

4. feladat: Lépcső (35 pont)

Egy lépcsőn N lépcsőfok van, egyszerre legfeljebb K lépcsőfokot léphetünk, de vannak hibás lépcsők, amelyekre tilos lépni.

Készíts programot (**lepcso.pas**, **lepcso.c**, ...), amely megadja hogy hányféleképpen mehetünk fel úgy a lépcső tetejére, hogy legfeljebb L lépést teszünk?

A **lepcso.be** szöveges állomány első sorában a lépcsők száma ($1 \leq N \leq 100$), az egyszerre léphető lépcsők száma ($1 \leq K \leq N$), a hibás lépcsők száma ($0 \leq H \leq N$) és a lépések száma ($1 \leq L \leq N$) van, egy-egy szóközzel elválasztva. A második sorban H egész szám, a hibás lépcsők sorszáma van, egy-egy szóközzel elválasztva.

A **lepcso.ki** szöveges állomány egyetlen sorába az a szám kerüljön, ahányféleképpen felmehetünk úgy a lépcső tetejére, hogy legfeljebb L lépést teszünk! Feltehető, hogy csak olyan esetekkel tesztlünk, amikor a megoldás elfér 64 bites egész típusú változóban.

Példa:

lepcso.be

6 2 1 4
3

lepcso.ki

3

5. feladat: Hálózat (30 pont)

Egy számítógépes hálózat csomópontokból és bizonyos csomópont-párokat összekötő, kétirányú adatátvitelt biztosító közvetlen vonalakkal épül fel. Minden közvetlen vonal adott átviteli sebességet biztosít. Adatátvitel természetesen közvetett módon is lehet, több közbülső csomóponton keresztül. Ekkor az átvitel sebességét az útvonalba eső közvetlen útvonalak átviteli sebességének minimuma adja.

Készíts programot (**halozat.pas**, **halozat.c**, ...), amely kiszámítja, hogy adott két csomópont között mekkora a lehető legnagyobb átviteli sebesség, és meg is ad egy megfelelő útvonalat!

A **halozat.be** szöveges állomány első sorában két egész szám van, egy szóközzel elválasztva, a csomópontok N ($N \leq 1000$) száma, a közvetlen vonalak M ($1 \leq M \leq 100000$) száma. A csomópontokat az $1, \dots, N$ számokkal azonosítjuk. A második sor két csomópont sorszámot tartalmaz, egy szóközzel elválasztva; **P Q**. A következő M sor mindegyike három egész számot tartalmaz, egy-egy szóközzel elválasztva, **u, v; s** ami azt jelenti, hogy közvetlen kétirányú átviteli vonal van kiépítve az **u** és a **v** csomópont között, aminek átviteli sebessége **s** ($1 \leq s \leq 1000$).

A **halozat.ki** szöveges állomány első sora azt a legnagyobb lehetséges átviteli sebességet tartalmazza, amelyen átvitelt lehetséges a bemenetben megadott **P** és **Q** csomópont között. A második sor egy olyan útvonalat tartalmazzon, amelyen a lehető legnagyobb átvitelt adja **P** és **Q** között. A sorban az első a **P**, az utolsó a **Q** pont legyen. Több megoldás esetén bármelyik megadható. Ha nem lehetséges átvitel, akkor az első és egyetlen sorba a 0 számot kell kiírni.

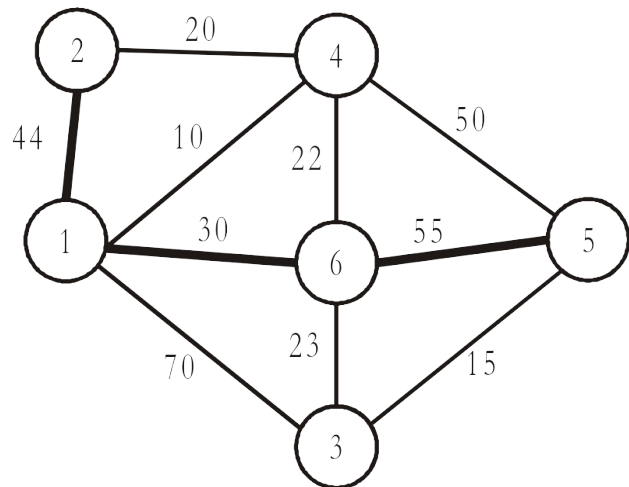
Példa:

halozat.be

```
6 10
2 5
1 2 44
2 4 20
1 4 10
1 6 30
6 4 22
4 5 50
5 3 15
6 3 23
1 3 70
6 5 55
```

halozat.ki

```
30
2 1 6 5
```



6. feladat: Építkezés (35 pont)

Egy nagyszabású építkezés terve N munkaelem elvégzését írja elő. Minden munkaelem pontosan egy nap alatt végezhető el, és egy napon csak egy munkaelemet lehet végezni. A terv azt is tartalmazza, hogy az egyes munkaelemek legkorábban melyik napon végezhető el, figyelembe véve a szükséges alapanyagok legyártását. Továbbá, bizonyos munkaelem csak akkor végezhető el, ha más munkaelemet már elvégeztek. Ezt nevezik megelőzési előírásnak.

Készíts programot (**epit.pas**, **epit.c**, ...), amely kiszámítja, hogy legkevesebb hány nap alatt lehet elvégezni az építkezést, és meg is ad egy ütemezést!

A **epit.be** szöveges állomány első sorában két egész szám van, egy szóközzel elválasztva, a munkaelemek N száma ($1 \leq N \leq 10000$) és a megelőzési előírások M ($1 \leq M \leq 100000$) száma. A munkaelemeket az $1, \dots, N$ számokkal azonosítjuk. A második sor pontosan N egész számot tartalmaz, egy-egy szóközzel elválasztva, az i -edik szám az i -edik munkaelem kezdési határideje. A további M sor mindegyike két munkaelem sorszámot tartalmaz egy szóközzel elválasztva, egy megelőzési előírást, ami azt jelenti, hogy az első számú munkaelemet előbb kell elvégezni, mint a második számút.

A **epit.ki** szöveges állomány első sora azt a legkisebb K számot tartalmazza, ahány nap alatt az összes munkaelemet el lehet végezni, betartva a feltételeket. A második sor pontosan N egész számot tartalmazzon, egy-egy szóközzel elválasztva. A sorban az i -edik szám annak a napnak a száma legyen, amelyik napon az i -edik munkaelemet elvégzik. Több megoldás esetén bármelyik megadható.

Példa:

epit.be

```
7 10
5 6 3 8 2 1 4
2 1
4 1
4 5
4 6
2 4
5 3
5 6
3 7
1 3
1 7
```

epit.ki

```
13
11 6 12 8 9 10 13
```

