$0101_2 = 5_{10}$	$0100_2 = 4_{10}$	$0110_2 = 6_{10}$
$0001_2 = 1_{10}$	$0000_2 = 0_{10}$	$0010_2 = 2_{10}$
$1001_2 = 9_{10}$	$1000_2 = 8_{10}$	$1010_2 = 10_{10}$

Observez qu'entre toute paire de bancs adjacents, les nombres diffèrent d'exactly un bit. La valeur maximale de cette réponse est 10, ce qui en fait une réponse optimale. Il existe d'autres réponses optimales – comme cette même solution, mais retournée verticalement ou horizontalement.

Voici une réponse non optimale dont la valeur maximale est 15 :

0110_2	0111_2	0101_2
1110_2	1111_2	1101_2
1010_2	1011_2	1001_2

Cette solution obtiendrait, en utilisant la formule de calcul du score, un score de 59.1% pour le fichier test.