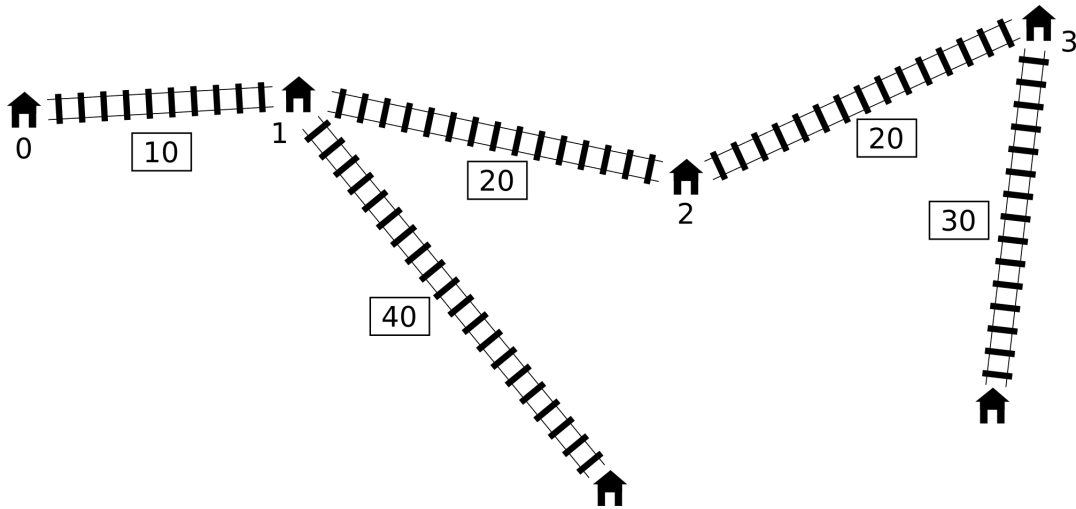


Kısayol

Kısayol

Pavelin bir oyuncak demiryolu vardır. Bu çok bir basit şey. Sadece bir ana çizgi vardır ve bu çizgide 0 -dan $n - 1$ -e kadar numarlandırılan n istasyon vardır. 0 ve $n - 1$ istasyonları çizginin iki uçlarındadır. i ve $i + 1$ -inci istasyonlarının arasındaki uzaklık l_i santimetredir ($0 \leq i < n - 1$).

Baş çizginin dışında bazı ikinci dereceli çizgiler olabilir. Her ikinci dereceli çizgi baş çizgideki bir istasyon ve yeni bir baş çizgide olmayan istasyonun arasındadır. (Bu yeni istasyonlar numarasızdır.) Baş çizgideki her istasyondan sadece bir ikinci dereceli çizgi çıkabilir. i -inci istasyonda başlayan ikinci dereceli çizginin uzaklığı d_i santimetredir. Biz eğer $d_i = 0$ olsa i -inci istasyondan ikinci dereceli çizgi çıkmıyor diye düşünüyoruz.



Şimdi Pavel bir kısayol kurmak istiyor: bu bir ekspres çizgi **baş çizgideki** iki istasyon arasında olacak (bu istasyonlar baş çizgide komşu olabilir). Ekspres çizginin uzaklığı c santimetredir, .

Demiryolunun her daire kesmesi iki tarafta kullanılabilir. İki istasyon arasındaki *uzaklık* demiryolunun parçaları boyunca olan en kısa yolun toplanmış uzaklığıdır. Demiryolunun *diametri* her iki istasyon arasındaki uzaklıkların en büyüğüdür.

Pavel demiryolu onun diametri en kısa olmak için kuruyor. Sizin onun en kısa kaç olabilirini söylemeniz lazım.

Uygulama detayları

Siz bu fonksiyonu yazmanız lazım

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

- o n : baş çizgideki istasyonlar sayısı,
- o l : baş çizgideki istasyonlar arasındaki uzaklıklar (uzunluğu $n - 1$ olan sıra),
- o d : ikinci dereceli çizgilerin uzaklıkları ((uzunluğu n olan sıra),
- o c : kısayolun uzaklığı.
- o bu fonksiyon demiryolun en kısa olabilen dimatrin getirmek lazım.

Programlama dilinde uygulama detayları için sağlanan şablon dosyalarını kullanınız.

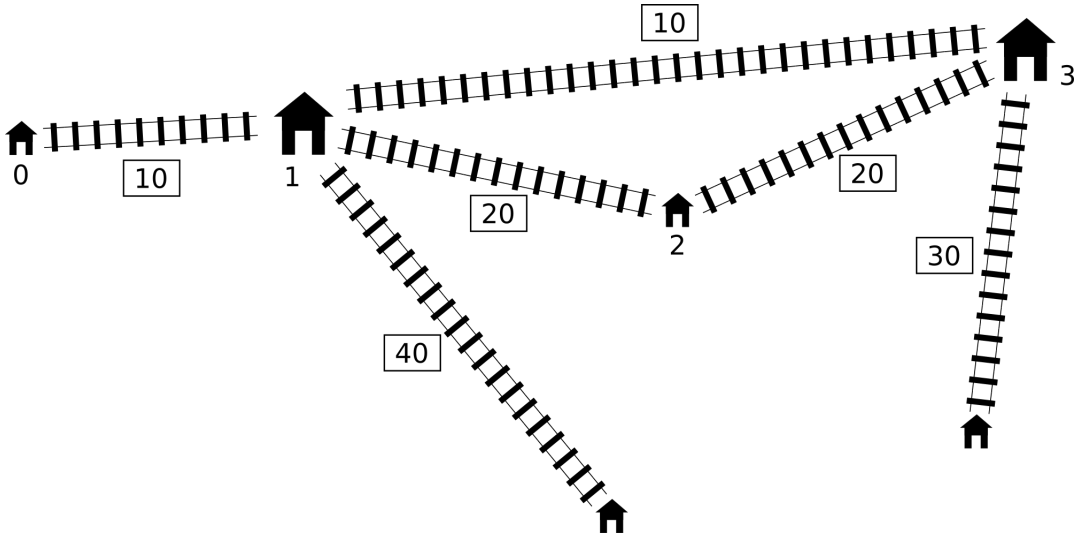
Examples

Example 1

Yukarıdaki demiryol için program bu fonksiyonu çağırarak:

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

En optimal çözüm olanı o aşağıdaki resimdeki gibi birinci ve üçüncü istasyonlar arasında kısayol kurmak.



Yeni demiryolunun diametresi 80 santimetredir, onun için fonksiyon 80 getirmek lazım.

Example 2

Yukarıdaki demiryol için program bu fonksiyonu çağırarak:

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

En optimal çözüm olanı o aşağıdaki resimdeki gibi birinci ve altıncı istasyonlar arasında kısayol kurmak.

Yeni demiryolunun diiametresi 110 santimetredir, onun için fonksiyon 110 getirmek lazım.

Example 3

Yukarıdaki demiryol için program bu fonksiyonu çağırarak:

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

En optimal çözüm olanı o aşğıdaki resimdeki gibi ikinci ve üçüncü istasyonlar arasında kısayol kurmak.

Yeni demiryolunun diiametresi 21 santimetredir, onun için fonksiyon 21 getirmek lazım.

Example 4

Yukarıdaki demiryol için program bu fonksiyonu çağırarak:

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

Burasa nasıl yapsak üzere diiametere 4 olarak kalır.

Subtasks

Her görev bölümünde $2 \leq n \leq 1000000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

1. (9 puan) $2 \leq n \leq 10$,
2. (14 puan) $2 \leq n \leq 100$,
3. (8 puan) $2 \leq n \leq 250$,
4. (7 puan) $2 \leq n \leq 500$,
5. (33 puan) $2 \leq n \leq 3000$,
6. (22 puan) $2 \leq n \leq 100000$,
7. (4 puan) $2 \leq n \leq 300000$.
8. (3 puan) $2 \leq n \leq 1000000$.

Sample grader

Basit grader programı girişı bu formatta okur:

- o çizgi 1: n ve c tam sayıları,
- o çizgi 2: l_0, l_1, \dots, l_{n-2} tam sayıları,
- o çizgi 3: d_0, d_1, \dots, d_{n-1} tam sayıları.