

4. feladat: Vám (35 pont)

A középkorban egyes városok vámot szedtek azon kereskedőktől, akik áthaladtak rajtuk. A városokat persze nem lehetett elkerülni, mert az utak városokon keresztül vezettek.

Írj programot (**vam.pas**, **vam.c**, ...), amely megadja, hogy a kereskedő az **A** városból a **B** városba milyen útvonalon menjen, ha útközben a lehető legkevesebb vámot akarja fizetni!

Bemenet

Az **vam.be** szöveges állomány első sorában a városok N száma ($1 \leq N \leq 100$), a közöttük levő utak M száma ($1 \leq M \leq 10000$), valamint a kezdő és a cél város sorszáma ($1 \leq A \neq B \leq N$) van. A második sorban N szám, az egyes városokban fizetendő adó van, egy-egy szóközzel elválasztva. A következő M sor mindegyikében két-két város sorszáma van szóközzel elválasztva, amelyek között van közvetlen, mindkét irányban járható út.

Kimenet

Az **vam.ki** szöveges állomány első sorába a legolcsóbb úton fizetendő adó összegét kell írni, a második sorba pedig azon városok sorszámaát, amelyeken keresztül vezet az **A**-ból **B**-be tartó legolcsóbb út (a haladás sorrendjében, azaz az első szám mindenképpen **A**, az utolsó pedig **B**). Ha nincs út **A**-ból **B**-be, akkor egyetlen sort kell kiírni, amelyben a -1-es érték szerepeljen!

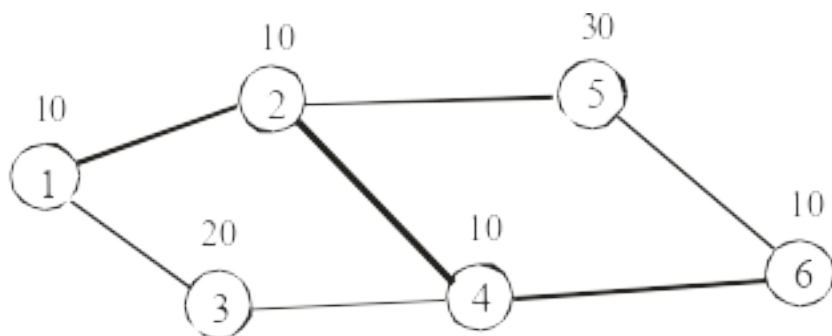
Példa bemenet és kimenet:

vam.be

```
6 7 1 6
10 10 20 10 30 10
1 2
1 3
2 4
3 4
2 5
4 6
5 6
```

vam.ki

```
20
1 2 4 6
```



5. feladat: Raktár (35 pont)

Egy vállalat az ország különböző városaiban levő üzemeiben gyárt alkatrészeket. A heti termelést egy adott városban létesítendő központi raktárban kívánja összegyűjteni. Bizonyos város-párokat kétirányú közvetlen út köt össze. Az úthálózat olyan, hogy bármely két város között pontosan egy útvonal létezik. Ha az U városból az ott lévő D mennyiségű alkatrészt a V városba szállítják, akkor ennek költsége $t(U,V)*D$, ahol $t(U,V)$ az U és V város távolsága. Az U és V város távolsága az a legkisebb k szám, amelyre létezik olyan $U=p_0, p_1, \dots, p_k=V$ sorozat, hogy a sorozatban egymást követő városok között van közvetlen út. A központi raktár helyét úgy kívánják meghatározni, hogy a szállítás összköltsége a lehető legkisebb legyen.

Írj programot (**raktar.pas**, **raktar.c**, **raktar.cpp**), amely kiszámítja a központi raktár helyét és a szállítás összköltségét!

Bemenet

A **raktar.be** szöveges állomány első sorában a városok N ($1 < N \leq 3000$) száma van. A második sor N darab pozitív egész számot tartalmaz, az i -edik szám az i -edik városban gyártott alkatrészek száma, ami nem nagyobb, mint 100 . A következő $N-1$ sor mindegyike két egész számot tartalmaz, két olyan város sorszámát, amelyeket közvetlen (mindkét irányban járható) út köt össze. A városokat az $1, \dots, N$ számokkal azonosítjuk.

Kimenet

A **raktar.ki** szöveges állomány első sorába a lehető legkisebb szállítási összköltséget kell írni. A második sorba annak a városnak a sorszámát kell írni, ahol a központi raktárt létesíteni kell. Ha több ilyen lenne, akkor közülük a legkisebb sorszámú várost kell kiírni.

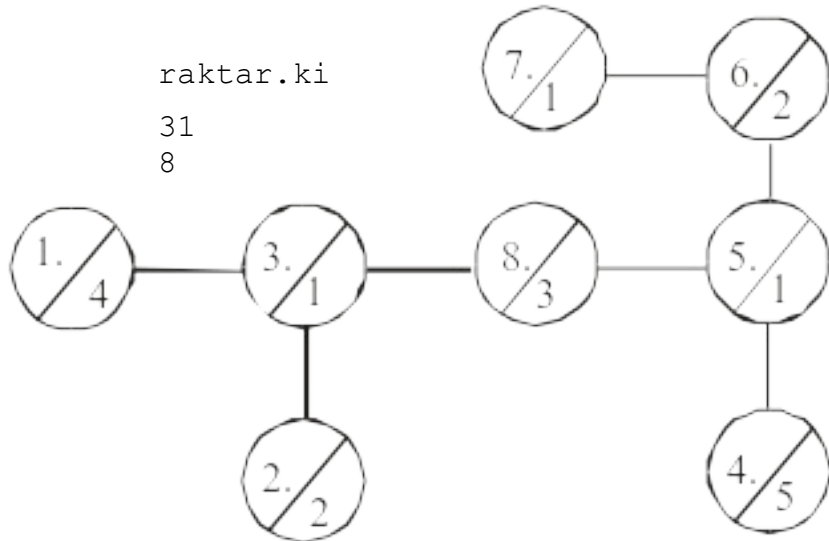
Példa bemenet és kimenet:

raktar.be

```
8
4 2 1 5 1 2 1 3
5 4
6 7
5 6
5 8
3 8
1 3
3 2
```

raktar.ki

```
31
8
```



6. feladat: Háromszög (30 pont)

Adott a síkon egy P ponthalmaz és egy ezektől különböző q pont. Ki kell választani a ponthalmaz három olyan a , b és c pontját, amelyekre teljesül, hogy a q pont az a , b és c pontok által meghatározott háromszögbe, vagy valamelyik oldalára esik, de a P ponthalmaz egyetlen más pontja sem esik a háromszögbe vagy oldalára.

Írj programot (**harom.pas**, **harom.c**, **harom.cpp**), amely kiszámít egy q -t tartalmazó háromszöget, van van ilyen!

Bemenet

A **harom.be** szöveges állomány első sora a P ponthalmaz pontjainak N ($3 \leq N \leq 8000$) számát tartalmazza. A második sor tartalmazza a kitüntetett q pont x - és y -koordinátáját. A következő N sor mindegyike két egész számot tartalmaz, egy-egy pont koordinátáit. Az $i+2$ -edik sorban lévő x és y ($-30000 \leq x, y \leq 30000$) az i -edik pont x - és y -koordinátája.

Kimenet

A **harom.ki** szöveges állomány első sorába három egész számot kell írni (egy-egy szóközzel elválasztva), a keresett háromszöget megadó pontok sorszámát, órajárással ellentétes felsorolásban. Ha nincs megoldás, akkor az első sorba három 0 számot kell írni. Több megoldás esetén bármelyik kiírható.

Példa bemenet és kimenet:

harom.be

```
5
4 4
0 1
2 2
6 1
2 6
6 3
```

harom.ki

```
4 3 5
```

