Tours

On vous donne une matrice $N \times M$ nommée A où chaque élément de 1 à $N \times M$ apparaît exactement une fois. On vous donne une tour qui se trouve sur la cellule contenant l'élément de valeur 1. Vous êtes également donné un K.

La tour peut sauter d'une cellule (r, c) à une cellule (r', c') si les deux conditions suivantes sont remplies :

- $(r, c) \neq (r', c')$, nous devons donc passer à une autre cellule,
- soit $r = r'$ soit $c = c'$, donc on se déplace uniquement sur la même ligne ou colonne,
- $0 < A[r'][c'] - A[r][c] \leq K$.

Trouvez pour chaque cellule de la matrice le nombre minimum de mouvements nécessaires à la tour pour l'atteindre depuis la cellule contenant 1, ou si elle n'est pas du tout atteignable.

Détails d'implémentation

Vous devez implémenter la fonction suivante :

```
int32[][] calculate_moves(int32[][] A, int32 K)
```

- A : tableau de longueur N de tableaux de longueur M décrivant le tableau.
- K : la contrainte de mouvement.
- La fonction doit renvoyer un tableau de longueur N de tableaux de longueur M contenant le nombre minimum de mouvements nécessaires pour atteindre cette cellule depuis la cellule 1. Si une cellule est inaccessible, la valeur de cette cellule doit être égale à -1.

Contraintes

- $1 \leq N, M \leq 2\,500$
- $1 \leq K \leq N \cdot M$
- $1 \leq A[i][j] \leq N \cdot M$
- Chaque entier de 1 à $N \cdot M$ apparaît dans A exactement une fois.

Sous-tâches

Sous-tâche	Score	Contraintes supplémentaires
1	9	$N = 1$
2	15	$N, M \leq 100$
3	11	$K = 1$
4	19	$K = N \cdot M$
5	15	$N, M \leq 500$
6	31	Aucune autre contrainte.

Exemples

Exemple 1

Considérez l'appel suivant :

```
calculate_moves([[8, 2, 4, 20, 5],
                 [14, 13, 1, 19, 7],
                 [15, 18, 12, 6, 11],
                 [10, 9, 3, 16, 17]], 5)
```

Nous avons $N = 4, M = 5, K = 5$.

La cellule contenant 1 , à partir de laquelle nous partons, est $A[1][2]$. Par conséquent, dans le tableau R renvoyé par `calculate_moves` , la valeur $R[1][2]$ doit être égale à 0 , car nous n'avons pas besoin de faire de mouvements pour atteindre cette cellule.

Pour atteindre $A[3][0] = 10$, on peut passer de $A[1][2] = 1$ à $A[0][2] = 4$ (car ils sont dans la même colonne, et $4 - 1 = 3 \leq 5$), alors de $A[0][2] = 4$ à $A[0][0] = 8$ (car ils sont dans la même rangée, et $8 - 4 = 4 \leq 5$), puis finalement de $A[0][0]$ à $A[4][0]$ (car ils sont dans la même colonne et $10 - 8 = 2$). Par conséquent, dans le tableau R , la valeur $R[4][0]$ devrait être égale à 3 .

Notez qu'il n'est pas possible d'atteindre $A[0][1] = 2$ de quelque manière que ce soit, donc la valeur $R[0][1]$ devrait être égale à -1 .

La procédure devrait renvoyer :

```
[[2, -1, 1, 6, 2],
 [4, -1, 0, 5, 3],
 [4, 5, 5, -1, 4],
 [3, -1, 1, -1, -1]]
```

Grader

Format d'entrée :

```
N M K
A[0][0] A[0][1] ... A[0][M-1]
A[1][0] A[1][1] ... A[1][M-1]
...
A[N-1][0] A[N-1][1] ... A[N-1][M-1]
```

Format de sortie :

```
R[0][0] R[0][1] ... R[0][M-1]
R[1][0] R[1][1] ... R[1][M-1]
...
R[N-1][0] R[N-1][1] ... R[N-1][M-1]
```

Ici, R est le tableau renvoyé par `calculate_moves` .