

A. 721. Legyen $n \geq 2$ pozitív egész szám, továbbá a_1, a_2, \dots, a_n olyan pozitív valós számok, melyek összege 1, négyzetösszege pedig S . Mutassuk meg, hogy ha $b_i = \frac{a_i^2}{S}$ ($i = 1, \dots, n$), akkor tetszőleges $r > 0$ mellett

$$\sum_{i=1}^n \frac{a_i}{(1-a_i)^r} \leq \sum_{i=1}^n \frac{b_i}{(1-b_i)^r}.$$

✱

Beküldési határidő: 2018. április 10.

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

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

✱



Informatikából kitűzött feladatok

I. 451. Kukac-robot egy számsoron szeretne végighaladni, végigkúszni. A robot onnan kapta a nevét, hogy mozgása a kukacokéra emlékeztet. A kúszás két fázisból áll: először összehúzza magát úgy, hogy testének első része helyben marad a számsoron és a végét előre húzza, amíg csak lehet; majd másodszor fordítva, testének utolsó pontja marad egy helyben és előre kinyúlik, amíg a szabályok engedik.

Szabályok:

- Összehúzott állapotban a Kukac-robot alatt lévő számok összege legalább K , és a lefedett számok száma kettőnél nem lehet kevesebb.
- Kinyújtott állapotban legfeljebb L lehet az alatta levő számok összege és nem nyúlhat öt számnál hosszabban.
- A Kukac-robot induló helyzete: az első két számon helyezkedik el összehúzott állapotban. (A kezdőállapotra a szabályokat nem kell vizsgálni.)
- Beérkezésnek számít az a kinyújtott állapot, amikor a számsor utolsó tagját lefedi.

Készítsünk programot **i451** néven, amely meghatározza, hogy Kukac-robot végig tud-e menni a számsoron és ha igen, akkor legkevesebb hány lépésben.

A program standard bemenetének első sorában 3 szám van: N ($10 \leq N \leq 10\,000$) a számsor hossza, K ($2 \leq K \leq 20$) összehúzott állapotban a Kukac-robot alatti és L ($K < L \leq 45$) a kinyújtott állapot alatti számok összege. Az ezt követő sorban a számsor tagjait adjuk meg szóközzel elválasztva, azaz N darab számot x_i ($0 \leq x_i \leq 9$).

A program írja ki a standard kimenetre, hogy legkevesebb hány megnyúlás alatt kúszik át a Kukac-robot a számsoron. Ha nem tud a számsoron végigkúszni, akkor a kimenet legyen -1 .

Példa a bemenetre	Kimenet
12 10 30 6 7 8 0 4 9 9 8 9 1 1 8	5
10 9 17 2 8 0 2 2 4 8 8 2 1	-1

Beküldendő egy tömörített `i451.zip` állományban a program forráskódja és rövid dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I. 452 (É). A feladat a brit Nemzeti Statisztikai Iroda által 2017. július 10-én közzétett, az Egyesült Királyságban dolgozó kelet-európai vendégmunkásokról szóló felmérésének feldolgozása táblázatkezelő program segítségével.

- Nyissuk meg a honlapunkról letölthető `eu_angliafelmeres.txt` tabulátorokkal tagolt UTF-8 kódolású szövegfájlt a táblázatkezelő program munkalapjára az A1-es cellától kezdődően. Munkánkat `i452` néven mentjük el a táblázatkezelő alapértelmezett formátumában.
- Írassuk ki képlet segítségével a D2:D29-es tartományban az eredeti forrásban használt jelöléseket (EU8, EU15, EU Other) az alábbiak szerint:

EU8	2004-ben csatlakozott országok
EU15	2004 előtt csatlakozott országok
EU Other	más évben csatlakozott országok, valamint Málta és Ciprus

- A 31. sortól kezdődően a forrás többi részében az angol országneveket cseréljük le a megfelelő magyar országnevekre.
- Az egyes országokat a brit Nemzeti Statisztikai Iroda több szempont alapján hasonlította össze. Mi hét szempontot fogunk vizsgálni, melyeket az A oszlopban lévő **Összehasonlítás** szó vezet be. Számozzuk meg ezeket 1-től kezdődően a következő formátumban: 1. összehasonlítás, . . . , az utolsó a 7. összehasonlítás legyen.
- Az 1. összehasonlítás az EU8 országokból érkezett vendégmunkások számának életkor szerinti eloszlását tartalmazza. Határozzuk meg korosztályonként a 39. sorban, hogy melyik országból érkezett a legtöbb vendégmunkás. Jelenítsük meg az ország neve alatt a 40. sorban, két tizedesjegy pontossággal, hogy ez az adott korosztály hány százaléka.
- A 2. összehasonlítást követő A47-es cellába számítsuk ki, hogy összesen hány ezer fő érkezett Angliába 2011-ig az EU8 és EU14 országaiból.
- A 3. összehasonlítás C50-es cellája a 16–64 éves korosztály létszámát tartalmazza. Az alatta lévő sorokban számítsuk ki, hogy hány főt jelentenek a megadott százalékok országonként.

8. A 4. összehasonlítás alatt feltételes formázás segítségével jelöljük meg piros háttérrel azon iparág nevét, ahol a legnagyobb, és kék színnel, ahol a legkisebb az eltérés az EU8 és a többi ország munkavállalói között.
9. A J60-as cellába írjuk be: „EU8 országon kívül”. Számítsuk ki az alatta lévő cellákban a „Kiskereskedelem és vendéglátás”, a „Gyári munka”, valamint a „Pénzügy” területén mennyi az EU8 országain kívüli országokból érkezett vendégmunkások száma. (Szlovénia nem szerepel a felsorolásban jogi problémák miatt.)
10. Az 5. összehasonlítás adatai alapján határozzuk meg, hogy melyik két ország között volt legtöbbször a legnagyobb különbség az egyes utazási típusokat tekintve. Írjuk ki képlettel a két ország nevét az A77-es cellába, valamint az eltérések számát az A78-as cellába.
11. A 6. összehasonlításnál – az alábbi mintának megfelelően – válaszoljunk a következő kérdésre: az Anglián kívül, más EU-s országokban élő angol népességnek hány százaléka van 15–64 év között, és ezek hány százaléka van abban az országban, ahol egyébként a legtöbb angol él, és ez melyik ország.

	A	B	C
79	6. összehasonlítás		
80	Angol nemzetségűek az EU más országaiban, év szerinti megoszlásban (fő)	kevesebb, mint 15 éves	15-29 év között
81	Csehország	294 fő	357 fő
82	Magyarország	247 fő	729 fő
83	Lengyelország	171 fő	281 fő
84	Szlovákia	247 fő	198 fő
85	Észtország	44 fő	132 fő
86	Szlovénia	34 fő	23 fő
87	Lettország	17 fő	39 fő
88	Litvánia	18 fő	28 fő
89	Az összes kint élő népességnek	-a van 15-64 év között, amiből	%
		-ban.	

12. A 7. összehasonlításnál a 28 napnál rövidebb utazásokat hasonlítjuk össze. Adjuk meg az A97-es cellában, hogy melyik ország esetén legkisebb az eltérés az oda beutazó angol nemzetségűek és az onnan Angliába beutazók száma között. (Az Angliába érkező utazók utazásainak számát az 5. összehasonlításnál találjuk.)
13. Készítsünk diagramot, amely összehasonlítja az angol nemzetségűek Litvániába, illetve a litvánok Angliába történő, 28 napnál rövidebb utazásainak számát. A diagram az összehasonlítást „vakációk”, „látogatások”, „üzleti út”, és „egyéb” kategóriánként szemléltesse. A diagram háttere legyen Litvánia zászlaja, a betűszínt úgy válasszuk meg, hogy minden adat olvasható legyen.

Beküldendő egy tömörített i452.zip állományban a megoldást adó táblázatkezelő munkafüzet és egy rövid dokumentáció, amely megadja a felhasznált táblázatkezelő nevét és verzióját.

Forrás: <https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/internationalmigration/articles/livingabroad/migrationbetweenbritainandtheeu8> (utolsó letöltés: 2017-11-26).

I. 453. Feladatunk egy síkbeli bolyongás szimulációja és az eredményének megmutatása. Egy $N \times M$ méretű négyzetháló celláiban egész számok vannak. A cellák jelentős részében a 0, míg néhány cellában a -5 vagy $+5$ található. Ezeket a cellákat megjelöljük, és a bennük lévő számot rögzítjük, vagyis nem fog változni

a szimuláció során. A kezdetben nulla értékű cellák mindegyikéből elindulunk, és egy ideig véletlenszerűen bolyongunk a cellák között, amíg meg nem érkezünk egy megjelölt cellába. Ekkor a kezdetben nulla értékű cella számához hozzáadjuk annak a megjelölt cellának a számát, ahová eljutottunk.

Ezt a bolyongást minden, kezdetben nulla értékű cellából S -szer végezzük el, tehát az ilyen cellák számértéke S -szer fog változni. A bolyongás során minden esetben egy celláról egy vele csúcsban vagy oldalánál érintkező szomszédos cellára lépünk. Ha az S -szeri bolyongást minden nem megjelölt cellára elvégeztük, akkor a szimuláció véget ér. Ekkor minden megjelölt cella értékét megszorozzuk S -sel. Az így kialakult számértékekhez rendeljük lineárisan egy színskáláról színeket, és a kapott képet jelenítsük meg grafikusan.

A program bemenete a négyzetháló mérete ($50 \leq N, M \leq 100$), az egy cellából induló bolyongások száma ($50 \leq S \leq 1000$), valamint a nem nulla kezdőértékű cellák száma ($1 \leq Z \leq 100$), és soronként egy-egy megjelölt cella koordinátái és -5 vagy $+5$ értéke.

A feladat megoldásaként a versenykiírásban szereplő eszközökkel elkészíthető alkalmazások mellett a webes vagy mobil applikációkat is elfogadjuk. A bemeneti adatok egyszerű bevitelét az alkalmazás jellegétől függően lehessen megadni.

Beküldendő egy `i453.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 fejlesztői környezetben fordítható.

I/S. 25. Egy cégnél N ember dolgozik összesen P projekten. Egy ember több projektcsoporthoz is tagja lehet, de egynek biztosan tagja. A vezetőség meg akarja hívni egy találkozóra a lehető legtöbb olyan munkatársat, akik közül senki nem dolgozik egy másik meghívottal sem közös projekten. Sajnos a vezetőség nem tudja megállapítani, hogy legfőljebb hány meghívott lesz, ezért segítségünket kérte. Készítsünk programot, amely a projektcsoporthoz tagjainak ismeretében meghatározza a meghívottak legnagyobb számát. A munkatársakat pozitív egész számokkal azonosítjuk.

A program standard bemenete N és P értéke, majd a következő P sor mindegyikében egy-egy projekten dolgozó munkatárs azonosítója szerepel szóközzel elválasztva. A program adja meg a standard kimeneten a meghívottak létszámának maximumát.

Példa (az újsor karaktereket / jelöli):

Bemenet	Kimenet
8 6 / 1 2 / 7 2 8 / 4 5 / 6 4 3 / 5 7 / 1 5 /	3 /

Korlátok: $2 \leq N \leq 1000$, $2 \leq P \leq 1000$.

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 P értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `is25.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

S. 124. Kertész barátunk telke téglalap alakú, oldalainak hossza a és b egység. A kertre terítsünk gondolatban egy egység oldalú négyzethálót. Barátunk szeretné a kert bizonyos részeit egység nagyságú, négyzet alakú járólappal lefedni, hogy a kertben utakat hozzon létre. A járólapok a négyzetháló egy-egy négyzetében helyezkednének el. A kert egyik kiválasztott sarkában lenne az első járólap, és a szemközti sarkában az utolsó. A járólapokon szépen át lehetne sétálni az első járólapról indulva a szemközti sarokba. Szabály, hogy az egyik járólapról a másikra csak akkor léphetünk, ha oldalaik érintkeznek, illetve séta közben mindig távolodnunk kell az első járólaptól. A kertész szeretné úgy elhelyezni a járólapokat, hogy a lehetséges séták száma éppen egy előre kitalált k érték legyen. Ugyanakkor szeretné az utak létrehozásához a legkevesebb járólapot vásárolni, de sajnos nem tudja meghatározni, hogy ez mennyi legyen. Készítsünk programot, amely megadja ezt a legkisebb értéket.

A program standard bemenete a telek a és b mérete, illetve a lehetséges séták

	→	→	→	→	
	↑				
	↑				
→	↑				

k száma. A program adja meg a standard kimeneten a k -féleképp sétálható utak közül a legkevesebb járólappal rendelkező utakhoz szükséges járólapok számát, illetve -1 -et, ha nem lehetséges k -féleképp bejárható úthálózatot készíteni. A mellékelt *ábra* a lenti példa egy lehetséges megoldását szemlélteti, illetve egy sétát mutat a nyilakkal.

Példa:

Bemenet	Kimenet
4 6 10	14

Korlátok: $2 \leq a, b \leq 20, 2 \leq k \leq 10^6$.

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. Részpontoszám kapható arra a programra, amely csak kisebb bemeneti értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `s124.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: 2018. április 10.

