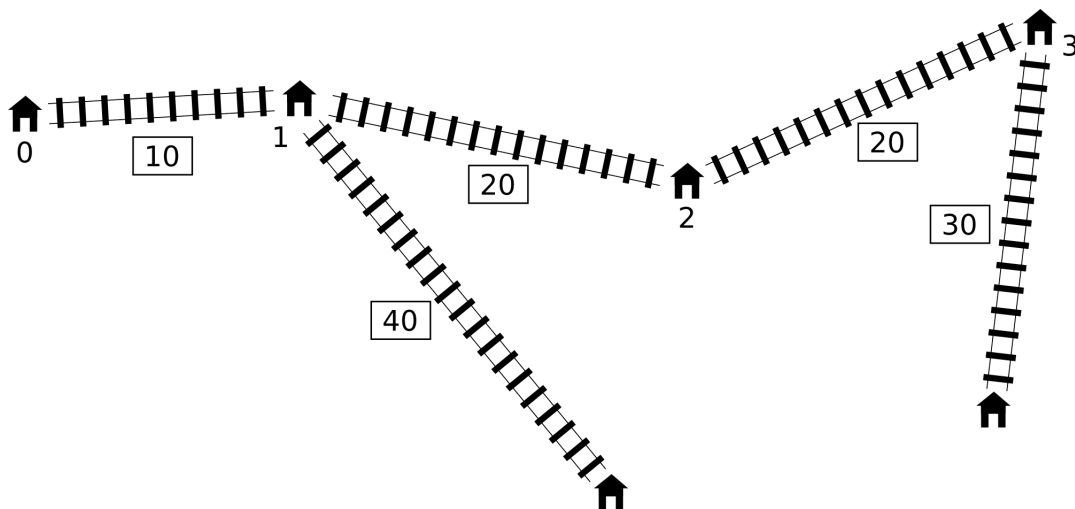


Atajo

Pavel tiene un tren de juguete. Es bastante sencillo. Hay una vía principal que consiste de (n) estaciones. Estas estaciones están numeradas desde el (0) hasta el $(n-1)$ en orden, a lo largo de la vía. Las estaciones (0) y $(n-1)$ están en los extremos de la vía principal. La distancia entre las estaciones (i) e $(i+1)$ es (l_i) centímetros ($(0 \leq i < n-1)$).

Además de la vía principal pueden existir vías secundarias. Cada vía secundaria es una vía de tren entre una estación en la vía principal y una nueva estación que no se encuentra en la vía principal. (Estas nuevas estaciones no tienen número.) Solamente puede salir a lo más una vía secundaria por cada estación en la vía principal. La longitud de la vía secundaria que comienza en la estación (i) es de (d_i) centímetros. Denotamos con $(d_i = 0)$ que no hay vía secundaria empezando en la estación (i) .



Pavel está planeando construir un atajo: una vía exprés entre dos diferentes (aunque tal vez contiguas) estaciones de la **vía principal**. La vía exprés tendría longitud de exactamente (c) centímetros, sin importar qué dos estaciones conecte.

Cada segmento de las vías del tren, incluyendo la nueva vía exprés, puede usarse en ambos sentidos. La *distancia* entre dos estaciones es la longitud mínima entre todos los caminos que conectan estas dos estaciones. El *diámetro* de la red de vías de tren es la distancia máxima entre todas las parejas de estaciones. En otras palabras, es el número (t) más pequeño tal que la distancia entre cualesquiera dos estaciones no es mayor que (t) .

Pavel quiere construir la vía exprés de tal forma que minimice el diámetro de la red de

vías de tren resultante.

Detalles de implementación

Deberás implementar la función

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

- **n**: el número de estaciones en la vía principal,
- **l**: las distancias entre las estaciones en la vía principal (arreglo de longitud $(n-1)$),
- **d**: las longitudes de las vías secundarias (arreglo de longitud (n)),
- **c**: la longitud de la nueva vía exprés.
- la función debe regresar el diámetro mínimo posible de la red de vías de tren después de añadir la vía exprés.

Por favor utiliza las plantillas de tu lenguaje de programación para ver los detalles de implementación.

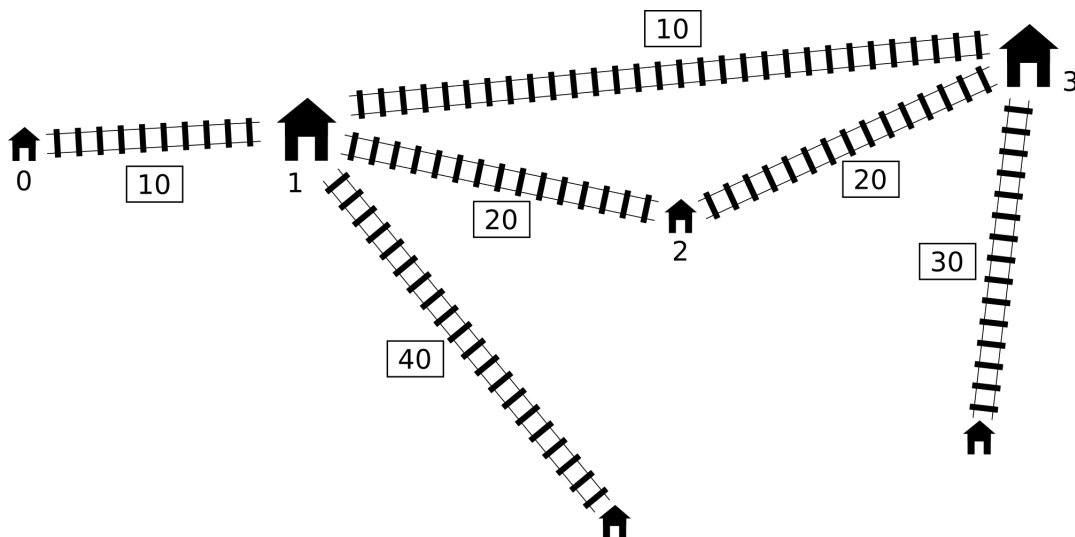
Ejemplos

Ejemplo 1

Para la red de vías de tren mostrada arriba, el evaluador haría la siguiente llamada a tu función:

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

La solución óptima es construir la vía exprés entre las estaciones 1 y 3, como se muestra debajo.



El diámetro de la nueva red de vías de tren es (80) centímetros, así que la función debería regresar (80) .

Ejemplo 2

El evaluador hace la siguiente llamada a tu función:

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

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

La solución óptima es conectar las estaciones (2) y (7) , en cuyo caso el diámetro es (110) .

Ejemplo 3

El evaluador hace la siguiente llamada a tu función:

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

La solución óptima es conectar las estaciones (1) y (2) , lo que reduce el diámetro a (21) .

Ejemplo 4

El evaluador hace la siguiente llamada a tu función:

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

Conectar cualesquiera dos estaciones con la nueva vía exprés (de longitud (3)) no mejora el diámetro inicial de la red de vías de tren, que es (4) .

Subtareas

En todas las Subtareas, $(2 \leq n \leq 1,000,000)$, $(1 \leq l_i \leq 10^9)$, $(0 \leq d_i \leq 10^9)$, $(1 \leq c \leq 10^9)$.

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

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

- línea 1: los enteros (n) y (c) ,
- línea 2: los enteros $(l_0, l_1, \dots, l_{n-2})$,
- línea 3: los enteros $(d_0, d_1, \dots, d_{n-1})$.