

## Regionális forduló

2015. november 28.

### 9-10. osztályosok feladata (I. kategória)

#### Feladat

A ProgTanKör több informatikai jellegű versenyt szervez. Ezen versenyek adatainak feldolgozásában és a versenyszervezési feladatokban kérjük a segítségeteket!

Olyan programot kell készítenek, amely a szervezők számára fontos adatokat teszi könnyen elérhetővé, segítve őket ezzel a szervezési feladatok áttekintésében.

#### Versenyegek

##### Programozói verseny: (kódja:1)

Típusa: csapatverseny (3 fős csapatok jelentkezhetnek)

Kategóriái: I. kategória (legfeljebb 10. osztályosok); II. kategória (van 10. osztályosnál idősebb)

Döntőbe jutók száma: Az I. kategóriából 2 csapat, a II. kategóriából 3 csapat.

Nevezési határideje: október 15.

##### Robotika verseny: (kódja: 4)

Típusa: egyéni verseny

Kategóriái: nincs kategória

Döntőbe jutók száma: 5 fő

Nevezési határideje: október 10.

#### Nevezések

Mindkét versenyre elektronikusan lehet jelentkezni. A ProgTanKör szervere egy szövegfájlba menti a beérkező nevezéseket. A fájl neve: **nevezes.txt**.

A nevezéseket tartalmazó fájl minden sora azonos szerkezetű, a verseny kódját, a csapat nevét, a nevezés idejét (4 karakter: hhhh alakban), az iskola nevét, a megye nevét, a diák/diákok nevét, évfolyamát és a felkészítő tanár nevét tartalmazza pontosvesszővel elválasztva. A csapatversenyen és az egyéni versenyen egyaránt meg kell adni egy „csapatnevet”, ami alapján a versenyző(ke)t azonosítani lehet. Minden csapatnak (egyéni versenyzőnek) pontosan 1 felkészítő tanára van.

A **nevezes.txt** két sora lehet:

*1;ugyesek;1010;Ugyesek Iskolaja;Pest megye;Kiss Vilmos;11;Nagy Arpad;11;Kiss Virag;12;Nagy Lajos  
1;programozok;1012;Programozok Iskolaja;Vas megye;Szabo Reka;10;Kovacs Erika;10;Nagy Eva;10;Kiss Janos*

A szövegfájlban nem szerepel ékezetes betű, a számok után sehol nincs pont.

Feltételezhetitek, hogy legfeljebb 50 jelentkezés van, egy versenyre egy csapat (egyéni versenyző) legfeljebb egy nevezést küld, a csapatnevek **mind különbözőek**, egy megyében nincs két azonos nevű iskola, és az **egy iskolából** jelentkezők között nincs két azonos nevű felkészítő tanár.

A nevezes.txt fájl az összes verseny-jelentkezést tartalmazza. A szerver minden üzenetet helyesen

dolgoz fel.

## A nevezések feldolgozása

A *nevezes.txt* fájl lezárása után a beérkező nevezéseket fel kell dolgozni.

A **programozói verseny** regionális fordulójának több helyszíne van. A jelentkezők adatainak ismeretében ki kell választani a regionális fordulót lebonyolító iskolát. Ha egy megyéből valamelyik versenyre érkezett nevezés, akkor abban a megyében lesz regionális forduló helyszín. A megye azon iskolája lesz a regionális forduló rendezője, ahonnan a legtöbb csapat nevezett. Ha több ilyen is van, közülük bármelyik kiválasztható.

A **programozói versenyre nevező** csapatok iskoláját értesíteni kell a regionális forduló helyszínéről.

A szerver automatikusan generál körlevelet a jelentkezőknek, nektek csak a megfelelő adatokat kell megadni: a versenyre jelentkező csapat neve, a versenyzők iskolájának neve, a regionális forduló iskolája; a megye neve.

Ha a programozói versenyre nincs érvényes jelentkezés, akkor is készülhet egy üres *regio.txt* fájl.

A robotika verseny első fordulójának helyszínét **nem nektek kell** kiválasztani.

A **regio.txt** két sora lehet:

*ugyesek;Ugyesek Iskolaja;Ugyesek Iskolaja;Pest megye  
programozok;Programozok Iskolaja;Programozok Gimnaziuma;Vas megye*

A *nevezes.txt* fájl az utolsó nevezési határidőt követően készült el, ezért minden addig beérkező nevezést tartalmaz. Előfordulhat, hogy a benne szereplő nevezések valamelyike az adott verseny esetében már érvénytelen, mert határidő után érkezett, vagy nem megfelelő az adott versenytípusra nevezők száma. Az ilyen esetben egy válaszüzenetet kell küldeni, amelyben a nevezés érvénytelenségének okát adjuk meg. A szerver generál ilyen körlevelet, nektek csak egy szövegfájlt kell készítenetek, amelyben az összes problémás nevezés esetében megadjátok az adatokat (verseny kódja, a csapat neve, iskola neve, a probléma oka – ha mindkét hiba előfordul, elég az egyiket megjelölni). A lehetséges okok, amelyeket a szerver a körlevélbe tud illeszteni: *hatarido, harom versenyzo kell, egy versenyzo kell*. Feltételezhetitek, hogy más típusú hiba nem fordul elő, a szerver formailag ellenőrzi az adatokat, és nem fogadja el az előzőtől eltérő hibákat tartalmazó nevezést.

Az **ervtelen.txt** két sora lehet:

*1;kesok;Elkesett Iskola;hatarido  
1;egyke;Sok Iskola;harom versenyzo kell*

Az érvénytelen nevezést leadó csapatok **ne** szerepeljenek a *regio.txt*-ben!

## A regionális fordulón elért pontszámok

A regionális fordulót követően a versenyhelyszínekről is üzenetként érkeznek az elért pontszámok, amelyekről a nevezésekhez hasonlóan a szerver egy szövegfájlt készít.

A fájl első két sora rendre programozó- és a robotika versenyeken résztvevő csapatok számát, a további sorai a résztvevő csapatok nevét és a pontszámát tartalmazzák a versenyek sorrendjében egymás után.

Az **eredmeny.txt** lehet:

```
7
6
ugyesek;100
programozok;101
...
```

Ahol a soron következő 5 sor még a programozói versenyre, a következő 6 sor pedig a robotika versenyre vonatkozik. A sorokban először a csapat neve, aztán pontosvesszővel elválasztva a regionális (első) fordulóban elért pontszáma szerepel.

## A beérkező pontszámok feldolgoása

A regionális versenyhelyszínekről érkező üzeneteket is fel kell dolgozni. A pontszámok alapján el kell dönteni, hogy mely csapatok jutnak egy adott verseny döntőjébe. A döntőbe jutó csapatok nevét versenyenként felsorolva a *donto.txt* fájlba kell írni. Egy versenyhez/versenyfajtaához tartozó adatok kerüljenek egy sorba! A sorok kezdődjenek a verseny kódjával, és folytatódjanak a csapatok nevével csökkenő pontszám szerint rendezve. A programozói verseny sorában a csapat neve után fel kell tüntetni a kategóriát is.

Ha holtverseny miatt nem lehet eldönteni, hogy melyik csapatok jussanak döntőbe, az adott verseny sorába azt kell kiírni: „Nem eldönthető.” (A versenybizottság dönti el.)

Előfordulhat, hogy valamelyik versenyre senki nem nevezett, vagy volt ugyan nevezés, de az első fordulóban mégsem vett részt senki. Az ilyen esetben nem kell az adott versenyre vonatkozó sort a *donto.txt* fájlban szerepeltetni. Az is lehetséges, hogy valamelyik versenyen az első fordulóban kevesebb csapat vesz részt, mint amennyi a döntőbe juthatna.

A döntőbe csak olyan csapat juthat, amelyik a regionális fordulóban pontot szerzett.

A **donto.txt** lehet:

```
1; ugyesek II; legjobbak II; nagyok II; programozok I; kezdok I
2; mobilosok; mivagyunk
3; alkotok; webesek; három
4; egyedul; robotos; robik; arduino; lego
```

## Statisztika készítése

Az összes adat feldolgoása után a képernyőn egy statisztikát kell megjeleníteni, amely tartalmazza

- a regionális fordulóban az egyes versenyekre (kategóriánként) nevező csapatok számát,
- a döntőbe jutó csapatok nevét és iskoláját versenyenként az regionális (első) fordulón elért pontszámaik szerinti csökkenő sorrendben,
- és azt, hogy a döntősök felkészítő tanárai hány versenyzőt készítettek fel. Minden versenyt

figyelembe véve az érvényesen jelentkező diákokat kell megszámolni; mindenkit annyiszor, ahány versenyre nevezett. (Ha valamelyik verseny döntőjébe jutó 3 fős csapatnál szereplő tanár neve egy másik ugyanazon iskolából jelentkező, nem döntős csapatnál és egy egyéni versenyzőnél is megjelenik, akkor ő 7 diákot készített fel.)

Nem szükséges mindhárom adatcsoportot egyszerre megjeleníteni.

## Beadandó

A program **forráskódja** (a programozási környezettől függően a forráskód több fájl is lehet, esetleg mappák is tartozhatnak hozzá), és ha erre a programozási környezet lehetőséget ad, a **lefordított fájl** (verseny.exe).

A program által használt külső fájlokat mindig az aktuális/alapértelmezett könyvtárba kell elhelyezni. (A program ne tartalmazzon abszolút elérési útvonalat!)

## Kódolási alapelvek

A forráskód minőségét is értékeljük.

Irányelvek, szempontok:

- Egységes kódolási szabályok az azonosítókra:
  - a változók egységes elnevezése (kis- és nagybetűk vagy más speciális karakterek használata),
  - az osztályok egységes elnevezése (objektum-orientált programnyelv esetén),
  - a függvények és eljárások tartalomra utaló elnevezése,
  - a programkód egységes strukturáltsága, tagoltsága.
- A kód minősége (könnyen – emberek számára – érhető illetve karbantartható kód):
  - áttekinthető, lehetőség szerint rövid eljárások, függvények, fájlok,
  - beszédes, tömör elnevezésű azonosítók,
  - objektum-orientált nyelveknél a globális változók mellőzése.
- Kommentezés:
  - A kommentezés elsődleges célja, hogy a programban a miért? kérdésre adjon választ. (A mit? kérdésre az azonosítók megfelelő elnevezése és a megfelelően strukturált kód, a hogyan? kérdésre pedig az áttekinthető forráskód ad választ.)
  - A túlzásba vitt kommentezés csökkenti az áttekinthetőséget, a túl kevés komment nehezíti a megértést.
  - Elvárás a változók, osztályok, függvények és eljárások szerepének rövid, értelem-szerű kommentezése.

Elérhető pontszám: 100 pont (Ebből a helyes dokumentálás – kommentezés – 15 pont)

Jó munkát kíván a versenybizottság!