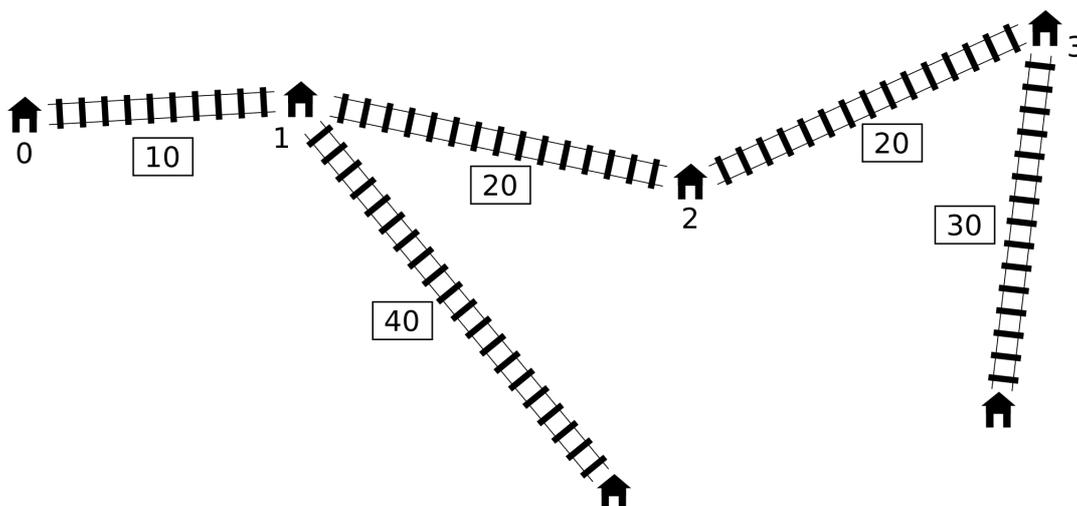


Raccourci

Pavel possède un jeu de chemins de fer. Ce jeu est très simple. Il y a une seule ligne principale qui consiste en n stations. Ces stations sont numérotées de 0 à $n - 1$ le long de la ligne. La distance entre les stations i et $i + 1$ est de l_i centimètres ($0 \leq i < n - 1$).

Hormis la ligne principale, il peut y avoir des lignes secondaires. Chaque ligne secondaire est une ligne de chemin de fer qui relie une station de la ligne principale et une nouvelle station qui n'est pas sur la ligne principale. (Ces nouvelles stations ne sont pas numérotées). Au plus une ligne secondaire peut commencer depuis chacune des stations de la ligne principale. La longueur d'une ligne secondaire commençant à la station i est d_i centimètres. On utilise $d_i = 0$ pour dénoter qu'il n'y a aucune ligne secondaire commençant à la station i .



Pavel a désormais prévu de construire un raccourci : une ligne express reliant deux stations différentes (potentiellement voisines) de **la ligne principale**. La ligne express aura une taille d'exactement c centimètres, indépendamment des deux stations qu'elle relie.

Chaque segment du chemin de fer, incluant la nouvelle ligne express, peut être utilisé dans les deux directions. La *distance* entre deux stations est définie comme la longueur du plus court chemin allant d'une station à l'autre via le chemin de fer. Le *diamètre* de l'ensemble du réseau ferré est défini comme la distance maximale parmi toutes les paires de stations. Autrement dit, il s'agit du plus petit nombre t , tel que la distance entre deux stations quelconques est au plus t .

Pavel veut construire une ligne express de telle sorte que le diamètre du réseau résultant soit minimisé.

Détails de l'implémentation

Vous devez implémenter la fonction

`int64 find_shortcut(int n, int[] l, int[] d, int c)`

- `n`: le nombre de stations sur la ligne principale,
- `l`: les distances entre les stations de la ligne principale (tableau de taille $n - 1$),
- `d`: les longueurs des lignes secondaires (tableau de taille n),
- `c`: longueur de la nouvelle ligne express.
- La fonction doit retourner le plus petit diamètre du réseau ferré qu'il est possible d'obtenir après ajout de la ligne express.

Veillez utiliser les fichiers templates fournis pour obtenir des détails de l'implémentation dans votre langage de programmation.

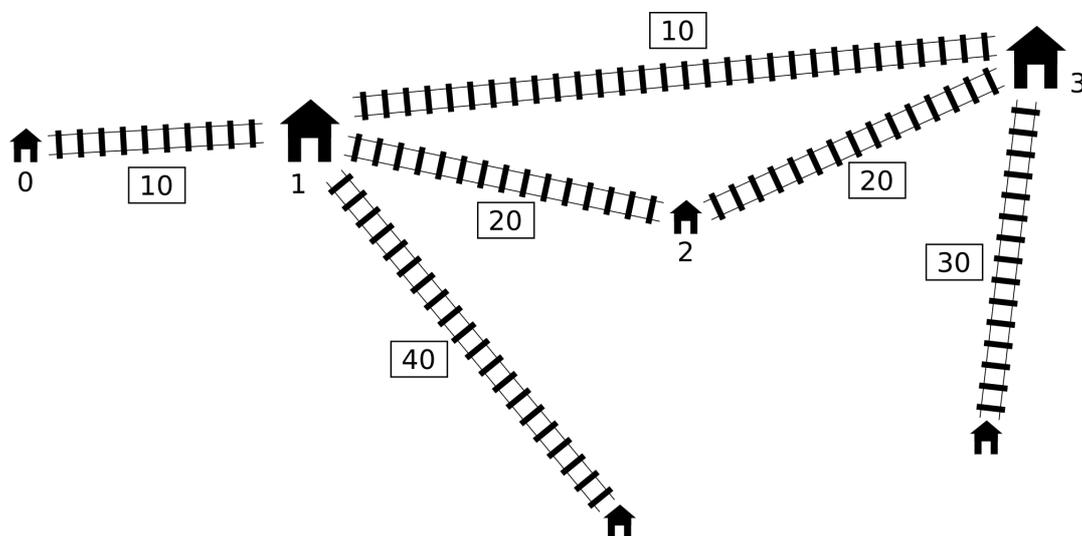
Exemples

Exemple 1

Pour le réseau ferré décrit ci-dessus, l'évaluateur ferait l'appel de fonction suivant :

`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`

La solution optimale est de construire une ligne express entre les stations 1 et 3, comme montré ci-dessous.



Le diamètre du nouveau réseau ferré est **80** centimètres, donc la fonction doit retourner **80**.

Exemple 2

L'évaluateur effectue l'appel de fonction suivant :

```
find_shortcut(9, [10, 10, 10, 10, 10, 10, 10, 10],
```

```
[20, 0, 30, 0, 0, 40, 0, 40, 0], 30)
```

La solution optimale est de relier les stations 2 et 7, auquel cas le diamètre est 110 .

Exemple 3

L'évaluateur effectue l'appel de fonction suivant :

```
find_shortcut(4, [2, 2, 2],  
[1, 10, 10, 1], 1)
```

La solution optimale est de connecter les stations 1 et 2, réduisant le diamètre à 21 .

Exemple 4

L'évaluateur effectue l'appel de fonction suivant :

```
find_shortcut(3, [1, 1],  
[1, 1, 1], 3)
```

Relier n'importe quelle paire de stations avec cette ligne express de taille 3 n'améliore pas le diamètre initial du réseau ferré, qui est de 4 .

Sous-tâches

Dans toutes les sous-tâches, $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$ et $1 \leq c \leq 10^9$.

1. (9 points) $2 \leq n \leq 10$,
2. (14 points) $2 \leq n \leq 100$,
3. (8 points) $2 \leq n \leq 250$,
4. (7 points) $2 \leq n \leq 500$,
5. (33 points) $2 \leq n \leq 3000$,
6. (22 points) $2 \leq n \leq 100\,000$,
7. (4 points) $2 \leq n \leq 300\,000$,
8. (3 points) $2 \leq n \leq 1\,000\,000$.

Evaluateur fourni (grader)

L'évaluateur fourni lit l'entrée selon le format suivant :

- ligne 1: les entiers n et c ,
- ligne 2: les entiers l_0, l_1, \dots, l_{n-2} ,
- ligne 3: les entiers d_0, d_1, \dots, d_{n-1} .