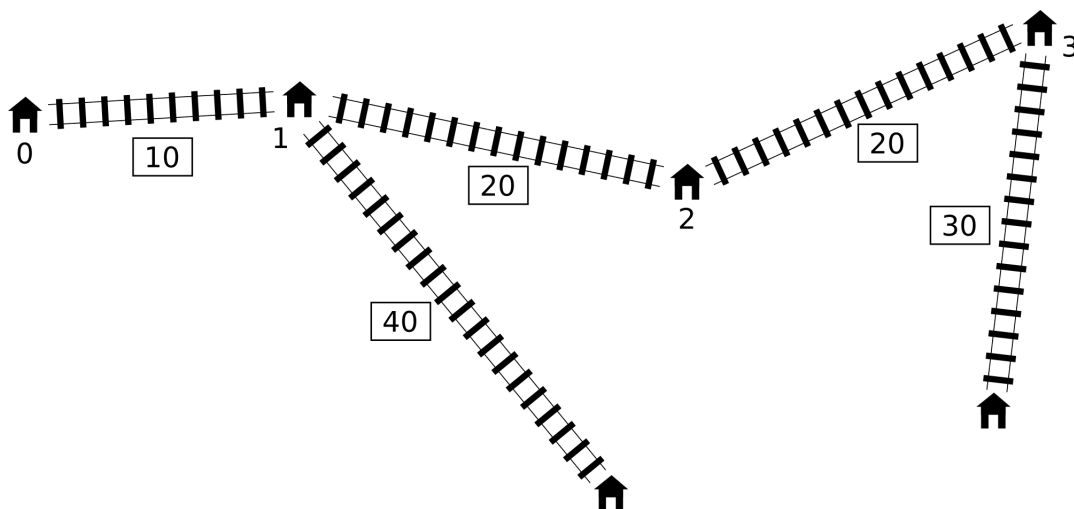


## Atajo

Pavel tiene un tren de juguete. Es muy simple. Hay una sola vía principal que consiste en  $n$  estaciones. Estas estaciones están numeradas desde  $0$  a  $n - 1$  en orden a lo largo de la vía. La distancia entre las estaciones  $i$  y  $i + 1$  es  $l_i$  centímetros ( $0 \leq i < n - 1$ ).

Aparte de la vía principal puede haber algunas vías secundarias. Cada vía secundaria es una vía de ferrocarril entre una estación en la vía principal y una nueva estación que no está en la vía principal. (Estas nuevas estaciones no están numeradas.) A lo sumo una vía secundaria puede comenzar en cada estación de la vía principal. La longitud de la vía secundaria a partir de la estación  $i$  es  $d_i$  centímetros. Utilizamos  $d_i = 0$  para indicar que no hay una vía secundaria partiendo de la estación  $i$ .



Pavel está planeando la construcción de un atajo: una vía rápida entre dos estaciones diferentes (posiblemente vecinas) de la vía principal. La vía rápida tendrá una longitud de exactamente  $c$  centímetros, sin importar cuales dos estaciones conecte.

Cada segmento de la vía férrea, incluyendo la nueva vía rápida, se puede utilizar en ambas direcciones. La *distancia* entre dos estaciones es la longitud más pequeña de una ruta que va de una estación a la otra a lo largo de las vías férreas. El *diámetro* de toda la red ferroviaria es la distancia máxima entre todos los pares de estaciones. En otras palabras, este es el menor número  $t$ , tal que la distancia entre cada par de estaciones es como máximo  $t$ .

Pavel quiere construir la vía rápida de una manera tal que se minimize el diámetro de

la red resultante.

## Detalles de la implementación

Debes implementar la función

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

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

Por favor utiliza los archivos de ejemplo previstos para los detalles de implementación en el lenguaje de programación que uses.

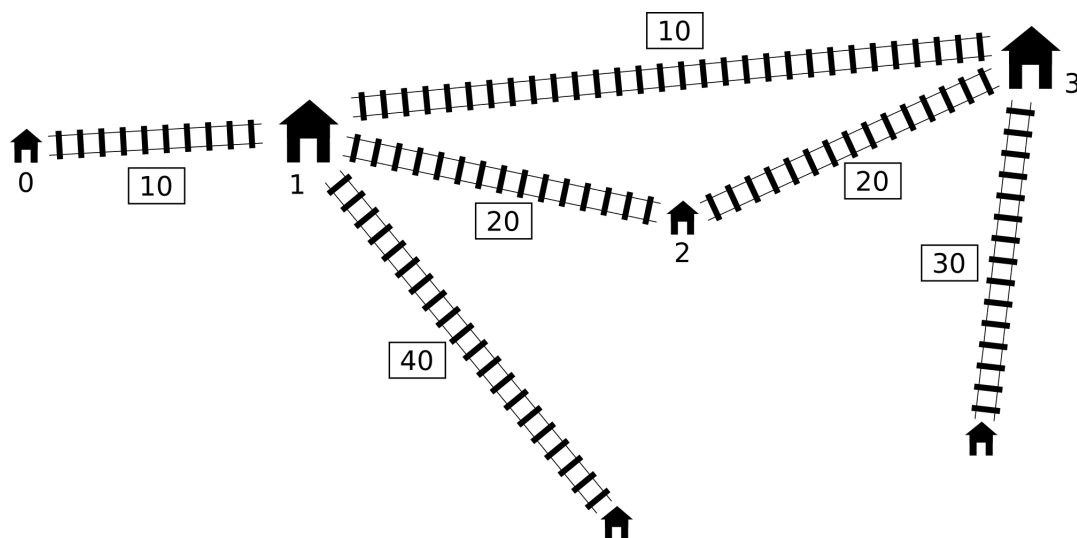
## Ejemplos

### Ejemplo 1

Para la red ferroviaria que se muestra arriba, el evaluador haría la siguiente llamada de función:

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

La solución óptima es la construcción de la vía rápida entre las estaciones 1 y 3, como se muestra a continuación.



El diámetro de la nueva red ferroviaria es **80** centímetros, por lo que la función debe devolver **80**.

### Ejemplo 2

El evaluador hace la siguiente llamada a la 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 la función:

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

La solución óptima es conectar las estaciones **1** y **2**, reduciendo el diametro a **21**.

### Ejemplo 4

El evaluador hace la siguiente llamada a la función:

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

Conectando dos estaciones cualquiera con la vía rápida de longitud **3** no mejora el diámetro inicial de la red ferroviaria, 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)  $2 \leq n \leq 1\,000\,000$ .

### El evaluador de ejemplo

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

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