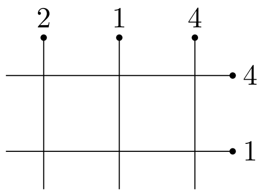
Political cost (cost)

Dans la ville où vous vivez, il y a N rues horizontales (numérotées de 0 à $N-1$) et M avenues verticales (numérotées de 0 à $M-1$). Chaque rue et chaque avenue a un *poids politique*, qui est l'importance du citoyen le plus important qui y habite. On représente les poids politiques comme deux tableaux $A[0\dots N-1]$ et $B[0\dots M-1]$ d'entiers entre 1 et K. Le schéma suivant représente une telle ville avec 2 rues et 3 avenues, de poids politiques $A = [1, 4]$ et $B = [2, 1, 4]$ respectivement.



Le maire veut organiser un défilé dans la ville. Si le défilé passe par l'intersection de la x-ème rue et de la y-ème avenue, le trafic des deux routes sera perturbé et le maire subira un *coût politique* de $\max(A[x], B[y])$. Si le défilé traverse plusieurs intersections, le coût politique sera le *maximum* des coûts de chaque intersection. Notez que les coûts ne sont pas additionnés : ce qui compte n'est pas combien de personnes le défilé dérange, mais l'importance du citoyen le plus important que le défilé dérange.

La *distance politique* entre deux intersections est le plus petit *coût politique* d'un défilé partant de la première intersection et arrivant à la deuxième intersection. Votre mission est de calculer la somme des distances politiques entre toutes les paires d'intersections de la ville.

Implémentation

Vous devez soumettre un unique fichier source `.cpp`.

☞ Parmi les fichiers joints du problème, vous trouverez un squelette de code `cost.cpp` avec un exemple d'implémentation.

Vous devez implémenter la fonction suivante :

```
C++ | int solve(int N, int M, int K, vector<int> A, vector<int> B);
```

— L'entier N représente le nombre de rues verticales.
— L'entier M représente le nombre d'avenues verticales.
— Le tableau A, indexé de 0 à $N-1$, contient les valeurs A_0, A_1, …, A_{N-1}, où A_i est le poids politique de la i-ème rue horizontale.
— Le tableau B, indexé de 0 à $M-1$, contient les valeurs B_0, B_1, …, B_{M-1}, où B_i est le poids politique de la i-ème avenue verticale.
— La fonction doit retourner la somme des distances politiques entre toutes les paires d'intersections, **modulo** 1000003.

L'évaluateur va appeler la fonction `solve` et va écrire sa valeur de retour dans le fichier de sortie.

Évaluateur

Le dossier du problème contient une version simplifiée de l'évaluateur du jury que vous pouvez utiliser pour tester votre solution localement. Cet évaluateur simplifié lit les données en entrée du fichier `stdin`, appelle les fonction que vous devez implémenter et enfin écrit la sortie dans `stdout`.

L'entrée contient 3 lignes :

- Ligne 1 : les entiers N, M and K.
- Ligne 2 : les entiers A_i, séparés par des espaces.
- Ligne 3 : les entiers B_i, séparés par des espaces.

La sortie est constituée d'une unique ligne, contenant la valeur renvoyée par la fonction `solve`.

Contraintes

- $1 \leq N \leq 3 \times 10^5$.
- $1 \leq M \leq 3 \times 10^5$.
- $1 \leq K \leq N + M$.
- $1 \leq A_i \leq K$ pour tout $i = 0, \ldots, N - 1$.
- $1 \leq B_i \leq K$ pour tout $i = 0, \ldots, M - 1$.

Score

Votre programme sera testé sur un ensemble de tests groupés par sous-tâche. Pour obtenir un score associé à une sous-tâche, vous devez résoudre correctement tous les tests qu'elle contient.

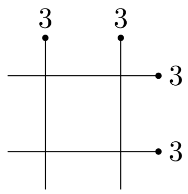
- **Sous-tâche 1 [0 points]**: Sample test cases.
- **Sous-tâche 2 [10 points]**: $N \leq 10^1, M \leq 10^1$.
- **Sous-tâche 3 [10 points]**: $N \leq 10^2, M \leq 10^2$.
- **Sous-tâche 4 [10 points]**: $N = 1, M \leq 10^4$.
- **Sous-tâche 5 [10 points]**: $N = 1, M \leq 10^5$.
- **Sous-tâche 6 [10 points]**: $N \leq 10^3, M \leq 10^3$.
- **Sous-tâche 7 [10 points]**: $N \leq 10^4, M \leq 10^4$.
- **Sous-tâche 8 [10 points]**: $N \leq 10^5, M \leq 10^5$ et les tableaux A et B sont non-décroissants, c'est-à-dire, si $i < j$, alors $A_i \leq A_j$ et $B_i \leq B_j$.
- **Sous-tâche 9 [10 points]**: $N \leq 10^5, M \leq 10^5, K \leq 10^1$.
- **Sous-tâche 10[10 points]**: $N \leq 10^5, M \leq 10^5$.
- **Sous-tâche 11[10 points]**: Pas de contraintes supplémentaires.

Exemples

stdin	stdout
2 2 4 3 3 3 3	48
1 3 4 2 2 3 1	25
2 3 5 1 4 2 1 4	135

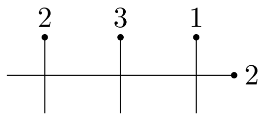
Explication

Dans le **premier exemple**, nous avons une ville avec 2 rues et 2 avenues, toutes avec un poids politique de 3 :



Il y a 16 paires différentes d'intersections. La distance politique entre deux intersections étant toujours de 3, la réponse est $3 \cdot 16 = 48$.

Dans le **second exemple** il y a 1 rue et 3 avenues, de poids politiques $A = [2]$ et $B = [2, 3, 1]$ respectivement :



Il y a 9 paires d'intersections. Trois de ces paires commencent et finissent à la même intersection et ont une distance politique de 2, 3 et 2 respectivement (l'avenue de droite a un poids de 1 mais le poids de l'unique rue horizontale est de 2, donc la distance politique est d'au moins 2). Pour chacune des paires restantes, le défilé doit traverser l'avenue du milieu et donc avoir une distance politique de 3. La somme totale est $2 + 3 + 2 + 6 \cdot 3 = 25$.

Le **troisième exemple** correspond à l'exemple donné dans l'énoncé du problème. Il y a 2 rues et 3 avenues. En étant patient, il est possible de vérifier que la somme des distances politiques des 36 paires d'intersections est 135.