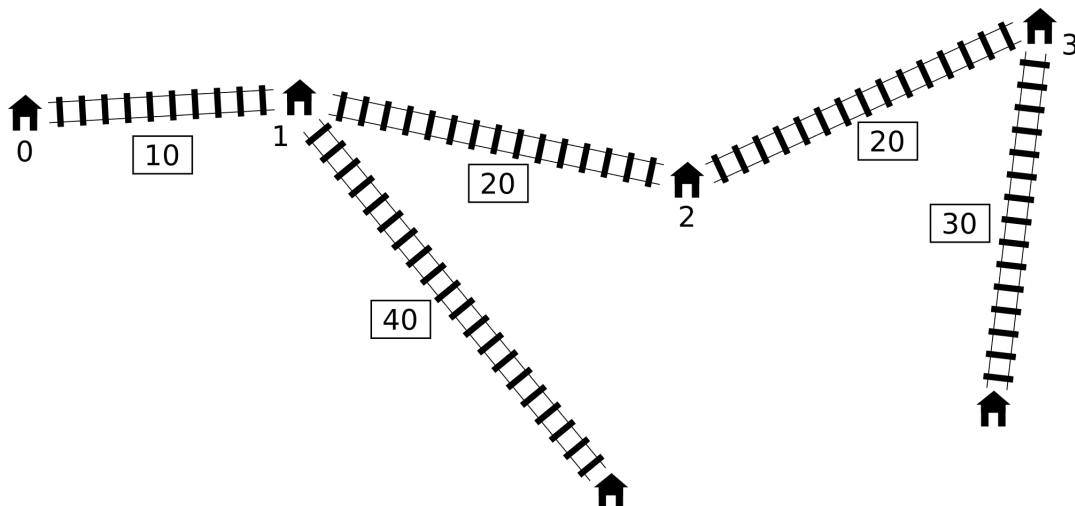


捷徑 Shortcut

Pavel 有一個非常簡單的鐵路玩具。它有一條含有 n 個車站的主幹線並且連續編號為 0 到 $n-1$ 。車站 0 和車站 $n-1$ 就在這條主幹線的兩端。其中車站 i 和車站 $i+1$ 之間的距離為 l_i 厘米 ($0 \leq i < n-1$)。

除了這條主幹線之外，這個鐵路也許會有些支線。每條支線都是主幹線中的一個車站和主幹線外的一個新車站之間的一條新鐵路。（這些新的車站不會被編號。）在主幹中一個車站最多只能有一條支線。在主幹線中車站 i 為起點的支線的長度為 d_i 厘米。我們用 $d_i = 0$ 來表示車站 i 沒有支線。



Pavel 現正計劃建造一條捷徑：一條在主幹線中任意二個不相同的車站之間（它們可能相鄰）的快速幹線。這條快速幹線不管是連接那二個車站，它的長度都將恰好是 c 厘米。

鐵路中的每一段，包括那條新的快速幹線，都能夠雙向行駛。任意二個車站的距離就是它們之間沿著鐵路由一個車站到另一個車站之間長度最短的路徑。在整個鐵路網絡之中距離最長的二個車站就叫做直徑。換句話說，存在一個最小值 t 使任意二個車站之間的距離都不會超過 t 。

Pavel 就是想建造一條快速幹線，使一個鐵路網絡的直徑最後能達到最小值。

程序實現詳情

你應該編寫下列函數：

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

- n : 主幹線中的車站數目，
- l : 主幹線中車站之間的距離（數組的長度為 $n-1$ ），

- d : 支線的長度 (數組的長度為 n),
- c : 新快速幹線的長度.
- 函數應該返回加入新快速幹線後鐵路系統直徑的最小值。

請使用適合你的程式語言所提供的模板來編寫你的程序。

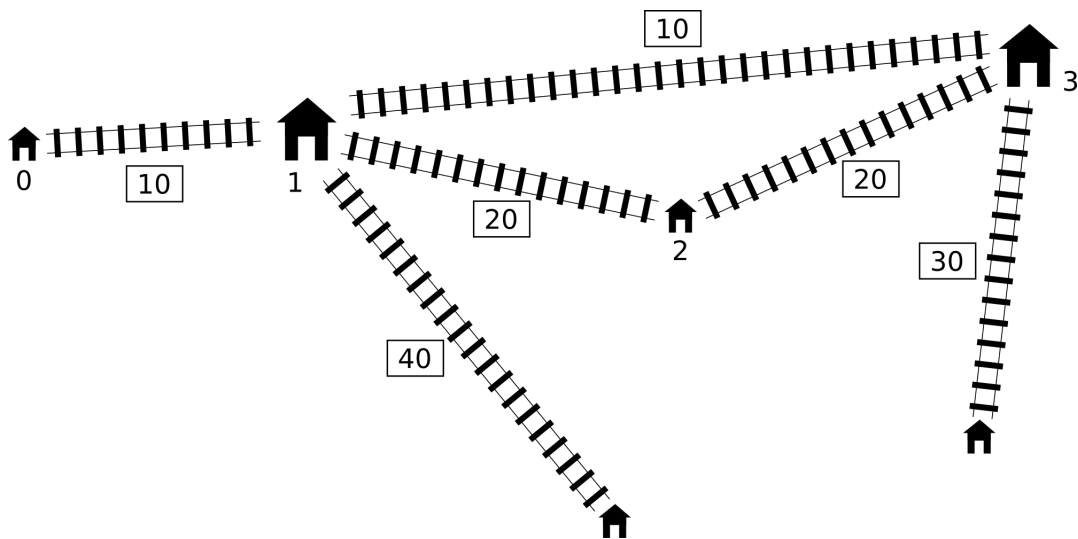
樣例

樣例 1

上圖所示是鐵路網絡，樣例評分程式會調用以下函數：

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

最優解是在車站1和車站3之間建造一條快速幹線，如下圖所示。



這個鐵路網絡中直徑的最小值是 80 厘米，所以函數應該返回數值 80。

樣例 2

樣例評分程式會調用以下函數：

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

最優解是連結車站 2 和車站 7，這個解的直徑是110。

樣例 3

樣例評分程式會調用以下函數：

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

最優解是連結車站 2 和車站 3，這樣直徑將被縮短到 21。

樣例 4

樣例評分程式會調用以下函數：

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

在任意二個車站中建立長度為 3 快速幹線都不會改進整個鐵路網絡的直徑，因此其直徑仍為初始值 4。

子任務

在所有子任務中 $2 \leq n \leq 1000000$ ， $1 \leq l_i \leq 10^9$ ， $0 \leq d_i \leq 10^9$ ， $1 \leq c \leq 10^9$ 。

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

樣例評分程式

樣例評分程式將會讀取以下格式的輸入：

- 第1行: 整數 n 和 c ，
- 第2行: 整數 l_0, l_1, \dots, l_{n-2} ，
- 第3行: 整數 d_0, d_1, \dots, d_{n-1} 。