



Distribution de bonbons

Stofl a trois sœurs – Ada, Binna et Claudia. Les trois ont collecté a , b et c bonbons respectivement. Comme elles se disputent souvent en cas d'inégalité entre elles, leur père leur donne des bonbons supplémentaires afin que chacune en ait la même quantité.

Le père a un stock illimité de bonbons. Il les distribue en un certain nombre de tours. Durant chaque tour, il prend deux bonbons, un dans chaque patte. Chacun de ces bonbons est donné à une des sœurs et chacun à une sœur différente. Ainsi, deux des sœurs reçoivent exactement un bonbon par tour.

Quel est le nombre minimum de tours requis pour obtenir une distribution égale de bonbons ?

Entrée

La seule ligne de l'entrée contient trois nombres entiers positifs séparés par des espaces, a , b and c .

Sortie

Imprime un seul nombre entier – le nombre minimum de tours nécessaires.

Si les nombres de bonbons ne peuvent pas être égalisés par ce procédé en un nombre fini de tours, imprime -1 à la place.

Limites

Il y a deux groupes de test, valant chacun 50 points.

- Dans le premier, $a, b, c \leq 10^4$.
- Dans le second, $a, b, c \leq 10^{18}$.

Exemples

Entrée	Sortie
1 2 3	3

Le père donne des bonbons deux fois à Ada et Binna et une fois à Ada et Claudia. Après ces trois tours, chaque souris aura quatre bonbons.

Entrée	Sortie
2 2 2	0

Les souris ont déjà le même nombre de bonbons et le père n'a donc rien à faire.



Ateliers

Il t'a été demandé d'organiser les ateliers de préparation pour l'année prochaine de la SOI. Pour t'assurer que chaque personne intéressée soit disponible pour au moins un atelier, tu veux en organiser autant que possible. C'est une tâche hardue, car chaque atelier requiert la présence d'un certain nombre d'organisateur. Plus précisément, il y a N tâches à exécuter pour chaque atelier et la tâche i requiert l'attention exclusive de a_i volontaires par atelier. Or, un nombre limité d'organisateur est disponible et chacun d'entre eux n'est prêt à aider que pour une tâche spécifique qui l'intéresse, et ce pour un seul atelier. En revanche, l'atelier auquel les organisateurs doivent participer leur est totalement égal. Il y a b_i organisateurs disponibles pour la tâche i .

Cela cause souvent des pénuries de volontaires pour certaines tâches en particulier – par exemple, l'équipe de cuisine n'est pas très populaire. Heureusement, les olympiades ont à leur disposition quelques bénévoles supplémentaires particulièrement dévoués et prêts à accomplir n'importe quelle tâche qui leur est demandée – appelons-les les "jokers". Ils sont au nombre de J . Chacun d'entre eux ne peut participer qu'à un atelier et n'est capable d'aider que pour une seule tâche lors de celui-ci.

Combien d'ateliers peux-tu organiser au total si tu attribues des tâches aux jokers de façon optimale?

Entrée

La première ligne contient N et J , le nombre de tâches par atelier et le nombre de jokers. La deuxième ligne contient N nombres entiers : a_0, \dots, a_{N-1} , le nombre de bénévoles nécessaires pour chaque tâche par atelier. La troisième ligne contient N nombres entiers : b_0, \dots, b_{N-1} , les nombres de bénévoles disponibles pour chaque tâche.

Sortie

Imprime le nombre maximal d'ateliers que tu peux organiser.

Limites

Il y a trois groupes de tests.

- Dans le premier, qui vaut 20 points, $1 \leq N \leq 5000$, $0 \leq J \leq 5000$ et $1 \leq a_i, b_i \leq 5000$.
- Dans le deuxième, qui vaut 30 points, $1 \leq N \leq 10^5$, $J = 0$ et $1 \leq a_i, b_i \leq 10^9$.
- Dans le troisième, qui vaut 50 points, $1 \leq N \leq 10^5$, $0 \leq J \leq 10^9$ et $1 \leq a_i, b_i \leq 10^9$.

Exemples

Entrée	Sortie
3 1 2 1 4 11 3 16	4

Tu as déjà assez de volontaires pour les tâches 0 et 2 pour quatre ateliers. Il te manque uniquement un organisateur pour la tâche 1, puisque $4 \cdot 1 - 3 = 1$, donc tu utilises ton seul joker pour cela.



Entrée	Sortie
4 10 7 4 6 3 6 8 7 1	1

Il y a déjà assez de volontaires pour les tâches 1 et 2 pour un atelier. Il y a largement assez de jokers disponibles pour organiser un atelier (il ne manque que trois bénévoles), mais il en faudrait 18 pour en organiser deux.

Péages

Au pays des souris, il y a N villes connectées par M segments d'autoroute appartenant à l'État et K segments d'autoroute appartenant à des privés. Chaque route peut être utilisée dans les deux sens et on peut atteindre toutes les villes depuis n'importe quelle ville en utilisant une (ou plusieurs) route(s). On peut utiliser les routes publiques gratuitement, mais il faut payer une taxe de **1 franc** par segment d'autoroute privée utilisé.

La souris Stolf vient de fonder une entreprise pour les conducteurs et a déjà reçu Q requêtes de la forme "Combien coûte le trajet le moins cher de la ville s à la ville t ?". Pourrais-tu aider Stolf à gérer ces requêtes le plus rapidement possible ?

Entrée

La première ligne d'entrée consiste en quatre entiers : le nombre de villes, N , le nombre de routes publiques, M , le nombre de routes privées, K , et le nombre de requêtes, Q . Les M lignes d'entrée suivantes consistent en deux entiers, a_i et b_i ($0 \leq a_i, b_i < N$, $a_i \neq b_i$), qui décrivent un segment d'autoroute *public* entre les villes a_i et b_i . Les K lignes d'entrée suivantes consistent en deux entiers, a_i et b_i ($0 \leq a_i, b_i < N$, $a_i \neq b_i$), qui décrivent un segment d'autoroute *privé* entre les villes a_i et b_i . Les dernières Q lignes d'entrée consistent de deux entiers, s_i et t_i ($0 \leq s_i, t_i < N$), qui décrivent les deux villes apparaissant dans une requête. Il peut y avoir plusieurs segments d'autoroute publics et privés entre les deux mêmes villes.

Sortie

Pour chacune des Q requêtes, retourne le prix de trajet entre les deux villes.

Limites

Il y a quatre groupes de tests, chacun valant 25 points.

- Dans le groupe 1, $1 \leq N \leq 500$, $0 \leq M \leq 10^4$, $0 \leq K \leq 500$, $1 \leq Q \leq 10^4$.
- Dans le groupe 2, $1 \leq N \leq 500$, $0 \leq M \leq 10^4$, $0 \leq K \leq 500$, $1 \leq Q \leq 10^6$.
- Dans le groupe 3, $1 \leq N \leq 10^4$, $0 \leq M \leq 10^5$, $0 \leq K \leq 10^3$, $1 \leq Q \leq 10^4$.
- Dans le groupe 4, $1 \leq N \leq 10^4$, $0 \leq M \leq 10^5$, $0 \leq K \leq 10^3$, $1 \leq Q \leq 10^6$.

Exemples

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



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



Parts de pizza

Exactement n souris ont participé à un atelier de la SOI et sont désormais affamées. Binna a donc prévu de commander des pizzas. Chaque pizza pèse exactement $A \cdot B$ grammes et Binna ne peut commander que des pizzas entières. La pizza peut être commandée comme A tranches de B grammes chacune ou comme B tranches de A grammes chacune. Binna peut décider comment découper chaque pizza indépendamment. La i -ème souris veut manger au moins g_i grammes de pizza mais est aussi d'accord d'en manger plus. En raison de considérations hygiéniques, deux souris ne peuvent pas partager des tranches, donc chaque souris doit recevoir une ou plusieurs tranches et les manger. Quel est le nombre minimal de pizzas que Binna doit commander ?

Entrée

La première ligne de l'entrée contient trois nombres entiers n, A, B – le nombre de souris et deux paramètres décrivant les pizzas comme expliqué ci-dessus. La deuxième ligne contient n nombres entiers g_i – le nombre minimal de grammes de pizza que la i -ème souris souhaite manger.

Sortie

Imprime un unique nombre entier – le nombre minimal de pizzas que Binna doit commander.

Limites

Dans chaque test, $1 \leq n, 1 \leq g_i \leq 10^9$ et $1 \leq A, B$.

- Sous-tâche 1 (15 points) : $n \leq 20, A, B \leq 30, g_i \leq 2 \cdot AB$.
- Sous-tâche 2 (15 points) : $n \leq 80, A, B \leq 100, g_i \leq 2 \cdot AB$.
- Sous-tâche 3 (15 points) : $n \leq 80, A, B \leq 100$.
- Sous-tâche 4 (20 points) : $n \leq 300, A, B \leq 400, g_i \leq 2 \cdot AB$.
- Sous-tâche 5 (5 points) : $n \leq 300, A, B \leq 400$.
- Sous-tâche 6 (20 points) : $n \leq 2 \cdot 10^4, A, B \leq 400$.
- Sous-tâche 7 (10 points) : $n \leq 3 \cdot 10^6, A, B \leq 300$.

Exemples

Entrée	Sortie
6 5 7 5 7 2 4 5 1	1

Binna peut commander une seule pizza coupée en 7 tranches de 5 grammes chacune. La deuxième souris reçoit deux tranches et chaque autre souris exactement une.

Entrée	Sortie
5 6 4 4 4 4 6 6	2

Une seule pizza pèse $4 + 4 + 4 + 6 + 6 = 24$ grammes, ce qui est suffisant, mais il est impossible de la découper de façon à ce que chaque souris reçoive assez de pizza. Donc Binna doit en commander deux.



Entrée	Sortie
5 6 5 12 15 15 10 6	2

Entrée	Sortie
2 8 10 800000000 320000000	14000000