12. feladat: Hálózat (50 pont)

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 (maximum 10 000-et), 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!

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

Könyvtár

- `kezdet` – a programod elején egyszer kell meghívni, megadja az épületek ***N*** számát ***(1≤N≤1000)***.
- `ajanlat` – három paraméterében megadja, hogy a következő kapcsolat az ***a*** és a ***b*** épületek között építhető ki ***(1≤a≠b≤n)***, ***c*** üzemeltetési költséggel ***(1≤c≤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≤a≠b≤n)*** beteszi az üzemeltetendők közé.
- `elvesz(a,b);` – az ***a*** és a ***b*** épületek közötti kapcsolatot ***(1≤a≠b≤n)*** kiveszi az üzemeltetendők közül.

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

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

Pascal program esetén:

```
uses halo;
```

A műveletek Pascal deklarációja

```
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);
```

Példa: Az `ajanlat` eljárás az alábbi adatokat kapja, amire a következő eljáráshívásokkal kell reagálni:

```
kezdet(12)
```

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

Időlimit: 7 mp.

Memórialimit: 32MB

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