

Footwork

Fichier d'entrée	Fichier de sortie	Limite de temps	Limite de mémoire
entrée standard	sortie standard	1 seconde	256 MiB

Un couloir peut être représenté par une grille de deux lignes N colonnes (numérotées de 1 à N de gauche à droite). Chaque case de la grille a une valeur entière, qui peut être *négative* :

- La i -ème case ($1 \leq i \leq N$) de la ligne du haut est A_i .
- La i -ème case ($1 \leq i \leq N$) de la ligne du bas est B_i .

Vous êtes un être humain ayant deux pieds. Vous commencez avec un pied sur la case A_1 et l'autre sur la case B_i . Vous devez vous déplacer afin d'avoir un pied sur la case A_N et l'autre sur la case B_N .

Vous vous déplacez en faisant des *pas* : à chaque pas, choisissez un pied et déplacez le sur une autre case, vers la droite, sur la même ligne. Après chaque pas, vos deux pieds doivent être à une distance d'au plus K colonnes. Plus formellement, si un de vos pied est sur la case A_i et l'autre sur la case B_j , alors $|i - j| \leq K$ doit être satisfait.

Vous *pouvez* avancer le même pied lors de plusieurs pas consécutifs.

À la fin, votre *score* est la somme des valeurs des cases sur lesquelles vous avez marché (en incluant les cases de début et de fin). Quel est le score maximum possible ?

Sous-tâches et Contraintes

Pour chaque sous-tâche, il est garanti que :

- $1 \leq N \leq 100\,000$.
- $1 \leq K \leq 100\,000$.
- $-10\,000 \leq A_i \leq 10\,000$.
- $-10\,000 \leq B_i \leq 10\,000$.

Des contraintes supplémentaires sont données dans le tableau ci-dessous.

Sous-tâche	Points	Contraintes supplémentaires
1	12	$K \leq 5$
2	24	$A_i = 0$ ou -1 , pour tout i . $B_i = 0$ ou -1 , pour tout i .
3	6	$N \leq 300$
4	20	$N \leq 3\,000$
5	38	Aucune contrainte supplémentaire.

Entrée

- La première ligne de l'entrée contient deux entiers, N et K .
- La deuxième ligne de l'entrée contient N entiers. Le i -ème d'entre eux est A_i .
- La troisième ligne de l'entrée contient N entiers. Le i -ème d'entre eux est B_i .

Sortie

La sortie doit contenir un seul entiers : le score maximum atteignable.

Entrée d'exemple 1

```
4 1
0 2 2 8
0 -10 5 2
```

Sortie d'exemple 1

19

Entrée d'exemple 2

```
7 2
0 -10 -6 2 -10 0 0
5 3 -2 -1 -10 -10 0
```

Sortie d'exemple 2

9

Explications

Dans l'exemple 1, le score maximum est de $0 + 2 + 2 + 8 + 0 + 5 + 2 = 19$.

Dans l'exemple 2, le score maximum est de $0 + 2 + 0 + 0 + 5 + 3 + -1 + 0 = 9$.

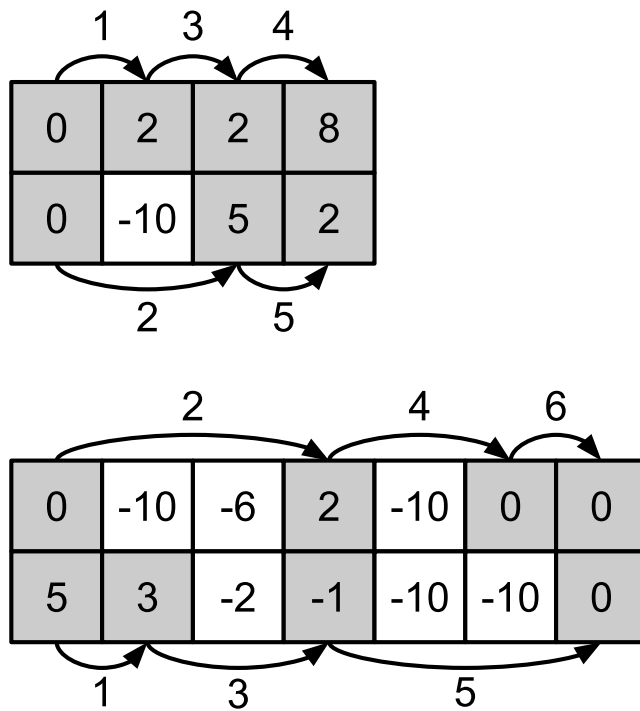


Figure 1: Les pas sont numérotés dans l'ordre dans lesquels vous devez les faire. Les cases grisées sont les cases sur lesquels vous avez marché.

Loss of Balance

Fichier d'entrée	Fichier de sortie	Limite de temps	Limite de mémoire
entrée standard	sortie standard	2 secondes	256 MiB

Un tableau A est (x, y) -acceptable, si il existe deux indices i et j tels que :

- $i < j$, et
- $A_i = x$ et $A_j = y$.

Un tableau A est K -équilibré si et seulement si :

- Tous les éléments du tableaux sont des entiers entre 1 et K .
- Chaque entier entre 1 et K apparaît *au moins une fois* dans A .
- A est (x, y) -acceptable et (y, x) -acceptable pour tous x et y (de 1 à K) avec $x \neq y$.

Par exemple :

- $[1, 2, 3, 4]$ n'est pas 3-équilibré, car il contient un 4.
- $[1, 3]$ n'est pas 3-équilibré, car il ne contient pas de 2.
- $[1, 2, 1, 3, 1]$ n'est pas 3-équilibré, car il n'est pas $(3, 2)$ -acceptable.
- $[1, 2, 3, 2, 1]$ est 3-équilibré.

Hugo avait un tableau A contenant N éléments qui est K -équilibré. Malheureusement, il a perdu son tableau et aimerait que vous l'aidiez à le retrouver.

Par chance, Hugo se rapelle de l'ordre relatif de certains des éléments de A . Plus précisément, il a un tableau B également de longueur N . Pour tous i et j :

- si $B_i = B_j$, alors $A_i = A_j$,
- si $B_i > B_j$, alors $A_i \geq A_j$ (inégalité **large**), et
- si $B_i < B_j$, alors $A_i \leq A_j$ (inégalité **large**).

Par exemple :

- Si $B = [1, 5, 2]$, alors $A_1 \leq A_3 \leq A_2$.
- Si $B = [6, 3, 7, 7, 3, 7]$, alors $A_3 = A_4 = A_6$ et $A_2 = A_5$. De plus, $A_5 \leq A_1 \leq A_6$.

Aidez Hugo à retrouver A , ou prévenez le que c'est impossible !

Sous-tâches et Contraintes

Dans toutes les sous-tâches, il est garanti que :

- $2 \leq N \leq 200\,000$.
- $2 \leq K \leq N$.
- $1 \leq B_i \leq 200\,000$, pour tout i .

Des contraintes supplémentaires sont données dans le tableau ci-dessous.

Sous-tâche	Points	Contraintes supplémentaires
1	10	$N \leq 10$ et $B_i \leq K$, pour tout i .
2	20	$B_i \leq K$, pour tout i .
3	30	$N \leq 3000$
4	10	$K \leq 5$
5	30	Aucune contrainte supplémentaire.

Entrée

- La première ligne de l'entrée contient deux entiers, N et K .
- La deuxième ligne de l'entrée contient N entiers. Le i -ème entier est B_i .

Sortie

La sortie doit contenir N entiers sur une seule ligne. Le i -ème de ces entiers doit être A_i , et chaque entier **doit** être entre 1 et K .

Le tableau A affiché doit être K -équilibré **et** doit correspondre à l'ordre relatif donné par B .

Si il existe plusieurs solutions, vous pouvez afficher **celle de votre choix**. Si il n'existe pas de solution, affichez -1 à la place.

Entrée d'exemple 1

```
5 3
5 300 900 300 6
```

Sortie d'exemple 1

```
1 2 3 2 1
```

Entrée d'exemple 2

```
10 5
1 2 3 4 5 6 7 8 9 10
```

Sortie d'exemple 2

```
-1
```

Entrée d'exemple 3

```
6 2
1 2 2 1 2 1
```

Sortie d'exemple 3

```
1 2 2 1 2 1
```

Explications

Dans l'exemple 1, la **seule** sortie correcte est $[1, 2, 3, 2, 1]$ (le tableau est 3-équilibré **et** il satisfait l'ordre relatif défini par B).

Notez que, par exemple, les trois sorties suivantes sont **incorrectes** :

- $[3, 1, 2, 3, 1]$ est 3-équilibré, mais ne satisfait pas l'ordre relatif de B
- $[1, 2, 3, 2, 2]$ satisfait l'ordre relatif de B , mais n'est pas 3-équilibré.
- $[1, 1, 2, 1, 1]$ satisfait l'ordre relatif de B , mais n'est pas 3-équilibré.

Dans l'exemple 2, il est impossible de créer un tableau croissant 5-équilibré.

Dans l'exemple 3, la seule sortie correcte est $[1, 2, 2, 1, 2, 1]$.

Notez que l'exemple 3 vérifie les contraintes de toutes les sous-tâches.

Emergency Reinforcement

Fichier d'entrée	Fichier de sortie	Limite de temps	Limite de mémoire
entrée standard	sortie standard	5 secondes	256 MiB

Dans un archipel, N îles (numérotées de 1 à N) sont connectées par E ponts (numérotés de 1 à E). Les ponts ont été construits par K entreprises (numérotées de 1 à K).

Le i -ème pont connecte les îles A_i et B_i , et a été construit par l'entreprise C_i . Deux îles peuvent être connectées par plusieurs ponts. Aucun pont ne connecte une île à elle même.

Bientôt, un tremblement de terre frappera l'archipel et détruira tous les pont ! Chaque entreprise a X euros qui peuvent être dépensés pour *renforcer* les ponts quelle a construit, et les sauver d'une destruction certaine. Pour renforcer le i -ème pont, l'entreprise C_i doit dépenser D_i euros.

Les entreprises souhaitent minimiser le nombre total de composantes connexes après le tremblement de terre. Deux île sont dans la même composante connexe s'il est possible de se rendre d'une île à l'autre en empruntant des ponts renforcés. Pouvez vous les aider ?

Note : lire attentivement la section Score ci-dessous.

Sous-tâches et Contraintes

Dans toutes les sous-tâches, il est garanti que :

- $1 \leq N \leq 10\,000$.
- $1 \leq E \leq 100\,000$.
- $1 \leq K \leq 5\,000$.
- $1 \leq X \leq 1\,000\,000\,000$.
- $1 \leq A_i \leq N$.
- $1 \leq B_i \leq N$.
- $A_i \neq B_i$, pour tout i .
- $1 \leq C_i \leq K$.
- $1 \leq D_i \leq X$.

Dans ce problème, chaque sous-tâche contient **exactement un fichier test**. Les fichiers tests sont téléchargeables depuis la page "Attachments".

Sous-tâche	Points	Contraintes supplémentaires
1	5	$N = 7$ et $E = 8$.
2	5	Entre deux îles, il existe toujours un unique chemin (séquence de ponts).
3	15	$K = 1$.
4	15	$B_i = N$, pour tout i . Si $A_i = A_j$ alors $D_i = D_j$, pour tout i, j .
5	15	$D_i = 1$, pour tout i . Chaque île est connectée à au plus deux autres îles.
6	15	$D_i = 1$, pour tout i .
7	15	-
8	15	-

Vous êtes encouragés à regarder le contenu des fichiers tests, et d'expérimenter sur votre ordinateur. Soumettre un code qui affiche une solution codée en dur (hardcoded) pour un des fichiers test **est** autorisé.

Entrée

- La première ligne de l'entrée contient quatre entiers, N , E , K et X .
- Suivent E lignes. La i -ème ligne contient quatre entiers A_i , B_i , C_i et D_i .

Sortie

Affichez une seule ligne, contenant jusqu'à E entiers, les ponts que vous souhaitez renforcer (dans n'importe quel ordre).

Score

Si vous :

- affichez deux fois le même pont, ou
- affichez un nombre strictement inférieur à 1 ou strictement supérieur à E , ou
- dépensez trop d'argent pour une des entreprises,

le score de votre solution sera de 0 pour cette sous-tâche.

Sinon, votre score sera calculé en fonction de sa distance à la solution optimale. Étant donné deux paramètres INF et SUP , si le nombre de composantes connexes après le tremblement de terre est X , votre score sur cette sous-tâche sera de :

$$\min(100, \max(0, 100 * (SUP - X) / (SUP - INF)))$$

Les paramètres de score de chaque sous-tâche sont donnés dans le tableau ci-dessous.

Sous-tâche	INF	SUP
1	1	4
2	382	1000
3	58	176
4	1	3
5	151	1000
6	1	1000
7	1	150
8	1	1100

Entrée d'exemple 1

```
10 11 8 1000
1 2 7 100
1 3 7 100
2 3 4 750
2 3 1 1000
4 3 4 750
4 5 3 600
5 6 3 601
5 6 3 602
6 4 3 603
4 6 3 604
7 8 3 100
```

Sortie d'exemple 1

```
2 4 5 7 11
```

Entrée d'exemple 2

```
4 5 5 12345
4 1 1 12345
1 2 2 12345
2 4 5 12345
4 3 2 12345
3 2 3 12345
```

Sortie d'exemple 2

```
1 2 3 5
```

Explications

Dans l'exemple 1, chaque entreprise a $X = 1000$ euros qu'elle peut dépenser:

- L'entreprise 1 renforce le 4-ème pont, ce qui coûte 1000 euros.
- L'entreprise 3 renforce les 7-ème et 11-ème ponts, ce qui coûte $601 + 100 = 701$ euros.
- L'entreprise 4 renforce le 5-ème pont, ce qui coûte 750 euros.
- L'entreprise 7 renforce le 2-ème pont, ce qui coûte 100 euros.

Après le tremblement de terre, il y aura 5 composantes connexes (deux d'entre elles sont des composantes de taille 1).

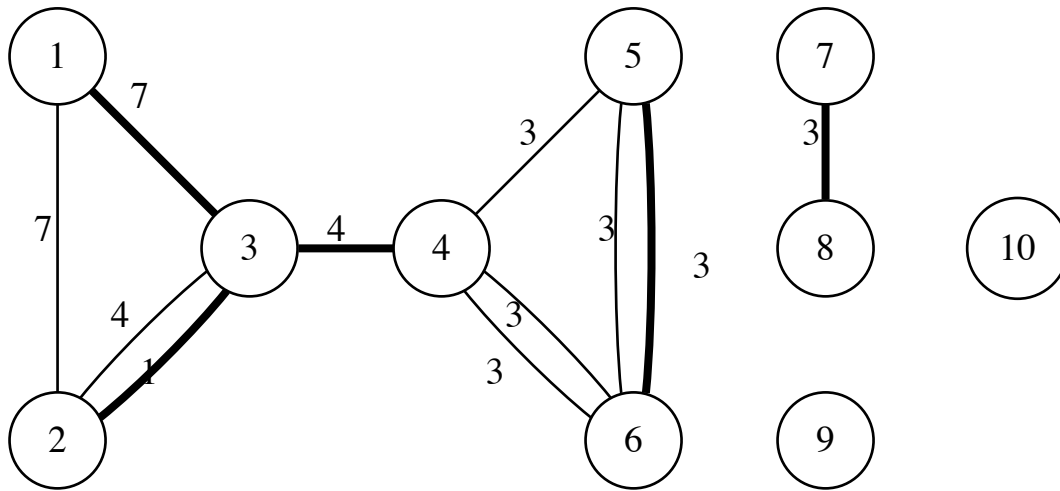


Figure 1: Exemple 1

Dans l'exemple 2, chaque entreprise a $X = 12345$ euros qu'elle peut dépenser:

- L'entreprise 1 renforce le 1-er pont, ce qui coûte 12345 euros.
- L'entreprise 2 renforce le 2-ème pont, ce qui coûte 12345 euros.
- L'entreprise 3 renforce le 5-ème pont, ce qui coûte 12345 euros.
- L'entreprise 5 renforce le 3-ème pont, ce qui coûte 12345 euros.

Après le tremblement de terre, il y aura 1 composante connexe.

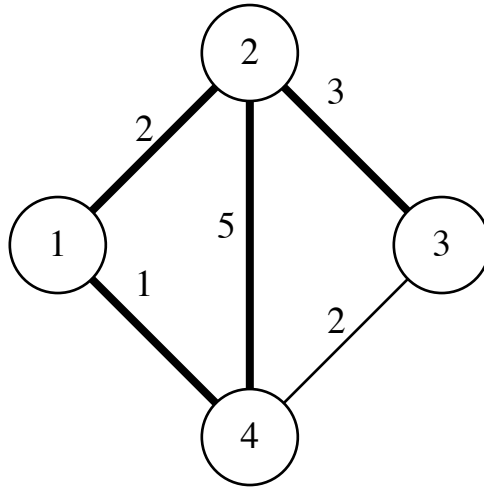


Figure 2: Exemple 2