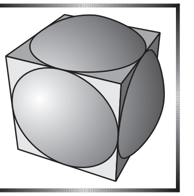

Az A pontversenyben kitűzött nehezebb feladatok (704–706.)

A. 704. Egy n hosszú oldalakkal rendelkező szabályos háromszög oldalainak n-edelőpontjain át behúztuk az oldalakkal párhuzamos egyenesek háromszögbe eső szakaszait. Tekintsük az így létrejövő $1+2+\ldots+(n+1)$ darab metszéspont alkotta pontrácsot. Melyek azok az n pozitív egészek, melyekre ez a pontrács olyan ponthármasokba partícionálható, melyek egy-egy egységnyi oldalú szabályos háromszög csúcsai?

Javasolta: *Alexander Gunning* (Cambridge, Egyesült Királyság)

A. 705. Legyen az ABC háromszög magasságpontja H, és legyen D egy, a csúcsoktól különböző pont a háromszög körülírt körén. Tegyük fel, hogy a BHD kör az AB egyenest $P\neq B$-ben, illetve a CHD kör az AC egyenest $Q\neq C$-ben metszi. Mutassuk meg, hogy a D pontot a körülírt körön mozgatva, D-nek PQ-ra vonatkozó tükörképe is egy rögzített körön mozog.

Javasolta: *Michael Ren* (Andover, Massachusetts, USA)

A. 706. Jelölje $\mathbb{Z}^+$ a pozitív egészek halmazát. Határozzuk meg az összes olyan $f:\mathbb{Z}^+\to\mathbb{Z}^+$ függvényt, melyre a következők teljesülnek:

- $f(mn)=f(m)f(n)$ minden $m,n\in\mathbb{Z}^+$-ra, illetve
- $f^{(n)}(n)=n$ minden $n\in\mathbb{Z}^+$-ra (azaz $f\Big(f\big(\ldots\big(f(n)\big)\ldots\big)\Big)=n$, ahol a zárójelpárok száma n).

(*Koreai feladat*)

Beküldési határidő: 2017. november 10.
Elektronikus munkafüzet: `https://www.komal.hu/munkafuzet`
Cím: KöMaL feladatok, Budapest 112, Pf. 32. 1518

Informatikából kitűzött feladatok

I. 436 (É). Ha egy szabályos dobókockát feldobunk, leesés után ugyanakkora valószínűséggel lesz a kocka tetején az első hat pozitív egész szám valamelyikének megfelelő számú pont. Erre a továbbiakban arab számjegyekkel hivatkozunk: 1, 2, 3, 4, 5, 6.

Ebben a feladatban szabályos dobókockával történő dobást szimulálunk, illetve az így kapott sorozatot elemezzük. Készítsünk programot *dobokocka* néven a következő feladatok megoldására.

1. Kérjünk be a felhasználótól egy tippet, majd szimuláljunk egy kockadobást szabályos dobókockával. Írassuk ki a képernyőre a felhasználó tippjét és a dobás eredményét is, majd tájékoztassuk a felhasználót az eredményről a következő formában: „Ön eltalálta." vagy „Ön nem találta el.". (Az ékezetmentes kiírás is elfogadott.)
2. Szimuláljunk egy $N(\leqslant 10\,000)$ dobásból álló kísérleti dobássorozatot, és az eredményt tároljuk el egy megfelelő típusú változóban. Az N értékét a felhasználótól kérjük be. Írassuk ki a dobássorozatot (elválasztójelek nélkül) a `kiserlet.txt` szöveges állomány első sorába is, ezt egy szóközzel elválasztva kövesse N értéke. A továbbiakban az így kapott sorozatot elemezzük.
3. Számoljuk meg, hogy a kísérlet során hányszor dobtuk az egyes számokat. Írassuk ki relatív gyakoriságukat két tizedesjegy pontossággal a `kiserlet.txt` fájl második sorába, egy-egy szóközzel elválasztva (pl. 1-16,51% 2-17,23% ...).
4. Hányszor fordult elő a kísérlet során, hogy egymás után pontosan két hatost dobtunk? Az eredményt írassuk a `kiserlet.txt` fájl harmadik sorába.
5. Hányszor fordult elő a kísérlet során, hogy a kocka két egymást követő dobás esetén két egymással szemben lévő oldalára esett? (Közismert, hogy a szabályos dobókocka szemben lévő oldalain szereplő számok összege 7.) A választ írassuk a `kiserlet.txt` fájl 4. sorába. (Például a 21612 sorozat kettőnek számít.)
6. Előfordult-e a kísérlet során, hogy hat egymást követő dobás során mind a hat lehetséges értékre sor került? Írassuk a választ (Igen vagy Nem) a `kiserlet.txt` fájl 5. sorába. Ha a válasz Igen, adjuk meg egy ilyen sorozat kezdetének a helyét is az Igen után egy szóközzel elválasztva. (A minta tagjainak számozását eggyel kezdjük.)
7. Előfordult-e a kísérlet során legalább M tagú palindrom? (Olyan részsorozat, amely elölről hátulra és hátulról előre olvasva megegyezik, például: 2345432.) Az M értékét kérjük be a felhasználótól. Írjuk a választ és M értékét egy szóközzel elválasztva a `kiserlet.txt` fájl 6. sorába (például: Nem 12). Ha a válasz Igen, adjuk meg egy ilyen sorozat kezdetének a helyét is egy szóközzel elválasztva.
8. Milyen hosszú volt a leghosszabb, azonos számjegyekből álló sorozat? Írassuk ki a választ a `kiserlet.txt` fájl 7. sorába, továbbá egy szóközzel elválasztva írassuk mellé egy ilyen részsorozat első tagjának helyét is.

Beküldendő egy `i436.zip` tömörített állományban a program forráskódja és dokumentációja, amely tartalmazza a megoldás rövid leírását, és megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 437. Magyarország fejedelmeit, királyait, uralkodóit, államfőit és miniszterelnökeit sorolja fel a MEK kisokos*. A feladat ezen adatok feldolgozása lesz táblázatkezelő program segítségével. A táblázat tartalmazza, melyik házból való uralkodóról van szó, vagy fejedelemről, államfőről, esetleg miniszterelnökről, a nevet, az uralkodóházat vagy egyéb megnevezést és hogy mettől meddig tartotta fenn a tisztséget. Vannak olyan személyek, akik megszakításokkal, de többször voltak hivatalban, ők többször szerepelnek a listában. A táblázatban szereplő utolsó államfő Göncz Árpád, utolsó miniszterelnök pedig Horn Gyula, egészítsük ki a táblázatot napjainkig.

1. Töltsük be a `magyarvezetok.txt` szövegfájlt a táblázatkezelő egy munkalapjára az A1-es cellától kezdődően. Munkánkat `i437` néven mentsük el a táblázatkezelő alapértelmezett formátumában.
2. Az E oszlop celláiban írassuk ki, mikor kezdte meg az uralkodását, mikor lépett hivatalba az adott személy. Másolható képletet használjunk és ügyeljünk arra, hogy vannak üres sorok a táblázatban.
3. Az F oszlop celláiba írassuk ki, meddig uralkodott, volt hivatalban az éppen aktuális ciklusában az adott személy.
4. A G oszlopban számítsuk ki, hány évig volt hivatalban az adott személy összesen élete során.
5. A minta alapján számítsuk ki a K oszlop adott celláit! K4-ben adjuk meg, hány évig volt vezető a leghosszabb időt ott töltő személy, a K5-ben pedig, hogy ki volt az.
6. A K7-es cellában adjuk meg ki volt, aki legtöbbször töltött be vezetői tisztséget, a K8-ba ki volt ez, a K9-be pedig, hogy milyen hivatalt töltött be. Ha több azonos is van, elég az elsőt kiírni.
7. A I12-es cellától lefelé gyűjtsük ki azokat a királyokat, vezetőket, akik többször voltak hivatalban, mellé, hogy összesen hányszor.
8. Feltételes formázással a B oszlopban azoknak az uralkodóknak a nevét piros háttérszínnel emeljük ki, akiknek a keresztneve először szerepel az ország történetében. Például: I. Béla.
9. Hasonlítsuk össze diagram segítségével a Habsburg-ház és a Habsburg–Lotaringiai-ház uralkodóinak trónon töltött évei számát, amelyből kiderül, mely ház uralkodói töltöttek több évet összesen a trónon.
10. Az első sorban lévő „Magyarország vezetői” cím link legyen, és a `http://mek.niif.hu/00000/00056/html/240.htm` oldalra mutasson.

Beküldendő egy tömörített `i437.zip` állományban a megoldást adó táblázatkezelő munkafüzet és egy rövid dokumentáció, amely megadja a felhasznált táblázatkelő nevét és verzióját.

*Forrás: `http://mek.niif.hu/00000/00056/html/240.htm`, utolsó letöltés: 2017-09-20.

	A	B	C	D	E	F	G	H	I	J	K
1		Magyarország vezetői									
2											
3	Fejedelmek										
4		Árpád	fejedelem	895-907			év			leghosszabb idő:	év
5		Zsolt	fejedelem	907-947			év			uralkodó:	
6		Fajsz	fejedelem	947			év				
7		Taksony	fejedelem	947-972			év			legtöbb uralkodás:	
8		Géza	fejedelem	972-997			év			legtöbbször uralkodott:	
9		István (Vajk)	fejedelem	997-1000			év			milyen uralkodó:	
10											
11	Árpád-házi királyok									akik többször uralkodtak	hányszor
12		I. (Szent) István	Árpád-házi király	1000-1038			év				
13		Péter	Árpád-házi király	1038-1041			év				
14		Aba Sámuel	Árpád-házi király	1041-1044			év				
15		Péter	Árpád-házi király	1044-1046			év				
16		I. András	Árpád-házi király	1046-1060			év				
17		I. Béla	Árpád-házi király	1060-1063			év				
18		Salamon	Árpád-házi király	1063-1074			év				
19		I. Géza	Árpád-házi király	1074-1077			év				
20		I. (Szent) László	Árpád-házi király	1077-1095			év				
21		(Könyves) Kálmán	Árpád-házi király	1095-1116			év				
22		II. István	Árpád-házi király	1116-1131			év				
23		II. (Vak) Béla	Árpád-házi király	1131-1141			év				

I. 438. Készítsünk táblázatkezelő alkalmazásban táblázatot vagy írjunk programot, amely egy kavicsot terítő robot munkáját vezérli.

A robot egy 10×10 cellás négyzetrácson mozoghat a szövegesen megadott utasítások szerint. A robot mozgása a lehető legegyszerűbb, mert egyszerre előre, hátra, illetve jobbra vagy balra (E, H, J és B) egy egységet tud lépni. Amikor a robot új cellába lép, köveket vesz fel, ha a kövek száma az adott cellában 1-nél több, és köveket tesz le, ha van nála kő, a cellában pedig éppen nincs. A robot a bal felső sarok cellájából indul, felfelé néz és nincs nála kavics. Működése során először lép és utána változtathatja a cellában a kavicsok számát. A vezérlés utasításainak száma legfeljebb 100.

A 10×10 cellás négyzetrács celláinak kavicsszáma és a robotot vezérlő utasítássor áll rendelkezésre a `terep.txt` állományban. Vagy töltsük be a táblázatkezelőbe az A1-es cellától kezdődően, vagy a program standard bemenetén adjuk meg a szóközökkel tagolt `terep.txt` állományt. A megoldás során a forrásadatok módosulása esetén is helyes eredményt kell kapnunk.

A táblázatkezelő az L1-es cellában, vagy a program a standard kimeneten jelenítse meg, hogy a vezérlés befejezése után hány kő van a robotnál.

Példa a bemenetre: **(amely 5×5 cellás a tömörség kedvéért)**	**Kimenet**
2 1 2 0 1 **0 2 0 3** 1 2 3 0 **0** 2 1 1 2 **2** 3 3 2 1 1 0 HJJJHHE	1

Beküldendő egy tömörített `i438.zip` állományban a táblázatkezelő munkafüzet, vagy a program forráskódja és rövid dokumentációja, amely megadja a táblázatkezelő alkalmazás nevét és verziószámát, illetve azt, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

A megoldáshoz szükséges letölthető állomány: `terep.txt`

I/S. 20. Egy elektromos terepjáró autóval szeretnénk eljutni egy dimbes-dombos területen az egyik helyről a másikra. A területet gondolatban $N \times M$ egyforma négyzetre osztjuk, és minden egyes négyzethez egy magassági adatot rendelünk. A négyzetek számozása a bal felső saroktól indul jobbra, illetve lefelé. Az autó útját úgy modellezzük, mintha egy-egy oldalukkal egymással érintkező négyzeteken haladna keresztül. Az egyik négyzetről a másikra történő mozgáskor az autó 1 egységnyit meríti az akkumulátorát, ha a két négyzet azonos magasságban van. Alacsonyabb magasságban lévő négyzetről magasabban lévő négyzetre mozgáskor a szintkülönbség kétszerese plusz 1 egységnyit merül az akkumulátor. Amikor az autó lefelé halad, akkor a szintkülönbség számértékének megfelelő egységnyit töltődik az akkumulátor, miközben 1 egységet merül. Az autó indulási pontja a térkép (s_i, s_j) négyzete, a cél a térkép (c_i, c_j) négyzete.

Kérdés, hogy legkevesebb hány egységnyi töltéssel kell rendelkeznie az autónak a kiindulási négyzetben, hogy eljusson a cél négyzetbe úgy, hogy közben egyszer sem kell külső energiával tölteni, csupán a szintkülönbség csökkenésekor kap energiát. Az autó nem tud továbbmenni, ha egy négyzetbe érve nem pozitív az energiája, ezért az csak a cél négyzetben lehet 0.

A feladatot megoldó program olvassa be a standard bemenetről a térképhez tartozó N és M értékét, majd a következő N sor mindegyikében M pozitív egész $h_{i,j}$ számot, melyek a térkép i-edik sorában és j-edik oszlopában lévő négyzet szintértékét adják, illetve a következő sorban az induló és cél négyzetek adatait: s_i, s_j, c_i, c_j. A program írja a szabványos kimenetre a legkisebb akkumulátor töltöttséget, amellyel az autó a kiindulási helyről a célba érhet.

Példa:

Bemenet	**Kimenet**
4 5 1 2 4 3 4 1 1 3 5 2 1 3 2 3 4 1 2 1 1 3 3 2 1 4	6

Korlátok: $1 \leqslant N, M \leqslant 100$, $1 \leqslant h_{i,j} \leqslant 1000$.

Értékelés: a megoldás lényegét leíró dokumentáció 1 pontot ér. További 9 pont kapható arra a programra, amely a korlátoknak megfelelő bemenetekre helyes kimenetet ad 1 másodperc futásidő alatt. Részpontszám kapható arra a programra, amely csak kisebb N és M értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `is20.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

S. 119. Ifjú hercegünk egy hegyi ösvényen készül átkelni. Az ösvény mentén manók állnak lesben, akikkel semmiképp nem jó találkozni. Szerencsére a manók mindegyike olyan, hogy egy bizonyos színt nem lát. Hercegünk ezért varratott magának mindegyik színből egy-egy köpenyt, így ha a megfelelő manó előtt elhaladva azt viseli, akkor a manó nem veszi észre. A köpenyek mindegyike karra terítve is könnyen vihető, és emellett tetszőleges számú köpeny – akár az összes – egymásra fölvéve hordható. Ha a herceg már visel egy vagy több köpenyt, akkor a következőt azok fölé tudja venni. Vetkőzéskor mindig a legfelső köpenyt tudja levenni, tehát ha szüksége van egy most nem legfelül viselt köpenyre, akkor az összes fölötte lévőt le kell vennie.

A herceg azt is megtudta, hogy milyen sorrendben következnek az egyes manók az ösvény mentén. Mivel nem szeretne sokszor öltözni, ezért szeretné tudni, hogy hogyan juthat túl a lehető legkevesebb számú öltözéssel az ösvényen. Az út megkezdése előtt nem viseli egyik köpenyt sem, és az ösvény után sem, tehát leveszi, ami még rajta van. Minden köpeny föl- vagy levétele egy öltözésnek számít.

A feladatot megoldó program olvassa be a standard bemenetről a manók N számát, illetve az általuk nem látott színek Z számát, majd a következő sorból N számú pozitív egészet: az i-edik szám az i-edik manó által nem látott szín m_i sorszáma. A program írja a standard kimenetre az ösvényen való áthaladáshoz szükséges legkevesebb öltözések számát.

Példák:

Bemenet	**Kimenet**
6 4 1 2 2 3 4 3	8
10 5 1 3 2 3 2 1 3 2 2 1	12

Korlátok: $1 \leqslant Z \leqslant N \leqslant 30$, $1 \leqslant m_i \leqslant Z$.

Értékelés: a megoldás lényegét leíró dokumentáció 1 pontot ér. További 9 pont kapható arra a programra, amely a korlátoknak megfelelő bemenetekre helyes kimenetet ad 1 másodperc futásidő alatt. Részpontszám kapható arra a programra, amely csak kisebb N értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `s119.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

❄

A feladatok megoldásai regisztráció után a következő címen tölthetők fel:
`https://www.komal.hu/munkafuzet`
Beküldési határidő: 2017. november 10.

❄