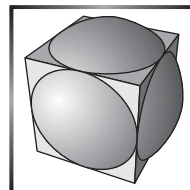


Az A pontversenyben kitűzött nehezebb feladatok (769–771.)



A. 769. Határozzuk meg azokat a három különböző pozitív egész számból álló (a, b, c) számhármassokat, melyekhez létezik olyan H részhalmaza a pozitív egész számoknak, hogy minden pozitív egész n -re az an , bn , cn számok közül pontosan egy van benne a H halmazban.

Javasolta: *Carl Schildkraut* (Massachusetts Institute of Technology)

A. 770. Határozzuk meg azokat az n pozitív egészeket, melyekre $n!$ két Fibonacci-szám szorzata.

A. 771. Legyen az ABC háromszög beírt köre ω , mely a BC oldalt a D pontban érinti. Az AD egyenes második metszéspontja az ω körrel legyen G . Az ω körhöz a G pontban húzott érintő messe az AB és AC oldalakat rendre az E és az F pontban. A DEF körülírt körének D -től különböző metszéspontja ω -val legyen M . A BCG körülírt körének a G -től különböző metszéspontja ω -val legyen N . Bizonyítandó, hogy az AD és MN egyenesek párhuzamosak.

Javasolta: *Györfly Ágoston* (Remeteszőlős)

Beküldési határidő: 2020. március 10.

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

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

Informatikából kitűzött feladatok



I. 502. A számegyenesen N darab intervallumot adunk meg. Az intervallumok nyíltak vagy zártak lehetnek, határpontjaik egészek vagy tizedes törtek. Jelölésük a szokásos módon történik (az informatikában megszokottabb tizedes pontot használva), például $[4, 5.2[$ vagy $] -3.3, 4.66[$, ahol az első intervallum balról zárt és jobbról nyílt, míg a második mindkét oldalról nyílt. Keressük meg a számegyenes azon egész számait, amelyek a legtöbb intervallumban vannak benne.

A program olvassa be a bemenet első sorából az intervallumok N számát ($2 \leq N \leq 100$), és a következő N sorból egyenként az intervallumokat. A bemenet intervallumai az egyszerűbb beolvasás céljából szóközzel tagoltak a mintának megfelelően, valamint mind a $[-1000, 1000]$ intervallum részei.

A kimenet egyetlen sorába írjuk ki növekvő sorrendben a legtöbb intervallumban szereplő egészeket. Az eredmény számait szóközzel határoljuk, az egymást követő egészek sorozatának csak szélső értékeit adjuk meg kötőjellel elválasztva.

Példa Bemenet (a / jel sortörést helyettesíti)	Példa Kimenet
3 / [-5.5 , 9] /] 7 , 11.1 [/] 3 , 5 [4 8 - 9

Beküldendő egy tömörített `i502.zip` állományban a megoldást tartalmazó forrásállomány, valamint egy rövid szöveges leírás, ami tájékoztat az alkalmazott programozási nyelvről és a fejlesztői környezetről.

Letölthető állományok: `i502beki.zip` (példa be- és kimenetek).

I. 503 (É). Magyarország közigazgatási helynévkönyve egy 1992 óta évente megjelenő kiadvány, mely tartalmazza minden település hivatalos megnevezését, megyei beosztását, a közös önkormányzati hivatalokat, a helységek járási besorolását, valamint a nemzetiségi önkormányzatokat. Közli a helységek 2018. január 1-jei területnagyság-, lakónépesség- és lakásszámadatát, továbbá a helységek KSH által kibocsátott településazonosító törzsszámát. Feladatunk ezen adatok feldolgozása lesz táblázatkezelő program segítségével.

- Töltsük be a települési adatokat tartalmazó `hnk1_2018.txt` szövegfájlt a táblázatkezelő egy munkalapjára az A1-es cellától kezdődően, a kódok jelentését tartalmazó `hnk2_2018.txt` szövegfájlt pedig egy másik munkalapra. A munkalapok neve legyen rendre **adat** és **statisztika**. Mindkét állomány pontosvesszővel tagolt, UTF-8 kódolású.
- Munkánkat `kozihelynev` néven mentjük el a táblázatkezelő alapértelmezett formátumában.
- A C3183-as cellától kezdődően egymás alá gyűjtsük ki a jogállásokat. Minden típust egyszer. A mellette lévő D oszlopban pedig másolható függvény segítségével adjuk meg, hogy az egyes jogállástípusokból hány található az országban.
- Hozzunk létre egy új oszlopot a jelenlegi H oszlop mögé és ebben az oszlopban jelenjen meg az önkormányzati hivatali kód jelentése, ami a statisztika munkalapon található.
- A **statisztika** munkalap C11:E11-es celláiban függvény segítségével adjuk meg Magyarország területét (hektárban), lakónépességét és a lakások számát. Ügyeljünk rá, hogy Budapest szerepel kerületenként lebontva is a listában.
- Tudjuk, hogy legtöbben Budapesten laknak. A **statisztika** munkalap B9-es cellájában egész mondatba foglalva, függvények segítségével a lakosságszámot is megadva, írjuk ki annak a településnek a nevét, ami a második legnagyobb lakosságszámú.
- Az előző mintájára a B10-es cellában adjuk meg annak a településnek a nevét, megyéjét, és lakosságszámát, ahol a legkevesebben laknak.

Budapest után a legtöbb	202214 fő	Budapest	megye	ben lakik.
A legkevesebben	fő	Magyar	ország	megyében található.

8. Az N oszlopban adjuk meg, hogy két tizedesjegyre felfelé kerekítve átlagosan hányan élnek egy lakásban.
9. Készítsünk egy 21 soros, 14 oszlopos táblázatot a **statisztika** munkalapra az eredeti táblázat mellé, úgy, hogy a G3-as cellába „Bács-Kiskun” megye kerüljön. Alatta gyűjtsük ki az **adatok** munkalap D oszlopában található „megye” neveket és rendezzük ABC-rendbe. A H2-es cellába a „bolgár” szó kerüljön, a mellette lévő cellákba a többi nemzetiségi önkormányzat az **adat** munkalapról. A táblázatot töltsük fel egyetlen másolható függvény használatával úgy, hogy megyénként adja meg az önkormányzatok számát.

	bolgár	görög	horvát	lengyel
Bács-Kiskun	1	1	1	1
Baranya	1	1	1	1
Békés	1	1	1	1
Borsod-Abaúj-Zemplén	1	1	1	1
Csongrád	1	1	1	1
Fejér	1	1	1	1
főváros	1	1	1	1
Győr-Moson-Sopron	1	1	1	1
Hajdú-Bihar	1	1	1	1
Heves	1	1	1	1

10. A **statisztika** munkalapon található C oszlopban adjuk meg, hogy az egyes önkormányzati hivatalokhoz hány lakás tartozik.

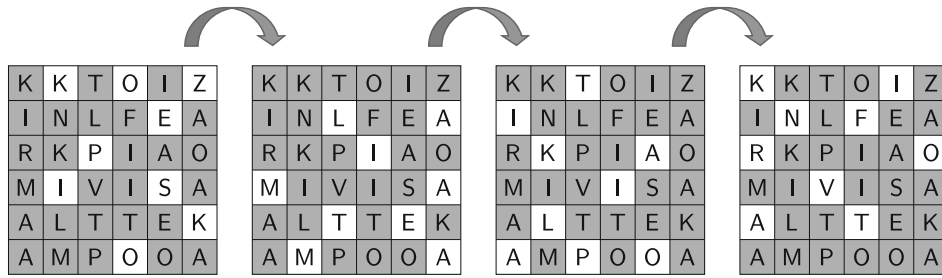
Forrás: http://www.ksh.hu/apps/shop.kiadvany?p_kiadvany_id=1039140&p_lang=HU (2019.09.11.).

Beküldendő egy tömörített i503.zip állományban a megoldást adó táblázatkezelő munkafüzet és egy rövid dokumentáció, amely megadja a felhasznált táblázatkezelő nevét és verzióját.

I. 504. A középkorban használt titkosítási eljárások egyike a Cardano-rács. A titkosítandó szöveget ebben az esetben négyzet alakban rendezik el, a szöveg rejtjelezése és visszafejtése pedig az erre illeszkedő, megfelelő helyeken kivágott rostély ablakain keresztül történik. A szövegből a rácsban csak a betűk szerepelnek, az írásjelek és szóközök nem. A rostély egy-egy helyzetében a látható betűket sorfolytonosan kiolvassuk, a rostély háromszori körbefordításával a négyzet minden betűje felhasználásra kerül. (A dupla betűket két karakterrel kódoljuk, például GY helyett G és Y.)

A módszer bemutatása az **I. 201.** feladatban szerepelt, érdemes belenézni a kitűnő mintamegoldásokba (<https://www.komal.hu/feladat?a=feladat&f=I201&l=hu>).

Készítsünk programot, amely a Cardano-rácscsal történő rejtjelezést és visszafejtést végzi el:



1. Olvassuk be és tároljuk el a Cardano-rácsot tartalmazó `cardano.txt` állományt, amely egy 6×6 -os rostélyt tartalmaz. Az átlátszó rácsponatokban 0 szerepel, a nem átlátszóokban 1.
2. Készítsünk eljárást `Forgat` néven, amely lehetővé teszi a Cardano-rács -90 fokos (vagyis az óramutató járásával egyező irányú) elforgatását.
3. Írassuk ki a képernyőre a megadott Cardano-rácsot, valamint annak -90 fokos elforgatását. A két rács egymás alatt jelenjen meg.
4. A fenti mintán látható szöveget a `titkos.txt` fájl tartalmazza 6×6 -os rácsokra bontva. Olvassuk be a fájl tartalmát, fejtjük vissza azt a megadott Cardano-rács segítségével a `Forgat` eljárás felhasználásával, majd a megfejtést sorfolytonosan írassuk ki a képernyőre.

Ha a titkosítandó szöveg „egy négyzetnél” hosszabb, akkor azt több négyzetre kell bontani. Ha nem tesz ki a szöveg utolsó része egy teljes négyzetet, akkor azt véletlenszerű karakterekkel töltik fel.

A Cardano-rács alkalmazásával találkozhatunk *Jules Verne: Sándor Mátyás* c. könyvében is. A könyvben szereplő titkosítandó szöveget a `nyilt.txt` UTF-8 kódolású állomány tartalmazza. A könyvben a titkosítást két lépésben végezték, az alábbiakban ezt kell végrehajtani:

5. Olvassuk be a `nyilt.txt` fájl tartalmát, majd fordítsuk meg a szöveget. A beolvasott, illetve a megfordított szöveget egyaránt írassuk ki a képernyőre.
6. A felcserélt karakterekből álló szöveget rejtjelezzük a megadott Cardano-rács alkalmazásával, és az eredményt 6×6 -os rácsban – a minta szerint – írassuk ki a `sandormatyas.txt` fájlba.

Beküldendő egy `i504.zip` tömörített állományban a program forráskódja és egy rövid leírás, ami megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

Letölthető fájlok: `cardano.txt`, `titkos.txt`, `nyilt.txt`.

I/S. 42. Egy út mindkét oldalán kilométerenként található egy-egy kilométerkő. N csirke szeretne átkelni az út egyik (ugyanazon) oldaláról a másikra. Mind-

R	H	G	A	A	Z
Ü	Y	G	G	R	É
A	F	X	S	G	M
N	T	L	Á	R	E
E	Z	L	F	T	É
S	E	R	É	O	G

L	K	A	E	E	N
N	J	E	Ö	L	M
S	N	E	Ö	O	L
T	K	E	Z	K	Y
G	A	G	E	A	E
E	N	R	G	É	L

S	L	Ö	Ö	É	Z
Z	K	B	N	E	T
E	A	K	Z	L	D
S	R	N	L	É	E
I	Á	I	M	R	L
N	T	E	Ö	Z	N

egyikről tudjuk, hogy melyik kilométerkötől indul és melyik kilométerkőhöz érkezik. Minden kilométerkötől legfeljebb egy csirke indul és minden kilométerkőhöz legfeljebb egy csirke érkezik. Ha két csirke útvonala keresztezi egymást, akkor találkozhatnak, összezavarodnak és esetleg nem érnek célba. Adjuk meg, hány csirke útja biztonságos, tehát hányat nem fenyeget a keresztezésből adódó veszély.

Bemenet: az első sor tartalmazza a csirkék N számát. A következő N sor mindegyike két számot tartalmaz, mely azt jelenti, hogy az i -edik csirke az A_i kilométerkötől indul és a B_i kilométerkőhöz érkezik.

Kimenet: az első sor tartalmazza azon csirkék számát, amelyek biztonságosan át tudnak kelni az úton.

Példa:

Bemenet (a / jel sortörést helyettesíti)	Kimenet
4 1 8 / 11 12 / 14 20 / 7 13	2

Korlátok: $1 \leq N \leq 10^5$, $1 \leq A_i, B_i \leq 10^9$. Időkorlát: 0,3 mp.

Értékelés: a pontok 50%-a kapható, ha $N \leq 10\,000$.

Beküldendő egy `is42.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

S. 141. Egy épület legfelső emelete N darab lépcsőfokra van a földszinttől. Balázs M napon át, minden nap felmegy a földszintről az épület legfelső szintjére. Az első nap maximum P darab lépcsőfokot tud lépni egy lépéssel. Mivel Balázs egy növekedésben levő tini, ezért a második nap már $P + 1$ fokot tud megtenni egy lépéssel, a harmadik nap $P + 2$ fokot, és így tovább. A legfelső szintről lefelé mindig lifttel közlekedik, csak felfelé lépcsőzik. Adjuk meg, hogy az M nap alatt legalább hány lépést tesz meg Balázs.

Bemenet: az első sor tartalmazza az N , M és P számokat ebben a sorrendben.

Kimenet: adjuk meg a minimálisan megtett lépések számát.

Példa:

Bemenet	Kimenet
12 4 2	16

Korlátok: $1 \leq N, M, P \leq 10^{12}$. Időkorlát: 0,4 mp.

Értékelés: a pontok 50%-a kapható, ha $N \leq 10^3$.

Beküldendő egy `s141.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: 2020. március 10.