

Kedves Versenyző! A megoldások értékelésénél csak a **programok futási eredményeit** vesszük tekintetbe. Ezért igen fontos a **specifikáció pontos betartása**. Ha a feladat szövege adatok valamilyen állományból történő beolvasását írja elő, és a program ezt nem teljesíti, akkor a feladatra nem adunk pontot (akkor sem, ha egyébként tökéletes lenne a megoldás); az objektív értékelés érdekében ugyanis a programszövegekben egyetlen karaktert sem javítunk, s az előre megadott javítási útmutatótól semmiben nem térünk el. 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 (sőt ezért plusz pont sem jár). Ha a programnak valamilyen állományra van szüksége, akkor azt mindig az aktuális könyvtárba kell rakni. Az állomány neve minden esetben rögzített.

1. feladat: Barátságok (20 pont)

Egy pszichológiai vizsgálatban azt jegyezték fel, hogy egy N tanulóból álló csoportban ki kit tart a legszimpatikusabbnak.

A „barat.be” állomány első sorában a tanulók száma ($2 \leq N \leq 1000$), a további N sorban a tanulók neve és a nekik legszimpatikusabb tanuló neve, egyetlen szóközzel elválasztva. A nevek szóközt nem tartalmazhatnak, hosszuk legfeljebb 20.

Készíts programot (Projekt név: barát (barat.cb; main.cpp)), amely a képernyőre és a „barat.ki” állományba írja (soronként) az alábbiakat:

A. Az első sorba a senkinek sem szimpatikus tanulók nevét, egy-egy szóközzel elválasztva.

B. A második sorba azokat a párokat, amelyek tagjai egymásnak a legszimpatikusabbak, de senki másnak nem szimpatikus egyik sem. A párok tagjai közé kötőjelet (–), a párok közé pedig szóközt kell írni.

C. A harmadik sorba a legkedveltebb tanuló nevét (akit a legtöbbnek adtak meg legszimpatikusabbnak), ha több van, akkor mindegyiket, egy-egy szóközzel elválasztva.

D. A negyedik sorba azt a legnagyobb számot, ahány csoportra lehet osztani a tanulókat úgy, hogy minden tanuló ugyanazon csoportba kerüljön, mint amelyikben a neki legszimpatikusabb van.

Példa:

barat.be:	barat.ki:
5	B
A C	D–E
B A	A
C A	2
D E	
E D	

2. feladat: Hálózati felügyelőprogram (28 pont)

Egy számítógép-hálózaton a szerver nyomon követi a felhasználók be- és kijelentkezését, melynek alapján naponta többféle jellemzőt kiszámíthatunk. Minden felhasználó munkája végén köteles kijelentkezni, valamint egyszerre csak egyetlen gépen jelentkezhet be. Ha az első adata egy kijelentkezés, akkor azt úgy kell érteni, hogy még az előző napon jelentkezett be, s ha nem jelentkezett ki, az azt jelenti, hogy még a következő napon is folytatja a munkáját.

Készíts programot (Projekt név: szerver (szerver.cb; main.cpp)), amely egy napi be- és kijelentkezési adatok alapján megadja a rendszer egy napi statisztikáját.

A „szerver.be” állomány minden sorában egy-egy be- vagy kijelentkezés adatai vannak. A sor első két karaktere a BE vagy a KI szó, majd ezt követi egy-egy szóközzel elválasztva a felhasználó azonosítója (legfeljebb 6 karakter), a művelet óra és perc adata.

Az állomány legfeljebb 3200 sort tartalmazhat, s legfeljebb 1500 felhasználói azonosítót adtak ki.

A „szerver.ki” állomány első sorába azon legkevesebb intervallumok számát kell írni, amely intervallum minden percében használta a rendszert legalább 1 felhasználó, s tőle egy-egy szóközzel elválasztva az intervallumok kezdetét és végét.

A második sorba azon intervallumok számát kell írni, amelyekben a rendszert a legtöbb felhasználó használta, s tőle egy-egy szóközzel elválasztva az intervallumok kezdetét és végét.

A harmadik sorba a legtöbb összesített rendszeridőt használó felhasználó azonosítóját kell írni. Ha több ilyen van, akkor mindet meg kell adni, egy-egy szóközzel elválasztva.

A negyedik sorba a tiltott tevékenységet végző felhasználók adatait (művelet, azonosító és idő) kell kiírni (bejelentkezik úgy, hogy előtte nem jelentkezett ki, illetve kétszer próbál kijelentkezni), egy-egy szóközzel elválasztva! A hibás műveleteket a szerver visszatüskésíti, ezek a többi részfeladat megoldásában nem játszanak szerepet.

Példa:

szerver.be	szerver.ki
BE ALFA 3 15	2 0 0 5 30 6 30 11 45
KI BETA 4 50	2 3 15 4 50 6 35 6 55
KI ALFA 5 30	ALFA
BE GAMMA 6 30	KI GAMMA 7 55
BE ALFA 6 35	
KI GAMMA 6 55	
KI GAMMA 7 55	
KI ALFA 11 45	

3. feladat: Katonák (28 pont)

Terepen $N \times N$ -es négyzetrácsos elrendezésben N katona helyezkedik el, különböző helyeken. A parancsnok olyan elrendezésbe kívánja parancsolni a katonákat, hogy minden sorban és minden oszlopban legyen katona. Egy időpontban egyszerre csak egy katona léphet a négy szomszédos mező valamelyikére, és minden egyes lépést egy időegység alatt hajt végre a katona. Az a cél, hogy a legrövidebb időn belül elérjenek egy kívánt elhelyezkedést. A parancsnok négy adatot: $x y i h$, tartalmazó parancsot adhat ki a katonáknak annak érdekében, hogy a kívánt elrendezést elérje, ahol

$x y$: a mozgatandó katona koordinátái

i : a mozgás iránya, ami lehet

L : lefelé, y-irányban csökkenően

F : felfelé, y-irányban növekvően

B : balra, x-irányban csökkenően

J : jobbra, x-irányban növekvően

h : a lépések száma a megadott irányban

Tehát minden egyes : $x y i h$ parancs végrehajtása h ideig tart.

Csak olyan parancsot lehet kiadni, amely végrehajtása során nem kerül két katona ugyanabba a pozícióba, és a mozgó katona nem is léphet át másik katonán.

Írj programot (Projekt név: katonak (katonak.cbp; main.cpp)), amely megoldja az alábbi 3 részfeladatot:

A. A legkisebb idő, amely a feladat megoldásához kell.

B. A parancsok sorozata, amely a megoldást adja.

C. A katonák lehetséges elhelyezkedése a B-beli parancsok alapján, ami a legrövidebb idő alatt elérhető.

A „katonak.be” állomány első sorában a katonák ($2 \leq N \leq 200$) száma van. A következő N sor a katonák kezdeti elhelyezkedését adja meg, egy sorban egy katona x, y koordinátái vannak egy szóközzel elválasztva.

A „katonak.ki” állomány első sora az A. részfeladat megoldását, azaz a legrövidebb időt tartalmazza. A második sorba a parancsok P számát kell írni, amivel elérhető, hogy minden sorban és minden oszlopban legyen katona. A harmadik sortól P sorban kell megadni a parancsokat. Az állomány utolsó sorába a C. részfeladat megoldását adó N számot kell írni egy-egy szóközzel elválasztva. Az i -edik szám azon katona y -koordinátája legyen, amelynek x koordinátája i a kívánt elhelyezkedésben.

Példa:

katonak.be	katonak.ki
6	8
1 2	6
2 4	4 3 J 2
3 4	3 2 J 2
3 5	3 4 J 1
4 3	1 2 L 1
3 2	3 5 F 1
	2 4 F 1
	1 5 6 4 2 3

4. feladat: Fogadás (24 pont)

Byteland ország követsége fogadást rendezett. A fogadásra N számú ország nagykövétét hívták meg. A vendégek különböző időpontokban érkeztek, és minden vendég azonos ideig volt jelen a fogadáson. A házigazda feljegyezte, hogy ki-kivel találkozott a fogadás során. Az előbb érkezett A vendég találkozott a később érkezett B vendéggel, ha A később távozott, mint B érkezett. Azt is tudjuk, hogy az első vendég érkezésétől az utolsó távozásáig minden időpontban jelen volt legalább egy vendég.

A feladat annak kiderítése, hogy a vendégek milyen sorrendben érkezhettek a fogadásra.

Írj programot (Projekt név: fogad (fogad.cbp; main.cpp)), amely kiszámítja a vendégek egy lehetséges érkezési sorrendjét (nem az érkezési időpontokat).

A „fogad.be” állomány első sora a vendégek ($2 < N \leq 200$) számát tartalmazza. A második sorban a fogadás alatt találkozott vendégpárok száma áll ($1 \leq M \leq 10000$). A következő M sor mindegyike egy $X Y$ számpárt ($1 \leq X, Y \leq N$) tartalmaz egy szóközzel elválasztva, ami azt jelenti, hogy az X és az Y vendég találkozott a fogadás során.

A „fogad.ki” állomány első és egyetlen sorába N számot kell írni egy-egy szóközzel elválasztva, ami a vendégek egy lehetséges érkezési sorrendje.

Példa:

fogad.be	fogad.ki
4	1 2 4 3 (3 4 2 1 is jó megoldás)
3	
1 2	
2 4	
3 4	