

Kedves Versenyző! A megoldások értékelésénél csak a **programok futási eredményeit** vesszük tekintetbe. Ezért igen fontos a **specifikáció pontos betartása**. A programokat csak a feladatkiírásban leírt szabályoknak megfelelő adatokkal próbáljuk ki, emiatt nem kell ellenőrizni, hogy a bemenő adatok helyesek-e, illetve a szükséges állományok léteznek-e. Ha a programnak valamilyen állományra van szüksége, akkor azt mindig az aktuális könyvtárba kell rakni. Az állomány neve minden esetben rögzített.

1. feladat: Örök (25 pont)

Egy szakaszán egy folyó sok ágra szakad, majd az ágak találkozásával újra egyetlen folyóként folytatja útját. Az egyes ágak szétválásánál és találkozásánál őrhelyeket létesítettek. Ha az első elágazásnál a folyóra teszünk egy csónakot, akkor az valamely útvonalon el fog jutni az utolsó találkozási pontig, minden egyes ágot adott idő alatt megtevé. Ha egy őrt az A. időponttól a B. időpontig alkalmazunk, akkor az alkalmazási ideje $B-A+1$.

Készíts programot, amely megadja minden őrhelyre, hogy ott mennyi ideig kell őrt alkalmazni, hogy az esetlegesen elhaladó csónakot biztosan megfigyelhessük!

Az `orok.be` állomány első sorában az őrhelyek száma ($1 \leq N \leq 10\,000$) és az ágak száma ($1 \leq M \leq 100\,000$) van. A következő M sorban az egyes ágak két végpontján levő őrhelyek sorszáma ($1 \leq A_i, B_i \leq N$) és e két őrhely közötti út megtételéhez szükséges idő ($1 \leq T_i \leq 100$) van

Az `orok.be` állomány egyetlen sorába pontosan N számot kell írni! Az i. szám az i. őrhelyen levő őrt alkalmazási ideje legyen!

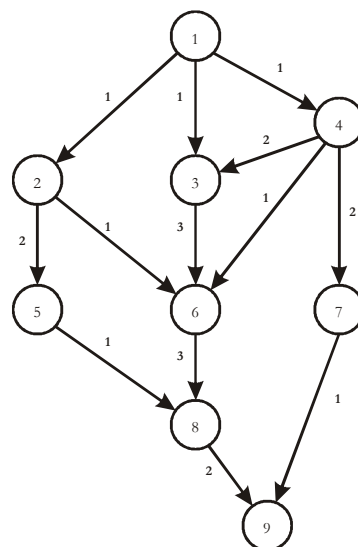
Példa:

`orok.be`

```
9 13
1 2 1
1 3 1
1 4 1
4 3 2
2 5 2
2 6 1
3 6 3
4 6 1
4 7 2
5 8 1
6 8 3
7 9 1
8 9 2
```

`orok.ki`

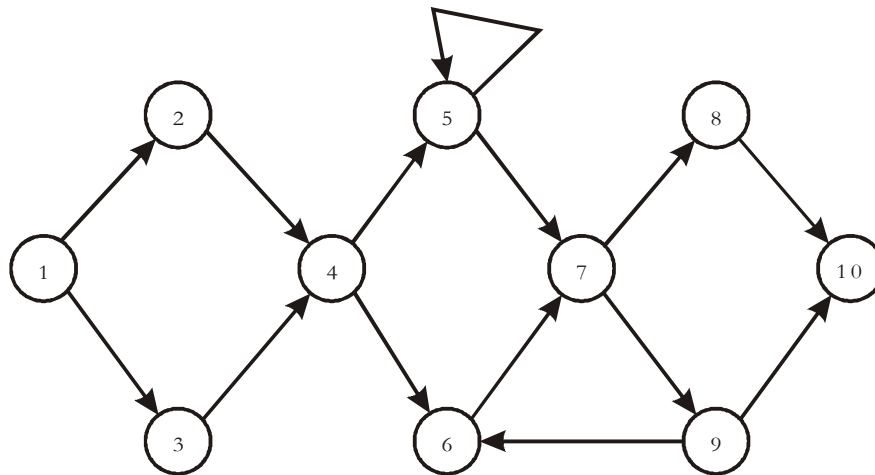
```
1 1 3 1 1 5 1 6 8
```



2. feladat: Programgráf (25 pont)

Egy programgráf olyan irányított gráf, amelynek pontjai a program utasításai, élei pedig az utasítások sorrendjét határozzák meg. A programgráf bármely pontja biztosan elérhető a kezdőpontból és a végpont biztosan elérhető bármely pontból.

Készíts programot, megadja egy programgráf azon pontjait, amelyek a programgráfot két független, legalább egy pontot tartalmazó részre osztják!



A `program.be` állomány első sorában a pontok száma ($1 \leq N \leq 1000$) és az élek száma ($1 \leq M \leq 100\,000$) van. Tudjuk, hogy az 1. pont a kezdőpont és az N. pont a végpont. A következő M sorban egy-egy él két végpontja sorszáma van ($1 \leq A, B \leq N$).

A `program.ki` állomány első sorába a programgráfot két részre osztó pontok számát kell kiírni, a második sorába pedig ezen pontok sorszámaát, növekvő sorrendben!

Példa:

`program.be`

10 14

1 2

1 3

2 4

3 4

4 5

4 6

5 5

5 7

6 7

7 8

7 9

9 6

8 10

9 10

`program.ki`

1

4

3. feladat: Szavak (25 pont)

Adott szavak egy halmaza, valamint a felhasználható betűk és pontértékük. A szavak csak az angol ábécé betűit tartalmazzák.

Készíts programot, amely megadja a felhasználható betűkből kirakható legmagasabb pontértékű szavak vagy szópárok számát!

A `szavak.be` állomány első sora felhasználható kisbetűkből álló karaktersorozat tartalmazza. A második sorban az angol ábécé betűinek pontértéke található (26 természetes szám). A harmadik sorban a felhasználható szavak száma van ($1 \leq N \leq 40\,000$).

Az utolsó N sorban soronként egy-egy legalább 3 és legfeljebb 7 betűből álló szó következik.

A szavak.ki állomány első sorába az elérhető maximális pontszámot kell írni! A második sorba az ennyi pontot érő megoldások száma kerüljön!

Példa:

```
szavak.be
faalpkp
1 1 1 1 1 2 1 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1
7
falat
falap
fal
ablak
pap
lapka
kalap

szavak.ki
7
3
```

4. feladat: Tornasor (25 pont)

Testnevelés órán a tanulók tornasorba állnak. Legelől áll a legmagasabb, akit a többiek magasság szerint csökkenő sorrendben követnek. Feltehető, hogy nincs két egyforma magasságú tanuló.

Készíts programot, amely megadja, hogy hányféle sorrendbe rendezhetők át úgy a tanulók, hogy pontosan K tanuló előtt álljon nála magasabb!

A tornasor.be állomány első sorában a tanulók száma ($1 \leq N \leq 20$) és a K értéke ($1 \leq K \leq N-1$) van.

A tornasor.ki állomány egyetlen sorába azon sorrendek számát kell írni, amelyekben pontosan K tanuló előtt áll nála magasabb tanuló!

Példa:

```
tornasor.be
3 1

tornasor.ki
4
```