



Oktatási Hivatal

A 2010/2011 tanévi Országos Középiskolai Tanulmányi Verseny döntő 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). A kimeneti állományban a feladatlírásban szereplőn túl semmilyen más karakter nem lehet (még láthatatlan karakter sem). **A programod nem írhat semmit a képernyőre és nem olvashat semmit a billentyűzetről!** Ha a programnak valamilyen állományra van szüksége, akkor azt mindig az aktuális könyvtárba kell rakni. Az állományok neve minden esetben rögzített.

A programod úgy kell lefordítanod, hogy a futtatásához semmilyen keretrendszerre ne legyen szükség!

A programod akkor értékeljük, ha 10 másodpercen belül eredményt ad.

1. feladat: Párok (15 pont)

Egy rendezvényre sok vendéget hívtak meg. Minden vendég előre megadta, hogy mikor érkezik, és mikor távozik. A rendezők fényképeken akarják megörökíteni a résztvevőket. A rendezőknek két betartandó kikötése van:

1. Minden képen pontosan két vendég legyen rajta.
2. Minden vendég legfeljebb egy képen szerepelhet.

Természetesen két vendég csak akkor szerepelhet azonos képen, ha van olyan F időpont, amikor mindketten jelen vannak. Egy vendég akkor és csak akkor van jelen az F időpontban, ha az E érkezési és T távozási idejére teljesül, hogy $E \leq F$, és $F < T$.

Készíts programot (`parok.pas`, `parok.c`, ...), amely kiszámítja, hogy legjobb esetben hány fénykép készülhet, és megadja, hogy mely párok szerepeljenek egy képen!

A `parok.be` szöveges állomány első sorában egy egész szám van, a vendégek N száma ($1 \leq N \leq 30\,000$). A vendégeket a sorszámukkal azonosítjuk 1 -től N -ig. A további N sor mindegyikében két egész szám van egy szóközzel elválasztva; egy vendég E érkezési és T távozási ideje ($1 \leq E < T < 20\,000$). Az állomány $i+1$ -edik sorában van az i sorszámú vendég adata.

A `parok.ki` szöveges állomány első sorába a lehetséges legtöbb készíthető fényképek M számát kell írni! A további M sor mindegyikébe egy számpárt kell írni, egy szóközzel elválasztva, azon két vendég sorszámát (tetszőleges sorrendben), akik egy képen szerepelnek. Több megoldás esetén bármelyik megadható.

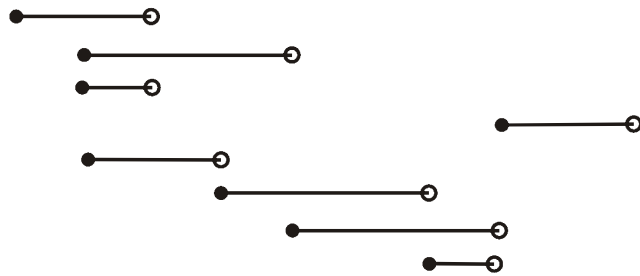
Példa:

parok.be

8
1 3
2 5
2 3
8 10
2 4
4 7
5 8
7 8

parok.be

3
3 1
5 2
6 7



2. feladat: Sudoku (15 pont)

A Sudoku játék 4x4-es változatában egy 4x4-es táblázatot kell kitölteni úgy, hogy minden sorban, minden oszlopban és mind a négy sarok 2x2-es részben az 1,2,3,4 számok mindegyike előforduljon!

Készíts programot (sudoku.pas, sudoku.c, ...), amely adott, nem teljesen kitöltött táblázatra kiszámítja, hogy hány féleképpen lehet kitölteni úgy, hogy szabályos sudoku táblázat legyen !

A sudoku.be szöveges négy sort tartalmaz, minden sorban négy egész szám van egy-egy szóközzel elválasztva. A számok értéke 0,1,2,3,4 lehet. A 0 azt jelenti, hogy a táblázat azon elemét meg kell adni a megoldásban.

A sudoku.ki szöveges állomány első és egyetlen sorába egy egész számot kell írni, ahány féleképpen kitölthető a bemeneti táblázat szabályos sudoku táblázattá!

Példa:

sudoku.be

0 3 0 1
0 1 0 3
3 0 1 0
0 2 0 4

sudoku.ki

2

Segítségként a 2 megoldás:

2	3	4	1
4	1	2	3
3	4	1	2
1	2	3	4

4	3	2	1
2	1	4	3
3	4	1	2
1	2	3	4

3. feladat: Kemence (15 pont)

Cserép korszók kiégetésére szakosodott vállalkozó egy kemencét üzemeltet. Az égetésre érkező korszókat az érkezés sorrendjében kell a kemencében kiégetni. Egy menetben legfeljebb **K** korszó rakható a kemencébe. Minden korszóra adott a minimális és maximális égetési idő percben kifejezve. Továbbá, minden korszóra adott egy **H** határidő, ami azt jelenti, hogy a munka megkezdésétől számítva, a **H** időpontig el kell készülnie a kiégetésének. Figyelembe kell venni, hogy egy menet előkészítése 1 percet vesz igénybe!

Készíts programot (kemence.pas, kemence.c, ...), amely kiszámítja, hogy a követelmények betartásával legkevesebb mennyi idő alatt lehet kiégetni az összes korszót és meg is ad egy helyes beosztást!

A kemence.be szöveges állomány első sorában két egész szám van, a korszók N száma ($1 \leq N \leq 10000$), és a kemence K kapacitása ($1 \leq K \leq 100$). A következő N sor mindegyike három egész számot tartalmaz egy-egy szóközzel elválasztva. Az első szám *min*, a minimális, a második szám *max* a maximális égetési idő (percben megadva) $1 \leq \text{min} \leq \text{max} \leq 1000$). A sorban a harmadik szám a határidő értéke percben megadva. A korszókat a sorszámukkal azonosítjuk 1-től N -ig az érkezésük sorrendjében.

A kemence.ki szöveges állomány első sorába két egész számot kell írni egy szóközzel elválasztva, az összes korszó kiégetéséhez szükséges legkisebb időt, és az égetési menetek M számát. A következő M sor mindegyike két egész számot tartalmazzon egy szóközzel elválasztva, u v , ami azt jelenti, hogy ebben a menetben az $u, u+1, \dots, v$ sorszámú korszók kerülnek a kemencébe ($1 \leq u \leq v \leq N$, $v \leq u+K-1$)! Több megoldás esetén bármelyik megadható. A feladat minden tesztesetre megoldható.

Példa:

kemence.be	kemence.ki
4 3	9 3
1 2 4	1 2
2 3 3	3 3
3 4 8	4 4
1 2 9	

4. feladat: Képzés (15 pont)

Egy vállalat fel akarja készíteni a dolgozóit egy új szoftver használatára. Arra nincs lehetőség, hogy minden dolgozó részt vegyen kiképzésen. Ezért az igazgató elhatározta, hogy a lehető legkevesebb dolgozó vegyen részt kiképzésen, de teljesüljön, hogy ha egy dolgozó nem vett részt a kiképzésen, akkor a közvetlen főnöke biztosan részt vett. A vállalat hierachikus felépítésű, tehát az igazgató kivételével (akinek nincs főnöke) minden dolgozónak pontosan egy közvetlen főnöke van, továbbá az igazgató mindenkinek a főnöke (közvetlenül, vagy közvetve).

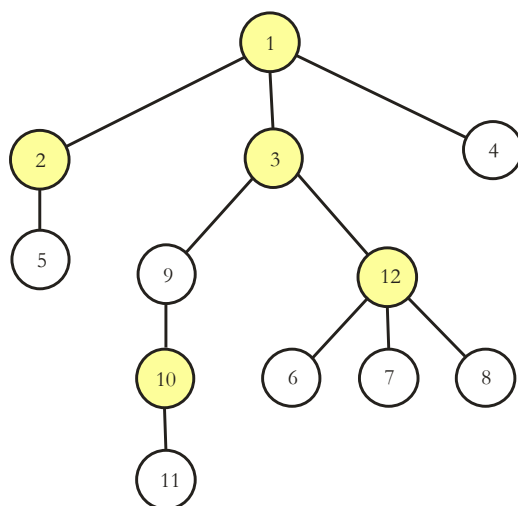
Készíts programot (kepzes.pas , kepzes.c, ...), amely kiszámítja, hogy legkevesebb hány dolgozónak kell részt vennie képzésen, és meg is adja hogy kiknek!

A kepzes.be szöveges állomány első sorában a dolgozók N száma ($1 \leq N \leq 100000$) van. A dolgozókat a $1, \dots, N$ számokkal azonosítjuk, az igazgató azonosítója I . A második sor pontosan N egész számot tartalmaz (egy-egy szóközzel elválasztva), az i -edik szám az i azonosítójú dolgozó közvetlen főnökének azonosítója. Mivel az igazgátónak nincs főnöke, ezért az első szám 0.

A kepzes.ki szöveges állomány első sorába azt az M számot kell írni, ahány dolgozónak rész kell vennie kiképzésen! A második sorba M számot kell írni egy-egy szóközzel elválasztva, azon dolgozók sorszámait, akik részt vesznek kiképzésen! A számok sorrendje közböbs. Több megoldás esetén bármelyik megadható.

Példa:

kepzes.be	kepzes.ki
12	5
0 1 1 1 2 12 12 12 3 9 10 3	1 2 3 10 12



5. feladat: Csoport beosztás (15 pont)

Egy kiránduláson résztvevő tanulókból két csapatot kell képezni. A két csapatot úgy kell képezni, hogy ha X és Y barátok, akkor azonos csapatba kerüljenek, de ha nem kedvelik egymást, akkor nem kerülhetnek egy csapatba.

Készíts programot (`csoport.pas`, `csoport.c`, ...), amely kiszámít egy, a feltételeknek megfelelő csapatbeosztást!

A `csoport.be` szöveges állomány első sorában három egész szám van egy szóközzel elválasztva, a tanulók N száma ($1 \leq N \leq 500$) és a baráti párok M száma ($1 \leq M \leq 20000$) és azon párok K száma, akik nem kedvelik egymást. A tanulókat az $1, \dots, N$ számokkal azonosítjuk. A következő $M+K$ sor mindegyike két egész számot tartalmaz (egy szóközzel elválasztva): A következő M sor a baráti párokat, az azt követő K sor pedig azon párokat tartalmazza, akik nem kedvelik egymást.

A `csoport.ki` szöveges állomány első sorába az egyik, a második sorába a másik csapatba sorolt tanulók azonosítóit kell írni egy-egy szóközzel elválasztva, tetszőleges sorrendben. Több megoldás esetén bármelyik megadható. Ha nincs megoldás, akkor a kimenet első és egyetlen sorába a -1 számot kell írni.

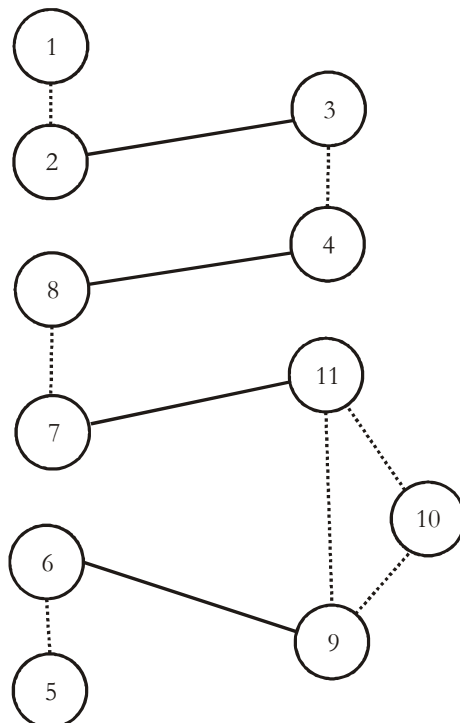
Példa:

`csoport.be`

```
11 7 4
1 2
3 4
5 6
9 10
10 11
11 9
7 8
2 3
6 9
8 4
7 11
```

`csoport.ki`

```
1 2 5 6 7 8
3 4 9 10 11
```



Elérhető összpontszám: 75 pont + 25 pont a 2. fordulóból