

4. feladat: Bírálók (30 pont)

N könyvet 2 bírálónak (A és B) adnak ki, hogy mindegyikről bírálatot készítsenek. Az i . könyv elolvasása az A bírálónak A_i , a B bírálónak pedig B_i ideig tart. A két bíráló egyszerre ugyanazt a könyvet nem olvashatja, és egyszerre mindegyikük legfeljebb 1 könyvet olvashat. Egy könyv olvasását bármelyikük abbahagyhatja, s egy későbbi időpontban onnan folytathatja. A könyvek között az első K fontos könyv. Ezekre teljesülni kell, hogy bármelyik csak akkor olvasható, ha mindkét bíráló elolvasta már az összes, nála kisebb sorszámú könyvet.

Feladat:

Készíts programot (BIRALO.PAS vagy BIRALO.C), amely megadja, hogy a két bíráló mikor végezhet a lehető leghamarabb!

Bemenet:

A BIRALO.BE állomány első sorában a könyvek száma ($1 \leq N \leq 1000$) és a fontos könyvek száma ($1 \leq K \leq N$) van. A következő N sor A és B számára az egyes könyvek elolvasásához szükséges időt tartalmazza, egy szóközzel elválasztva ($1 \leq A_i, B_i \leq 100$).

Kimenet:

A BIRALO.KI állományba azt az időtartamot kell írni, ami alatt a két bíráló a leghamarabb végezhet.

Példa:

BIRALO.BE	BIRALO.KI
3 2	15
5 5	
4 1	
3 4	

5. feladat: Barátok (30 pont)

Egy osztályba N tanuló jár. Mindenkinek vannak barátai. Egyikük születésnapjára kap egy könyvet, amelyről úgy gondolja, hogy mindenkinek el kellene olvasnia. Ezért egy N elemű láncot kell szervezni, ahol mindenki a barátai közül valakinek adja tovább a könyvet, amely a végén visszajut a tulajdonosához.

Feladat:

Készíts programot (BARAT.PAS vagy BARAT.C), amely megadja a láncot!

Bemenet:

A BARAT.BE állomány első sorában a tanulók száma és a barátok minimális száma ($1 \leq N \leq 100$, $2/3 * N \leq K < N$) van egy-egy szóközzel elválasztva. A következő N sor mindegyike legalább K számot tartalmaz, egy-egy szóközzel elválasztva, az egyes tanulók barátainak sorszámát.

Kimenet:

A BARAT.KI állomány első sorába a lánc tagjait kell írni, feltéve, hogy az 1-es sorszámtól indult el a könyv, egy-egy szóközzel elválasztva.

Példa:

BARAT.BE	BARAT.KI
5 3	1 5 3 2 4
3 4 5	
4 3 5	
1 2 4 5	
1 3 2	
1 2 3	

6. feladat: Játék (40 pont)

Tekintsük azt a kétszemélyes játékot, amelyet egy $N \times N$ -es négyzethálós táblán lehet játszani. A tábla bizonyos mezőin pozitív egész számok vannak elhelyezve. A két játékos egyetlen bábut mozgat felváltva lépkedve. Egy lépésben egyet lehet lépni a bábuval szomszédos mezőre vagy lefelé, vagy jobbra. A játék akkor ér véget, amikor a második játékos a tábla jobb alsó mezőjére lép. A bábu a játék kezdetén a tábla bal felső sarkában van. Az első játékos megszerzi mindazon pontokat, amelyek olyan mezőn vannak, amire lépett. A játék célja az, hogy az első játékos, aki a játékot kezdi, a lehető legtöbb összpontot szerezzék meg. A második játékos arra törekszik, hogy lépéseivel akadályozza az első játékost a legjobb eredmény elérésében.

Feladat:

Írj programot (JATEK.PAS vagy JATEK.C), amely az első játékos játékát valósítja meg. A második játékos játékát a GEP.TPU modul valósítja meg.

A játékot azzal kell kezdeni, hogy az első játékos végrehajtja a `Kezd` eljárást. Az első játékos saját lépését a `Lep(L)` eljárás végrehajtásával közli a második játékosal, ahol `L` értéke 'L' vagy 'J', attól függően, hogy milyen irányba lép. A második játékos válaszlépését a `GepLep(L)` eljárással kérdezheti le, az `L Char` típusú változóban kapja meg a választ, aminek értéke szintén 'L' vagy 'J' lehet, az 'L' azt jelenti, hogy a második játékos lefelé, a 'J' pedig, hogy jobbra lépett. (A `GepLep` eljárás mindig a második játékos utoljára végrehajtott lépését adja.).

Bemenet:

A JATEK.BE szöveges állomány első sorában két szám van: N M egy szóközzel elválasztva, N a táblamérete ($1 < N \leq 100$). A következő M sorban található a kezdeti táblaállás, megadva azokat a mezőket, amelyek tartalma nem 0. Ezek három számot tartalmaznak: i j k , ami azt jelenti, hogy az i -edik sor j -edik oszlopában a táblán a k szám van. (A sorokat fentről-lefelé, az oszlopokat balról-jobbra sorszámozzuk.)

Kimenet:

A program nem írhat kimeneti állományba. A játék végeredményét a GEP modul feljegyzi.

Példa:

A program az alábbi szerkezetű lehet:

```

Program jatek;
uses gep;
begin {program}
  Kezd;
  while ... do
  begin
    ...
    Lep(L1);
    ...
    GepLep(L2);
    ...
  end;
end. {program}

```

