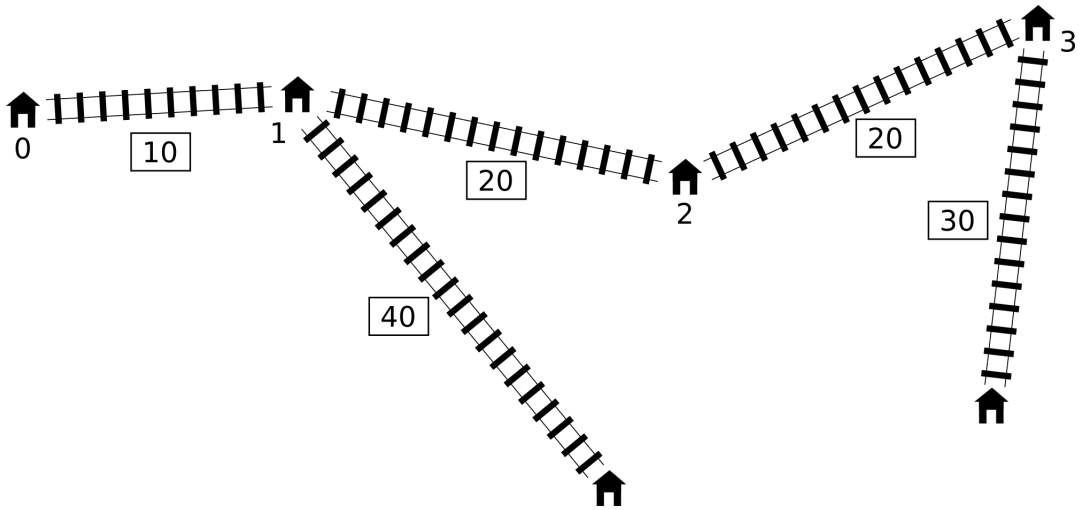
捷徑(Shortcut)

Pavel有一個非常簡單的玩具鐵道系統，這個系統有一條主線包含 n 個車站，這些車站以 0 到 $n-1$ 連續編號，車站 0 與車站 $n-1$ 是主線的兩個端點。第 i 站與第 $i+1$ 站之間的距離是 l_i 公分($0 \leq i < n-1$)。

除了主線之外，可能有若干支線。每一個支線都是從主線的某一站到一個新的車站(新車站不在主線上，且這些新車站並未被編號)。每一個主線上的車站最多只會是一條支線的起點，以第 i 號主線車站為起點的支線長度為 d_i ，我們以 $d_i = 0$ 表示沒有任何支線以主線第 i 號車站為起點。



Pavel目前正在計畫建造一條捷徑：此捷徑是一條快捷線用來連結兩個不同的**主線**車站(此兩站可能相鄰)，不管接到哪兩站，快捷線的長度恰好是 c 公分。

鐵道系統的每一段都是雙向的，包含新加入的快捷線也是雙向的。兩站之間的距離(distance)是沿著此鐵道系統連接此兩站的最短路徑長度，而在所有兩兩車站距離中的最大值稱為整個鐵道系統的直徑(diameter)，也就是說，直徑即為滿足任兩站距離都不超過 t 的最小 t 值。

Pavel想要建造一條快捷線使得加入此快捷線後的鐵道系統直徑(diameter)為最小。

實作細節

你必須實作此函式

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

- n: 主線上的車站數量，
- l: 主線上相鄰兩站之間的距離 (長度為 $n-1$ 的陣列),
- d: 支線的長度(長度 n 的陣列),

- c: 新建快捷線的長度。
- 此函式必須回傳加入新快捷線後的最小可能直徑。

在你實作細節時，請使用所提供該程式語言的樣板檔案(template files)。

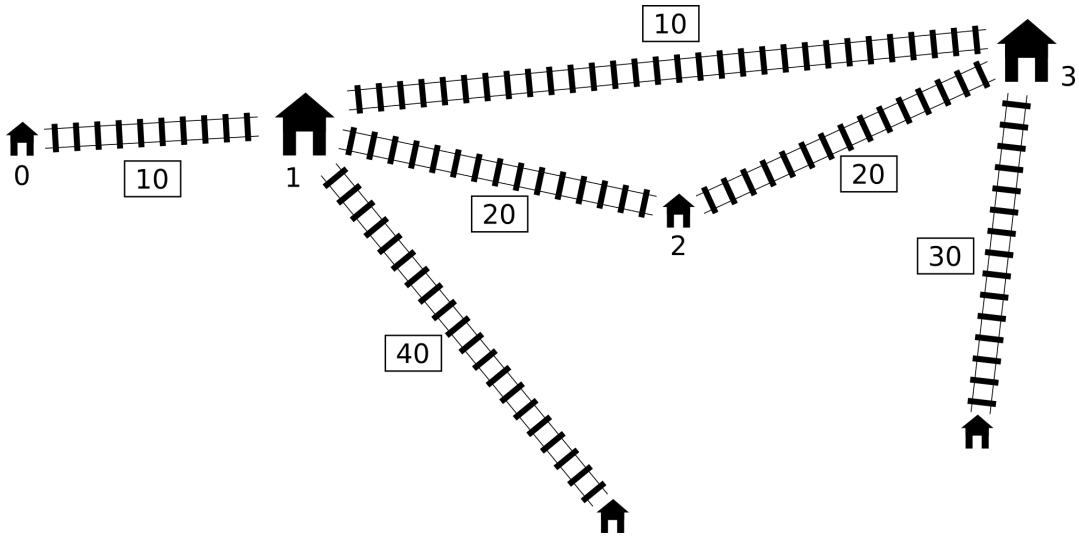
範例

範例1

以前面所舉的鐵道系統為例，評分程式(grader)將會進行以下的函式呼叫：

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

最佳解是建造車站1與車站3之間的快捷線，如下所示。



建造後新系統的直徑(diameter)為 80 公分，所以函式應該回傳 80。

範例2

評分程式(grader)若進行以下的函式呼叫：

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

最佳解是連結車站 2 與車站 7 ，直徑為 110。

範例3

評分程式(grader)若進行以下的函式呼叫：

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

最佳解是連結車站 1 與車站 2 ，可降低直徑為 21 .

範例4

評分程式(grader)若進行以下的函式呼叫：

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

以長度 3 的快捷線連接任兩主線車站無法改善原系統的直徑，該直徑為 4。

子任務(Subtasks)

在所有子任務中，$2 \leq n \leq 1\,000\,000$，$1 \leq l_i \leq 10^9$，$0 \leq d_i \leq 10^9$，$1 \leq c \leq 10^9$。

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

範例評分程式(Sample grader)

範例評分程式以下列格式讀取輸入資料：

- 第1行: integers n and c,
- 第2行: integers $l_0, l_1, \ldots, l_{n-2}$,
- 第3行: integers $d_0, d_1, \ldots, d_{n-1}$.