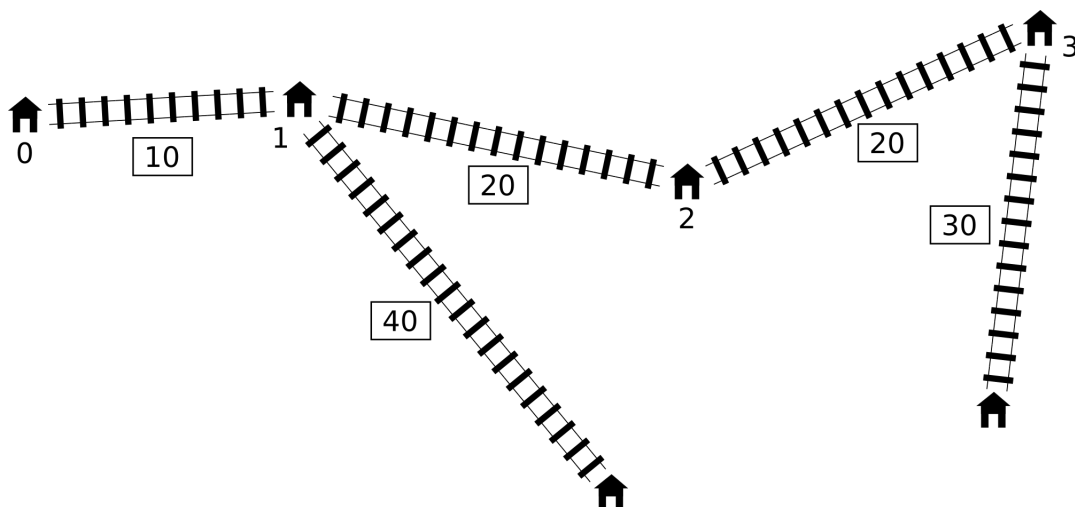


Shortcut

Pavel har en modelljernbane. Den er veldig enkel. Den består av en hovedlinje med n stasjoner. Disse er nummerert i rekkefølge fra 0 til $n - 1$ og ligger i rekke langs hovedlinja. Avstanden mellom stasjonene i og $i + 1$ er l_i centimeter ($0 \leq i \leq n - 1$).

I tillegg til hovedlinjen så er det noen sidelinjer. Hver sidelinje er en toglinje som forbinder en stasjon på hovedlinja med en ny stasjon som ikke befinner seg på hovedlinja. (Disse nye stasjonene er ikke nummererte.) Det kan ikke være flere enn én sidelinja som begynner i samme stasjon på hovedlinja. Lengden av sidelinjen som begynner på stasjon i er d_i centimeter. Vi bruker $d_i = 0$ for å betegne at det ikke er noen sidelinje som begynner i stasjon i .



Pavel planlegger å bygge en snarvei: dette skal være en ekspresslinje som forbinder to forskjellige stasjoner på **hovedlinja** (som potensielt allerede er naboer).

Ekspresslinjen vil ha en lengde på nøyaktig c centimeter, uavhengig av hvilke to stasjoner den forbinder.

Hvert segment på toglinjen, inkludert den nye ekspresslinjen, kan brukes i begge retninger. *Avstanden* mellom to stasjoner er den korteste lengden av en rute som går fra den ene av stasjonene til den andre langs toglinjene. *Diameteren* på hele tognettverket er den største avstanden blandt alle par med stasjoner. Med andre ord så er diameteren det minste tallet t slik at avstanden mellom hvert par av stasjoner er maksimalt t .

Pavel ønsker å bygge ekspresslinjen på en slik måte at diameteren av det resulterende nettverket er så liten som mulig.

Implementasjonsdetaljer

Du skal implementere funksjonen

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

- n : antall stasjoner langs hovedlinja,
- l : avstanden mellom stasjonene langs hovedlinja (lengden på arrayet er $n - 1$),
- d : lengden av sidelinjene (lengden på arrayet er n),
- c : lengden av den nye ekspresslinja,
- funksjonen skal returnere den minste mulige diameter på tognettverket etter at ekspresslinja har blitt lagt til.

Bruk templatfilene for detaljer på implementeringen i ditt programmeringsspråk.

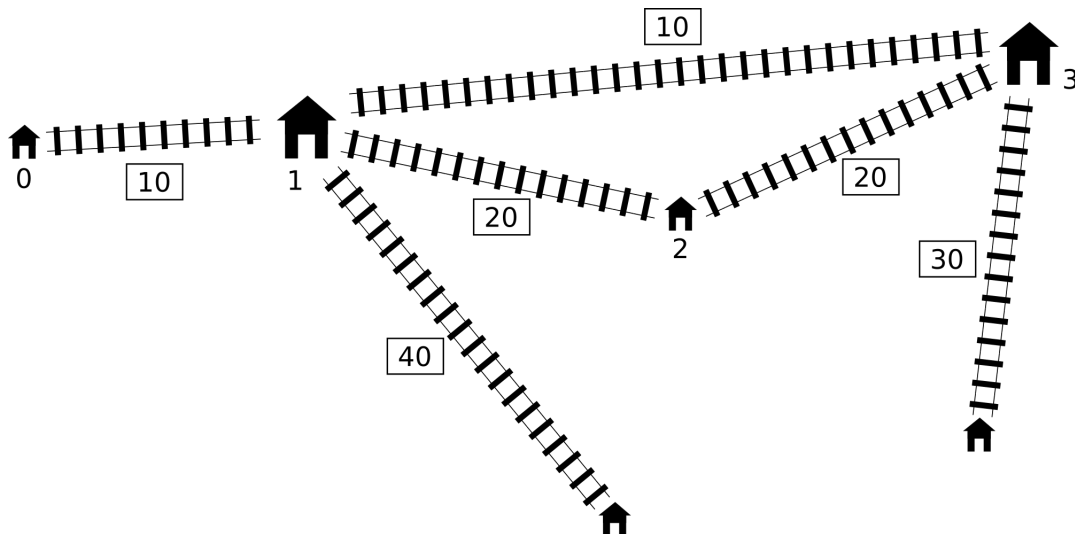
Eksempler

Eksempel 1

For tognettverket vist ovenfor så vil graderen gjøre dette funksjonskallet:

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

Den optimale løsningen er å bygge ekspresslinjen mellom stasjoner 1 og 3, som vist under.



Diameteren på dette nettverket er 80 centimeter, så funksjonen skal returnere 80.

Eksempel 2

Graderen gjør dette funksjonskallet:

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

Den optimale løsningen er å forbinde stasjonene **2** og **7**, i hvilket tilfelle diameteren blir **110**.

Eksempel 3

Graderen gjør dette funksjonskallet:

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

Den optimale løsningen er å forbinde stasjonene **1** og **2**, og dermed redusere diameteren til **21**.

Example 4

Graderen gjør dette funksjonskallet:

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

Å forbinde noen av stasjonene med en ekspresslinje med lengde **3** vil ikke forbedre den initielle diameteren av tognettverket som er **4**.

Subtasks

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

Sample grader

Eksempelgraderen leser input i følgende format:

- linje 1: heltall n og c ,
- linje 2: heltall l_0, l_1, \dots, l_{n-2} ,
- linje 3: heltall d_0, d_1, \dots, d_{n-1} .