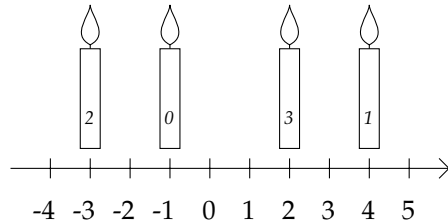


Bougies

La souris Binna aime les bougies et a décidé de les utiliser pour remplacer les ampoules dans sa chambre : elles sont plus jolies et sentent bon. Elle a placé N bougies le long d'un mur linéaire. La i -e bougie est placée exactement x_i mètres à droite du milieu du mur. Il peut y avoir plusieurs bougies sur la même coordonnée, et x_i est négatif si la i -e bougie est du côté gauche. Dans la visualisation ci-dessous, $N = 4$ et $x_0 = -1, x_1 = 4, x_2 = -3, x_3 = 1$.



Le problème est qu'allumer les bougies prend beaucoup de temps à Binna quand elle se réveille. Elle doit allumer au moins K bougies pour avoir assez de lumière dans la chambre. Elle veut le faire aussi vite que possible, donc elle veut marcher aussi peu que possible, en partant devant son lit à la coordonnée s . Elle se déplace seulement sur un axe le long du mur et peut allumer la bougie i quand elle se tient sur la coordonnée x_i . Peux-tu l'aider à déterminer la plus petite distance qu'elle devra marcher pour allumer K bougies ?

Entrée

Sur la première ligne, il y a trois nombres entiers, N , le nombre de bougies, K , le nombre de bougies que Binna veut allumer, et s , la coordonnée du lit de Binna.

Sur la deuxième ligne, il y a N nombres entiers x_0, \dots, x_{N-1} , les coordonnées de chacune de N bougies.

Sortie

Imprime un nombre entier, la distance minimale que Binna doit parcourir pour allumer K bougies en partant de s .

Limites

Il y a cinq sous-tâches. Dans chaque sous-tâche, $1 \leq N \leq 10^6$, $1 \leq K \leq N$, $-10^9 \leq x_i \leq 10^9$ et $-10^9 \leq s \leq 10^9$.

- Dans la sous-tâche 1, qui vaut 5 points, $N = K = 1$.
- Dans la sous-tâche 2, qui vaut 40 points, $N \leq 10^4$.
- Dans la sous-tâche 3, qui vaut 10 points, $x_i - x_{i-1} = c$ pour un nombre entier constant $c \geq 0$ et pour tout $1 \leq i < N$. De plus, pour un $0 \leq j < N$, $x_j = s$.
- Dans la sous-tâche 4, qui vaut 5 points, $x_i - x_{i-1} = c$ pour un nombre entier constant $c \geq 0$ et pour tout $1 \leq i < N$.
- Dans la sous-tâche 5, qui vaut 40 points, il n'y a pas d'autre restriction.



Exemples

Entrée	Sortie
6 3 9 11 20 6 13 2 16	7

Binna commence à la position 9 et allume les bougies aux positions 11, 13, et 16. Elle peut aller tout droit, et la distance totale est 7.

Entrée	Sortie
6 4 0 -3 7 5 -7 2 -8	12

Binna commence à la position 0, allume la bougie à la position 2, repart en arrière et allume les bougies aux positions -3, -7 et -8. La distance totale est 12.

Ordre de départ

La souris Stofl a organisé une grande fête dans sa grande maison. Il a invité N souris (numérotées de 0 à $N - 1$) et elles sont bien évidemment toutes venues. Elles sont arrivées dans un ordre bien précis. À la fête, il y a une grande pile de boulettes de fromage. À chaque fois qu'une souris arrive à la fête, elle reçoit une boulette de fromage par chacun de ses amis déjà présents à la fête au moment de son arrivée. Note que deux souris amies de Stofl ne sont pas forcément amies entre elles. Une souris peut donc ne pas recevoir de boulette de fromage de chaque autre souris déjà présente, mais chaque souris recevra au moins une boulette de fromage, car elles sont toutes amies avec Stofl.

Malheureusement, toutes les bonnes choses ont une fin, y compris les meilleures fêtes. Cependant, les amis de Stofl ne veulent pas partir si leur ordre de départ n'est pas équitable : les souris doivent partir dans le même ordre que celui dans lequel elles sont arrivées.

Il reste un petit problème : toutes les souris présentes ont oublié l'ordre d'arrivée ! La seule chose dont elles se souviennent est (bien évidemment) combien de boulettes de fromage elles ont reçu à leur arrivée.

Stofl a donc trouvé une parade : il n'a qu'à prétendre se souvenir de l'ordre d'arrivée. Il lui suffit de vérifier que, si ses amis étaient effectivement arrivés dans cet ordre, ils auraient reçu le bon nombre de boulettes de fromage.

Étant donné le nombre de boulettes de fromage reçus par chaque ami de Stofl, détermine un ordre d'arrivée possible.

Entrée

La première ligne de l'entrée contient un entier N – le nombre d'amis de Stofl présents à la fête. La deuxième ligne contient N entiers entre 1 et N , le i -ème entier est le nombre de boulettes de fromage reçu par le i -ème ami.

La troisième ligne de l'entrée contient un entier M – le nombre d'amitiés supplémentaires parmi les amis de Stofl.

La j -ème des M lignes suivantes contient deux entiers a_j et b_j ($0 \leq a_j < b_j < N$), qui indiquent que la souris a_j et la souris b_j sont amies l'une avec l'autre. Chaque paire (a_j, b_j) apparaît au plus une fois.

Note que Stofl est ami avec toutes les autres souris, mais que ces amitiés ne sont pas incluses dans l'entrée.

La dernière ligne de l'entrée contient un entier L qui est soit 0, soit 1, et dont la signification est expliquée plus bas.

Sortie

Imprime une ligne contenant les N premiers entiers non négatifs dans un ordre dans lequel les amis de Stofl auraient pu arriver à la fête. S'il y a plusieurs solutions et $L = 0$, tu peux imprimer n'importe laquelle d'entre elles. Par contre, si $L = 1$, il t'est demandé de trouver la solution *lexicographiquement minimale*.

(Une solution valide p_0, p_1, \dots, p_{N-1} est dite lexicographiquement minimale si pour toute autre solution valide q_0, q_1, \dots, q_{N-1} il existe un index i tel que $p_i < q_i$ et $p_j = q_j$ pour tous les indices $j < i$. Il t'est donc demandé de trouver la *plus petite* solution, où les solutions sont comparées au premier indice où elles diffèrent.)

Limites

Il y a 5 sous-tâches, qui rapportent les points donnés ci-dessous.

- Dans la sous-tâche 1, qui rapporte 30 points, $1 \leq N \leq 1\,000$, $M = N(N - 1)/2$ et n'importe quelle solution valide sera acceptée ($L = 0$).



- Dans la sous-tâche 2, qui rapporte 20 points, $1 \leq N \leq 1\,000$, $0 \leq M \leq N(N - 1)/2$ et n'importe quelle solution valide sera acceptée ($L = 0$).
- Dans la sous-tâche 3, qui rapporte 20 points, $1 \leq N \leq 100\,000$, $0 \leq M \leq 500\,000$ et n'importe quelle solution valide sera acceptée ($L = 0$).
- Dans la sous-tâche 4, qui rapporte 15 points, $1 \leq N \leq 1\,000$, $0 \leq M \leq N(N - 1)/2$ et uniquement la solution lexicographiquement minimale sera acceptée ($L = 1$).
- Dans la sous-tâche 5, qui rapporte 15 points, $1 \leq N \leq 100\,000$, $0 \leq M \leq 500\,000$ et uniquement la solution lexicographiquement minimale sera acceptée ($L = 1$).

Dans tous les cas de test, il est garanti qu'il existe au moins une solution. En particulier, le nombre de boulettes de fromage reçu par chaque ami de Stofl est entre 1 et le nombre total de leurs amis invités à la fête.

Exemples

Entrée	Sortie
5 1 4 3 5 2 10 0 1 0 2 0 3 0 4 1 2 1 3 1 4 2 3 2 4 3 4 0	0 4 2 1 3

Cet exemple satisfait les contraintes de la sous-tâche 1.

Selon l'entrée, la souris 0 a reçu 1 boulette de fromage, la souris 1 en a eu 4, la souris 2 en a eu 3, la souris 3 en a eu 5 et la souris 4 en a eu 2.

De plus, toutes les paires de souris sont amies.

Comme $L = 0$, il n'est pas nécessaire de trouver la solution lexicographiquement minimale.

La sortie décrit le scénario où la souris 0 arrive en premier, puis la 4, la 2, la 1, et enfin la 3.

Cette sortie est une solution valide, car les souris auraient effectivement reçu le bon nombre de boulettes de fromage si elles étaient arrivées dans cet ordre :

Quand la souris 0 arrive, elle reçoit 1 boulette de fromage de la part de Stofl.

Quand la souris 4 arrive, elle reçoit 2 boulettes de fromage, un de la part de Stofl et un de la part de la souris 0.

Quand la souris 2 arrive, elle reçoit 3 boulettes de fromage, de la part de Stofl, la souris 0 et la souris 4.

Quand la souris 1 arrive, elle reçoit 4 boulettes de fromage, de la part de Stofl, la souris 0, la souris 4 et la souris 2.

Quand la souris 3 arrive, elle reçoit 5 boulettes de fromage, de la part de chaque autre souris..



Entrée	Sortie
5 1 4 2 3 1 6 2 3 0 1 1 2 0 2 1 4 3 4 0	0 4 2 1 3

Cet exemple satisfait les contraintes des sous-tâches 2 et 3.

Selon l'entrée, la souris 0 a reçu 1 boulette de fromage, la souris 1 en a eu 4, la souris 2 en a eu 2, la souris 3 en a eu 3, et la souris 4 en a eu 1.

Dans cet exemple, toutes les paires de souris ne sont pas forcément amies : la souris 0 n'est pas amie avec les souris 3 et 4. De plus, la souris 1 n'est pas amie avec la souris 3, et la souris 2 n'est pas amie avec la souris 4.

De nouveau, $L = 0$, et il n'est donc pas nécessaire de trouver la solution lexicographiquement minimale.

La sortie est la même que celle de l'exemple précédent.

Cette sortie est une solution valide, car les souris auraient effectivement reçu le bon nombre de boulettes de fromage si elles étaient arrivées dans cet ordre :

Quand la souris 0 arrive, elle reçoit 1 boulette de fromage de la part de Stofl.

Quand la souris 4 arrive, elle reçoit 1 boulette de fromage, de la part de Stofl (elle n'est pas amie avec la souris 0).

Quand la souris 2 arrive, elle reçoit 2 boulettes de fromage, de la part de Stofl et de la souris 0 (elle n'est pas amie avec la souris 4).

Quand la souris 1 arrive, elle reçoit 4 boulettes de fromage, de la part de Stofl, la souris 0, la souris 4 et la souris 2.

Quand la souris 3 arrive, elle reçoit 3 boulettes de fromage, de la part de Stofl, la souris 2 et la souris 4 (elle n'est amie ni avec la souris 0, ni avec la souris 1).



Entrée	Sortie
5 1 4 2 3 1 6 2 3 0 1 1 2 0 2 1 4 3 4 1	0 2 4 1 3

Cet exemple satisfait les contraintes des sous-tâches 4 et 5.

L'entrée est la même que celle de l'exemple précédent.

Note que si nous échangeons l'ordre d'arrivée des souris 2 et 4 dans la sortie de l'exemple précédent, cela ne change pas le nombre de boulettes de fromage obtenu par les souris. En effet, elles arrivent l'une droit après l'autre, et elles ne se donnent pas de boulette de fromage puisqu'elles ne sont pas amies. Note that if we swap the order of arrival for Mouse 2 and Mouse 4

Ce nouvel ordre est donc également valide. En fait, pour cet exemple, ce nouvel ordre est la solution lexicographiquement minimale. Dans les sous-tâches 4 et 5, c'est la seule sortie qui remportera des points.

Comme $L = 1$, cet exemple est jugé comme les sous-tâches 4 et 5, et il est possible que ta solution ne réussisse pas cet exemple si elle ne peut pas résoudre les sous-tâches 4 et 5.

Timber!

La souris Binna a rejoint les Communautés Organisées et Unifiées pour la Protection de l'Environnement Sylvestre (COUPES), une organisation qui s'intéresse aux pratiques écologiques d'abattage des arbres.

Les principales convictions des membres des COUPES sont les suivantes :

1. Si vous abattez des arbres, rapportez au moins s unités de bois. En faisant de grosses récoltes, l'écosystème est très peu perturbé.
2. Essayez d'espacer les arbres abattus autant que possible. Soit d la distance minimale entre deux arbres qui ont été abattus. Trouvez le plus grand d possible.
3. Lorsqu'un arbre tombe, il faut crier "timber!".

La souris Binna a bien maîtrisé le point 3, mais n'arrive pas à trouver comment satisfaire les points 1 et 2. Elle aimerait donc que tu l'aides.

La forêt peut être modélisée par n arbres sur une ligne. Tu connais leurs tailles h_0, h_1, \dots, h_{n-1} ainsi que le paramètre s . Tu peux soit abattre un arbre, soit le laisser tel quel. La somme des hauteurs de tous les arbres abattus doit être au moins égale à s . Calcule le plus grand d tel que deux arbres abattus soient séparés d'au moins d et que la somme de leurs hauteurs soit d'au moins s .

Entrée

La première ligne de l'entrée contient deux entiers n et s , le nombre d'arbres et la somme minimale des hauteurs des arbres abattus.

La deuxième ligne contient les nombres $n, h_0, h_1, \dots, h_{n-1}$, les hauteurs des arbres.

Sortie

S'il est impossible d'obtenir s unités de bois dans la forêt, imprime une seule ligne contenant "Impossible".

S'il est possible d'atteindre s avec zéro ou un arbre, imprime une seule ligne contenant "Infinity".

Sinon, imprime une seule ligne avec un nombre entier d – le plus grand d tel qu'il est possible d'abattre des arbres séparés d'une distance d'au moins d et dont la somme des hauteurs est au moins s .

Limites

Il y a 7 sous-tâches. Dans toutes les sous-tâches, nous avons $1 \leq h_i \leq 10^9$ et $0 \leq s \leq 10^{16}$.

- Dans la sous-tâche 1, qui vaut 10 points, nous avons $1 \leq n \leq 10$.
- Dans la sous-tâche 2, qui vaut 10 points, nous avons $1 \leq n \leq 10\,000$ et tous les arbres ont une hauteur distincte et sont ordonnés de manière croissante (c'est-à-dire $h_0 < h_1 < \dots < h_{n-1}$).
- Dans la sous-tâche 3, qui vaut 10 points, nous avons $1 \leq n \leq 10\,000$.
- Dans la sous-tâche 4, qui vaut 10 points, nous avons $1 \leq n \leq 10^5$ et tous les arbres ont la même hauteur (c'est-à-dire $h_0 = h_1 = \dots = h_{n-1}$).
- Dans la sous-tâche 5, qui vaut 20 points, nous avons $1 \leq n \leq 10^5$ et tous les arbres ont une hauteur distincte et sont ordonnés de manière croissante (c'est-à-dire $h_0 < h_1 < \dots < h_{n-1}$).
- Dans la sous-tâche 6, qui vaut 20 points, nous avons $1 \leq n \leq 10^5$.
- Dans la sous-tâche 7, qui vaut 20 points, nous avons $1 \leq n \leq 10^6$.



Exemples

Entrée	Sortie
8 14 4 5 2 2 3 1 1 5	2

En abattant les arbres aux positions $[0, 2, 4, 7]$, on obtient exactement $4 + 2 + 3 + 5 = 14$ bois.

Entrée	Sortie
7 27 1 2 3 5 8 13 21	3

En abattant les arbres aux positions $[0, 3, 6]$, on obtient exactement $1 + 5 + 21 = 27$ bois. Cet exemple est une entrée valide pour les sous-tâches 2 et 5.

Entrée	Sortie
11 13 5 6 1 1 4 1 1 1 1 4 1	4

Entrée	Sortie
5 6 5 5 5 5 5	4

Entrée	Sortie
5 6 1 1 1 1 1	Impossible

Même en abattant tous les arbres, il n'y a pas assez de bois.

Entrée	Sortie
5 5 5 5 5 5 5	Infinity

Abattre un arbre suffit, donc la distance jusqu'à un autre arbre abattu est infinie.

Entrée	Sortie
4 4 1 1 1 1	1

Jeu de rôles

Stofl la souris a beaucoup joué, ces derniers temps, à des RPG (des jeux de rôles pour ordinateur). Il a investi des centaines d'heures dans un jeu appelé *L'Effet des dieux primordiaux*. Tous les amis de Stofl s'inquiètent : ils ont peur qu'il ne quitte plus jamais sa chambre. Peux-tu aider Stofl à terminer le dernier niveau du jeu pour qu'il puisse à nouveau sortir de chez lui ?

Dans le jeu, Stofl maîtrise deux attaques : il peut attaquer à l'aide de son bras gauche ou de son bras droit. S'il utilise son bras gauche, il doit attendre l secondes avant de le faire à nouveau. S'il utilise son bras droit, il doit attendre r secondes avant de le faire à nouveau. (C'est ce qu'on appelle généralement un *cooldown* ou temps de recharge.) Note que les deux bras ne s'affectent pas mutuellement : même si Stofl attend de pouvoir réutiliser son bras droit, il peut encore utiliser son bras gauche s'il est prêt, et vice-versa.

Stofl veut actuellement terminer un niveau avec n monstres, où le i -e monstre apparaît t_i secondes après qu'il commence le niveau (les monstres sont donnés dans un ordre temporel non décroissant d'apparition). Quand un monstre apparaît, Stofl peut le frapper avec un de ses deux bras. S'il le fait, le monstre meurt. Sinon, le monstre frappe Stofl puis disparaît.

Stofl commence avec un nombre (entier et positif) de points de santé h , symbolisant son état de santé. Chaque fois qu'un monstre frappe Stofl, ses points de santé diminuent de 1. Si ses points de santé atteignent 0, Stofl meurt.

Quel est le nombre minimal de points de santé (la valeur minimale de h) que Stofl doit avoir au début du niveau pour survivre ?

Remarque : note que la limite de mémoire est plus basse qu'habituellement.

Entrée

Sur la première ligne, il y a trois nombres entiers n , l et r , dénotant le nombre de monstres et le temps de recharge du bras gauche et du bras droit. Sur la deuxième ligne, il y a n nombres entiers t_i , dénotant l'instant auquel le i -e monstre apparaît. Les monstres sont donnés dans un ordre temporel non décroissant (c'est-à-dire que si $i < j$, alors $t_i \leq t_j$).

Sortie

Imprime un unique nombre entier, la quantité minimale de points de santé dont Stofl a besoin au début pour survivre au niveau.

Limites

Il y a cinq sous-tâches. Dans chacune d'entre elles, $1 \leq n \leq 10^6$, $0 \leq t_0 \leq t_1 \leq \dots \leq t_{n-1} \leq 10^9$ et $1 \leq l, r \leq 10^9$.

- Dans la sous-tâche 1, valant 10 points, $t_i < 10^9$ et $r = 10^9$.
- Dans la sous-tâche 2, valant 15 points, $l = r$.
- Dans la sous-tâche 3, valant 10 points, $n \leq 16$.
- Dans la sous-tâche 4, valant 20 points, $n \leq 3 \cdot 10^3$.
- Dans la sous-tâche 5, valant 15 points, $n \leq 2 \cdot 10^4$.
- Dans la sous-tâche 6, valant 30 points, il n'y a pas de restrictions additionnelles.



Exemples

Entrée	Sortie
6 3 5 1 2 4 5 6 11	2

Cet exemple satisfait les contraintes pour les sous-tâches 3, 4, 5 et 6.

Si Stofl commence avec 2 points de santé, il peut utiliser son bras droit pour attaquer le premier, le cinquième et le sixième monstres, et son bras droit pour attaquer le deuxième et le quatrième monstres. Il ne se fait alors frapper que par le troisième monstre, ce qui réduit ses points de santé à 1. Mais cela ne le tue pas, donc Stofl survit.

Entrée	Sortie
5 5 3 1 2 4 7 14	1

Cet exemple satisfait les contraintes pour les sous-tâches 3, 4, 5 et 6.

Si Stofl commence avec 1 point de santé, il peut utiliser son bras droit pour frapper le premier, le troisième et le cinquième monstres, et son bras gauche pour frapper le deuxième et le quatrième monstres. Stofl ne se fait alors frapper par aucun monstre et il survit.

Entrée	Sortie
7 4 1000000000 0 4 8 10 12 14 16	2

Cet exemple satisfait les contraintes pour les sous-tâches 1, 3, 4, 5 et 6.

Entrée	Sortie
9 4 4 1 2 3 6 7 7 9 11 13	4

Cet exemple satisfait les contraintes pour les sous-tâches 2, 3, 4, 5 et 6.