

Egy 32 sakköző között zajló versenyt a következők szerint rendeztek meg a szervezők. Az első fordulóban csoportokat hoztak létre, majd a csoportokon belül körmérkőzéseket játszottak egymással a játékosok. A versenyzőknek kezdetben 100 pontjuk volt. A játszmák győztese minden mérkőzés után megkapta az ellenfél pontjainak 20-ad részét, és pontszámát fölfelé egészre kerekítették. A vesztes pontjai nem változtak. Döntetlen eredmény esetén a játszmát újra kellett játszani.

Az első forduló után kialakult pontszámok és a magyar versenyzők nevei megtalálhatók a `sakk.txt`, honlapunkról letölthető szöveges állományban. A fájl minden sora egy versenyző adatait tartalmazza, a pontszám, a vezeték és a keresztnév vagy keresztnévek egymástól egy-egy szóközzel vannak elválasztva. A fájl csak ékezetek nélküli karaktereket tartalmaz, nincsenek benne fölösleges sorok vagy szóközők, a versenyzők neve előtt nincs előtag (pl. Dr. vagy ifj.). Egy rövid részlet a fájlból:

345 Kovacs Andras Gyorgy  
378 Kiss-Kerekes Endre

A második fordulóban egyenes kieséssel folytatták a versenyt a játékosok. A sakközőkat pontszám szerint csökkenő sorrendbe állították, és sorrendben az első játékos játszott a legutolsóval, a második az utolsó előttivel, és így tovább. Az azonos pontszámú versenyzőket nevük szerint ABC rendbe sorolták be. Az első körben az így kialakított párosok egy játszmát játszottak egymással. A győztesek a fentiek alapján növelték pontjaikat, a vesztesek kiestek a további játékból. Ezután ismét elvégezték a sorba rendezést, és a második körben a még játékban lévő játékosok játszottak egymással az előbbi módon. Minden további körben ugyanígy történt a párosok kiválasztása és a pontok számítása, egészen a döntő játszmáig.

Készítsünk programot `i397` néven, amely megoldja a következő feladatokat. Az egyes feladatok megoldása előtt írjuk ki a feladat sorszámát a képernyőre, pl. „3. feladat:”. A nevek kiírásakor minden feladatban a keresztnévek helyett csak azok első betűjét és egy pontot jelenítsünk meg, figyelve a magyar nyelvben használt kettős betűkre.

1. Olvassuk be a `sakk.txt` állományt, és tároljuk el az adatokat.
2. Írjuk ki a képernyőre a versenyzők adatait a beolvasás szerinti sorrendben a következő formában:

Kovacs A. Gy. (345)  
Kiss-Kerekes E. (378)

3. A második fordulóban a párosok közötti játszmákat szimuláljuk úgy, hogy a játékosnak az aktuális pontszámával arányos esélye legyen a győzelemre. Legyen az egyik versenyző jelenlegi pontja  $A$ , a másik versenyzőé  $B$ . Generáljunk egy véletlen egész számot 1 és  $A + B$  között. Az első versenyző a győztes, ha a véletlenszám 1 és  $A$  közötti, és a másik versenyző a győztes, ha  $A + 1$  és  $B$  közötti. Készítsünk `ElsoNyer` néven logikai értéket visszaadó függvényt, amelynek két egész paramétere két játékos pontszáma. A függvény szimulálja a játszmát véletlenszám generálásával, majd adjon vissza igaz értéket, ha az első játékos nyert, és hamis értéket, ha a második.
4. Játsszuk le a második forduló első körét az előbb elkészített függvény segítségével. Írjuk a képernyőre a sakközők kör végén érvényes adatait. Egy-egy sorban az egymás ellen játszó szerepeljenek a 2. feladat szerinti formában, három vonallal (`--`) elválasztva.
5. Adjuk meg, hogy melyik versenyző és ki ellen szerezte a legtöbb pontot a második forduló első körében. A választ foglaljuk mondatba, pl. „Kiss-Kerekes E. (396) szerezte a legtöbb pontot (18) Kovacs A. Gy. (345) ellen”.
6. Játsszuk végig a második forduló második körét is, és írjuk ki a `kor2.txt` állományba a kör nyolc játszmája után érvényes eredményeket és a párosítást a 4. feladatban leírtaknak megfelelően.
7. Soroljuk föl azokat a második körben kiesett sakközőkat, akik több pontot szereztek, mint a legkisebb pontszámot szerző továbbjutó, és írjuk a képernyőre nevük rövidített alakját egymástól vesszővel elválasztva.
8. Játsszuk végig a további köröket is, és írjuk a képernyőre a döntőben részt vevő két sakköző adatait az utolsó kör előtt, illetve a kör után.

Beküldendő egy tömörített `i397.zip` állományban a program forráskódja, valamint a program rövid dokumentációja, amely a fentiekén túl megadja, hogy a forrás mely fejlesztői környezetben fordítható.