Problem J5/S2: Escape Room

Problem Description

You have to determine if it is possible to escape from a room. The room is an M-by-N grid with each position (cell) containing a positive integer. The rows are numbered $1, 2, \ldots, M$ and the columns are numbered $1, 2, \ldots, N$. We use (r, c) to refer to the cell in row r and column c.

You start in the top-left corner at $(1, 1)$ and exit from the bottom-right corner at (M, N). If you are in a cell containing the value x, then you can jump to any cell (a, b) satisfying $a \times b = x$. For example, if you are in a cell containing a 6, you can jump to cell $(2, 3)$.

Note that from a cell containing a 6, there are up to four cells you can jump to: $(2, 3)$, $(3, 2)$, $(1, 6)$, or $(6, 1)$. If the room is a 5-by-6 grid, there isn't a row 6 so only the first three jumps would be possible.

Input Specification

The first line of the input will be an integer M ($1 \leq M \leq 1000$). The second line of the input will be an integer N ($1 \leq N \leq 1000$). The remaining input gives the positive integers in the cells of the room with M rows and N columns. It consists of M lines where each line contains N positive integers, each less than or equal to $1\,000\,000$, separated by single spaces.

For 1 of the 15 available marks, $M = 2$ and $N = 2$.

For an additional 2 of the 15 available marks, $M = 1$.

For an additional 4 of the 15 available marks, all of the integers in the cells will be unique.

For an additional 4 of the 15 available marks, $M \leq 200$ and $N \leq 200$.

Output Specification

Output `yes` if it is possible to escape from the room. Otherwise, output `no`.

Sample Input

```
3
4
3 10  8 14
1 11 12 12
6  2  3  9
```

Output for Sample Input

```
yes
```

La version française figure à la suite de la version anglaise.

Explanation of Output for Sample Input

Starting in the cell at $(1, 1)$ which contains a 3, one possibility is to jump to the cell at $(1, 3)$. This cell contains an 8 so from it, you could jump to the cell at $(2, 4)$. This brings you to a cell containing 12 from which you can jump to the exit at $(3, 4)$. Note that another way to escape is to jump from the starting cell to the cell at $(3, 1)$ to the cell at $(2, 3)$ to the exit.

Notes

1. The online grader begins by testing submissions using the sample input. All other tests are skipped if the sample test is not passed. If you are only attempting the first three subtasks (the first 7 marks), then you might want to handle the specific values of the sample input as a special case.

2. For the final subtask (worth 2 marks), if you are using Java, then `Scanner` will probably take too long to read in the large amount of data. A much faster alternative is `BufferedReader`.

La version française figure à la suite de la version anglaise.

Problème J5/S2: Le jeu d'évasion

Énoncé du problème
Vous devez déterminer s'il est possible de s'échapper d'une salle. La salle est en forme de grille $M \times N$. Chacune des cases de la grille contient un entier strictement positif. Les rangées de la grille sont numérotées $1, 2, \ldots, M$ tandis que les colonnes sont numérotées $1, 2, \ldots, N$. Les coordonnées (r, c) représentent la case qui est située dans la rangée r de la colonne c.

Vous commencez le jeu à partir de la case $(1, 1)$ au coin supérieur gauche de la grille et vous vous échappez de la salle en atteignant la case (M, N) au coin inférieur droit de la grille. Si vous vous trouvez sur une case contenant la valeur x, alors vous pouvez accéder à n'importe quelle case (a, b) telle que $a \times b = x$. Par exemple, une case contenant un 6 vous donne accès à la case $(2, 3)$.

On remarque qu'une case contenant un 6 nous donne accès à quatre cases, soient les cases $(2, 3)$, $(3, 2)$, $(1, 6)$ et $(6, 1)$. Si la salle est en forme de grille 5×6, il n'y a pas de 6^e rangée, donc l'accès ne sera limité qu'aux trois premières cases, soient les cases $(2, 3)$, $(3, 2)$ et $(1, 6)$.

Précisions par rapport aux données d'entrée
La première ligne des données d'entrée contient un entier M qui vérifie $1 \leq M \leq 1000$. La deuxième ligne des données d'entrée contient un entier N qui vérifie $1 \leq N \leq 1000$. Le restant des données d'entrée contient les entiers strictement positifs dans les cases de la salle à M rangées et à N colonnes. On s'attend donc à ce que le restant des données d'entrée contienne M lignes et que chaque ligne contienne N entiers strictement positifs chacun étant séparé des autres par un espace; chacun de ces entiers étant au plus égal à $1\,000\,000$.

Pour 1 des 15 points disponibles, $M = 2$ et $N = 2$.

Pour 2 autres points parmi les 15 points disponibles, $M = 1$.

Pour 4 autres points parmi les 15 points disponibles, les entiers dans les cases seront tous différents les uns des autres.

Pour 4 autres points parmi les 15 points disponibles, $M \leq 200$ et $N \leq 200$.

Précisions par rapport aux données de sortie
Les données de sortie devraient afficher `yes` s'il est possible de s'échapper de la salle, sinon elles devraient afficher `no`.

Exemple de données d'entrée
```
3
4
3 10  8 14
1 11 12 12
6  2  3  9
```

English version appears before the French version

Exemple de données de sortie

```
yes
```

Justification des données de sortie

En commençant par la case $(1, 1)$ qui contient l'entier 3, on peut emprunter le chemin $(1, 1) \rightarrow (1, 3) \rightarrow (2, 4) \rightarrow (3, 4)$ afin de s'échapper de la salle selon le raisonnement suivant: l'entier 3 dans la case $(1, 1)$ nous donne accès à la case $(1, 3)$, l'entier 8 dans la case $(1, 3)$ nous donne accès à la case $(2, 4)$, l'entier 12 dans la case $(2, 4)$ nous donne accès à la case $(3, 4)$ d'où on peut s'échapper de la salle. Remarquons aussi qu'on pourrait s'échapper de la salle en empruntant le chemin suivant: $(1, 1) \rightarrow (3, 1) \rightarrow (2, 3) \rightarrow (3, 4)$.

Notes

1. Le correcteur informatisé commence par tester les soumissions à l'aide de l'échantillon d'entrée. Si l'échantillon testé n'est pas accepté, alors tous les autres tests sont ignorés. Ainsi, si vous ne tentez que les trois premières sous-tâches (les 7 premiers points), ensuite vous pourriez vouloir traiter les valeurs spécifiques de l'échantillon d'entrée en tant que cas particulier.

2. Pour la dernière sous-tâche (qui vaut 2 points), si vous utilisez Java, alors le `Scanner` va probablement prendre trop de temps pour lire la masse de données. Une alternative beaucoup plus rapide est le `BufferedReader`.

English version appears before the French version