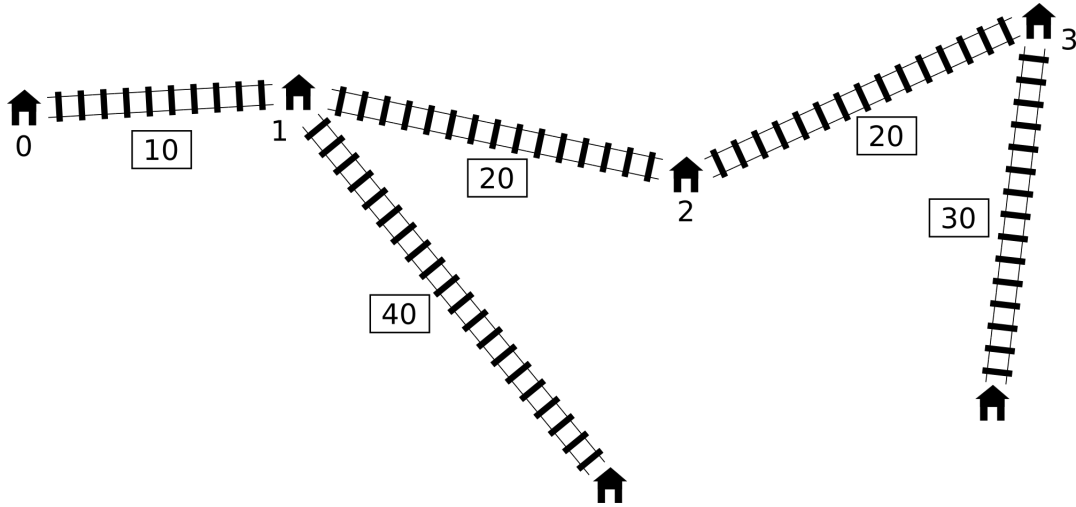


Qısayol

Pavelin oyuncaq dəmiryolu var. O, çox sadədir. 0 -dan $n - 1$ -dək ardıcıl nömrələnmiş n sayda stansiyadan ibarət bir baş xətt var. 0 və $n - 1$ stansiyaları baş xəttin iki başında yerləşir. i və $i + 1$ stansiyaları arasındakı məsafə l_i santimetrdir ($0 \leq i < n - 1$).

Baş xətdən kənarında ikinci dərəcəli xətlər ola bilər. Hər bir ikinci dərəcəli xətt baş xətdə olan stansiya ilə baş xətt üzərində olmayan yeni stansiya arasında dəmiryolu xəttidir. (Bu yeni stansiyalar nömrələnməyib.) Baş xətdə yerləşən hər bir stansiyadan ən çoxu bir ikinci dərəcəli xətt başlaya bilər. i stansiyasından başlanan ikinci dərəcəli xəttin uzunluğu d_i santimetrdir. i stansiyasından başlanan ikinci dərəcəli xəttin olmadığını göstərmək üçün $d_i = 0$ yazısından istifadə olunur.



Pavel indi bir qısayol - **baş xəttin** iki fərqli stansiyası (qonşu da ola bilər) arasında ekspres xətt qurmağı planlaşdırır. Hansı iki stansiyanı birləşdirməsindən asılı olmayaraq, ekspres xəttin uzunluğu dəqiq c santimetr olmalıdır.

Yeni ekspres xətt də daxil olmaqla, dəmiryolunun hər bir segmenti hər iki istiqamətə istifadə oluna bilər. İki stansiya arasında *məsafə* dedikdə, bir stansiyadan o birinə gedən ən qısa marşrutun uzunluğu nəzərdə tutulur. Bütün dəmiryol şəbəkəsinin *diametri* dedikdə, bütün stansiya cütlükləri arasında maksimal məsafə nəzərdə tutulur. Başqa sözlə, bu, elə ən kiçik t ədədidir ki, istənilən iki stansiya arasındakı məsafə ən çoxu t ola bilər.

Pavel ekspres xətti elə yolla qurmaq istəyir ki, yekun şəbəkənin diametri minimum olsun.

Implementation details

You should implement the function

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

- n : number of stations on the main line,
- l : distances between stations on the main line (array of length $n - 1$),
- d : lengths of secondary lines (array of length n),
- c : length of the new express line.
- the function should return the smallest possible diameter of the railway network after adding the express line.

Please use the provided template files for details of implementation in your programming language.

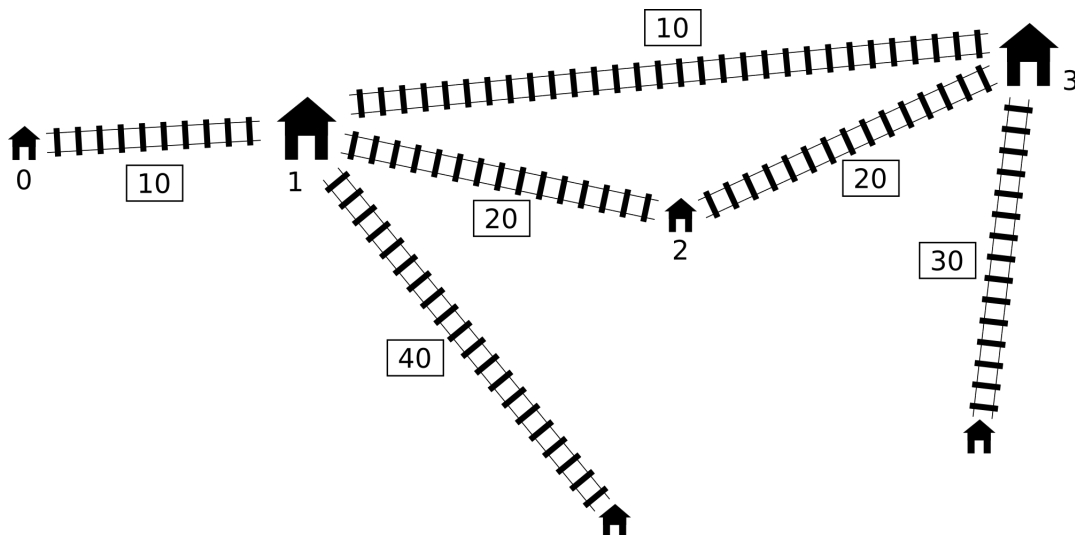
Examples

Example 1

For the railway network shown above, the grader would make the following function call:

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

The optimal solution is to build the express line between stations 1 and 3, as shown below.



The diameter of the new railway network is **80** centimeters, so the function should return **80**.

Example 2

The grader makes the following function call:

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

The optimal solution is to connect stations **1** and **6**, in which case the diameter is **110**.

Example 3

The grader makes the following function call:

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

The optimal solution is to connect stations **2** and **3**, reducing the diameter to **21**.

Example 4

The grader makes the following function call:

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

Connecting any two stations with the express line of length **2** does not improve the initial diameter of the railway network which is **4**.

Subtasks

In all Subtasks $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

1. (9 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) $2 \leq n \leq 1\,000\,000$.

Sample grader

The sample grader reads the input in the following format:

- line 1: integers n and c ,
- line 2: integers l_0, l_1, \dots, l_{n-2} ,
- line 3: integers d_0, d_1, \dots, d_{n-1} .