



A 2011/2012. tanévi
Országos Középiskolai Tanulmányi Verseny
második fordulójának feladatlapja

INFORMATIKÁBÓL
II. (programozás) kategóriában

Munkaidő: 300 perc

Elérhető pontszám: 75 pont (+ az első fordulóban szerzett pontszám 25 %-ának egészre kerekített értéke)

A VERSENYZŐ ADATAI

A versenyző neve: évf.: oszt.:

Az iskola neve:

Az iskola címe: irsz. város

..... utca hsz.

Megye:

A felkészítő tanár(ok) neve:

Középiskolai tanulmányait a 12. / 13. évfolyamon fejezi be.*

**A megfelelő szám bekarikázandó.*

	RVB-pontszám
1. feladat	
2. feladat	
3. feladat	
4. feladat	
5. feladat	
1. fordulóból	
Összesen:	
Javította:	

A 2011/2012 tanévi Országos Középiskolai Tanulmányi Verseny második fordulójának feladatai

II. (programozás) kategória

Kedves Versenyző! A megoldások értékelésénél csak a **programok futási eredményeit** vesszük tekintetbe. Ezért igen fontos a **specifikáció pontos betartása**. Ha például 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 pontozóknak a programszövegekben egyetlen karaktert sem szabad javítaniuk, s az előre megadott javítási útmutatótól semmiben nem térhetnek 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 állományok neve minden esetben rögzített. **Csak olyan programokat értékelünk, amelyek 1 percen belül adnak végeredményt!**

1. feladat: Előrejelzés (15 pont)

Ismerjük N településre az M napos időjárás előrejelzést, ezek alapján keressük településeket.

Készíts programot (`jelzes.pas`, ...), amely megad négyféle értelmezés szerint egy-egy települést:

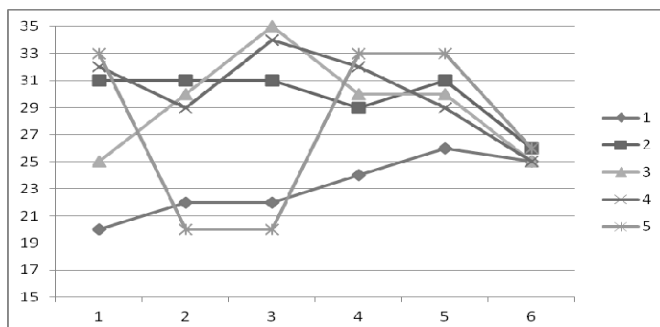
- A. azt a települést, amelyre a legnagyobb és a legkisebb előrejelzés eltérése a legnagyobb;
- B. az a települést, amelyre van olyan település, ahol minden nap nála hidegebb várható;
- C. egy olyan települést, amelyben a leghosszabb időszakon belül várható folyamatosan K fok feletti hőmérséklet;
- D. azt a települést, amelyre a legtöbb napon fordul elő, hogy a várt hőmérséklet nagyobb minden más, arra a napra előrejelzett hőmérsékletnél.

A `jelzes.be` szöveges állomány első sorában a települések száma ($1 \leq N \leq 1000$), a napok száma ($1 \leq M \leq 1000$) és a hőmérséklet korlát van ($20 \leq K \leq 50$), egy-egy szóközzel elválasztva. A következő N sor mindegyikében M egész szám van, egy-egy szóközzel elválasztva: az i-edik település j-edik napra várt hőmérséklete.

A `jelzes.ki` szöveges állomány négy sorába egy-egy település sorszámát kell írni! Az első sorba az A, a másodikba a B, a harmadikba a C, a negyedikbe pedig a D értelmezés szerinti települést. Ha több megoldás van, bármelyik megadható. Ha nincs megoldás (B,C,D részfeladatban), abba a sorba -1-et kell írni!

Példa:

<code>jelzes.be</code>	<code>jelzes.ki</code>
5 6 30	5
20 22 22 24 26 25	2
31 31 31 29 31 26	2
25 30 35 30 30 25	5
32 29 34 32 29 25	
33 20 20 33 33 26	



2. feladat: Verseny (15 pont)

Egy kieséses versenyben ismerjük a csapatok mérkőzéseit: ki kit győzött le.

Írj programot (`verseny.pas, ...`), amely megadja:

A. azt a csapatot, amely a kiesettek közül a legtöbbször győzött;

B. a legtöbb csapatot közvetlenül vagy közvetve legyőző csapatot;

C. a következő mérkőzést játszó két csapatot, amely két olyan versenyben levő csapat legyen, amely eddig közvetve vagy közvetlenül a lehető legkevesebb csapatot győzte le!

A `verseny.be` szöveges állomány első sorában a csapatok száma ($2 \leq N \leq 1000$) és a mérkőzések száma van ($1 \leq M < N$), egy szóközzel elválasztva. A következő M sor mindegyikében két csapat I és J sorszáma van ($1 \leq I \neq J \leq N$), ami azt jelenti, hogy az I -edik csapat legyőzte a J -edik csapatot.

A `verseny.ki` szöveges állomány első sorába a kiesettek közül legtöbb győzelmet szerző csapat sorszámát kell írni (-1-et, ha nincs ilyen csapat)! A második sorba azt a csapatot, amely a legtöbb más csapatot győzte le közvetve vagy közvetlenül, a harmadik sorba a szabály szerint a következő mérkőzést játszó két csapat sorszámát, egy szóközzel elválasztva! Ha több megoldás van, bármelyik kiírható, a harmadik sorban az egyetlen -1 szám álljon, ha nincs megoldás!

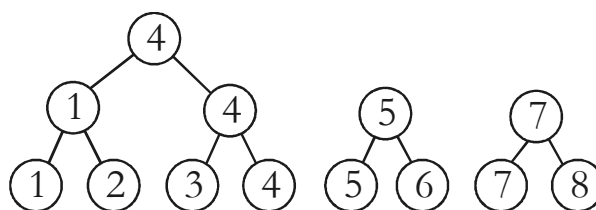
Példa:

`verseny.be`

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

`verseny.ki`

```
1
4
5 7
```



3. feladat: Üvegválogatás (15 pont)

Egy üzletben válogatás nélkül gyűjtötték össze az üres üvegeket N ládában. Mivel az üvegfajták száma is N , ezért az azonos fajtájú üvegeket egy-egy ládában el lehet tárolni. Ezért szét akarják válogatni az üvegeket, hogy minden ládában csak azonos fajtájú legyen, de úgy, hogy a lehető legkevesebb üveget kelljen átrakni más ládába.

Készíts programot (`valogat.pas, ...`), amely kiszámítja, hogy legkevesebb hány darab üveget kell másik ládába átrakni, és megadja, hogy ehhez az egyes fajtákat melyik ládába kell gyűjteni!

A `valogat.be` szöveges állomány első sorában egy egész szám, a ládák N száma ($1 \leq N \leq 8$) van. A következő N sor mindegyike pontosan N nemnegatív egész számot tartalmaz egy-egy szóközzel elválasztva. Az $I+1$ -edik sorban a J -edik szám az I -edik ládában lévő J -fajta üvegek száma. Minden szám értéke legfeljebb 5000. A ládákat és az üvegfajtákat is az $1, \dots, N$ számokkal azonosítjuk.

A `valogat.ki` szöveges állomány első sora egy egész számot tartalmazzon, a kívánt szétválogatáshoz szükséges átrakások minimális számát! A második sor pontosan N egész számot tartalmazzon (egy-egy szóközzel elválasztva): az I -edik szám annak az üvegfajtának a sorszáma legyen, amelyet az I -edik láda rakunk!

Példa:

`valogat.be`

```
3
15 30 8
55 80 10
50 60 12
```

`valogat.ki`

```
182
3 2 1
```

4. feladat: Párosítás (15 pont)

Egy raktárból N boltba kell kiszállítani ládába csomagolt árut. Minden boltba pontosan két ládát kell vinni. A ládák az előkészítés időrendi sorrendjében egymás mellett egy sorban vannak, mindegyikre ráragasztva annak a boltnak a sorszáma, ahova szállítani kell. A raktárosnak át kell rendezni a ládák sorrendjét, hogy az egy boltba kerülő ládák egymás mellett legyenek. Az átrendezés során egy lépésben két láda helyét cserélheti meg.

Készíts programot (`parosit.pas, ...`), amely kiszámítja, hogy legkevesebb hány cserével lehet kialakítani a kívánt sorrendet!

A `parosit.be` szöveges állomány első sorában a boltok N ($2 \leq N \leq 20000$) száma van. A második sor pontosan $2N$ pozitív egész számot tartalmaz egy-egy szóközzel elválasztva, az $1, \dots, N$ számok mindegyike pontosan kétszer fordul elő a sorban.

A `parosit.ki` szöveges állomány első és egyetlen sora egy egész számot tartalmazzon, azt a legkisebb M számot, amire a bemenetben megadott ládásor M számú cserével a kívánt sorrendbe rakható!

Példa:

<code>parosit.be</code>	<code>parosit.ki</code>	Magyarázat:
4	3	1 3 2 1 3 4 4 2
1 3 2 1 3 4 4 2		1 1 2 3 3 4 4 2
		1 1 2 3 3 2 4 4
		1 1 2 2 3 3 4 4

5. feladat: Számjáték (15 pont)

Tekintsük a következő egyszemélyes játékot: A játék kezdetén egy sorban leraknak N darab pozitív egész számot. A játékos legfeljebb L lépést tehet. Egy lépésben a még a táblán lévő számsorból H darab egymás melletti számot levehet, a levett számok a pontszámához adódnak. A levett számok helye üresen marad, és lépés során a szomszédos számok között nem lehet üres hely. A játékosnak az a célja, hogy a lehető legtöbb pontot szerezze.

Készíts programot (`szamos.pas, ...`), amely kiszámítja, hogy legjobb esetben hány pontot szerezhet a játékos!

A `szamos.be` szöveges állomány első sorában három egész szám van, a kezdeti számsorozat számainak N száma, a lépések maximális L száma és az egyszerre levehető számon H darabszáma ($1 \leq N \leq 3000$, $1 \leq L \leq 1000$, $2 \leq H \leq N$). A második sor tartalmazza a kezdeti játékállást, azaz N pozitív egész számot egy-egy szóközzel elválasztva. Minden szám értéke legfeljebb 5000.

A `szamos.ki` szöveges állomány első sora egy egész számot tartalmazzon, a játékban elérhető lehető legtöbb pont értékét! A második sor egy olyan lépéssort tartalmazzon, amelylyel a maximális pontszám elérhető! Egy lépést a lépésben levett számsor első elemének sorszáma legyen!

Példa:

<code>szamos.be</code>	<code>szamos.ki</code>
8 2 3	32
1 6 8 7 6 2 1 8	2 6

Elérhető összpontszám: 75 pont + 25 pont az 1. fordulóból