

**A. 744.** Mutassuk meg, hogy bármely páratlan  $N > 5$  egész számhoz léteznek olyan  $\mathbf{u}, \mathbf{v}, \mathbf{w}$  vektorok a (három dimenziós) térben, amelyek páronként merőlegesek egymásra, nem párhuzamosak egyik koordináta-tengellyel sem, a koordinátáik egész számok, és  $|\mathbf{u}| = |\mathbf{v}| = |\mathbf{w}| = N$ .

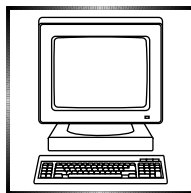
A 2018. évi Kürschák-verseny 2. feladata alapján

**A. 745.** Egy konvex poliéder minden lapja egy óramutatót hordoz; a mutatók mindig valamelyik élben szomszédos lap felé mutatnak. Minden perc végén valamelyik lap mutatója – az órajárás szerinti irányban – elfordul a következő lap felé úgy, hogy szomszédos lapok mutatói soha nem mutatnak egymás felé. Mutassuk meg, hogy van olyan mutató, amely csak véges sokszor fordul el.

**Beküldési határidő: 2019. március 10.**

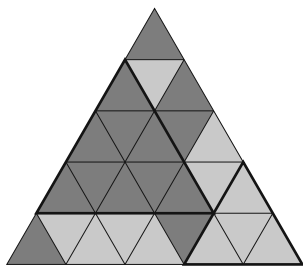
**Elektronikus munkafüzet: <https://www.komal.hu/munkafuzet>**

**Cím: KöMaL feladatok, Budapest 112, Pf. 32. 1518**



## Informatikából kitűzött feladatok

**I. 475.** Egy  $N$  oldalhosszúságú szabályos háromszög két fajta színű, egységnyi oldalú szabályos háromszögekből áll. Vizsgáljuk az  $N$  oldalhosszúságú háromszögben kirajzolódó egyszínű, szabályos háromszögeket. Készítsünk programot `i475` néven, amely megadja mindkét színhez a lehető legnagyobb területű szabályos háromszög oldalának hosszát.



A program olvassa be a standard bemenet első sorából az  $N$  oldalhosszúságot ( $1 \leq N \leq 100$ ), majd a következő  $N$  sorból a háromszög adott szintjén lévő egységnyi háromszögek színének kezdőbetűjét ( $\mathbf{k}$  = kék,  $\mathbf{s}$  = sárga).

A program a standard kimenetre írja ki a kék, majd a következő sorba a sárga háromszögekből álló legnagyobb egyszínű, szabályos háromszög oldalhosszúságát.

*Példa:*

Standard bemenet (a / jel sortörést helyettesít)	Standard kimenet
5	3
k / ksk / kkkks / kkkkss / ksssksss	2

Beküldendő egy tömörített `i475.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztő környezetben fordítható.

**I. 476 (É).** A főváros tömegközlekedési forgalomirányítási rendszerében a járművek fedélzeti számítógépekkel rendelkeznek. Egy adott időpontban az összes jármű néhány adatát lekérdezték, amelyek rendelkezésünkre állnak a `jarmu.txt`, `status.txt`, `tarolas.txt` és a `telep.txt` állományokban. Az állományok tabulátorral tagolt, UTF-8 kódolású szövegfájlok, az első sorok a mezőneveket tartalmazzák.

Készítsünk új adatbázist `i476` néven. A honlapunkról letölthető adatállományokat importáljuk az adatbázisba a forrásállományokkal azonos néven. Beolvasáskor állítsuk be a megfelelő adattípusokat és kulcsokat. A táblákba ne vegyünk fel új mezőt.

#### Táblák:

##### **jarmu** (rendszer, típus, fajta)

rendszer	a jármű rendszáma, villamos, trolibusz pályaszáma, hajó neve (szöveg), ez a kulcs;
típus	a jármű típusának jelölése (szöveg);
fajta	a jármű fajtája, pl.: autóbusz, troli, fogaskerekű, ... (szöveg).
<b>statusz</b> (az, rendszer, beido, tarolas az)	
az	a bejelentkezés azonosítója (szám), ez a kulcs;
rendszer	a jármű azonosítója (szöveg);
beido	az utolsó bejelentkezés dátuma (dátum);
tarolas az	az utolsó telephely azonosítója, amelyen tartózkodott (szám).

##### **tarolas** (az, telep\_az, uzemag)

az	a jármű tárolásának azonosítója (szám), ez a kulcs;
telep_az	a járműtelep azonosítója (szám);
uzemag	a járműtelep üzemágának rövidítése (szöveg).

##### **telep** (az, nev, kerulet, cim)

az	a járműtelep azonosítója (szám), ez a kulcs;
nev	a telep nevének rövidítése (szöveg);
kerulet	a telep kerülete (szöveg);
cim	a telep címe (szöveg).

Készítsük el a következő feladatok megoldásait. Az egyes lekérdezéseknél ügyeljünk arra, hogy mindig csak a kért értékek jelenjenek meg és más adatok viszont ne. Megoldásainkat a zárójelben lévő néven mentjük el.

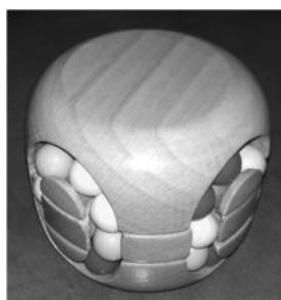
1. Soroljuk fel lekérdezés segítségével a villamosok és autóbuszok kivételével azon járművek rendszámát, típusát és fajtáját, amelyek utolsó bejelentkezése 2018. augusztus előtti. (1regi)
2. Melyik kerületben vannak a villamos járműtelepek? A listában minden kerület egyszer jelenjen meg. (2kocsizin)
3. Lekérdezéssel határozzuk meg azoknak a járműveknek a rendszámát és típusát, amelyekhez nincs járműtelep megadva. (3hiany)

4. Adjuk meg azoknak a busz járműtelephelyeknek a név rövidítését, kerületét és címét, ahol az Ikarus egyik típusa sincs. (4nincsikarus)
5. A járműparkban van néhány olyan típus, amiből csak néhány darab van. Határozzuk meg a legkisebb darabszámú autóbusz típusát és azt, hogy az hány telephelyen van. (5keves)
6. Paraméteres lekérdezés segítségével adjuk meg a paraméterként megadott rendszámú járművel azonos napon és órában bejelentkezőket. A listában a rendszámuk, a típusuk és a fajtájuk jelenjen meg. (6egyszerre)
7. Határozzuk meg lekérdezés segítségével az oszlopok sorrendjétől eltekintve a minta szerint, hogy a fővárosban melyik járműfajtából hány darab jármű van. (7összesites)

Autóbusz (db) ▾	Villamos (db) ▾	Trolibusz (db) ▾	Fogaskerekű (db) ▾	Hajó (db) ▾
1662	566	134	12	7

Beküldendő egy tömörített i476.zip állományban az adatbázis, valamint egy rövid dokumentáció, amely megadja az alkalmazott adatbázis-kezelő nevét és verziószámát.

**I. 477.** A bal oldali képen látható egyszemélyes játék egyszerűbb változatát mutatja a jobb oldali kép.



Az egyszerűsített forma négy egymásba fonódó körből áll, amelyek mentén egymástól egyenlő távolságra egy-egy kör került elhelyezésre a minta szerint. Mind-egyik kör – egymástól függetlenül –  $60^\circ$ -kal elforgatható, természetesen egyszerre csak egy. A forgatás az éppen rajta lévő kis köröket „magával viszi”, így néhány forgatás után a kezdetben rendezett színösszeállítás (a kisebb körök mentén 6, azonos színű kör van) már igen nagy összevisszaságot mutathat.

A feladatunk a síkbeli változat grafikus megvalósítása. A használó választhatson: a rendezett állapotból indulva szabadon használja a játékot, vagy kérhet egy véletlenszerű (de megoldható) állapotot, amelyből indulva előállíthatja a rendezett állapotot. (Rendezett állapotnak azt tekintjük, ha a kis körök mentén egyező színek vannak, függetlenül azok elhelyezkedésétől.)

A feladat megoldását a versenykiírásban szereplő eszközökön túl a webböngészőben, vagy mobil eszközökön futó applikációval is meg lehet adni. A feltétel, hogy a megoldás tesztelhető legyen Windows, Linux vagy Android alapú operációs rendszer alatt ingyenes eszközökkel, kiegészítők telepítése nélkül.

Beküldendő egy `i477.zip` tömörített állományban a program forráskódja és a működéséhez szükséges egyéb fájlok, továbbá a hozzá kapcsolódó felhasználói dokumentáció, valamint a leírás, amely tartalmazza, hogy a forrásállomány melyik fejlesztő környezetben fordítható. Szükség esetén a tesztkörnyezetet is pontosan meg kell adni.

Az értékelésben 7 pont jár a feladat leírásának megfelelő megoldásokért, 3 pont pedig a megoldás kifinomultsága, ötletessége, használhatósága alapján történő differenciálásra szolgál.

**I/S. 33.** Egy adatsokaságban  $N$  féle adat van. Tudjuk minden adatról, hogy hányszor szerepel az adatsokaságban, és tudjuk azt is, hogy egy adott típusú adat törlése vagy beszúrása mennyibe kerül. Adjuk meg minden adattípusra, hogy minimum milyen költséggel érhető el adatok törlésével és beszúrásával, hogy ez az adattípus (is) az adatsokaság módusza legyen.

*Bemenet:* az első sorban az adattípusok  $N$  száma szerepel. A második sorban  $N$  darab szám: az  $i$ . szám azt mondja meg, hogy az  $i$ . adattípus hányszor szerepel az adatsokaságban. A harmadik sorban  $N$  darab szám van: az  $i$ . szám azt mondja meg, hogy az  $i$ . adattípusú adat beszúrása vagy törlése mennyibe kerül.

*Kimenet:* egy sorba írjunk ki  $N$  darab számot: az  $i$ . szám annak a minimális költsége, ami szükséges ahhoz, hogy az  $i$ . adattípus módusz legyen.

*Példa:*

Standard bemenet (a / jel sortörést helyettesít)	Standard kimenet
11	0 17 3 0 64 71 24 27 16 0 14
8 4 5 8 1 1 2 2 4 8 1	
1 6 1 2 17 24 4 5 4 1 2	

*Korlátok:*  $2 \leq N \leq 10^5$ ,  $0 \leq a$  a 2. és 3. bemeneti sorban levő számok  $\leq 10^9$ .  
*Időlimit:* 0,5 mp.

*Értékelés:* a pontok 20%-a kapható, hogyha  $N \leq 100$ , a 2. és 3. bemeneti sorban levő számok  $\leq 100$ ; további 10% kapható, ha a 2. sor számai egyenlők; további 20% kapható, ha  $N \leq 1000$ ; további 10% kapható, ha a 2. és 3. bemeneti sorban levő számok  $\leq 10^6$ ; további 40% kapható az eredeti korlátokra.

Beküldendő egy `is33.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztő környezetben futtatható.

**S. 132.** Egy ország  $N$  városa között az elektromos energiát vezetékeken szállítják. Minden vezeték két várost köt össze, elektromos energiát előállító erőmű csak a 0 indexű városban van. Két várost több vezeték is összeköthet. Egy város akkor kap áramot, ha a vezetékekkel közvetlenül vagy közvetetten összeköttetésben van az erőművel. Egy vezeték nélkülözhetetlen, ha meghibásodásakor lesz olyan város, ami nem kap áramot.

Az országnak túl költséges karbantartani az összes vezetékét, ezért úgy döntenek, hogy  $Q$  darab vezetékeltávolítanak a hálózathoz. Kíváncsiak vagyunk, hogy kezdetben hány nélkülözhetetlen vezeték van, és hogy az egyes vezetékek eltávolításával mennyivel nő a nélkülözhetetlen vezetékek száma. A vezetékek eltávolítása során sosem lesz olyan város, ami nem kap áramot.

*Bemenet:* az első sorban a városok  $N$  száma, a vezetékek  $M$  száma és az eltávolítandó vezetékek  $Q$  száma található. A városokat és vezetékeket is 0-tól indexeljük. A következő  $M$  sor mindegyike két város indexét tartalmazza, amiket összeköt az adott vezeték. Az utolsó  $Q$  sor mindegyike egy vezeték indexét tartalmazza, amit eltávolítanak.

*Kimenet:* Az első sorba írjuk ki, hogy kezdetben hány nélkülözhetetlen vezeték van. A következő  $Q$  sorba pedig azt, hogy az egyes vezetékek eltávolításával mennyi lesz a nélkülözhetetlen vezetékek száma.

*Példa:*

Standard bemenet (a / jel sortörést helyettesít)	Standard kimenet
17 28 4	0
0 10 / 4 1 / 13 14 / 8 12 / 0 1 / 0 2	1
0 5 / 1 3 / 7 1 / 4 1 / 2 8 / 5 2 / 3 6	1
3 7 / 3 11 / 8 5 / 9 5 / 10 6 / 6 10 / 7 11	1
9 12 / 15 9 / 10 13 / 13 10 / 11 14 / 11 16	1
15 12 / 14 16	
0 / 2 / 1 / 3	

*Korlátok:*  $3 \leq N \leq 10^5$ ,  $N - 1 \leq M \leq 10^6$ ,  $0 \leq Q \leq 10^5$ . *Időlimit:* 0,5 mp.

*Értékelés:* a pontok 20% kapható, ha  $Q = 0$ -ra ad jó megoldást; további 20%, ha  $Q \cdot M \leq 10^6$ ; további 10%, ha az eltávolítások során végig maximum 2 nélkülözhetetlen vezeték van; további 50% az eredeti bemenetre.

Beküldendő egy `s132.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

A feladatok megoldásai regisztráció után a következő címen tölthetők fel:

<https://www.komal.hu/munkafuzet>

**Beküldési határidő: 2019. március 10.**