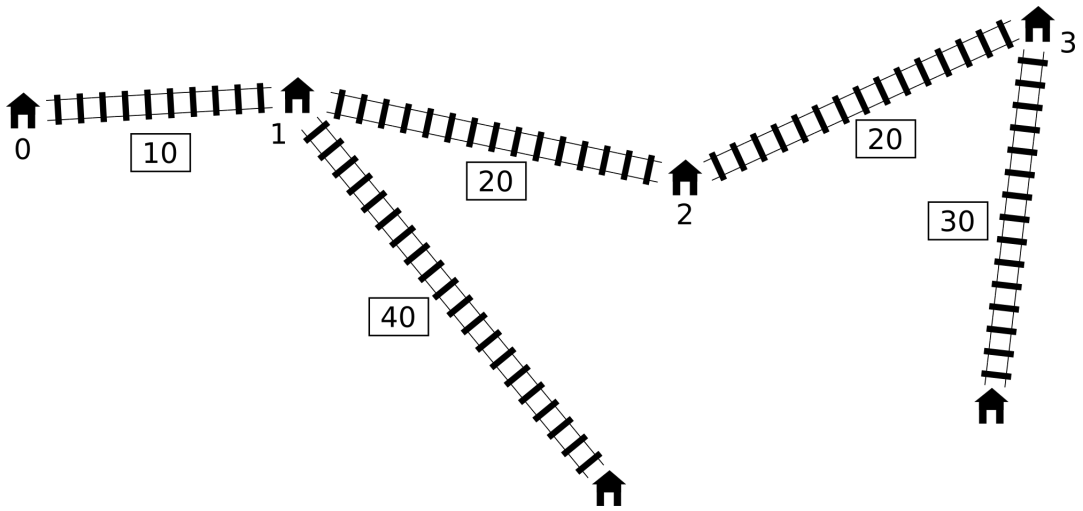
# Bližnjica

Pavel ima maketo železnice, ki je zelo preprosta. Obstaja ena glavna proga, sestavljena iz $n$ postaj. Postaje so označene od $0$ do $n-1$, v vrstnem redu po progi. Razdalja med postajama $i$ in $i+1$ je $l_i$ centimeterov ($0 \leq i < n-1$).
Od glavne proge se lahko odcepijo stranski tiri. Vsak stranski tir je železniška proga med postajo na glavni progi in novo postajo, ki ne leži na glavni progi (te nove postaje niso oštevilčene). Na vsaki izmed postaj na glavni progi se lahko prične največ en stranski tir. Dolžina stranskega tira, ki se prične na postaji $i$, je $d_i$ centimetrov. Z $d_i = 0$ označimo, da se na postaji $i$ ne prične stranski tir.




Pavel zdaj načrtuje izgraditi eno bližnjico: hitri tir med dvema (lahko tudi sosednjima) postajama **glavne proge**. Hitri tir bo imel dolžino natanko $c$ centimetrov, ne glede na izbiro dveh postaj, ki ju povezuje.

Vsak odsek železnice, vključujoč novi hitri tir, se lahko uporablja v obe smeri. *Razdalja* med dvema postajama je najmanjša dolžina poti, ki gre od ene postaje do druge, vzdolž železnice. **Premer** celotnega železniškega omrežja je največja razdalja med vsemi pari postaj. Z drugimi besedami, to je najmanjše število $t$, tako, da je razdalja med vsakim parom postaj največ $t$.

Pavel želi izgraditi hitri tir na tak način, da se premer končnega omrežja minimizira.

## Opombe k implementaciji

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

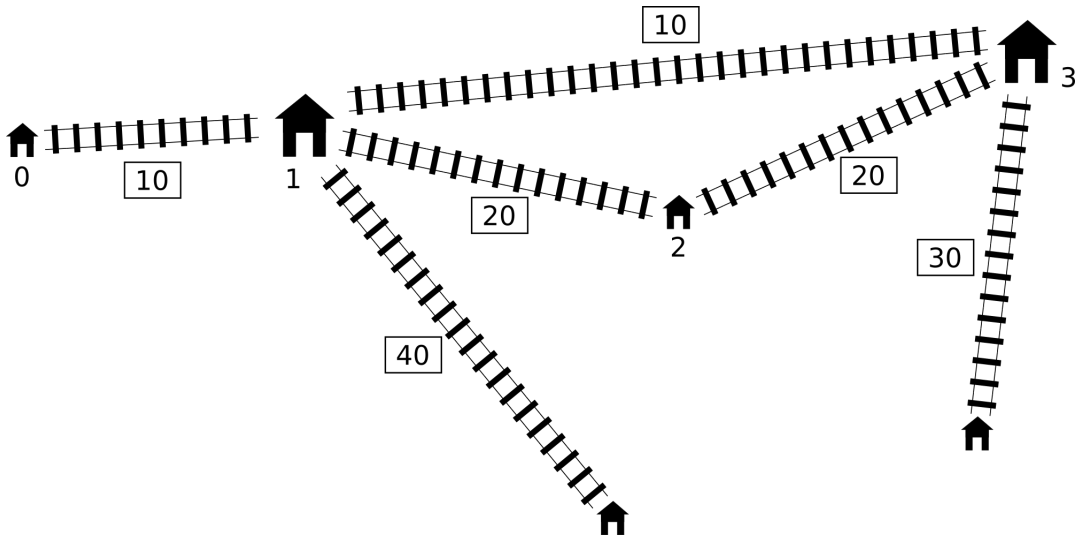
- `n`: število postaj na glavni progi,
- `l`: razdalje med postajami na glavni progi (polje dolžine $n-1$),
- `d`: dolžine stranskih tirov (polje dolžine $n$),
- `c`: dolžina novega hitrega tira.
- funkcija naj vrne najmanjši možni premer železniškega omrežja po izgradnji hitrega tira.

Prosim, uporabite predložne datoteke za več informacij o implementaciji v vašem programskem jeziku.

## Primeri

### 1. primer

Za zgoraj prikazano omrežje bi ocenjevalnik izvedel naslednji funkcijski klic:
`find_shortcut(4, [10, 20, 20], [0, 40, 0, 30], 10)`
Optimalna rešitev je izgraditi hitri tir med postajama $1$ in $3$, kot je prikazano spodaj.




Premer novega omrežja je $80$ centimetrov, zato bi funkcija morala vrniti $80$.

### 2. primer

Ocenjevalnik izvede naslednji funkcijski klic:

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

Optimalna rešitev je povezati postaji $2$ in $7$, v katerem primeru je novi premer $110$.

### 3. primer

Ocenjevalnik izvede naslednji funkcijski klic:

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

Optimalna rešitev je povezati postaji $1$ in $2$, s čimer se premer zmanjša na $21$.

### 4. primer

Ocenjevalnik izvede naslednji funkcijski klic:

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

Povezovanje katerih koli dveh postaj shitrim tirom dolžine $3$ ne izboljša začetnega premera železniškega omrežja, ki je $4$.

## Podnaloge

Pri vseh podnalogah velja $2 \leq n \leq 1\,000\,000$, $1 \leq l_i \leq 10^9$, $0 \leq d_i \leq 10^9$ in $1 \leq c \leq 10^9$.

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

## Vzorčni ocenjevalnik

Vzorčni ocenjevalnik bere vhod v naslednji obliki:

- $1$. vrstica: celi števili $n$ in $c$,
- $2$. vrstica: cela števila $l_0, l_1, \ldots, l_{n-2}$,
- $3$. vrstica: cela števila $d_0, d_1, \ldots, d_{n-1}$.