



Országos Középiskolai Tanulmányi Verseny, 2004/2005-ös tanév INFORMATIKA, II. (programozói) kategória második fordulójának javítási útmutatója

Kérjük a tisztelt kollégákat, hogy az egységes értékelés érdekében az alábbi eljárást alkalmazzák:

1. Az értékelő gépen hozzák létre a \NT2 könyvtárat.
2. Másolják be a \NT2 könyvtárba az NT2.EXE állományt, amely jelszóval védett önkibontós ARJ állomány (a tesztadatokat és az értékelő programot tartalmazza), és indítsák el az NT2 . EXE -g<jelszó> paranccsal (a jelszót a < jelek nélkül kell beírni). A NT2.EXE állományt és a jelszót mindenkihez időben eljuttatjuk.
3. Minden versenyző számára hozzanak létre egy külön könyvtárat, és ezekbe másolják be, majd fordítsák le a versenyzők programjait (a feladtleírásban szereplő néven).
4. Egy versenyző értékelése:
 - A. Az aktuális könyvtár legyen a versenyző könyvtára.
 - B. Adják ki a \NT2\T3 parancsot, amely lefuttatja a versenyző programjait minden tesztetere. Ha a végrehajtás megszakad, vagy meg kell szakítani, mert letelt a 60 másodperc, akkor ismét a T3 parancsot kell kiadni, mindaddig, amíg az „ÉRTÉKELÉS BEFEJEZŐDÖTT” üzenet meg nem jelenik a képernyőn. (A futtató tudja, hogy honnan kell folytatnia.). Ezt követően automatikusan elindul a megoldásokat értékelő program, amely összesítést készít a versenyző könyvtárában EREDMENY.TXT néven, és az eredményt a képernyőre is kiírja.

1. feladat: Fák (14 pont)

Egy fasorba N fát ültettek balról jobbra, egy vonalba. Mindegyik fának ismerjük a magasságát és a bal szélső fáról vett távolságát. Ha egy fát kivágunk, akkor az a jobboldali szomszédja felé dől, s amelyik szomszédját érinti, az is kidől.

Írj programot (*fak.pas* vagy *fak.c*), amely megadja, hogy minimum hány fát kell kivágunk ahhoz, hogy az összes fa kidőljön, s melyik fa kivágása okozza a legtöbb fa kidőlését!

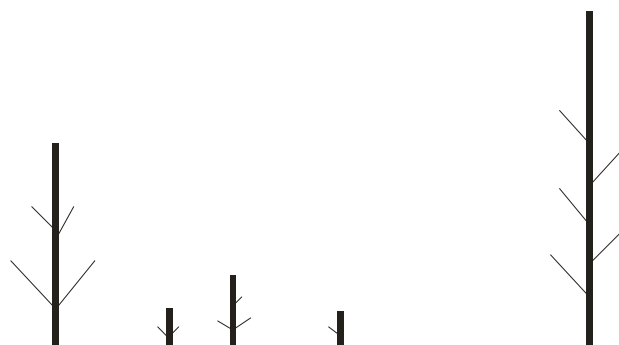
A *fak.be* állomány első sorában a fák N száma van ($1 \leq N \leq 1000$). A következő N sor mindegyike két számot, egy-egy fa leírását tartalmazza: a bal szélső fától vett T távolságát ($1 \leq T \leq 1000$) és a fa M magasságát ($1 \leq M \leq 100$). A fákat balról jobbra haladva adjuk meg.

A *fak.ki* állomány első sorába a minimálisan kivágandó fák számát kell írni, a második sorába pedig annak a kivágandó fának a sorszámát, amely kivágása esetén a legtöbb fa fog kidőlni.

Példa:

```
fak.be
5
0 6
3 1
5 2
8 1
15 10
```

```
fak.ki
3
1
```

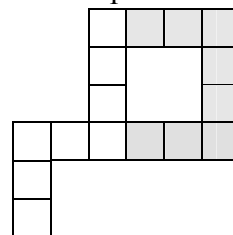


Értékelés:

Az első fa rádől mindegyikre	1+1 pont
Minden fa a következőre dől	1+1 pont
Két fát kell kivágni, minden fa a következőre dől	1+1 pont
Több fát kell kivágni, mindegyik a következőre dől	1+1 pont
Minden fát ki kell vágni	1+1 pont
Véletlen nagy teszt	1+1 pont
Véletlen nagy teszt	1+1 pont

2. feladat: Pince (16 pont)

Pincét fűrnak: az E,J,B utasítások hatására egy egységnyi részt fűrnak, jobbra, illetve balra fordulnak 90 fokkal. A ()-ben levő részek a főágról leágazást jelentenek, előbb mindig a főágot fűrjük végig, s utána kezdenek az elágazásokhoz. Szabályok: két ág nem érhet össze, sőt a sarkával sem találkozhat. Önmagában minden lépés biztosan olyan, hogy fűrni is kell, azaz teljesen nem mennek bele már kifűrt részbe (pl. nincs olyan sorozat, hogy EJJE).



Írj programot (*pince.pas* vagy *pince.c*), amely megadja hogy a pince helyes-e, s ha nem, akkor melyik lépésben találkoznak össze az ágak!

A *pince.be* állomány egyetlen sorában a pincét leíró karaktersorozat van (legfeljebb 10000 karakter a sor végéig). A pincefűrés közben az X- és az Y-koordináta nem lép ki a [-100..100] intervallumból. A kezdőkoordináta a (0,0).

A *pince.ki* állományba egyetlen számot tartalmazó sort kell írni: 0-t, ha a pince helyes, illetve az I számot, ha a pincét leíró karaktersorozatban az I-edik karakter hatására fűrt szakasznál találkoznak össze az ágak.

Példa:

```
pince.be                pince.ki
EEEJEEB (JEEEEEEEEBE) EEE    19
```

A példában az elágazás ásásának utolsó lépésében ütközünk a főág végébe, szürkítetten szerepel az elágazás.

Értékelés:

Egyenes pince elágazás nélkül	2 pont
Kanyargó pince, oldallal ér össze	2 pont
Kanyargó pince, sarokkal ér össze	2 pont
Elágazás a főág mellett megy (összeér)	2 pont
Elágazás főággal találkozik	2 pont
Elágazáson belül másik elágazás van, és az rossz	2 pont
Sarkos leágazás	2 pont
Sarkos leágazás	2 pont

3. feladat: Osztály (15 pont)

Egy osztályba N tanuló jár. Minden tanuló ismeri néhány osztálytársának telefonszámát.

Írj programot (*osztaly.pas* vagy *osztaly.c*), amely megadja azt a tanulót, akitől egy hír az ismert telefonokon keresztül továbbadva előbb-utóbb az osztály legtöbb tanulója-hoz eljut!

Az *osztaly.be* állomány első sorában a tanulók N száma ($1 \leq N \leq 100$) van. A következő N sor mindegyike egy-egy tanuló által ismert telefonszámú tanulókat ír le, az állomány $i+1$ -edik sorában azoknak a tanulóknak a sorszáma van, akiét az i -edik tanuló ismeri. Mindegyik sorban legfeljebb $N-1$ különböző egész szám van, egy-egy szóközzel elválasztva és 0-val zárva: az ismert telefonszámú tanulók sorszáma.

Az *osztaly.ki* állományba egyetlen sort kell írni, annak a tanulónak a sorszámát, akitől a legtöbb tanulóhoz eljuthat egy hír. Ha több ilyen tanuló van, akkor bármelyik sorszáma kiírható.

Példa:

<i>osztaly.be</i>	<i>osztaly.ki</i>
5	1
3 5 0	
3 4 0	
0	
2 3 0	
2 0	

Értékelés:

Egyetlen 0 befokú van, s tőle mindenhova eljut a hír	2 pont
Egyetlen 0 befokú van, de tőle nem mindenhova jut el a hír, ő a megoldás	2 pont
Egyetlen 0 befokú van, de tőle nem mindenhova jut el a hír, más a megoldás	2 pont
Egyetlen kör van	2 pont
Több független kör van, egyikbeli bármelyik a megoldás	2 pont
Több kör van, az egyikből a többibe is el lehet jutni, abból kell választani	2 pont
Véletlen nagy teszt	3 pont

4. feladat: Vonat (15 pont)

Egy hosszú vasútvonal mentén N város helyezkedik el, minden városnak pontosan egy vasútállomása van a vonalon. Ismerjük a vonalon közlekedő vonatokat. Minden vonat adott i -edik városból indul és adott j -edik városba közlekedik ($i < j$) és közben nem áll meg egyetlen közbülső állomáson sem. Az I . városból indulva, vonattal közlekedve a lehető legtöbb várost szeretnénk meglátogatni.

Írj programot (*vonat.pas* vagy *vonat.c*), amely kiszámítja, hogy az I . városból indulva mennyi a legtöbb meglátogatható város, és meg is ad egy útvonalat, amelyen haladva a legtöbb város meglátogatható!

Az *vonat.be* állomány első sorában két egész szám van, a városok N száma ($1 \leq N \leq 200$) és a járatok M ($1 \leq M \leq 3000$) száma. A további M sor mindegyike két egész számot tartalmaz (egy szóközzel elválasztva), az első szám i , a járat indulási, a második szám j a járat érkezési állomása ($1 \leq i < j \leq N$). Az állomány $i+1$ -edik sora az i -edik járat adatát tartalmazza.

Az *vonat.ki* állományba első sorába egyetlen egész számot kell írni, a legtöbb meglátogatható város K számát, beleértve az 1. induló várost is! A második sor pontosan $K-1$ számot tartalmazzon (egy-egy szóközzel elválasztva), a járatok bemenetbeli sorszámaikat az utazás sorrendjében. Több megoldás esetén bármelyik kiírható.

Példa:*vonat.be*

5 7
 1 2
 1 3
 2 4
 3 5
 2 3
 4 5
 3 4

vonat.ki

5
 1 5 7 6

Értékelés:

Nincs elágazás	1+0 pont
Az úthálózat fa	1+1 pont
Az úthálózat egyszerű rács	1+1 pont
Az úthálózat rács	1+1 pont
Az úthálózat teljes	1+1 pont
Véletlen közepes teszt	1+2 pont
Véletlen nagy teszt	1+2 pont

5. feladat: Játék (15 pont)

Tekintsük a Solitaire játéknak azt a változatát, amelyet **6x6**-os négyzetrácsos táblán lehet játszani. A táblára három fekete korongot helyeznek három különböző mezőre, ez a kezdeti játékkállás. A játék során minden lépésben egy korongot lehet mozgatni az alábbi szabály szerint.

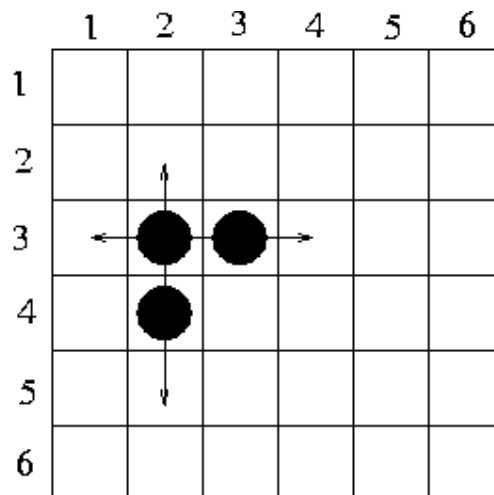
- Csak üres mezőre lehet lépni.
- A négy szomszédos mező valamelyikére lehet lépni, balra, jobbra, felfelé vagy lefelé.
- Ha a lépés irányába eső szomszédos mezőn van korong, akkor azt az egy korongot át lehet lépni.

A a (3,2) mezőn álló korong négy lehetséges lépése: (2,2), (3,1), (5,2), (3,4), mint az ábrán látható.

Írj programot (*jatek.pas* vagy *jatek.c*), amely kiszámítja, hogy adott kezdeti játékkállásból legkevesebb hány lépés végrehajtásával lehet eljutni adott végállásba!

Az *jatek.be* két sort tartalmaz, az első sor a kezdeti játékkállást, a második pedig a végállást írja le. Mindkét sor 6 egész számot tartalmaz, a három korong koordinátáit. Az *i*-edik (*i*=1,2,3), számpár az *i*-edik korong sor, illetve oszlopkoordinátáját jelenti. A sorokat fentről lefelé, az oszlopokat balról jobbra sorszámozzuk 1-től 6-ig. A három korong sorrendje közömbös a végállásban!

Az *jatek.ki* állomány első és egyetlen sora egy egész számot tartalmazzon, azon legkevesebb lépések számát, amennyi lépéssel el lehet jutni a kezdeti játékkállásból a végállásba!



Példa:*jatek.be*

3 2 3 3 4 2

2 3 3 3 3 4

jatek.ki

3

Értékelés:

Nem kell (lehet) átlépni korongot 1 pont

Függőlegesen lehet haladni 1 pont

Vízszintesen lehet haladni 1 pont

Kevés lépés kell 1 pont

A korongok szétszórtan vannak induláskor 2 pont

A korongok szétszórtan vannak a célban 2 pont

Középről indulunk 2 pont

Sarokból középre kell menni 2 pont

Véletlen általános teszt 3 pont

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