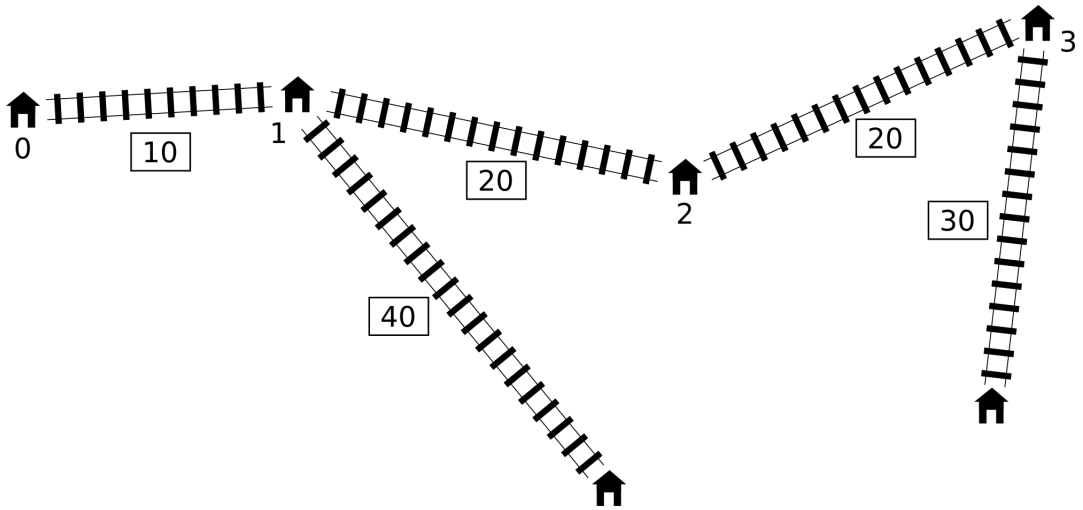
Shortcut

파벨은 장난감 기차길이 하나 있다. 기차길은 매우 단순하게 생겼다. 기차길에는 중앙 라인이 한 줄로 있는데, 중앙 라인에는 n개의 기차역이 있으며 기차길 위의 순서 대로 0부터 $n-1$까지 번호가 붙어 있다. 기차역 i번과 $i+1$번의 거리는 l_i 센티미터이다 $(0 \leq i < n-1)$.

중앙 라인 이외에 곁가지들이 있다. 하나의 곁가지는 중앙 라인에 있는 기차역 중 하나에서 시작하여 다른 새로운 (즉, 중앙 라인에 없는) 기차역으로 연결된다. (새로운 기차역들에는 번호가 없다.) 중앙 라인에 있는 기차역 하나에는 최대 하나의 곁가지가 연결된다. 기차역 i번에 연결된 곁가지의 길이는 d_i로 표현한다. 만약 $d_i = 0$이면 기차역 i번에 연결된 곁가지가 없다는 뜻이다.



파벨은 지름길을 하나 만들려고 한다. 지름길은 **중앙 라인**에 있는 두 개의 기차역을 연결한다. 두 기차역이 이미 인접한 경우에도 지름길을 만들 수 있다. 지름길은 어떤 기차역이 지름길로 연결되는지와 무관하게 그 길이가 정확히 c 센티미터이다.

새로 만든 지름길을 포함해서, 모든 기차길은 양방향으로 통행이 가능하다. 두 기차역 간의 **거리**는 한 기차역에서 다른 기차역으로 기차길을 이용해서 이동할 수 있는 가장 가까운 길의 길이를 말한다. 전체 기차길의 **지름**은 모든 쌍의 기차역 간의 거리들 중 최대값이다. 즉, 지름은 어떤 두 기차역 간의 거리도 t보다 작거나 같게 되는 가능한 t 중 가장 작은 것이다.

파벨은 전체 기차길의 지름이 가장 작게 될 수 있도록 지름길을 건설하려고 한다.

Implementation details

다음 함수를 구현하여야 한다.

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

- `n`: 중앙 라인의 기차역 개수,
- `l`: 중앙 라인에 있는 기차역 간의 거리들 (크기 $n-1$인 배열),

- d: 곁가지들의 길이 (크기 n인 배열),
- c: 새로운 지름길의 길이.
- 이 함수는 지름길을 추가하여 달성할 수 있는 최소의 지름의 값을 리턴하여야 한다.

사용하는 언어 별로 제공되는 template를 참고하여 구현의 디테일을 확인하라.

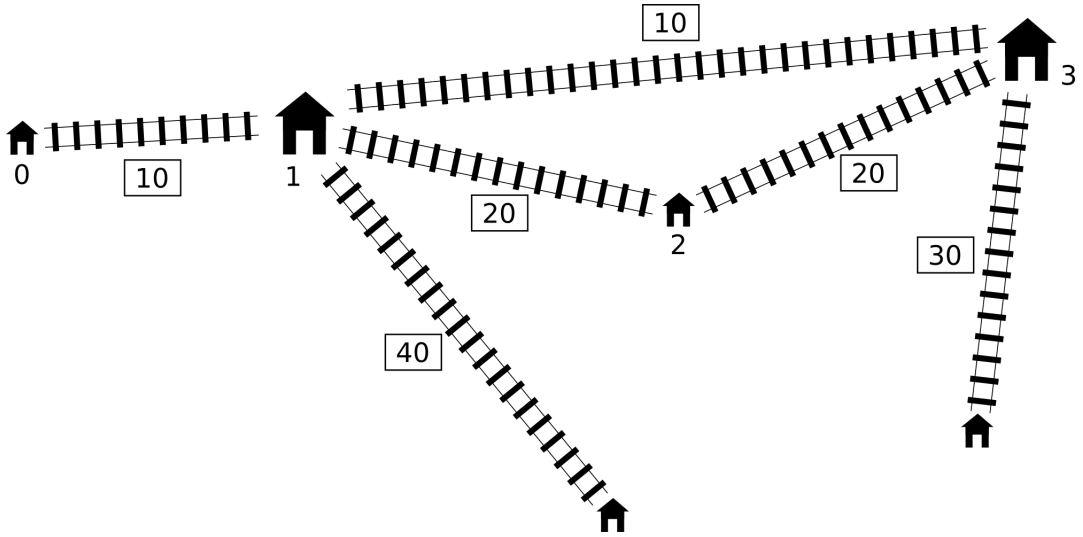
Examples

Example 1

위에서 보인 기차길의 경우 그레이더는 다음과 같은 함수 호출을 한다:

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

이 경우의 최적의 해답은 아래 그림과 같이 1번과 3번 기차역을 연결하는 지름길을 건설하는 것이다.



이때 새로운 기차길의 지름은 80이다. 따라서 함수는 80을 리턴하여야 한다.

Example 2

그레이더가 다음과 같은 함수 호출을 한다:

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

이 경우의 최적의 해답은 2번과 7번 기차역을 연결하는 지름길을 건설하는 것이다. 이때 지름은 110이다.

Example 3

그레이더가 다음과 같은 함수 호출을 한다:

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

이 경우의 최적의 해답은 1번과 2번 기차역을 연결하는 것이다. 이때 지름은 21이다.

Example 4

그레이더가 다음과 같은 함수 호출을 한다:

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

이 경우, 어떤 방식으로 길이 3인 지름길을 건설하여도 지름이 줄어들지 않고 4로 유지된다.

Subtasks

모든 Subtask에서 $2 \le n \le 1\,000\,000$, $1 \le l_i \le 10^9$, $0 \le d_i \le 10^9$, $1 \le c \le 10^9$.

1. (9 points) $2 \le n \le 10$,
2. (14 points) $2 \le n \le 100$,
3. (8 points) $2 \le n \le 250$,
4. (7 points) $2 \le n \le 500$,
5. (33 points) $2 \le n \le 3000$,
6. (22 points) $2 \le n \le 100\,000$,
7. (4 points) $2 \le n \le 300\,000$,
8. (3 points) $2 \le n \le 1\,000\,000$.

Sample grader

Sample grader는 다음의 형식으로 입력을 읽어들인다:

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