

Tondeuse à gazon

Après ses aventures à la forteresse de Poenari, Vlad rentre chez lui et, en véritable Roumain, sa première pensée est qu'il doit nourrir son cheval. Le cheval n'est pas très difficile en matière de nourriture, alors Vlad utilise sa pelouse comme principale source de nourriture pour lui.

Pour cette tâche, Vlad dispose d'une tondeuse à gazon ayant une capacité de c . Il a décidé de diviser sa pelouse en n allées, numérotées de 0 à $n - 1$, qu'il doit tondre dans cet ordre. Chaque allée i contient une quantité d'herbe non tondue $v[i]$ et, pour des raisons inconnues, il faut $a[i]$ secondes à Vlad pour pousser la tondeuse sur cette allée.

Après être poussé sur quelques allées, la tondeuse peut atteindre sa capacité maximale ; dans ce cas, elle cesse de tondre l'herbe et en laisse une partie sur l'allée. Chaque fois que cela se produit, son réservoir collecteur doit être vidé, ce qui prend b secondes et ne peut être fait qu'à la fin d'une allée. Si le réservoir se remplit pendant que Vlad pousse la tondeuse sur l'allée i , il doit continuer à pousser la tondeuse jusqu'à la fin de l'allée, vider le réservoir, puis pousser la tondeuse sur l'allée une fois de plus (ou autant de fois que nécessaire) afin de éliminer l'herbe restante. Par exemple, si pour une allée i il faut y pousser la tondeuse 3 fois pour éliminer toute l'herbe, cela prendra $a[i] + b + a[i] + b + a[i]$ secondes. **Après avoir tondu toute la pelouse, la tondeuse doit être vidée.**

Après avoir beaucoup réfléchi et s'être plaint que cela lui prendrait beaucoup trop de temps pour finir de tondre, Vlad en est arrivé à la conclusion qu'il pourrait parfois être plus efficace, en termes de temps, de vider le réservoir même avant qu'il n'atteigne sa capacité maximale. Cependant, il ne sait pas quelle est la meilleure stratégie à adopter. Il demande donc votre aide.

Étant donné la quantité d'herbe sur chaque allée, le nombre de secondes nécessaires pour pousser la tondeuse sur chaque allée, la capacité du réservoir et le temps qu'il faut pour le vider, trouvez la meilleure façon pour Vlad de terminer la tonte de sa pelouse en un minimum de temps.

Détails d'implémentation

Vous devez implémenter la procédure suivante.

```
long long mow(int n, int c, int b, std::vector<int> & a, std::vector<int> & v);
```

(note du traducteur : le nom de la fonction "mow" signifie "tondre" en français)

- n : le nombre d'allées de la pelouse
- c : la capacité totale du réservoir collecteur
- b : le nombre de secondes nécessaires pour vider le réservoir

- a : vecteur de longueur n décrivant le temps nécessaire pour pousser la tondeuse sur chaque allée
- v : vecteur de longueur n indiquant la quantité d'herbe sur chaque allée
- Cette procédure doit retourner un entier unique correspondant au temps minimum pour tondre la pelouse.
- Cette procédure est appelée exactement une fois pour chaque cas de test.

Limites

- $1 \leq n \leq 200\,000$
- $1 \leq a[i] \leq 10^9$ (pour chaque i tel que $0 \leq i < n$)
- $1 \leq v[i] \leq 10^9$ (pour chaque i tel que $0 \leq i < n$)
- $1 \leq b \leq 10^9$
- $1 \leq c \leq 10^9$
- Il est garanti que le résultat correct sera au plus 10^{18}

Sous-tâches

1. (9 points) Toutes les valeurs données ($n, b, c, a[i]$ et $v[i]$) sont au plus 200
2. (16 points) $n, c \leq 5000$ et $v[i] \leq 5000$ pour tout $0 \leq i < n$
3. (36 points) $c \leq 200\,000$
4. (17 points) $a[0] = a[1] = \dots = a[n-1]$
5. (22 points) Aucune contrainte supplémentaire

Exemples

Exemple 1

Considérez l'appel suivant :

```
mow(3, 5, 2, {2, 10, 3}, {2, 4, 6})
```

Dans cet exemple, il y a 3 allées, le réservoir a une capacité de 5 et il faut 2 secondes pour le vider.

Pour cet exemple, Vlad tondra la première allée en 2 secondes. La quantité d'herbe dans la tondeuse sera de 2. Ensuite, il videra la tondeuse en 2 secondes. Sur la première allée, il passe donc 4 secondes.

Il passera ensuite sur la deuxième allée. Il tondra 4 unités d'herbe. Il choisira de ne pas vider le réservoir après avoir terminé cette allée. Le temps passé sur la deuxième allée est de 10 secondes.

Pour la troisième allée, il commence à tondre. Après une unité d'herbe, sa tondeuse est pleine ; il doit donc aller jusqu'à la fin de l'allée, vider la tondeuse, puis recommencer à tondre la troisième allée. N'oubliez pas qu'une fois toute la pelouse tondue, la tondeuse doit être vidée. Le temps passé sur la troisième allée est de $3 + 2 + 3 + 2 = 10$ secondes.

Au total, il passe $4 + 10 + 10 = 24$ secondes. On peut prouver qu'il s'agit de la stratégie optimale que Vlad utilise pour tondre la pelouse.

Exemple 2

Considérez l'appel suivant :

```
mow(4, 10, 4, {1, 2, 1, 4}, {3, 2, 6, 7})
```

Dans cet exemple, il y a 4 allées, le réservoir a une capacité de 10 et il faut 4 secondes pour le vider.

La stratégie optimale consiste simplement à pousser la tondeuse sur les 3 premières allées ; ensuite, le réservoir sera plein et le vecteur des quantités d'herbe sera $[0, 0, 1, 7]$. Après cela, le réservoir doit être vidé, puis les 2 dernières allées seront tondues et le réservoir sera vidé à nouveau à la fin.

Le nombre total de secondes sera $a[0] + a[1] + a[2] + b + a[2] + a[3] + b = 17$.

Grader d'exemple

Le grader d'exemple lit l'entrée au format suivant :

- ligne 1 : $n \ c \ b$
- ligne 2 : $a[0] \ a[1] \ \dots \ a[n-1]$
- ligne 3 : $v[0] \ v[1] \ \dots \ v[n-1]$

et affiche le résultat de l'appel à `mow` avec les paramètres correspondants.