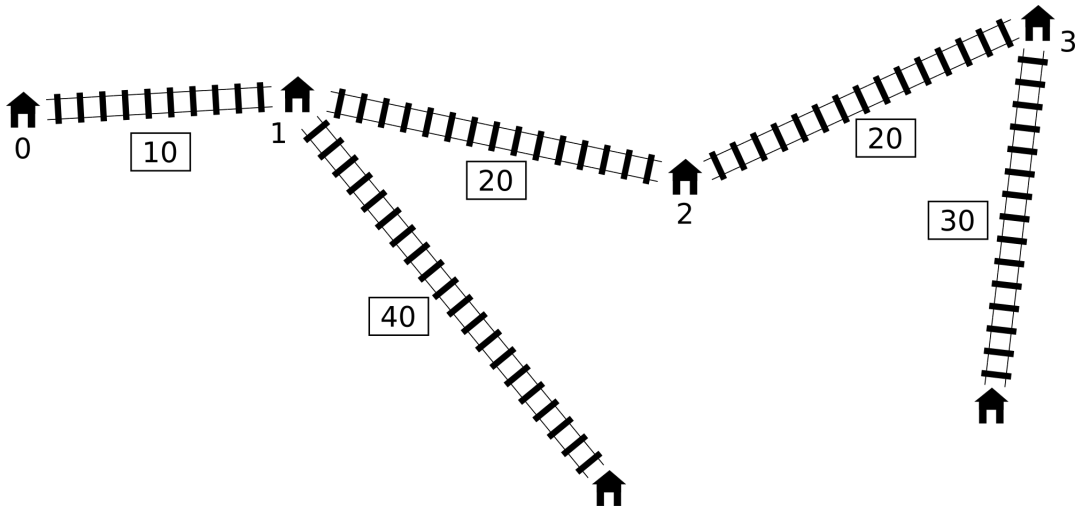
Jalan Pintas

Pavel memiliki sebuah rel mainan. Relnya sangat sederhana. Terdapat sebuah lintasan utama berisi n stasiun yang secara berurutan dinomori dari 0 sampai $n-1$. Stasiun 0 dan $n-1$ berada pada kedua ujung dari lintasan utama. Jarak antara stasiun i dan $i+1$ adalah l_i centimeter ($0 \leq i < n-1$).

Selain lintasan utama mungkin juga terdapat beberapa lintasan sekunder. Setiap lintasan sekunder adalah sebuah lintasan rel di antara sebuah stasiun pada lintasan utama dan sebuah stasiun baru yang tidak terletak pada lintasan utama. Panjang dari lintasan sekunder yang dimulai pada stasiun i adalah d_i centimeter. Kita menggunakan $d_i = 0$ untuk menandakan bahwa tidak terdapat jalur sekunder yang dimulai dari stasiun i.



Pavel sekarang merencanakan untuk membangun satu jalan pintas: sebuah lintasan ekspres di antara dua stasiun berbeda **pada lintasan utama** (mungkin saja bersebelahan). Lintasan ekspres akan memiliki panjang tepat c centimeter, tanpa peduli dua stasiun apa yang akan dihubungkannya.

Tiap segmen dari rel, termasuk lintasan ekspres yang baru, dapat digunakan pada kedua arah. *Jarak* antara dua stasiun adalah panjang rute terpendek yang berjalan dari satu stasiun ke stasiun lain sepanjang rel. *Diameter* dari seluruh jaringan rel adalah jarak maksimum dari semua pasang stasiun. Dengan kata lain, jika diameter tersebut dinyatakan dengan bilangan terkecil t, maka jarak antara setiap pasangan stasiun yang ada paling besar adalah t.

Pavel ingin membangun lintasan ekspres sedemikian sehingga diameter dari hasil jaringan adalah seminimal mungkin.

Rincian Implementasi

Anda harus mengimplementasikan fungsi
`int64 find_shortcut(int n, int[] l, int[] d, int c)`

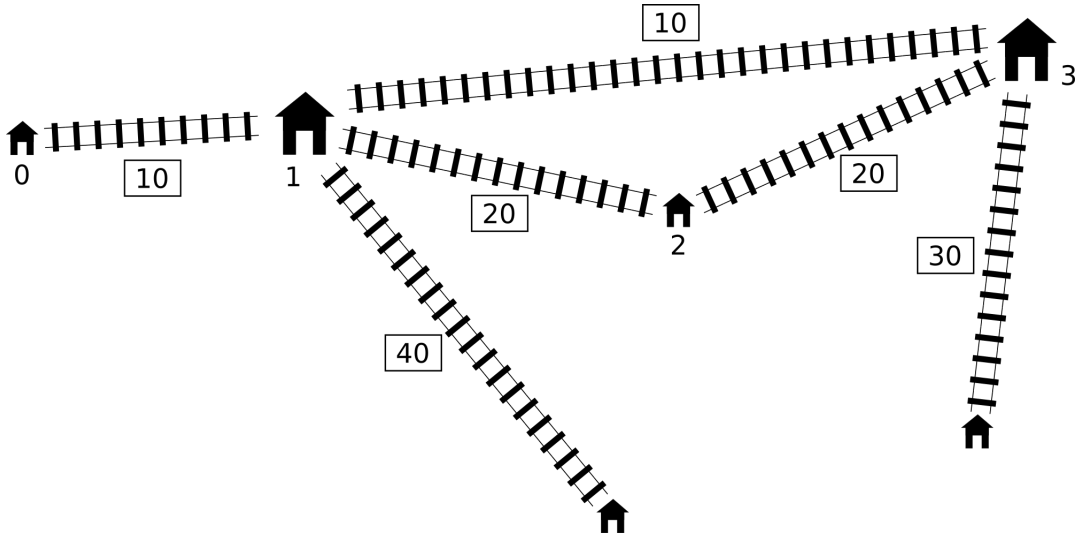
- `n`: banyaknya stasiun pada lintasan utama,
- `l`: jarak antara stasiun pada lintasan utama (array dengan panjang $n-1$),
- `d`: panjang lintasan sekunder (array dengan panjang n),
- `c`: panjang lintasan ekspres baru.
- fungsi ini harus mengembalikan diameter terkecil yang mungkin dari jaringan rel setelah menambahkan lintasan ekspres.

Gunakan file template yang sudah disediakan untuk implementasi rinci dari bahasa pemrograman yang Anda pakai.

Contoh

Contoh 1

Untuk jaringan rel ditunjukan di atas, grader akan melakukan pemanggilan fungsi berikut:
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`
Solusi optimal adalah membangun lintasan ekspres antara stasiun 1 dan 3, seperti ditunjukan di bawah.



Diameter dari jaringan rel yang baru adalah 80 centimeter, sehingga fungsi harus mengembalikan 80.

Contoh 2

Grader melakukan pemanggilan fungsi berikut:

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

Solusi optimal adalah menghubungkan stasiun 1 dan 6, yang mana diameternya adalah 110.

Contoh 3

Grader melakukan pemanggilan fungsi berikut:

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

Solusi optimal adalah menghubungkan stasiun 2 dan 3, mengurangi diameter menjadi 21.

Contoh 4

Grader melakukan pemanggilan fungsi berikut:

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

Menghubungkan dua stasiun manapun dengan lintasan ekspres dengan panjang 2 tidak akan memperbaiki diameter awal dari jaringan rel yaitu 4.

Subtasks

Pada semua subtask $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

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

Grader

Grader membaca masukan dengan format berikut:

- baris 1: bilangan integer n dan c,
- baris 2: bilangan integer $l_0, l_1, \ldots, l_{n-2}$,
- baris 3: bilangan integer $d_0, d_1, \ldots, d_{n-1}$.