

Hálózat

N épület között mikrohullámú hálózatot építenek ki. Az egyes épületek összekötése ingyenes, de az összekötésnek üzemeltetési költsége van. Folyamatosan kapunk ajánlatokat, hogy mely kapcsolatok építhetők ki, milyen üzemeltetési költséggel és azt várjuk válaszként, hogy milyen kapcsolatokat üzemeltessünk és melyeket szüntessünk meg azért, hogy a legolcsóbb legyen az üzemeltetés és a legtöbb helyre lehessen – esetleg másokon keresztül – üzenetet küldeni. Ugyanarra a két épületre korábbi ajánlatnál olcsóbb ajánlat is jöhet. Ha olcsóbb ajánlat jön egy üzemelő kapcsolatra, akkor először a meglevőt kell megszüntetni, majd az olcsóbbat elfogadni.

Készíts programot, amely megadja, hogy mikor mely kapcsolatokat üzemeltetünk!

Könyvtár

A program megvalósításához a `halo` könyvtár műveleteit kell használni.

- `kezdet`: a programod elején egyszer kell meghívni, megadja az épületek számát ($1 \leq N \leq 1000$).
- `ajanlat(a, b, c)`: három paraméterében megadja, hogy a következő kapcsolat az a és a b épületek között építhető ki ($1 \leq a \neq b \leq n$), c üzemeltetési költséggel ($1 \leq c \leq 10\,000$). Ez az eljárás befejezi a programod futását, ha nincs több ajánlat.

Minden ajánlat után a következő két eljárás segítségével kell közölnöd a rendszerrel, hogy mely új kapcsolatokat kell üzembe állítani és melyeket kell megszüntetni, hogy az üzemeltetési költség minimális legyen:

- `hozzaad(a, b)`; az a és a b épületek közötti kapcsolatot ($1 \leq a \neq b \leq n$) beteszi az üzemeltetendők közé.
- `elvesz(a, b)`; az a és a b épületek közötti kapcsolatot ($1 \leq a \neq b \leq n$) kiveszi az üzemeltetendők közül.

Megjegyzés: Hibás `hozzaad` vagy `elvesz` eljáráshívás esetén a program automatikusan befejeződik.

A műveletek Pascal deklarációja:

```
uses halo;  
function kezdet: longint;  
procedure ajanlat(var a,b,c: longint);  
procedure hozzaad(a,b: longint);  
procedure elvesz(a,b: longint);
```

A műveletek C/C++ deklarációja:

```
#include "halo.h"  
int kezdet();  
void ajanlat(int &a, int &b, int &c);  
void hozzaad(int a, int b);  
void elvesz(int a, int b);
```

Használat

Letölthető egy minta `halo` modul C++ és Pascal programja. A `kezdet` a standard bemenetről egy egész számot olvas be, az épületek számát. Az `ajanlat` egy a, b, c számhármast olvas be és ad vissza a paramétereiben. Leáll, ha $a=0$.

Példa

Az ajánlat eljárás az alábbi adatokat kapja, amire a következő eljáráshívásokkal kell reagálni:

kezdet(12)

```
ajanlat(1,2,5)    ⇒ hozzáad(1,2)
ajanlat(6,7,5)    ⇒ hozzáad(6,7)
ajanlat(6,10,4)   ⇒ hozzáad(6,10)
ajanlat(11,7,4)   ⇒ hozzáad(11,7)
ajanlat(10,11,3)  ⇒ elvesz(6,7); hozzáad(10,11)
ajanlat(1,5,6)    ⇒ hozzáad(1,5)
ajanlat(2,6,7)    ⇒ hozzáad(2,6)
ajanlat(2,5,4)    ⇒ elvesz(1,5); hozzáad(2,5)
ajanlat(6,1,4)    ⇒ elvesz(2,6); hozzáad(6,1)
ajanlat(7,11,1)   ⇒ elvesz(7,11); hozzáad(7,11)
ajanlat(4,8,10)   ⇒ hozzáad(4,8)
ajanlat(0,0,0)    ⇒ a futás véget ér, ez az érték nem jut vissza a programodba
```

Korlátok

A megoldás program nem olvashat, és nem írhat semmilyen állományt!

Időlimit: 7.0 mp.

Memórialimit: 32 MiB

Pontozás: A tesztek 40%-ában $N \leq 100$, 60%-ában $N \leq 500$. A tesztek 60%-ában az ajánlatokban szereplő (a,b) párok mind különbözőek.