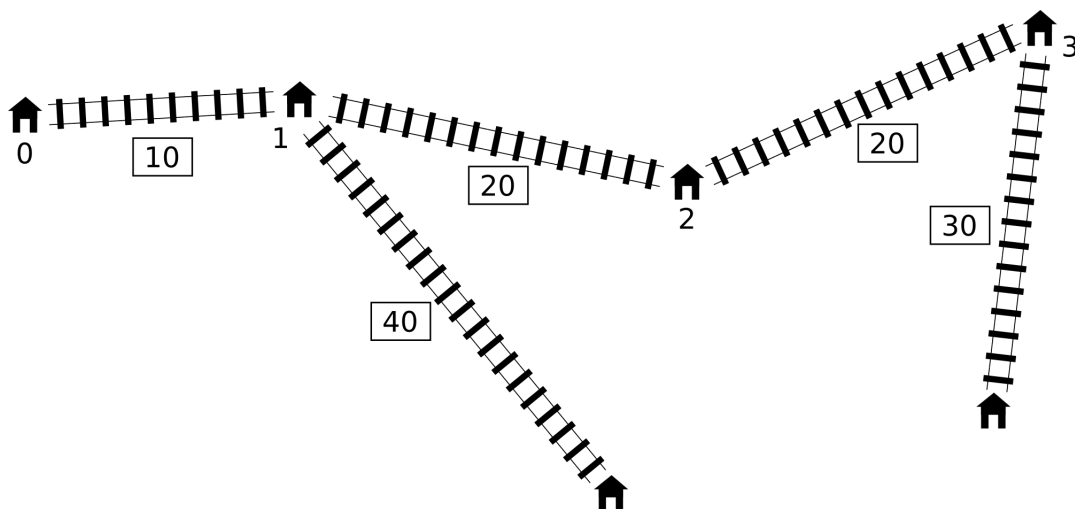


Кратенка

Дмитри си има играчка-железница. Истата е многу едноставна. Постои една главна железничка линија која има n станици. Станиците се нумерирани со целите броеви од 0 до $n - 1$, по редослед - од првата кон последната станица долж линијата. Растојанието помеѓу станиците i и $i + 1$ е l_i сантиметри ($0 \leq i < n - 1$).

Покрај главната железничка линија, може да постојат и некои споредни линии. Секоја споредна линија е железничка линија помеѓу станица од главната линија и уште една, „нова“ станица, која што не се наоѓа на главната линија (овие „нови“ станици не се нумерирани). Од секоја станица на главната линија може да започнува најмногу една споредна линија. Должината на споредната линија која започнува од станицата i е d_i сантиметри. Со $d_i = 0$ ќе означуваме дека не постои споредна линија која започнува од станицата i .



Дмитри сега планира да изгради една кратенка (анг. shortcut): експресна линија помеѓу две различни (може да бидат и соседни!) станици од **главната линија**. Експресната линија ќе биде долга точно c сантиметри, независно од тоа кои две станици ќе ги поврзува.

Секој сегмент од железницата, вклучувајќи ја и новата експресна линија, може да се користи во двете насоки. *Растојание* помеѓу две станици е најмалата должина на железничка рута која што оди од едната до другата станица. *Дијаметар* на целата железничка мрежа е максималното растојание помеѓу

сите парови од станици. Со други зборови, дијаметарот на мрежата е најмалиот број t , таков што растојанието помеѓу кој било пар од станици е најмногу t .

Дмитри сака да ја изгради експресната линија на таков начин што ќе го минимизира дијаметарот на новодобиената железничка мрежа.

Детали за имплементација

Треба да ја имплементирате функцијата

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

- n : број на станици на главната железничка линија,
- l : растојанија помеѓу станиците на главната железничка линија (низа со должина $n - 1$),
- d : должини на споредните линии (низа со должина n),
- c : должина на новата експресна линија.
- функцијата треба да го враќа најмалиот можен дијаметар на железничката мрежа после додавањето на експресната линија.

Ве молиме да ги користите дадените темплејт датотеки за детали околу имплементацијата во вашиот програмски јазик.

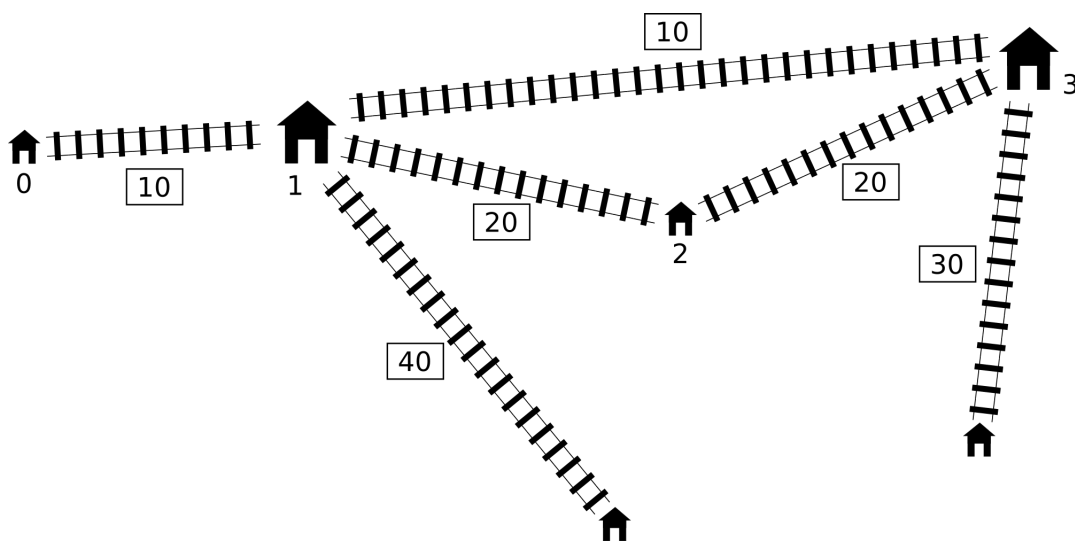
Примери

Пример 1

За железничката мрежа прикажана погоре, оценувачот би го извршил следниот функциски повик:

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

Оптималното решение е експресната линија да се изгради помеѓу станиците 1 и 3, како што е прикажано подолу.



Дијаметарот на новодобиената железничка мрежа е **80** сантиметри, па функцијата треба да врати **80**.

Пример 2

Оценувачот го извршува следниот функциски повик:

```
find_shortcut(9, [10, 10, 10, 10, 10, 10, 10, 10],  
               [20, 0, 30, 0, 0, 40, 0, 40, 0], 30)
```

Оптималното решение е да се поврзат станиците **2** и **7**, и во овој случај дијаметарот е **110**.

Пример 3

Оценувачот го извршува следниот функциски повик:

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

Оптималното решение е да се поврзат станиците **1** и **2**, со што дијаметарот се редуцира на **21**.

Пример 4

Оценувачот го извршува следниот функциски повик:

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

Поврзувањето на кои било две станици со експресна линија со должина **3** нема да го подобри иницијалниот дијаметар на железничката мрежа кој што изнесува **4**.

Подзадачи

Во сите подзадачи $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

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

Оценувач

Оценувачот чита влезни податоци во следниот формат:

- линија 1: цели броеви n и c ,
- линија 2: цели броеви l_0, l_1, \dots, l_{n-2} ,
- линија 3: цели броеви d_0, d_1, \dots, d_{n-1} .