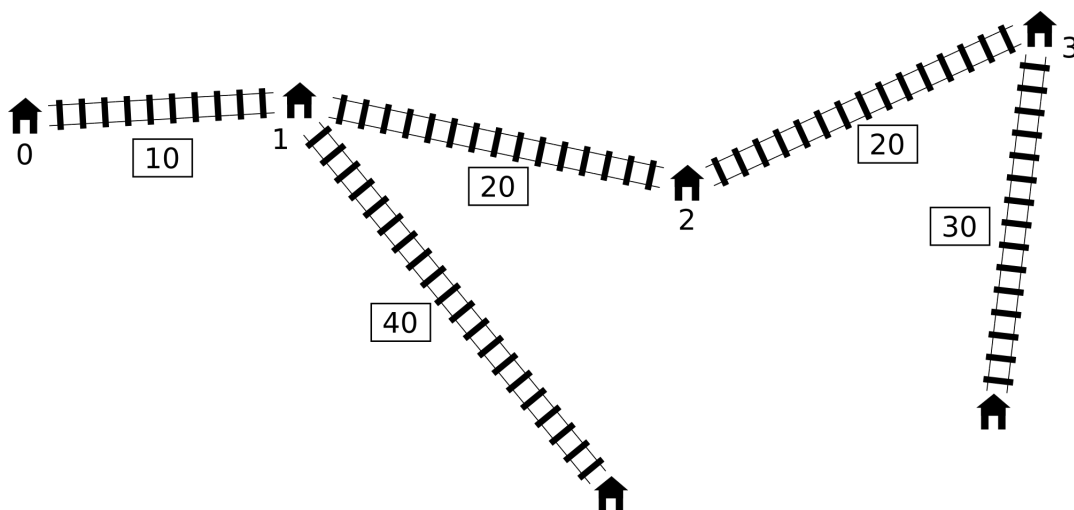


Экспресс-линия

У Павла есть игрушечная железная дорога. Она очень просто устроена. У дороги есть одна главная линия, состоящая из n станций, последовательно пронумерованных от 0 до $n - 1$ в порядке следования вдоль линии. Расстояние между станциями с номерами i и $i + 1$ равно l_i сантиметров ($0 \leq i < n - 1$).

Кроме главной линии существуют несколько второстепенных. Каждая второстепенная линия состоит из одного участка железной дороги, соединяющего станцию на главной линии и новую станцию, которая не находится на главной линии. Эти новые станции не имеют номеров. От каждой станции на главной линии может начинаться не более одной второстепенной линии. Длина второстепенной линии, начинающейся в станции с номером i , равна d_i сантиметров. Для обозначения отсутствия второстепенной дороги, начинающейся в станции i , используется $d_i = 0$.



Теперь Павел хочет построить один дополнительный участок железной дороги: экспресс-линию между двумя различными (возможно соседними) станциями, расположенными **на главной линии**. Экспресс-линия будет иметь длину ровно c сантиметров, независимо от того, какие две станции она будет соединять.

Каждый участок железной дороги, включая новую экспресс-линию, может быть использован в обоих направлениях. Расстоянием между двумя станциями называется минимальная длина пути, который проходит от одной станции до другой вдоль участков железной дороги. Диаметр железной дороги называется максимальное из расстояний между парами станций. Другими

словами, это наименьшее число t , такое что расстояние между любыми двумя станциями не больше чем t .

Павел хочет построить экспресс-линию таким образом, чтобы итоговый диаметр был как можно меньше.

Детали реализации

Вы должны реализовать функцию

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

- n : количество станций на главной линии,
- l : расстояния между станциями на главной линии (массив длины $n - 1$),
- d : длины второстепенных линий (массив длины n),
- c : длина новой экспресс-линии.
- Функция должна вернуть минимальный возможный диаметр железной дороги после постройки экспресс-линии.

Используйте предоставленные шаблоны решений для получения деталей реализации на выбранном вами языке программирования.

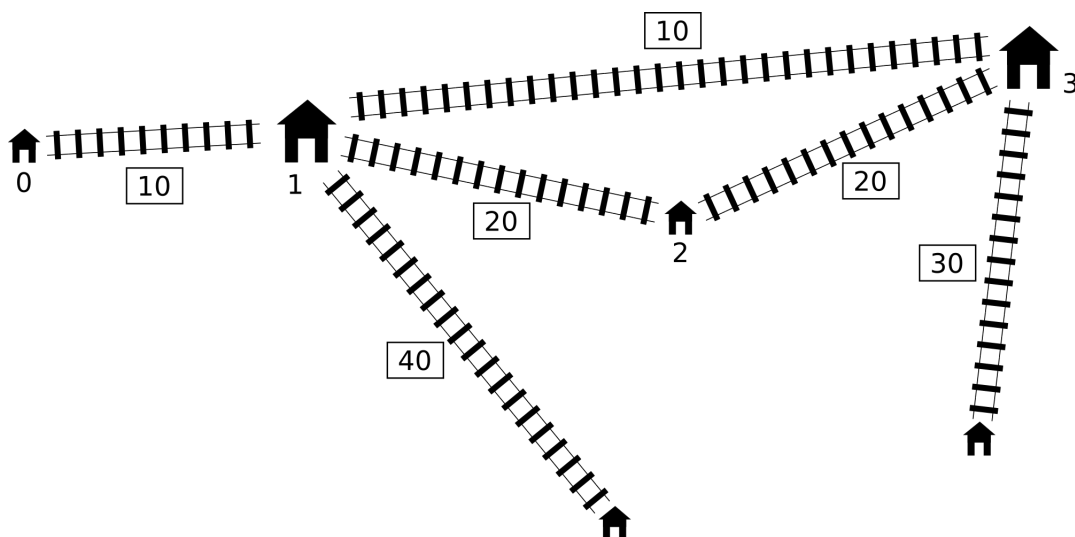
Примеры

Пример 1

Для дороги, показанной на рисунке ниже, будет сделан следующей вызов функции:

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

Оптимальное решение -- построить экспресс-линию между станциями 1 и 3, как показано ниже.



Диаметр получившейся железнодорожной дороги равен **80** сантиметрам, поэтому функция должна вернуть **80**.

Пример 2

Если будет сделан вызов функции:

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

Оптимальное решение -- соединить станции **2** и **7**, в этом случае диаметр равен **110**.

Пример 3

Если будет сделан вызов функции:

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

Оптимальное решение -- соединить станции **1** и **2**, уменьшая диаметр до **21**.

Пример 4

Если будет сделан вызов функции:

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

Соединение любых двух станций экспресс-линией длины **3** не уменьшает изначальный диаметр железнодорожной сети равный **4**.

Система оценивания

Во всех подзадачах $2 \leq n \leq 1000000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$, $1 \leq c \leq 10^9$.

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

Пример проверяющего модуля

Пример проверяющего модуля читает входные данные в следующем формате:

- Строка 1: целые числа n и c .
- Строка 2: целые числа l_0, l_1, \dots, l_{n-2} .
- Строка 3: целые числа d_0, d_1, \dots, d_{n-1} .