A. 712. Egy pozitív valós számokból álló szigorúan monoton növekvő $a_1, a_2, \ldots$ sorozatot *törpének* nevezünk, ha tetszőleges $c > 0$-hoz megadható N, melyre $a_n < cn$ áll fenn $n = N, N+1, \ldots$ esetén. Továbbá azt mondjuk, hogy a_n *sipka,* ha $1 \leqslant i \leqslant n-1$ esetén $a_{n-i} + a_{n+i} < 2a_n$.

Igaz-e, hogy minden törpe sorozatnak végtelen sok sipkája van?

Beküldési határidő: 2018. január 10.
Elektronikus munkafüzet: `https://www.komal.hu/munkafuzet`
Cím: KöMaL feladatok, Budapest 112, Pf. 32. 1518

Informatikából kitűzött feladatok

I. 442 (É). Zümike, a kedves konyhalégy, és Ati bácsi, a gazda időnként üldözős játékot játszanak. A játék úgy indul, hogy miután Zümike leszállt a konyhaasztalra, addig billegeti a szárnyait, amíg Ati bácsi elő nem veszi a légycsapót. Amint Zümike meglátja a felé közeledő légycsapót, felszáll, és addig repdes össze-vissza a konyhában, amíg Ati bácsi el nem veszíti a nyomát. Ilyenkor Zümike valahová leszáll, és várja, hogy Ati bácsi újra megtalálja. A játék rendszerint úgy ér véget, hogy Ati bácsi megunja a keresést és elteszi a légycsapót.

Egyik alkalommal Maci, a sarokban lakó keresztespók feljegyezte a játék menetét: egy hosszú pókfonálra másodpercenként 0-t vagy 1-et írt. 0-t akkor, ha az adott másodperc elején Zümike pihent, 1-et, ha repült. A rögzített adatokat a `naplo.txt` fájl tartalmazza. Olvassuk be a fájl adatait, és válaszoljunk az alábbi kérdésekre (a számozást minden esetben kezdjük 1-től):

1. Hány alkalommal szállt fel Zümike?
2. Hány percig tartott összesen a játék?
3. A játék során hány másodpercet repült Zümike összesen? Mennyi ideig tartott átlagosan egy repülés? Az eredményt két tizedesjegyre kerekítve írassuk ki.
4. Néha Zümike akkor is megijedt és felszállt, amikor Ati bácsi nem volt a közelében. Ilyenkor legfeljebb 3 másodpercet töltött a levegőben, egyébként azonban jóval többet. Hány „téves riasztása” volt Zümikének?
5. Hányadik másodpercben kezdődött és milyen hosszú volt Zümike leghosszabb ideig tartó repülése? Ha több ilyen is volt, mindegyiket jelenítsük meg.
6. Zümike „sikernek” érzi, ha két repülési szakasz között többet tudott elrejtőzve pihenni, mint a két repülési szakasz időtartama (külön-külön). Hányadik másodperceben kezdődött a legrövidebb „siker”? (Feltételezhetjük, hogy legfeljebb egy megoldás van.)
7. Ati bácsi „jó sorozatnak” tartja, ha sikerült elérnie, hogy Zümike egymást követő levegőben töltött időszakai egyre hosszabbak legyenek. Melyik másod-

percben kezdődött, és hány tagból állt a leghosszabb „ jó sorozat”? (Feltételezhetjük, hogy egy megoldás van.)

8. Készítsünk `sorrend.txt` néven szövegfájlt, amelybe soronként kiírjuk Zümike repülésének időtartamait növekvően rendezve, és mindegyik mellett

   `óra:perc:másodperc - óra:perc:másodperc`

   alakban feltüntetjük, hogy az a játék során mikor kezdődött, illetve fejeződött be. Ha egy adott időtartamhoz több alkalom is tartozik, akkor mindegyiket tüntessük fel egy-egy szóközzel elválasztva.

Beküldendő egy `i442.zip` tömörített állományban a program forráskódja és dokumentációja, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

Letölthető fájl: `naplo.txt`

I. 443. Bergengóciában a király – érezve, hogy megöregszik – megajándékozná egy vagy több fiát adott összértékű kincseivel. Egyik kincsesládájából legfeljebb három különböző értékű tárgyat választana, így egyenlő ajándékokkal biztos nem lepi meg három deli fiát. A király már régóta gondolkodik ezen, így kitalálta, hogy pontosan hány tallér értékben fog ajándékozni, se többel, se kevesebbel. Ha ezt nem lehet pontosan megvalósítani, akkor inkább lemond az ajándékozásról.

A kincstárnok pontosan feljegyezte, hogy a ládában 10 különböző, ismert értékű aranytárgy van. Segítsünk a kincstárnoknak egy táblázatkezelő segítségével az elosztásban. Készítsünk táblázatot, amelybe ha beírjuk a király által elosztandó tárgyak összértékét tallérban, és a ládában lévő 10 aranytárgy értékét, akkor megadja, hogy elvégezhető-e az elosztás, és ha igen, akkor hogyan. A megoldásban saját függvény vagy makró nem használható. Munkánkat `i443` néven mentsük el a táblázatkezelő alapértelmezett formátumában.

	A	B	C	D	E
1	301		Egyedül	Ketten	Hárman
2				1 300	1 100 200
3					
4	Tárgyak értéke				
5	1				
6	11				
7	50				
8	100				
9	161				
10	189				
11	200				
12	300				
13	378				
14	412				
15					

Alakítsuk ki a minta szerinti táblázatszerkezetet. Segédszámításokat az E oszloptól jobbra végezhetünk, amelyek értelmezését feliratokkal segítsük. A megoldás során képletet, függvényt, hivatkozást használjunk, hogy a király által meghatározott összeg és a láda kincseinek változását a megoldás kövesse.

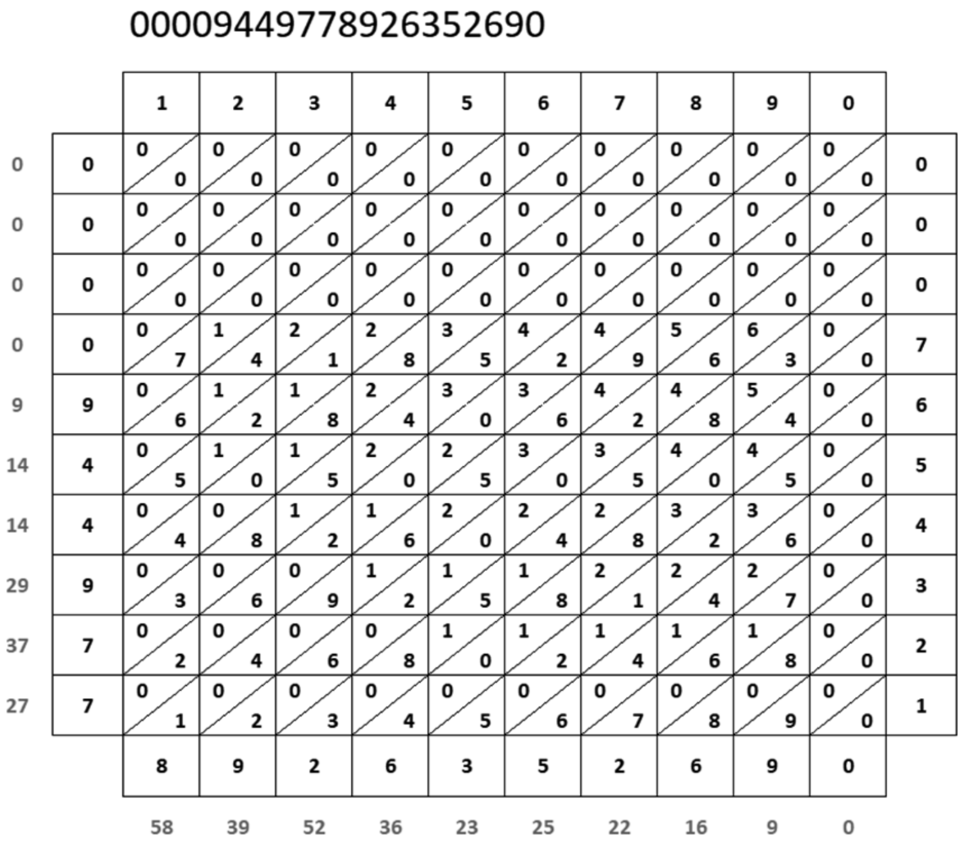
Az A1 cellába, a király parancsának megfelelően, az elosztandó tárgyak összes értékét tallérban írjuk be. Alatta soroljuk fel a kincsesládában lévő aranytárgyak értékét.

A C2:E2 tartomány celláiban jelenítsük meg, hogy egy, kettő vagy három tárgy ajándékozásakor milyen értékű ékszereket kell a ládából kivenni. Ha valamelyik elosztás nem valósítható meg, akkor a cella üresen jelenjen meg, különben a tallérban értendő értékeket szóközzel válasszuk el.

Beküldendő egy tömörített `i443.zip` állományban a táblázatkezelő munkafüzet, illetve egy rövid dokumentáció, amelyben szerepel a megoldáskor alkalmazott táblázatkezelő neve, verziószáma.

I. 444. A számok szorzására használt *gelosia-módszer* vagy *rácsos módszer* először Indiában, Perzsiában, Kínában és az arab kultúrkörben jelent meg. Európában a XIV. század elején vált ismertté, nevét a korai olasz építészet geometrikus, osztott rácsos ablakkereteinek nevéből kapta.

Az alábbi *ábrán* az $1\,234\,567\,890 \cdot 7\,654\,321 = 9\,449\,778\,926\,352\,690$ szorzat meghatározását látjuk gelosia-módszerrel:



A módszer lényege a következő. Elhelyezzük vízszintesen (balról jobbra haladva) a szorzandót, függőlegesen (felülről lefelé haladva) a szorzót, majd kiegészítjük az ábrát függőleges és vízszintes vonalakkal a mintának megfelelően. Így egy mátrixot kapunk. A mátrix celláit az átlók meghúzásával az ábrának megfelelően felosztjuk.

A mátrix minden egyes mezőjébe beírjuk a hozzá tartozó oszlop és sor szorzatát úgy, hogy az egyeseket az alsó, a tízeseket a felső háromszögbe írjuk.

A következő lépésben átlónként összeadjuk az átlókon elhelyezkedő számokat. Ezt a mátrix mentén, a jobb alsó saroktól kezdve a bal felső sarok felé haladva végezzük. Az összeg utolsó jegyét a mátrix mellé írjuk, a többi jegyből képezett számot pedig a következő átlós összeghez adjuk hozzá.

A szorzatot a mátrix mentén a bal felső sarokból indulva a jobb alsó sarok felé haladva olvashatjuk le.

Készítsünk táblázatkezelővel táblázatot vagy írjunk programot, amely két, legfeljebb 10 jegyű számot a fenti eljárással összeszoroz. Mindkét esetben gondoskodjunk a módszer szemléltetésére a fenti ábrának megfelelő megjelenítéséről is!

Beküldendő egy `i444.zip` tömörített állományban a program forráskódja vagy a munkafüzet, továbbá program esetén a dokumentáció, amely tartalmazza, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

I/S. 22. Egy N hosszúságú, egységnégyzetekből álló szalagon korongok helyezkednek el. Minden korong egy-egy négyzetben van, ugyanakkor egy négyzetben legföljebb egy korong található, vagy a négyzet üres. A szalag első P négyzetében piros korongok találhatók, míg utolsó K négyzetében kékek. Célunk az, hogy a piros, valamint a kék korongokat a szalag tőlük távolabbi végéhez sorakoztassuk föl. Akkor teljesítettük a feladatunkat, ha a szalag első K számú négyzetében kék, az utolsó P számú négyzetében piros korong található.

A korongokkal kétféle mozgatást tudunk végezni bármelyik irányba. Az egyik lehetőség, hogy egy koronggal a szomszédos üres négyzetre lépünk. A másik lehetőség, hogy egy koronggal a szomszédos korongon át a következő szomszédos, üres mezőre ugrunk. Több korongot vagy üres négyzetet nem lehet átugrani, de bármelyik korong átugorhatja bármelyik másikat függetlenül a színétől.

Készítsünk programot, amely megadja, hogy legkevesebb hány mozgással teljesíthetjük a célt, amely mindig teljesíthető. A program standard bemenete az N, K és P egészek. A program standard kimenete egy sorban a szükséges legkevesebb mozgások, majd azon belül az ugrások és a lépések száma.

Példák:

Bemenet	**Kimenet**
`3 1 1`	`3 1 2`
`6 2 2`	`10 6 4`

Korlátok: $1 \leqslant P < P + K < N \leqslant 1000$.

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 $N \leqslant 10$, $N \leqslant 30$, $N \leqslant 100$ értékek esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `is22.zip` tömörített állományban a megoldást leíró dokumentáció és a program forráskódja.

S. 121. Egy $N \times M$-es négyzetháló mezőire pozitív egész számokat írtunk 1-től 6-ig. A bal felső négyzeten elhelyeztünk egy kockát, amelynek oldalaira szintén egy-egy pozitív egész számot írhatunk 1-től 6-ig. A négyzetháló négyzetei és a kocka oldalai azonos méretűek, a kocka bármely lapja pontosan a négyzetháló négyzetére illeszkedik. A kockát el kell juttatni a négyzetháló jobb alsó sarkába a következő két művelet megfelelő sorrendben történő többszöri alkalmazásával:

1. Az álló kocka a négyzethálóra merőleges forgástengelye körül derékszögben (akár többször is) elforgatható.
2. A kocka az egyik négyzethálón fekvő éle körül derékszögben elforgatható úgy, hogy egy másik lapján álljon, de csak akkor, ha azonos számok kerülnek egymásra.

Tehát a kocka minden helyzetében – induláskor is – olyan lapon áll, amelynek száma egyezik annak a négyzetnek a számával, amelyen áll. A négyzetháló négyzetein található számok ismeretében adjuk meg, hogy milyen számokat írjunk a kocka lapjaira, hogy az eljusson a jobb alsó négyzetre.

A program standard bemenete N és M, majd a következő N sor mindegyikében M darab szám van 1-től 6-ig, amelyek a négyzetháló számai. A program standard kimenete vagy 0, ha nem létezik megfelelő feliratozás a kockára, vagy a kockán szereplő számok. A kezdő helyzetben álló kocka oldalain az alsó, a felső, majd rendre a négy egymáshoz csatlakozó oldalán lévő számokat adjuk meg. Több megoldás esetén elegendő egy alkalmas feliratozást megadni.

Példa bemenet	**Példa kimenet**

`4 5`
`1 2 3 1 4`
`5 3 6 3 5`
`3 4 1 1 4`
`3 4 3 6 4`
	`1 3 4 6 2 4`

Korlátok: $1 \leqslant N, M \leqslant 100$.

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, M érték esetén ad helyes eredményt 1 másodpercen belül.

Beküldendő egy `s121.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. január 10.

❄