Informatikából kitűzött feladatok

I. 550. Egy bolha tartózkodik a számegyenes 0 pontján. Kétféle mozgásra képes: B hosszút tud ugrani balra, vagy J hosszút tud ugrani jobbra ($1 \leqslant B, J \leqslant 100$ egész számok). Adjuk meg, hogy eljuthat-e a bolha egy tetszőleges C ($-100 \leqslant C \leqslant$ $\leqslant 100$) számhoz. Ha eljuthat, akkor adjuk meg a legrövidebb ugrássorozat hosszát, valamint azt, hogy hány balra és hány jobbra ugrással lehet eljutni 0-tól C-ig.

Készítsünk programot, amely a standard bemenet első sorából beolvassa B, J és C értékét, majd a standard kimenet egyetlen sorába írja a legrövidebb ugrássorozat hosszát, a balra, valamint a jobbra ugrások számát, illetve 0-t, ha a C számhoz a bolha nem tud eljutni. Ha több ugrássorozat van, amellyel a C számhoz a legkevesebb ugrással el lehet jutni, akkor bármelyik megadható.

Példák:

Bemenet	Kimenet
`5 3 10`	`6 1 5`
`21 73 50`	`20 15 5`
`48 82 73`	`0`

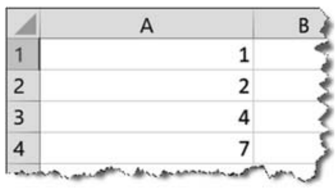
Beküldendő egy tömörített `i550.zip` mappában a megoldást adó program forráskódja és egy rövid dokumentáció, amely megadja, hogy a forrásállomány melyik fejlesztői környezetben fordítható.

(*A 2021 októberében kitűzött* **K. 701.** *feladat alapján*)

I. 551. Különböző vastagságú falakat szeretnénk készíteni. Ehhez háromféle, 1, 2 és 3 cm vastagságú lapok állnak rendelkezésre.

Táblázatkezelő program segítségével oldjuk meg a falak készítéséhez használandó lapok számának és sorrendjének számításához kapcsolódó következő feladatokat.

1. Hozzuk létre a táblázatkezelőben a **fal** nevű állományt a program alapértelmezett formátumában.
2. Nevezzük át az első munkalapot **sorrend** névre.
3. Határozzuk meg az A1:A60-as tartomány celláiban, hogy hányféle lényegesen különböző sorrendben lehet balról jobbra összeilleszteni a lapokat úgy, hogy a fal vastagsága az adott cella sorának értékével egyezik meg. (Lényegesen különböző a sorrend, ha az egymás után elhelyezett lapok vastagsága legalább egy helyen eltér, például 3 cm-es falvastagságnál 4 lényegesen különböző összeállítás van: $1, 1, 1$; $1, 2$; $2, 1$ és 3.)



Minta a **sorrend** *munkalaphoz*

4. Hozzuk létre a **max10** nevű munkalapot és állítsuk be a mintaképen látható szürke hátterű cellákat és formátumukat.

5. Ezen munkalap B, C és D oszlopában a második sortól kezdve soroljuk fel, hogy hány 3, 2 és 1 cm-es lapot használhatunk fel a fal elkészítéséhez, ha a fal vastagságát az A2 cellába írt pozitív egész szám adja és mindhárom vastagságból legfeljebb 10–10 lapot használhatunk fel.
6. A kép A3 cellájában olvasható üzenet csak akkor jelenjen meg, ha a megadott falvastagság eléréséhez legalább egyféle lapból 10-nél több darabra lenne szükség. Ebben az esetben a többi cella maradjon üres a B–F oszlopokban a második sortól kezdve.

	A	B	C	D	E	F	G
1	méret:	3	2	1	sorrend	összes sorrend:	
2	100						
3	Túl vastag!						
4							
5							

7. Az E oszlopban határozzuk meg, hogy az adott darabszámokból hány lényegesen különböző összeállítás lehetséges. Például 14-es falvastagság összeállítható többek között 4 db 3 cm-es és 1 db 2 centiméteres lapból. Ezek 5 lehetséges sorrendben tehetők egymás után: $3, 3, 3, 3, 2$; $3, 3, 3, 2, 3$; $3, 3, 2, 3, 3$; $3, 2, 3, 3, 3$ és $2, 3, 3, 3, 3$. Ezért ebben a sorban az E oszlopba 5 kerül, természetesen képlet felhasználásával.
8. Az F2 cellában összesítsük az E oszlopban felsorolt esetenkénti sorrendeket.

	A	B	C	D	E	F
1	méret:	3	2	1	sorrend	összes sorrend:
2	14	4	1	0	5	3110
3		4	0	2	15	
4		3	2	1	60	
5		3	1	3	140	
6		3	0	5	56	
7		2	4	0	15	
8		2	3	2	210	
9		2	2	4	420	
10		2	1	6	252	
11		2	0	8	45	
12		1	5	1	42	
13		1	4	3	280	
14		1	3	5	504	
15		1	2	7	360	
16		1	1	9	110	
17		0	7	0	1	
18		0	6	2	28	
19		0	5	4	126	
20		0	4	6	210	
21		0	3	8	165	
22		0	2	10	66	
23						
24						

Minta a **max10** *munkalaphoz*

Segédszámításokat mindkét munkalapon a P oszloptól kezdődően végezhetünk. A megoldáshoz makró vagy más program nem használható, csak a táblázatkezelő beépített függvényei.

Beküldendő egy `i551.zip` tömörített állományban a munkafüzet és egy rövid dokumentáció, amely megadja, hogy a megoldás milyen táblázatkezelő program melyik verziójában készült és egy kb. ötsoros magyarázat a megoldások módszeréről.

I. 552 (É). Az emberiség történelme során többféle számírás és számolási módszer alakult ki. A helyiértékes számrendszer használatának elterjedése előtt a számokat különféle jelekkel, jelcsoportokkal írták le. Az ókori görögöknél az i. e. V. században kialakult *alfabetikus* számírásban például az ABC betűi jelentették a számokat. A megfeleltetés alfa(A) = 1, béta(B) = 2, … egészen kilencig, majd ióta(I) = 10, kappa(K) = 20 stb. A betűkkel nem jelölt számokat a betűkből álló „szavak" segítségével adták meg: a szó értéke a benne szereplő betűk számértékének összegével volt egyenlő.

Alkalmazzuk az alfabetikus számítást az angol ABC betűire, vagyis A = 1, B = 2, … és végül Y = 600 és Z = 700. A következő feladatokban ezekkel a „számokkal" kell számolni és műveleteket végezni. A számokat minden esetben az angol ABC nagybetűivel jelöljük. A feladatok megoldása során törekedjünk a mintának megfelelő input/output megvalósítására. A bemeneteket nem kell ellenőrizni, azok a leírásnak megfelelő, helyes értékek. A kommunikáció során az ékezetmentes kiírás is elfogadott.

1. Írjuk ki az angol ABC nagybetűit.
2. Írjuk ki a fenti nagybetűkkel jelölt számokat.
3. Kérjünk be a felhasználótól egy számot jelentő betűt, és adjuk meg a számértékét.
4. Kérjünk be a felhasználótól egy számot jelentő szót (legföljebb 10 betű), és adjuk meg a számértékét.

A számok felírása nem egyértelmű, például a 31 felírható LA, AJK vagy akár JDGJ alakban is.

5. Kérjünk be a felhasználótól két számot jelentő szót (legföljebb 10 betű), és adjuk meg, hogy egyenlő értékű-e a két szó.
6. Kérjünk be a felhasználótól egy tízes számrendszerben felírt számot, és adjuk meg a neki megfelelő legrövidebb szavak egyikét.
7. Kérjünk be a felhasználótól egy tízes számrendszerben felírt számot, és adjuk meg a neki megfelelő hárombetűs szavak mindegyikét.

Minta:

```
1. feladat:
Az ABC: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
2. feladat:
A számok: 1 2 3 4 5 6 7 8 9 10 20 30 40 50 60 70 80 90 100 200 300
400 500 600 700 800
3. feladat:
Kérek egy betűt:K
```

```
Értéke: 20
4. feladat:
Kérek egy szót:EMU
Értéke: 345
5. feladat:
Kérek egy szót:KSM
Kérek egy másik szót:OLP
Egyenlőek
6. feladat:
Kérek egy számot:256
Egy legrövidebb szó: TNF
7. feladat:
Kérek egy számot:195
A számnak megfelelő hárombetűs szavak: ERS ESR RES RSE SER SRE
```

Beküldendő egy `i552.zip` tömörített állományban a megoldás forráskódja, valamint egy rövid dokumentáció, amely megadja, hogy a program melyik fejlesztői környezetben futtatható.

I/S. 58. Egy épület alaprajza egy N oszlopból és N sorból álló négyzetráccsal szemléltethető. A négyzetrács minden egységnégyzete kétféle lehet: foglalt, vagy szabad. Foglalt egységnégyzeteknek megfelelő területekre nem léphet senki, szabad egységnégyzeteknek megfelelő területeken viszont szabadon lehet sétálni.

Tűzriadó esetén az épület minden szabad négyzettel adott területén egy fluoreszkáló nyíl jelenik meg, amely mutatja, merre kell továbbhaladni, hogy biztonságosan elhagyhassuk az épületet. Egy nyíl négy különböző irányba mutathat az alaprajzon: fel, jobbra, le, balra.

Miután felfestésre kerültek a nyilak, tűzriadót tartanak az épületben, hogy kipróbálják, mindenhova jó irányba mutató nyíl került-e. A próba kezdetén minden szabad egységnégyzetnek megfelelő területen pontosan egy ember áll. Minden időegység alatt mindenki megpróbálja végrehajtani az alatta lévő nyíl által mutatott parancsot.

Az alaprajzot és a tűzriadó tervét egy T tömb írja le. Az alaprajzon az i-edik sor j-edik egységnégyzetének állapotát a $T[i][j]$ tömbelem értéke adja: '`*`' = foglalt, '`F`' = felfele nyíl, '`J`' = jobbra nyíl, '`L`' = lefele nyíl és '`B`' = balra nyíl.

Ha egy ember alatt lévő nyíl egy szabad területre mutat, akkor oda átlép az illető. Ha foglalt cellára mutat, akkor helyben marad. Ha az épületből kifelé mutat, akkor biztonságban kijut az épületből és többé nem megy vissza oda a próba során. A próba M időegységig tart. Adjuk meg, hogy M időegység után hányan vannak még az épületben.

A bemenet első sorában az N és M egész szám található. A következő N sor mindegyikében N karakter található: az i-edik sor j-edik eleme $T[i][j]$.

A kimenet egyetlen sorában egy szám szerepeljen: az M időegység után az épületben lévő személyek száma.

Példa:

Bemenet (a `/` sortörést helyettesít)	**Kimenet**
`4 6 / JJFF / F*BB / FLJJ / FBJB`	`6`

A kezdetben a $T[2][3]$, $T[2][4]$, $T[3][2]$, $T[4][2]$, $T[4][3]$, $T[4][4]$ egységnégyzeteken álló emberek maradnak az épületben 6 időegység után.

Korlátok: $1 \leqslant N \leqslant 750$, $T[i][j] \in \{$'`*`', '`F`', '`J`', '`L`', '`B`'$\}$. Időlimit: 0,3 mp.

Értékelés: a pontok 50%-a kapható, ha $1 \leqslant N \leqslant 30$.

Beküldendő egy `is58.zip` tömörített állományban a megfelelően dokumentált és kommentezett forrásprogram, amely tartalmazza a megoldás lépéseit, valamint megadja, hogy a program melyik fejlesztői környezetben futtatható.

S. 157. Álmos és Sára kaptak egy nagy tábla mogyorós csokoládét karácsonyra. A csokoládé $N \times N$ db négyzet alakú mezőből áll. Egyes mezőkben van mogyoró, más mezőkben nincs. Úgy szeretnék elosztani a csokoládét, hogy a kapott mogyorós részek számának különbsége minél kisebb legyen. Nem szeretnék azonban túl sok részre vágni a csokoládét, így először N hosszú és egy mező széles sávokra vágják, majd minden sávot középen kettévágnak és az egyik felét Álmos, a másikat Sára kapja (N páros szám). Azt, hogy ki melyik részt kapja, minden sávra külön-külön eldönthetjük. A sávokra vágást soronként és oszloponként is el lehet végezni, így mindkét lehetőséget meg kell vizsgálni.

Készítsünk programot, amely egy tábla mogyorós csokoládéra megadja, mennyi a legkisebb különbség, ami Álmos és Sára mogyorót tartalmazó mezőinek száma között lehet. Azt is adjuk meg, hogy ehhez soronként vagy oszloponként kell-e felvágni a csokoládét.

Bemenet: az első sor tartalmazza a méretet megadó N számot. A következő N sor mindegyike N számot tartalmaz. Ezek mindegyike 0, ha nincs mogyoró a mezőn és 1, ha van.

Kimenet: az első sorba egy `S` karakter írjunk, ha a legkisebb különbséget soronkénti felvágással is el lehet érni, különben pedig egy `O` karaktert. A második sorba az elérhető legkisebb különbség kerüljön.

Példa:

Bemenet (a `/` jel sortörést helyettesít)	**Kimenet** (a `/` jel sortörést helyettesít)
`4 / 1 0 1 0 / 1 1 0 1` `0 1 1 0 / 1 1 0 1`	`S / 0`

Magyarázat: A csokoládé mindkét felvágással igazságosan elosztható.

Korlátok: $2 \leqslant N \leqslant 500$, N páros. Időlimit: 1 mp.

Értékelés: a pontok 30%-a kapható, ha $N \leqslant 10$.

Beküldendő egy `s157.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: 2022. január 15.

❄