

Kedves Versenyző! A megoldások értékelésénél csak a **programok futási eredményeit** vesszük tekintetbe. Ezért igen fontos a **specifikáció pontos betartása**. Ha 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 programszövegekben egyetlen karaktert sem javítunk, s az előre megadott javítási útmutatótól semmiben nem térünk 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 állomány neve minden esetben rögzített.

1. feladat: Kockavilág (25 pont)

Van N ($1 \leq N \leq 100$) db azonos méretű kockánk (1, 2, ..., N számokkal jelöljük). A kockák vagy az asztalon vannak, vagy egy másik kocka tetején.

Van egy robotkar, ami képes fentről megfogni a legfelső kockát és azt egy másik kocka tetejére vagy az asztalra helyezni. A cél az, hogy a robotkar segítségével úgy mozgassuk a kockákat, hogy a 1. legyen legalul, a 2. az 1. kockán legyen, stb.

Készíts programot (Projekt név: kocka (kocka.cbp; main.cpp))!

A **kocka.be** bemeneti állomány első sorában N értéke van. A további N sorban található, hogy az adott kocka melyik másik kocka tetején van. Ez a szám 0, ha a kocka az asztalon van. Tehát az állomány i -edik sorában lévő szám azt adja meg, hogy az i -1-edik kocka melyik kocka tetején van. Ha egy kocka a másik tetején van, az csak úgy lehet, hogy az érintkező oldaluk teljesen fedik egymást.

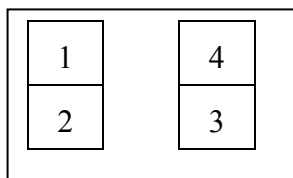
A **kocka.ki** állomány első sorába a HIBAS szöveget, ha a **kocka.be** valami oknál fogva nem megfelelő, egyébként a HELYES szót.

Az állomány második sorától kezdődően a legkevesebb lépésszámú megoldást kell írni! (Ha több ilyen is van, akkor csak az egyiket.) Minden sorban két szám szerepeljen egy szóközzel elválasztva: melyik kockát melyikre kell tenni!

Példa:

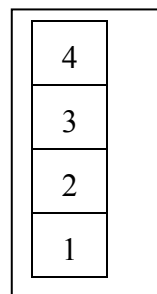
kocka.be

4
2
0
0
3



kocka.ki

HELYES
1 0
2 1
4 0
3 2
4 3



2. feladat: Malacpersely (25 pont)

Mohó Marci malacperselyben gyűjti pénzét. Csak fémpénzeket rakott a perselybe, de nem jegyezte fel, hogy milyeneket. Felírta azonban az üres persely súlyát, így meg tudja állapítani a perselyben lévő pénzek összsúlyát. Ismeri továbbá az egyes pénzermékek egyedi súlyát és értékét. Szeretné kiszámítani, hogy mennyi az a legkisebb érték, amelyet a perselye biztosan tartalmaz. Egy adott típusú pénzerméből (címletből) több is lehet a perselyben.

Készíts programot (Projekt név: malac (malac.cbp; main.cpp)), amely kiszámítja, hogy legkevesebb mekkora értéket tartalmaz a malacpersely!

A **malac.be** bemeneti állomány első sorában van a perselyben lévő pénzek S ($1 \leq S \leq 10000$) összsúlya. A második sorban a pénzérme fajták (címletek) N ($1 \leq N \leq 100$) száma található. A további N sor mindegyike két pozitív egész számot tartalmaz egy szóközzel elválasztva. Az első szám egy pénzérme értéke (nem nagyobb, mint 200), a második szám pedig a pénzérme súlya (nem nagyobb, mint 1000).

A **malac.ki** állomány egyetlen sort tartalmazzon, azt a legkisebb értéket, amelyet a malacpersely biztosan tartalmaz!

Példa:

malac.be	malac.ki
15	8 (6 db 1 forintos és 1 db 2 forintos)
4	
1 2	
2 3	
5 6	
10 4	

3. feladat: Üzenetek (25 pont)

A Kozmosz Rt. a stratégiai fontosságú üzenetek továbbítására saját rendszert dolgozott ki. Ha valaki, aki részt vesz a rendszerben és üzenetet kap, köteles azt továbbítani a számára előírt embereknek. A társaságnál az ezirányú kötelezettséget a következőképpen jelölték:

János (Géza, István, Mónika)

Géza (Éva, Lajos)

Jelentésük: Jánosnak továbbítania kell az üzenetet Gézának, Istvánnak és Mónikának, illetve Gézának továbbítania kell az üzenetet Évának és Lajosnak.

A társaság ezen szabályokat egy globális, összevont üzenetközvetítési szabállyal írja le, ami a példa esetében:

János (Géza (Éva, Lajos), István, Mónika) .

Az első ember (János) fogja az üzenetet megkapni a vezérigazgatóságtól, majd továbbítja azokat a számára kiírt embereknek, akik szintén továbbadják azt. Azon emberek, akiknek nincs kijelölve senki, természetesen nem adják tovább az üzenetet senkinek.

A társaság az üzenetközvetítés hatékonyságáról szeretne informálódni, megadott lánc esetén. Készíts programot, amely kiszámolja az alábbiakat:

- A. Egy embernek maximum hány másikkal kell közvetlenül átadnia az üzenetet?
- B. Legfeljebb hány emberen keresztül jut el az üzenet valakihez?
- C. Hány olyan ember van, akinek nem kell továbbítania az üzenetet?

Készíts programot (Projekt név: uzenet (uzenet.cbp; main.cpp))!

Az **uzenet.be** állomány első sora tartalmazza a szabályzatot, melyben maximum 1000 ember neve szerepel. A neveket az angol abc kis és nagy betűi jelölik, a szabályzat nem tartalmaz szóköz karaktert. A szabályzatban legalább egy ember szerepel. A szabályzatot leíró karaktersorozatot a # karakter zárja.

Az **uzenet.ki** állomány első sorába az A, a második sorába a B, a harmadik sorába pedig a C kérdésre adott választ kell írni.

Példa:

uzenet.be:	
Miklos (Peter (Balazs, Zsoka), Eva, Ferenc (Agi (Teri, Zsuzsa),	
Laci, Magdi, Dora)) #	
uzenet.ki:	
4	(mert Ferenc 4 embernek adja)
3	(mert Terihez Mikloson, Ferencen es Agin keresztül jut el)
8	

4. feladat: Autópálya (25 pont)

Bergengóciában nevezetességeit egy kör alakú autópálya köti össze, amely mentén benzinkutak sorakoznak. Berg Egon elhatározta, hogy körbeautózik az úton. Az autóját üres tankkal tudja csak szállítani az egyik benzinkúthoz. Tudjuk, hogy melyik benzinkútnál mennyi benzin van és ismerjük a benzinkutak egymástól való távolságát.

Készíts programot (Projekt név: korut (korut.cbp; main.cpp)), amely megadja, hogy melyik kúttól kell indulnia Egonnak az ott található teljes benzinkészlettel (biztosan magával tudja vinni), hogy úgy tudjon körbeautózni, hogy autójából ne fogyjon ki a benzin. Az autópálya egyirányú, az I. benzinkúttól csak az I+1. felé lehet indulni, valamint az N.-től az 1. felé.

A **korut.be** állomány első sorában két egész szám áll egyetlen szóközzel elválasztva, N és M, ahol N a kutak számát jelöli ($1 \leq N \leq 16000$), M pedig azt, hogy az autó egy liter benzinnel hány km-t tud megtenni ($1 \leq M \leq 100$). A következő N sor mindegyike két egész számot tartalmaz egy szóközzel elválasztva; az első a következő kút távolságát adja meg km-ben (legfeljebb 1000 km), a második pedig az itt fellelhető benzin mennyiségét (legfeljebb 200 liter).

A **korut.ki** állomány első sorába az IGEN szót kell írni, ha valamelyik kúttól kezdve az autópálya körbeutazható, egyébként pedig a NEM szót. A második sorba azon kutak számát kell írni, melyek bármelyikéből indulva körbe lehet autózni. Ha sehonnan sem lehet körbejutni, akkor ide annak a kútnak a sorszámát kell írni, ahonnan a legmeszszebbre el lehet jutni, azaz a legtöbb kutat lehet érinteni. (Ha több megoldás is van, közülük egyet kell megadni.)

Példa:

korut.be	korut.ki
8 10	IGEN
200 25	4 6 5
200 15	
100 5	
200 20	
300 30	
400 45	
200 20	
200 20	