



**Eötvös Loránd Tudományegyetem**  
**Informatikai Kar**  
**Média- és Oktatásinformatika Tanszék**

---

**Tehetséggondozás az informatikában:**  
**TÁBLÁZATKEZELÉS**

**– SZAKDOLGOZAT –**

Készítette: **Molnár Katalin**  
(Tanári MA, matematika-informatika szak)

Témavezető: **Gudenus László**  
(Média- és Oktatásinformatika Tanszék, mestertanár)

**Budapest, 2012**



# EÖTVÖS LORÁND TUDOMÁNYEGYETEM

## EREDETISÉGNYILATKOZAT

### Tanári mesterszakos szakdolgozat tanulmány részéhez

A hallgató neve: Molnár Katalin

A hallgató EHA-kódja: MOKOABT.ELTE

A tanulmány címe: Tehetséggondozás az informatikában: TÁBLÁZATKEZELÉS

Az ELTE tanári mesterszakos hallgatójaként büntetőjogi felelősségem tudatában kijelentem és aláírással igazolom, hogy a szakdolgozat részét képező tanulmányom saját, önálló szellemi munkám, az abban hivatkozott, nyomtatott és elektronikus szakirodalom felhasználása a szerzői jogok általános szabályinak megfelelően történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- a szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- a tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén tanulmányom visszautasításra kerül, és ilyen esetben fegyelmi eljárás indítható.

Budapest, 2012.11.19.

.....  
aláírás

# TARTALOMJEGYZÉK

ELŐSZÓ.....	4
BEVEZETŐ .....	5
TÁBLÁZATKEZELÉS HALADÓ SZINTEN.....	7
<b>1. Az adatok importálása és előkészítése .....</b>	<b>7</b>
1.1. Importálás <i>txt</i> -ből vagy <i>docx</i> -ből .....	7
1. példa: Nemes 2010-2011 2. forduló, 4/C (A légielő pilótáinak kiképzése).....	7
2. példa: OKTV 2010-2011 1. forduló, 5/B (Tisztavetés).....	9
1.2. Importálás weblapból.....	11
3. példa: Nemes 2010-2011 3. forduló, 4/A (Eszmei érték) .....	11
1.3. Importálás <i>pdf</i> állományból .....	15
4. példa: Nemes 2009-2010 3. forduló, 4/C (Holdak) .....	16
1.4. Az importált adatok szerkezetének átalakítása függvényekkel .....	18
5. példa: OKTV 2009-2010 1. forduló, 5/H (Karácsonyi síverseny) .....	18
<b>2. Dátum- és időkezelés.....</b>	<b>20</b>
6. példa: Nemes 2010-2011 1. forduló, 5/A, B (A szabadságharc kronológiája).....	24
<b>3. Cellaformázás.....</b>	<b>27</b>
3.1. A tartalom igazítása .....	27
7. példa: Nemes 2010-2011 1. forduló, 5/C (A szabadságharc kronológiája).....	27
3.2. Sortörés cellán belül.....	28
8. példa: Nemes 2010-2011 2. forduló, 5/C (A légielő pilótáinak kiképzése).....	28
3.3. Karakterformázás .....	28
3.4. Oszlopszélesség, sormagasság.....	29
3.5. Formátumkódok .....	30
3.5.1. Mértékegységek .....	30
3.5.2. Számok formázása .....	32
3.5.3. Dátumformátumok.....	36
3.5.4. Időformátumok .....	37
3.5.5. Feltételek használata .....	38
9. példa: Egy saját példa: „ <i>többlet és hiány</i> ” típusú cellaformázás.....	38
3.5.6. Színek megadása.....	40
10. példa: Nemes 2009-2010 3. forduló, 5/E (Holdak) .....	41
3.6. Feltételes formázás .....	42
11. példa: Egy tetszőleges táblázat minden második sorának árnyékolása.....	45
12. példa: Nemes 2010-2011 1. forduló, 5/C (A szabadságharc kronológiája) .....	46
13. példa: Egy saját példa: Négyzetszámok .....	48
14. példa: OKTV 2009-2010 2. forduló, 3/O (Afrimpex) .....	50

<b>4. Adatok áttekintését segítő eszközök.....</b>	<b>54</b>
4.1. Rendezés .....	54
15. példa: Nemes 2009-2010 1. forduló, 5/A (Locsolóverseny).....	54
16. példa: Nemes 2008-2009 3. forduló, 4/F (ECDL).....	57
17. példa: Nemes 2009-2010 2. forduló, 4 (Damimpex).....	59
4.2. Autoszűrő.....	61
4.3. Irányított szűrő.....	62
18. példa: OKTV 2009-2010 1. forduló, 5/E (Karácsonyi ajándékok).....	63
19. példa: OKTV 2009-2010 1. forduló, 5/F (Karácsonyi ajándékok).....	64
20. példa: Egy saját példa: Jedlik-díj .....	66
4.4. Ismétlődések eltávolítása .....	70
4.5. Kimutatás .....	71
21. példa: Nemes 2009-2010 1. forduló, 5/D (Locsolóverseny).....	71
4.6. Részösszeg.....	74
22. példa: Nemes 2009-2010 1. forduló, 5/C (Locsolóverseny).....	75
<b>5. Diagram és formázása .....</b>	<b>77</b>
23. példa: OKTV 2010-2011 1. forduló, 4 (Virágkarnevál).....	77
24. példa: OKTV 2010-2011 1. forduló, 5/G (Tisztavatás).....	81
25. példa: OKTV 2009-2010 3. forduló, 4 (Szimuláció).....	82
26. példa: OKTV 2010-2011 2. forduló, 3/I (Űrhajósok kiképzése).....	85
<b>6. Helyettesítő karakterek használata.....</b>	<b>89</b>
6.1. A <i>Keresés és csere</i> ablakban.....	89
6.2. Függvények feltételt megadó argumentumában .....	90
6.3. Cella tartalmát vizsgáló, IGAZ/HAMIS értéket visszaadó képletben.....	90
6.4. Feltételes formázásnál.....	90
6.5. Szűrők feltételeként .....	90
<b>7. Függvények.....</b>	<b>91</b>
7.1. Összetett képletek szerkesztése .....	92
27. példa: Egy saját példa: Cukrászda.....	93
28. példa: Nemes 2009-2010 2. forduló, 4 (Damimpex).....	94
7.2. Adatbázisfüggvények.....	97
29. példa: Nemes 2009-2010 3. forduló, 5/D (Holdak).....	97
7.3. Tömbképletek .....	99
30. példa: Nemes 2009-2010 3. forduló, 5/D (Holdak).....	101
31. példa: OKTV 2010-2011 1. forduló, 5/I (Tisztavatás).....	102
32. példa: Nemes 2010-2011 3. forduló, 4/E (Eszmei érték).....	105
7.4. Szövegkezelés .....	106

<b>8. Lehetőségelemzés .....</b>	<b>107</b>
8.1. Célérték keresése .....	107
33. példa: Nemes 2010-2011 2. forduló, 4/J (A légielő pilótáinak kiképzése) .....	107
8.2. Solver .....	109
34. példa: Szendvicsek .....	111
35. példa: Nemes 2008-2009 3. forduló, 5 (Regionális forduló) .....	115
8.3. Adattábla .....	121
36. példa: Nemes 2010-2011 2. forduló, 4/K (A légielő pilótáinak kiképzése) .....	122
<b>9. Adatok érvényessége és lapvédelem .....</b>	<b>123</b>
37. példa: Egy saját példa: Virágrendelés .....	124
<b>10. Vezérlők .....</b>	<b>128</b>
10.1. Beviteli lista .....	128
38. példa: Nemes 2009-2010 3. forduló, 5/E (Holdak) .....	128
<b>KONKLÚZIÓ .....</b>	<b>130</b>
<b>TÁRGYMUTATÓ .....</b>	<b>131</b>
<b>IRODALOMJEGYZÉK.....</b>	<b>132</b>

## Előszó

Írásom alapja az *Eötvös Loránd Tudományegyetemre* beadott szakdolgozatom, melyet a *matematika-informatika tanári MA* szak elvégzésekor készítettem. Azóta történhetek rajta módosítások, és a tárhelyen található feladatmegoldások is gyarapodhattak. A legfrissebb példány elérése: <http://alkverseny.elte.hu>.

A segédanyag – mind ez a dokumentum, mind a hozzá tartozó feladatmegoldások – **mindenki számára ingyenesen használható, a forrás megjelölése mellett.** Amennyiben munkámat értékesnek találja, vagy szeretne támogatni annak továbbfejlesztésében, kérem, segítsen diákhitelém törlesztésében, akár csak egy csekély összeggel is:

- Kedvezményezett számlaszáma: 10032000-01800126-00000000
- Kedvezményezett neve: Diákhitel Központ Zrt
- Közlemény: elotorlesztes 13372487 Molnar Katalin

Amennyiben van valamilyen észrevétele, megkeresését szívesen fogadom a [molnarkatanak@gmail.com](mailto:molnarkatanak@gmail.com) e-mail címen (a levél tárgya minden esetben legyen *Tehetseggondozas\_Tablazatkezeles*).

Köszönettel:

*Molnár Katalin*

## **Bevezető**

Szakedolgozatom elsődleges célja, hogy segítse az informatika alkalmazói OKTV-re illetve OKATV-re készülő diákokat a felkészülésben, tanáraikat pedig a felkészítésben.<sup>1</sup> Ezzel egyszerre segítséget kívánok nyújtani mindenkinek, aki az informatika középszintű érettségi, illetve az ECDL szintjén túlmutató ismereteket és rutint szeretne szerezni a táblázatkezelés témakörében.

Azért választottam ezt a célkitűzést, mert azt tapasztalom, hogy a könyvesboltok polcain és az interneten is számos olyan szakirodalom található, mely a táblázatkezelés alapjait mutatja be. Olyannal viszont még nem találkoztam, ami ennél magasabb szintű, de mégis könnyen érthető, gyakorlatias, és átfog minden – a mindennapi életben is felhasználható – témakört.

A versenyeken használható *Microsoft* és *Libre* szoftvercsomag is, ennek ellenére a korábbi OKTV/OKATV versenyeken indulók döntő többsége a *Microsoft Office* programcsomag valamely verzióját választotta. Szakedolgozatomban én is ezt a szoftvercsomagot használom, még hozzá a jelenlegi legfrissebb, 2010-es verzióját. Éppen ezért, egyszerűsítésként a fájlok típusát *Microsoft*-os kiterjesztésükkel jelzem majd. Ennek ellenére a szabad szoftvereket használók számára sem haszontalanok a feladatleírásaim, hiszen azok gyakran egy az egyben átültethetőek más programokra is.

Szeretném előrebecsátani, hogy a szakedolgozatomban szereplő linkek, ha a beadás idejéig aktuálisak is voltak, nem garantált, hogy nem változtak azóta, legyen szó akár a honlap címéről, akár a tartalmáról. Ugyanígy, a közben eltelt idő esetleg egyéb változásokat is hozhatott magával, például a szoftverek frissülését, újabb csomagok megjelenését stb. Kérem az olvasót, ne feledkezzen meg erre tekintettel lenni!

Szakedolgozatomban az egyszerűsége, érthetősége törekszem, hogy mind a középiskolás korosztály, mind pedig a táblázatkezelés témakörében már valamennyire jártas felnőtt olvasó számára könnyen emészthető legyen. Nem írok általában a versenyekről, hiszen akit ezen információk érdekelnek, az úgyis utánanéző az interneten.<sup>2</sup>

---

<sup>1</sup> OKTV: az informatika OKTV alkalmazás kategóriája 11.-12. osztályos tanulóknak, OKATV: Nemes Tihamér Országos Informatikai Tanulmányi Verseny alkalmazás kategóriája, 9.-10. osztályos tanulóknak.

<sup>2</sup> Például itt: <http://tehetseg.inf.elte.hu>

Pedagógiai tanácsokat sem fogalmazok meg konkrétan, a táblázatkezelés általam helyesnek talált tanítási szempontjait csak a sorok közé bújtatva, a témakörök felépítésével, a magyarázatok megfogalmazásával, a típushibák megemlítésével sugalmazom.

Munkám során igyekeztem minél inkább gyakorlatorientált lenni. Az egyes fejezeteket és alfejezeteket a feladatok megoldásához és megértéséhez feltétlenül szükséges elméleti bevezetővel kezdem, bemutatva ezzel az *Excel* főbb funkcióit. Ezt követően a versenyfeladatokból merített példákon keresztül mélyítem el az anyagot, ezek megoldását bőséges magyarázatokkal látom el.

Szakdolgozatomhoz természetesen tartozik jónéhány versenypélda feladatlap, forrásfájlokkal, pontozási útmutatóval és az általam készített megoldásokkal. Ezek évek és fordulók szerint csoportosítva egy-egy mappában, a CD-mellékleten és az <http://alkverseny.elte.hu> oldalon találhatóak, gyakran hivatkozom majd rájuk. Ugyanitt helyeztem el pár saját példát is, valamint ezen irományt elektronikus formában.

Terveim közt szerepel, hogy a későbbiekben folytatom munkámat – újabb példákkal, esetleg fejezetekkel egészítem majd ki. Ezért érdemes az előbb megjelölt honlapot figyelemmel kísérni, és mindig a legfrissebb példányt letölteni.

Mint az a fentiekből is kitűnik, munkám nem azon olvasóknak készült, akik még csak most ismerkednek a táblázatkezeléssel. Helyettük azoknak ajánlom, akik legalább az ECDL vizsga, még inkább a középszintű érettségi letételéhez szükséges tudásszinttel rendelkeznek<sup>3</sup>, mivel ezekre az ismeretekre építek. Akiknek ezek az alapok hiányoznak, ajánlom az ECDL példatárat és az *Érettségi felkészítő feladatgyűjteményt* a szükséges jártasság megszerzéséhez.

Végezetül ajánlom szaktársaim hasonló jellegű, de más témakörben megírt szakdolgozatait<sup>4</sup>:

- *Barta Anita: Szövegszerkesztés és prezentációkészítés*
- *Csongrádi Tamás: Adatbázis-kezelés*
- *Kiss Csaba: Web- és képszerkesztés*

---

<sup>3</sup> Lásd itt: <http://njszt.hu/ecdl/kovetelmenyek> és itt: [http://www.oh.gov.hu/letolt/okev/doc/erettsegi\\_40\\_2002\\_201201/informatika\\_vk.pdf](http://www.oh.gov.hu/letolt/okev/doc/erettsegi_40_2002_201201/informatika_vk.pdf).

<sup>4</sup> A négy munka együtt lefedi az alkalmazói verseny összes témakörét: <http://alkverseny.elte.hu>.



## **Táblázatkezelés haladó szinten**

### **1. Az adatok importálása és előkészítése**

Az adatokat, amelyekkel dolgozni fogunk, különböző forrásokban biztosítják számunkra. A legtöbb feladat azzal kezdődik, hogy ezeket importáljuk. Ez érettségien, ECDL-en sem ritka feladat, ott azonban az oszlopok típusának beállításánál, esetleg a karakterkódolás kiválasztásánál nehezebb dolgunk nem szokott lenni. Ezekben a versenyeken, vagy a mindennapi élet valós megoldandó problémái esetén ritkán van ilyen egyszerű dolgunk. A hangsúly nem is a nehézségen van, hanem a kreativitáson és az időn: nem mindegy, mennyit töltünk el a folyamat megtervezésével, nem mindegy, hogy a kiválasztott módszer mennyi időt igényel, és az sem, hogy esetleg milyen későbbi problémákhoz vezet valamilyen megfontolatlan lépés.

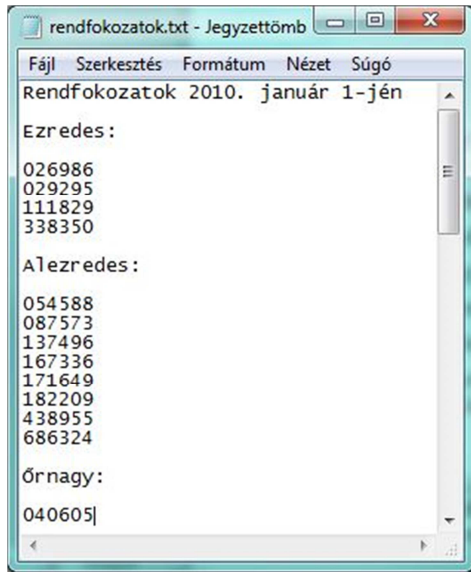
Ez egy eléggé összetett témakör, gyakran épül olyan ismeretekre, melyeket majd csak a későbbiekben tárgyalok. Ennek ellenére mégis ezt választottam első fejezetnek, mert a feladatok megoldása mindig ezzel a résszel kezdődik. Igyekeztem olyan példákat választani, melyek kevésbé alapoznak a további témakörökre, ugyanakkor a későbbi fejezetekben előfordulhat, hogy a példa éppen egy importálási feladat lesz, ami az adott témakörben való jártasságot méri fel.

#### **1.1. Importálás *txt*-ből vagy *docx*-ből**

Elsőként egy nagyon egyszerű példát mutatok, ahol semmi más dolgunk nincs, mint egy kicsit átrendezni az importált adatokat.

##### **1. példa: Nemes 2010-2011 2. forduló, 4/C (A légielő pilótáinak kiképzése)**

A forrásban rangonként csoportosítva szerepelnek a pilóták azonosítói, míg mi az azonosítók egy már adott sorrendjét szeretnénk használni, azonosítónként felsorolva a hozzájuk tartozó rangot. Az egyik legegyszerűbb eljárás a következő:



A forrás: rendfokozat.txt

Azonosító	Rendfokozat 2010. január 1-jén
026986	Ezredes
029295	Ezredes
040605	Őrnagy
054588	Alezredes
087573	Alezredes
111829	Ezredes
137496	Alezredes
143679	Őrnagy
167336	Alezredes
171649	Alezredes

A cél: Január munkalap

- Importáljuk az adatokat egy félreeső oszlopba, mondjuk F-be! A kódolás legyen megfelelő (például ANSI), hogy minden ékezetes és egyéb speciális karakter helyesen jelenjen meg!
- Ahhoz, hogy a nullával kezdődő azonosítók a mintának megfelelően nézzenek ki, az összes adatot szöveggként érdemes importálni. Ehhez a döntéshez azonban előbb meg kell róla győződni, hogy a későbbiekben nem lesz olyan feladat, amelyhez számként lenne az azonosítókra szükség. Ha így lenne, akkor érdemesebb *Általános*ként importálni (hiszen ugyanazon oszlopban szerepel szám és szöveg is), majd a *Cellaformázás*nál beállítani az azonosítókra, hogy fixen hat hosszúak legyenek (formátumkód: 000000).

Akkor sem kell kétségbe esni, ha csak később jövünk rá, hogy inkább számként lenne az azonosítókra szükség. Az *Excel* képes automatikusan számmá konvertálni a szöveggként tárolt azonosítókat, ha például megszorozzuk őket eggyel.

Gond inkább abból szokott lenni, ha eleve olyan forrást kapunk, ahol a számok - amikkel természetesen dolgoznunk is kell - szöveggként vannak eltárolva. Erre jobb esetben a cella bal felső sarkában megjelenő zöld háromszög is felhívja a figyelmet. Ha nem választjuk az *Átalakítás számmá* opciót, a függvényeink #ÉRTÉK! hibaüzenettel térhetnek vissza. Erre a jelenségre láthatunk példát a 2008-2009-es év Nemes 3. fordulójának 4. feladatában, a *D* részben.

- Az *A* oszlopot elkészíthetjük a feladat megelőző részeinek felhasználásával, a *Képzés* munkalap *A* oszlopának másolásával (értékmásolás helyett helyesebb hivatkozást használni, még ha ennek hiánya nem is okoz problémát).

Oszlopok másolása helyett alkalmazhatunk munkalapmásolást is, ami sok esetben előnyösebb lehet, hiszen megtartja a formázásokat is, a szükséges adatokon túl. Ehhez nem kell mást tenni, mint a *Ctrl* gombot nyomva tartva elvonszolni a *Január* munkalapot, mintha csak át akarnánk helyezni. Utána persze ki kell törölni a szükségtelen adatokat, de ez azért jóval gyorsabb lehet, mint a kulcsin újbóli beállítása. Ebben a módszerben az lehet a buktató, hogy olykor kimondottan kéri a feladat (de egyébként is érdemes így gondolkodni), hogy minden további munkalap az első munkalapra épüljön, esetünkben a *Képzés* munkalapra. A legegyszerűbb tehát, ha a munkalap másolását követően minden adatot kijelölünk (*Ctrl+A*), majd törölünk (*Delete*) az új munkalapon, hogy csak a formázás maradjon. Utána **Csatolt beillesztéssel** másolhatjuk a rangokat az eredeti munkalapról.

- A rangok mögül töröljük a kettőspontot, és másoljuk a megfelelő azonosítók mögé!
- INDEX ( ) -HOL .VAN ( ) -nal töltjük ki a C oszlopot!

	A	B	C	D	E	F	G
1						Rendfokozatok 2010. január 1-jén	
2	Azonosító						
3	026986		Ezredes				
4	029295		Ezredes				
5	040605		Őrnagy			026986	Ezredes
6	054588		Alezredes			029295	Ezredes
7	087573		Alezredes			111829	Ezredes
8	111829		Ezredes			338350	Ezredes
9	137496		Alezredes				
10	143679		Őrnagy				
11	167336		Alezredes				
12	171649		Alezredes			054588	Alezredes
13	182209		Alezredes			087573	Alezredes
14	201461		Őrnagy			137496	Alezredes

A 3. és 4. lépés

- A C oszlop érték szerinti másolásával megkapjuk a B oszlopot, a felesleges oszlopokat törölhetjük.

Egy másik, még egyszerűbb megoldás az előzőekben F és G oszlopban előkészített adatok rendezése azonosítók szerint. Ettől ugyanis az üres sorok egy helyre kerülnek, és mivel az A oszlopban is rendezetten vannak az azonosítók, elég az így kapott összefüggő, rangokat tartalmazó tartományt áthelyezni a B oszlopba.

## 2. példa: OKTV 2010-2011 1. forduló, 5/B (Tisztavatás)

A forrás most egy *doc* fájl, mely táblázatos formában tartalmazza az adatokat. A hagyományos módon ezt nem tudjuk forrásként felhasználni, hiszen a szövegből történő importáláskor nincs lehetőség *doc* fájl kitallózására. Vágólap segítségével azonban

megoldható a dolog: megnyitjuk a forrást *Word*-del, kijelöljük a táblázatot, vágólapra helyezük, majd egy *Excel* munkalapon állva *Ctrl+V*-t nyomunk. Ezt követően csak át kell csoportosítani az adatokat, és helyreállítani a formázást, hiszen az elég furcsán került át.

Választhatunk azonban egy másik módszert is, mely használata akkor jön jól, amikor nagyon sok, számunkra felesleges formázás van beállítva a *Word*-táblázatra. A formázás teljes elvesztésével ugyanis lehet, hogy gyorsabban végzünk a feladattal. A következőkben ennek lépései olvashatóak.

Tancoport	Szak, szakirány
<b>Kossuth-Lajos-Hadtudományi-Kar</b>	
NAKV0171	Katonai-vezetői, gépesített lövész szakirány BSc
NAKV0271	Katonai-vezetői, harckocsizó BSc
NAKV0371	Katonai-vezetői, felderítő BSc
NAKV0471	Katonai-vezetői, tüzér BSc
NAKV7571	Katonai-vezetői, repülésirányító szakirány, légiforgalmi specializáció BSc
NAKV7572	Katonai-vezetői, repülésirányító szakirány, vadászirányító spec. BSc
<b>Bolyai-János-Katonai-Műszaki-Kar</b>	
NBHBK671	Had-és biztonságtechnikai mérnöki, katonai elektronikai szakirány, katonai informatika spec.
NBHBK371	Had-és biztonságtechnikai mérnöki, katonai elektronikai szakirány, híradó spec.
NBHBH371	Had-és biztonságtechnikai mérnöki, haditechnikai szakirány, vegyvédelmi technikai spec.

*A forrás egy doc fájl, benne táblázattal. A két kar adatainak külön munkalapra kell kerülniük.*

- Nyissuk meg a forrást *Word*-del!
- Jelöljük ki a táblázatot, és az *Elrendezés* fülön válasszuk a **Szöveggé alakítás** opciót! Az alapértelmezettként felkínált *tabulátor* jel megfelelő lesz oszlophatárolónak.
- Válasszuk a *Mentés másként* opciót, a fájl típusának *txt*-t adjunk meg! Ezzel a dokumentum elveszíti minden formázási beállítását (erre egy ablak figyelmeztet is minket), de most nekünk éppen ez a célunk, úgyhogy mentsünk bátran!
- Így már könnyedén importálhatunk a *txt*-ből, ezt követően pedig csak az adatok elosztására van szükség (hiszen azok két munkalapra kerülnek), és pár felesleges sor törlésére.

## 1.2. Importálás weblapból

Gyakori, hogy az adatok forrása valamilyen weblap. Az *Excel* erre külön lehetőséget nyújt: az *Adatok/ Külső adatok átvétele* csoportban kiválasztható a weblapforrás is. Itt a *cím* mezőbe megadhatunk tényleges internetes vagy helyi linket, vagy vonszolással az ablakba „ejtethünk” egy korábban már lementett weblapot. Ezt követően a nyíl (➡) ikonokkal tudjuk kijelölni, hogy a weblap mely részeire van szükségünk.

Ez a módszer azonban problémás lehet, ha a weblap a szükséges adatokon túl más értékeket is tartalmaz, vagy az előbb említett ikonnal nem tudjuk kijelölni, amit szeretnénk. A megoldásnak ekkor többféleképp is nekifoghatunk, de a **Keresés és csere** funkcióra szinte minden esetben szükségünk lesz. Erre láthatunk egy példát a következőkben:

### 3. példa: Nemes 2010-2011 3. forduló, 4/A (Eszmei érték)

Először védett növények nevét, latin nevét és eszmei értékét kell kinyernünk három *htm* fájlból. Ezek hasonló felépítésűek, így érdemes velük egyszerre dolgozni.

Algarvei hasadtfogúmoha - <i>Fissidens algarvicus</i> Értéke: 5.000 Ft
Alhavasi szőrössüvegúmoha - <i>Orthotrichum rogeri</i> Értéke: 5.000 Ft
Arnoldi-hasadtfogúmoha - <i>Fissidens arnoldii</i> Értéke: 5.000 Ft

*A forrás: novenyek1.htm, novenyek2.htm, foknoveny.htm*

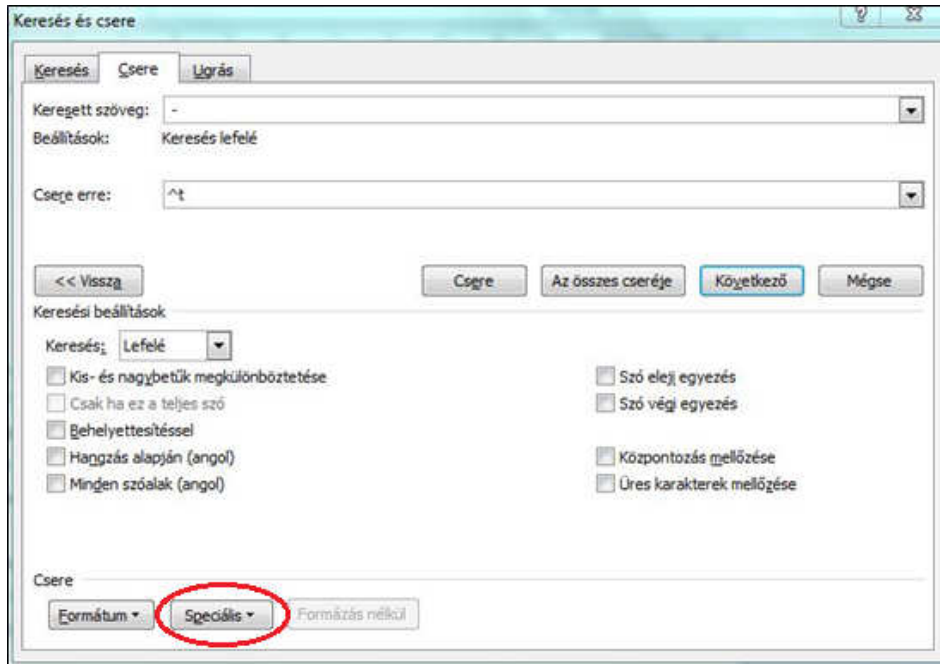
	A	B	C	D
	Kód	Magyar név	Latin név	Érték
1				
2				
3	679592	Gímpáfrány	Asplenium scolopendrium	2 000 Ft
4	610841	Hegyi csipkeharaszt	Selaginella helvetica	2 000 Ft
5	902141	Kígyónyelv	Ophioglossum vulgatum	2 000 Ft
6	823241	Rucaöröm	Salvinia natans	2 000 Ft
7	664265	Adriai varjúháj	Sedum acre subsp. neglectum	2 000 Ft
8	648460	Árokvirág	Samolus valerandi	2 000 Ft
9	630093	Balatoni galaj	Galium austriacum	2 000 Ft

*A cél: Eszmei érték munkalap*

- Nyissuk meg a *htm*-eket, a számunkra szükséges tartalmat tegyük vágólapra! (Az „oszlop” kijelölése úgy a legegyszerűbb, ha egérrel kijelöljük az első karaktert, majd a *Shift* nyomva tartása mellett kattintunk az utolsó „sor” végén.)
- Illesszük be a tartalmakat egymás alá egy *Word* dokumentumba, de a formázás elhagyásával! Célunk az, hogy ebből három oszloppá alakítható, csak a számunkra szükséges adatokat tartalmazó forrást hozzunk létre. A külön oszlopokba tartozó adatok elválasztásához *tabulátor* jelet fogunk beilleszteni a megfelelő helyekre.

A továbbiakban úgy fogunk dolgozni, hogy a vágólapról beillesztett adathalmazt a megfelelő formájúra hozzuk, a *Keresés és csere* funkcióval. Ennek megkezdéséhez mindig át kell nézni a beillesztett forrást, hogy lássuk, milyen sorrendben érdemes haladni, milyen karaktercsoportokat kell lecserélni vagy törölni. Ez persze azon is múlik, hogy mi került a vágólapra, ami viszont függ a használt böngészőtől. A továbbiakban a megoldás a *Firefox 14.0.1*-hez illeszkedik, a többi böngésző esetén kisebb változások lehetnek, melyek értelemszerű módosításokat igényelnek.

- Az első két oszloptartalommal nincs teendőnk, őket most a „-” karaktersorozat választja el, ezeket cseréljük *tabulátor* jelre a *Keresés és csere* funkcióval! Az elgépelés elkerülése végett a cserélendő karaktersorozatokat gyakran érdemesebb vágólapra másolni, majd beilleszteni a *csere* ablakba. Szintén jó tanács, hogy a *csere* összes elfordulásra történő alkalmazása előtt nézzük át a dokumentumot, hogy nem fogunk-e olyat is lecserélni, amit nem szeretnénk, vagy legalább ne egyszerre cseréljünk mindent, hanem először csak egyesével haladjunk.



A **Keresés és csere** ablaka. A tabulátorjelet szimbolizáló „^t” kódot a **Speciális** gomb listájából is kiválaszthatjuk.

- A következő oszlopban az *érték* lesz, de az „Értéke:” kifejezésre nincs szükségünk. Nem kell *bekezdés vége* jel se, így a keresett szöveg: „^p·Értéke: ”, ezt cseréljük szintén *tabulátor* jelre! Ellenőrizve munkánkat látjuk, hogy a dokumentum utolsó részén (*foknoveny.htm*) ez nem működött, ott ugyanis három szóköz is bekerült a sor elejére. Cseréljük ezt is!
- Mivel az értékekkel majd számolni szeretnénk, töröljük a „·Ft” kifejezéseket, a *Keresés és csere* funkcióval ez is csak egy kattintás!
- Gond adódna, ha bent hagynánk az értéket tagoló pontokat, ezeket szintén egy gombnyomásra cserélhetnénk, de vigyázni kell, hiszen számtalan növény latin nevében szerepel a „subsp.” pontot tartalmazó kifejezést, ezeket a pontokat természetesen nem szabad törölni. Használhatjuk a *Következő* gombot, hogy egyesével dönthessük el a pontokról, kívánjuk-e megtartani őket, de ez most elég hosszadalmas lenne. A dokumentumot átvizsgálva („·”-ra keresve) meggyőződhetünk róla, hogy csak az előbb említett megtartani való kifejezés tartalmaz pontot, így a legegyszerűbb, ha töröljük mindet, majd a „subsp” szócskákhoz visszahelyezzük őket: a keresett szöveg „subsp”, cseréljük arra, hogy „subsp.”!

- A fokozottan védett növényekkel ismét további teendők van, itt a százezres értékeket szóközzel tagolják. Mielőtt kétségbeesnénk, hogy a másutt szükséges szóközők megőrzése végett egyesével kell törölni az ilyen szóközőket, észre kell venni, hogy ezek az értékek kerek számok, így alkalmazhatjuk a „0·0” karaktorsorozat cseréjét a „00”-ra, hiszen más helyen ez a karaktorsorozat nem fordul elő.
- A felesleges, üres „sorokat” (valójában: bekezdéseket) is el kell távolítani, a keresett szöveg „^p^p”, amire cseréljük: „^p”. Ügyeljünk, hogy olykor a második *bekezdés vége* jel előtt előfordul egy vagy akár több szóköz is, ezeket is törölni kell.
- Még így is maradt némi „szemét” a dokumentumban, például kategórianévek, felesleges szóközők stb. Ezeket is javítsuk, belátásunk szerint kereséssel, vagy akár „kézzel”! Mentsük a dokumentumot formázás nélküli formátumban, például *txt*-ben!
- Az elkészült forrásfájlunkat importáljuk egy *Excel* munkafüzetbe a *B3*-as cellától!

Ha weblap a forrás, érdemes megfontolni, nincs-e könnyebb dolgunk, ha közvetlenül a weblap forrásszövegével dolgozunk? Olykor bizony egyszerűbb lehet, és csak minimális *html* ismeretet igényel.

A kódok és a becsült példányszám importálása következik, az *A* és *E* oszlopba. Ezeket az adatokat az *nlista.txt* tartalmazza, pont a megfelelő sorrendben, erre a feladat szövege ki is tér. Nincs egyszerű dolgunk, mert az adatok csupán egy szóközzel elválasztva, ömlesztve vannak, nekünk meg két „oszlopra” szedve kellenének. Többféleképp is lehet próbálkozni, de ha nem a megfelelő eszközt választjuk, a munka könnyen elég macerássá válhat.

Próbálkozzunk először az eddig jól bevált módszerrel, a *Word Keresés és csere* funkciójával.

- Nyissuk meg a forrást szövegszerkesztőben, és vizsgáljuk meg az adatokat! Látható, hogy eleinte a példányszám százasokra van kerekítve, míg az azonosítók csak nagyon ritkán végződnek két nullára.
- Cseréljük „00”-t „00^p”-re! A dolog csak néhány helyen hibádzik, ezeket javítsuk kézzel!



- A végénél (véltetőleg a fokozottan védett, kevés példányszámú növények esetén) már csak tízesekre vannak kerekítve a példányszámok, itt alkalmazhatjuk az előbbi módszert, egyetlen nullával. Ügyeljünk rá, hogy az előbbi, már kész részt ne bántsuk, például a javítandó rész kijelölésével!
- Kézi javításra megint szükség van, de szerencsére nem sokra.
- Importáljunk tetszőleges félreeső helyre, majd másoljuk az oszlopokat a megfelelő helyekre! Ehhez érdemes a kijelölést a következőképp végezni: álljunk az oszlop első cellájára, és tartsuk nyomva a *Shift*-et (kijelölés), illetve nyomjunk *Ctrl+lefele nyilat* (ugrás a tartomány aljára).

Látható, hogy ezzel így nagyon sok a munka. De nem kell feltétlenül a *Word*-höz ragaszkodni, vagy legalábbis nem kell teljesen készre alakítanunk a forrást (az előbbi módszer eredménye egy *tabulátor* jellel kettéválasztott, két szavas bekezdéseket tartalmazó szöveg, melyet könnyedén két oszlopba importálhatunk). Elég, ha a szóközöket lecseréljük *bekezdés vége* jelre, és így importáljuk az adatokat egy *Excel* munkafüzetbe, egyetlen oszlopba. Innentől kezdve csak két oszlopra kell választani, amit a *MARADÉK ( )* függvény segítségével könnyedén meg is tehetünk:

$B1=HA ( MARADÉK ( SOR ( A1 ) ; 2 ) = 1 ; A1 ; " " )$ , ill.  
 $C1=HA ( MARADÉK ( SOR ( A1 ) ; 2 ) = 0 ; A1 ; " " )$

Az üres soroktól gyorsan megszabadulhatunk, az *Adatok áttekintését segítő eszközök* című fejezetben részletesen ismertetett *Ismétlődések eltávolítása* módszerrel. Alkalmazásához készítsünk előbb érték szerinti másolatot a *B* és *C* oszlopkról a *D* és *E* oszlopokba, majd az *E* oszlop tartalmát illesszük egy sorral feljebb! Az így kapott két oszlopos táblázatból már csak az üres sorokat törli az eszköz.

Ne lepődjünk meg, ha munkafüzetünk nem teljesen hasonlít a mintaként adott képre! Ha megfelelő sorrendben illesztettük be a *htm*-ek tartalmát, akkor az első növény az *Algarvei hasadtfogúmoha* lesz, nem pedig a *Gímpáfrány*, ahogy azt a minta mutatja! Ez előfordul néha, ezért is szerepel minden verseny szövegében a figyelmeztetés:

- „Vedd figyelembe, hogy a minták mindenütt a formátumra utalnak, a számított értékek esetében nem (vagy nem feltétlenül) a helyes eredményt mutatják!”

### 1.3. Importálás *pdf* állományból

Ritkán, de előfordul az is, hogy a forrást *pdf* formátumban adják meg.

**4. példa: Nemes 2009-2010 3. forduló, 4/C (Holdak)**

Ez a feladat jórészt csak az importálásról, a táblázat elkészítéséről szól, mindenféle számolás nélkül. A feladat nem egyszerű, sok tervezést és munkát igényel.

A *pdf* már maga tartalmazza a holdakat jellemző adatok kívánt elrendezését, de ezt egyrészt nem egyszerű átvinni táblázatkezelőbe, másrészt a *pdf* is rengeteg hibát tartalmaz, melyre a feladat szövege külön fel is hívja a figyelmet:

- „több helyen nem csak a sorok végén, de a sorok belsejében is vannak kitöltetlen („üres”) cellák,
- sajnos, a forrás nem mindenütt következetes a tizedesjel használatában,
- egyes cellákban egy kis □-et látsz a számok között; ez egy normálalakban megadott számra utal (például a Hold sorának utolsó előtti cellájában ez az érték valójában  $7,35 \cdot 10^{25}$ )”

PN4)	1989	Voyager2, Synnott	0.429	61950	2.50	0.05	0.0001	0			$3,7 \square 10^{21}$	0,000
IN2)	1989	Voyager2, Reitsema	0.554	73550	2.97	0.20	0.0014	0			$4,3 \square 10^{21}$	0,000
PN1)	1989	Voyager2, Synnott	1.121	117640	4.75	0.06	0.0004	0			$4,0 \square 10^{22}$	0,000
	1846	Lassell	5.877	354800	14.33	157	0.001	0	65	34.5	$5 \square 10^{25}$	0,77
	1949	Kuiper	$3 \square 0,1 \square 3$	5513000	222.75	29	0.75	0	65			
	1978	Christy	6.387	19640	17.06	0	0.0001	15	55		$1,22 \square 10^{24}$	0,01

*holdak.pdf, és hibái*

Nézzünk egy lehetséges megoldást!

- Másoljuk a fejléc alatti, holdakról szóló részt vágólapra, és illesszük be szövegszerkesztőbe!
- Vizsgáljuk meg a szöveget, vessük össze a mintával!
- Látjuk, hogy az első oszlopban csak ritkán van adat, ezért jobb, ha ezzel külön nem is foglalkozunk, hiszen kézzel is gyorsan be tudjuk majd írni a megfelelő helyre. A legjobb, ha ki is töröljük őket, hogy ne okozzon kárt a későbbi munkánkban.
- Sajnos előfordulhat, hogy a táblázat pár sora itt két külön sorba kerül, ezeket legjobb, ha „kézzel” ellenőrizzük és javítjuk.

Észrevehető, hogy az oszlophatárolók akár a szóközök is lehetnek, kivéve pár adatot a holdak vagy a felfedezők nevét tartalmazó oszlopokból. A terv, hogy előbb

lecserélünk minden szóközt *tabulátor* jelre, mint oszlophatárolóra, majd amelyiket kell, visszacseréljük szóközre.

- Cseréljük le az összes szóközt *tabulátor* jelre!
- Egy hold neve akkor áll két szóból, ha a neve után zárójelben ott van az azonosítója is. Tehát az olyan *tabulátor* jeleket, melyeket egy nyitózároljel követ, nem kellett volna cserélni, javítsuk! Keresett szöveg „^t(”, csere erre: „(”. A „Helene” holdnál ez nem elég, ezt javítsuk kézzel, vagy válasszunk más módszert.
- Néhány holdnak több felfedezője is volt, az ő nevük vesszővel van elválasztva. Cseréljük a „^t” karaktersorozatokat „;”-ra, az újabb javítás érdekében. (Egy kis segítség: a „vessző + *tabulátor* jel”-et cseréljük „vessző + szóközre”.)
- Gyakoriak az „et al.”-ra végződő felfedezők, ezeket is egy oszlopban kell hagyni: szintén behelyettesítéssel keressük, hogy „^tet^tal.” és cseréljük arra, hogy „.et al.”. Ezzel korrigáltuk a szóközők lecsereléséből adódó hibáinkat.
- Már nincs más hátra az oszlopok kialakításából, mint hogy megkeressük azt a pár sort, amelynél a felfedezéshez nincs sorszám, itt tegyünk be egy-egy *tabulátor* jelet, hogy ne csússzon el a sor.
- Most jön az adatok javítása, hiszen a feladat külön felhívta a figyelmet a hibákra. Az üres cellákkal most nem muszáj törődni, ezt majd egyszerűbb táblázatos formában látva javítani. Javítsuk inkább a tizedesponttal írt adatokat tizedesvesszősre, hogy az *Excel* számként ismerje fel őket! Ügyelni kell az egyéb *pont* karakterekre, ezért érdemes egyesével haladni, vagy akár ki lehet próbálni más stratégiát is, mint például visszacserélni pontra azokat, amiket kell.
- A normálalakban feltüntetett adatok most ilyen formában vannak: „7,35↯→1025”, vagyis a □-ből *feltételes elválasztó* jel lett a beillesztés során.<sup>5</sup> Jól tudjuk, hogy a normálalak csak egy cellaformázási beállítás lesz, tehát az lesz a helyes eljárás, ha kiszámoljuk a megfelelő értékét, és ezt tároljuk el, majd később a kívánt alakúra formázzuk. Az értékét képlettel fogjuk kiszámolni, valahogy így:  $7,35 \cdot 10^{25}$ . Bontsuk tehát a karaktersorozatot három részre (7,35; 10; 25)! Ezek

---

<sup>5</sup> Nem kizárt, hogy itt is függ a használt szoftverektől, mi kerül a vágólapra. A továbbiakban ügyelj erre, és ha szükséges, bíráld felül a leírtakat!

majd külön oszlopokba kerülnek, hogy ki tudjuk számolni a hozzájuk tartozó értéket. Ehhez a feltételes kötőjelek törlendők, de akkor már egyúttal a „10” mögé is betehetünk egy *tabulátor* jelet. A keresett szöveg tehát: „^t10” (hiszen a feltételes elválasztójel kódja „^”), csere erre: „^t10^t”.

- Menthetjük a forrást, természetesen *txt*-be.
- Importáljuk az adatokat a forrásból, a fejléc alá, a *B* oszlopba! Az *A* oszlop adatait pótoljuk kézzel, és ahol kell, alakítsuk ki az üres cellákat, hogy minden adat a saját oszlopába kerüljön!
- Számoljuk ki a normálalakban írandó adatok értékét, ezt képletmásolással nyugodtan végig is „húzhatjuk”, nem törődve a #SZÁM! hibaiüzenettel. Értékmásolással illesszük a megfelelő oszlopba, majd töröljük ki azt a két oszlopot, amivel számoltunk! Ügyeljünk az „Encke-résben” adat esetleges felülírására! Ezt követően az *Excel Keresés és csere* funkciójával könnyen megszabadulhatunk a #SZÁM! hibaiüzenetektől.
- Az adatok importálása, előkészítése itt véget ér, persze a feladat további részét képezi a megfelelő formázások beállítása. Erre a *Cellaformázás* fejezet *Formátumkódok* című alfejezetében térek ki.

#### 1.4. Az importált adatok szerkezetének átalakítása függvényekkel

Úgy gondoltam, ez a téma is megér egy fejezetet, még ha nem is egy újabb forrástípust mutat be. Már az eddigi példákban is láthattuk, hogy az adatokat milyen sokféleképpen elő lehet készíteni, és, hogy olykor többre megyünk *Excel*-függvények használatával, mint például a *Word Keresés és csere* funkciójával. A következő példában egy olyan esetet mutatok be, amikor más eszközzel talán neki se lehet fogni a feladatnak, mint függvények használatával.

#### 5. példa: OKTV 2009-2010 1. forduló, 5/H (Karácsonyi síverseny)

A forrásban sorfolytonosan vannak az adatok, melyeket *Pascal háromszög*-szerűen kell elhelyezni:



A következő feladat, hogy az *egyese*k helyére a kívánt értékek kerüljenek. Ehhez előbb arra lesz szükség, hogy a sorfolytonos adatsorozat hányadik helyén található az egyes sorok első adata.

- Készítsük el a sorok sorszámát tartalmazó oszlopot AA1-től!

=SOR ( )

- Az előbb készített oszlopot felhasználva számoljuk ki az egyes sorokhoz, hogy az importált adatsorozat hányadik elemétől kezdődnek majd benne az adatok:

AB1=1

AB2=2

AB2=SZUM( \$AA\$1 : AA2 ) +1 , ...

- A HA ( ) függvényben cseréljük ki az 1-eseket a 15. sor azon elemére, melynek majd a sor első helyén kell állnia!

INDIREKT( CÍM( 15 ; \$AB1 ) )

- Egészítsük ezt ki azzal, hogy mindig az annyiadik elemet adjuk meg a 15. sor kijelölt elemétől kezdve, mint ahányadik eleménél járunk a sornak!

INDIREKT( CÍM( 15 ; \$AB1+( OSZLOP ( ) +SOR ( ) -14 ) / 2 ) )

A feladat lényegi részével készen vagyunk, már csak másolnunk kell a kapott **értékeket** az *sms* munkalapra.

A bemutatott példák segítségével látható, hogy az adatok importálása leginkább nem a technikai tudást és a kevésbé ismert funkciók használatát igényli, sokkal inkább **a kreativitás és az előre tervezés készsége a fontos**. Sok múlik azon, hogy mekkora jártasságot szereztünk az adatok importálásában, mennyire van gyakorlatunk a feladatok lépésekre bontásában, a típushibák kiküszöbölésében, és **a leggyorsabb megoldás megtalálásában**, hiszen minden további táblázatkezelés feladat erre épül.

## 2. Dátum- és időkezelés

Habár dátum- és időkezelésre középszintű érettségien és ECDL vizsgán is szükség van, és így alapszintű ismeretnek számít, úgy gondoltam, mégis megéri ezt a témakört átismételni. Annak ellenére, hogy igazából egy cellaformázási beállításról van szó,

mégis külön fejezetet kapott, ahelyett, hogy a *Cellaformázás* egy alfejezete lenne. A tapasztalatom ugyanis az, hogy nagyon sok diák fejében téves elképzelések élnek, melyekkel számos esetben sokáig „ki is húzzák”. Gyakori, hogy a könnyebb feladatokat sikeresen meg tudják oldani, ezért aztán nem értik, hogy nehezebb feladatok esetén miért nem működik a „megoldásuk”.

Az alapvető problémát **a cella tartalmának és formátumának összemosódása** okozza, ami általános probléma, de **típusan dátumok esetén** jelentős. Lássunk egy konkrét példát! Ha azt látni egy cellában, hogy „2012. június 30.”, néhányan azt hiszik, hogy ez a cella tényleges tartalma, mintha csak szöveg lenne. Ez már csak azért is meggondolatlan szemlélet, mert akkor például a „2012.06.30.” már egy másik érték lenne, holott tudjuk jól, hogy csak a *Cellaformázás*ban választottunk egy másik megjelenítési formát. Még egyébként ez is rendben van a többségnél. Ami viszont tényleg gyakran hiányzik a diákok (ne adjisten tanárok) fejéből, hogy „akkor mi is van valójában abban a cellában?”.

Próbáljuk ki, hogy mi történik, ha a dátumot tartalmazó cella formátumát *Általánosra* állítjuk! Azt látjuk, hogy „41090”, melyből könnyen levonhatjuk azt a következtetést, hogy valójában egy számról van szó. Ez így is van: **az Excel a dátumokat számként tárolja**, melynek értéke független a megjelenített alakjától.

Mielőtt folytatnánk ezt a gondolatmenetet, vegyünk egy kis kitérőt!

Ne essünk abba a hibába, hogy azt hisszük, ez minden esetben így van, vagyis hogy a formátumot *Általánosra* állítva mindig a cella valódi tartalma lesz látható. Vizsgáljuk meg a következő esetet: *B3*-ba beírjuk, hogy 0,333333333333333, *B4*-et pedig előbb *Tört* formátumúra állítjuk, majd rögzítjük benne az  $1/3$  értéket. Ezt követően kiszámoljuk mindkét érték háromszorosát a *C* oszlopba. Habár a *B4* cella tartalmát *Általánosra* állítva 0,333333333333333-at kapnánk (mint amennyi *B3* értéke), mégsem egyezik meg a két cella háromszorososa: a *B4* valódi tartalma  $1/3$  (tehát egy művelet), nem pedig a tizedes tört.

B4		fx		0,333333333333333	
	A	B	C	D	
1					
2		a	3a		
3		0,333333333333333	0,99999999999999900000		
4		1/3	1,00000000000000000000		
5					

A tört számok tárolása nem tizedes tört alakban történik, még ha a formátum Általánosra állításakor ezt is látjuk a szerkesztőlécen és a Cellaformázás ablakban.

Próbáljuk ki, mi történik, ha megcseréljük a két formátumot! Ha B3-at Tört formátumúra állítjuk, megtévesztő módon **1/3-ot látunk a cellában, holott a valódi értéke maradt a korábbi tizedes tört.**<sup>8</sup> A B4 cellában lévő törtet (hányadost) Általános formátumúra állítva **a megjelenített érték egy tizedes tört, de a valódi tartalom továbbra is 1/3**, így annak háromszorosa is maradt 1. Tehát a cella formátumának átállítása nincs hatással annak tartalmára.

B4		fx		0,333333333333333	
	A	B	C	D	
1					
2		a	3a		
3		1/3	0,99999999999999900000		
4		0,333333333	1,00000000000000000000		
5					

A formátum átállítása nincs hatással a tartalomra.<sup>9</sup>

Érdeemes ezt követően azt is kipróbálni, mi történik, ha a B4 cellán állva a szerkesztőlécre kattintunk, majd *Enter*-t ütünk!

Látható, hogy az *Excel* mennyire nem következetes a valódi értékek és a formátumok megjelenítése terén. Ha egy cella formátumát Általánosra állítjuk, általában a valódi tartalmat látjuk, de például törtek esetén nem: a hányados értékét, tizedes tört formában. Ugyanez dátumoknál éppen a valódi tartalmat mutatja meg.

<sup>8</sup> Sőt: már 0,33-ot is egyharmadként jelenít meg, ha a formátumot törtre állítjuk. Ennek az oka az, hogy ilyenkor az *Excel* megkeresi a számhoz legközelebbi lévő törtet, amit ismer, és ezt fogja mutatni.

<sup>9</sup> Nem úgy, mint a dátum formátumnál: egy dátumnak érzékelt szöveg könnyen dátummá, vagyis számmá konvertálódik, így a cella eredeti tartalma, a szöveg, végleg elveszhet.



A szerkesztőlécen is általában a valódi tartalmat láthatjuk, de kivételek itt is vannak: most nem csak a tört, hanem a dátum is kilóg a sorból: előbbinél a hányados értéke látszik (de több tizedesjeggyel, mint amennyit az *Általános* formátum mutat), utóbbinál egy formátum.

Ezek az eltérő módszerek könnyen összezavarják a felhasználókat, és hátráltatják annak megértésében, hogy egy cella értéke és formátuma két különböző dolog. Ha azonban tisztában vagyunk ezekkel a zavaró tényezőkkel, könnyedén megoldhatjuk a felmerülő problémákat.

A kis kitérő után, térjünk vissza a dátumokhoz!

Sokáig nehézséget okozott a dátumok kezelése az informatika világában, és habár vannak elterjedt módszerek a probléma megoldására, gyakori, hogy különböző programok eltérően kezelik a dátumokat. A bonyodalmat az okozza, hogy ezeket a dátumokat nem csak megjeleníteni szeretnénk, hanem természetesen számolni is kellene velük. Például: hány nap telik el „január 1.” és „március 1.” között? Január 31 napos, február 28, így a válasz 59, feltéve, hogy nem szökőévről beszélünk. Látható, hogy az ilyen jellegű kérdésekre így elég nehézkes lehet választ adni, nem is beszélve a dátumok eltárolásának problematikájáról. Az *Excel* esetében a következő a stratégia: „1900. január 1.”-t „1”-gyel azonosítjuk, „1900. január 2.”-t „2”-vel, és így tovább, **minden naphoz tartozik egy sorszám.**<sup>10</sup> Ezzel könnyű megválaszolni az előbbi kérdést, és még sok hasonlót is. A „konverzió”, vagyis az általunk használt formátumban való megjelenítés az *Excel* dolga, így nekünk igazából nem muszáj a számokkal dolgoznunk, de mindenképp ismerni kell, mi áll a háttérben!

Ami a már emlegetett félreértéseket tovább súlyosbíthatja, hogy úgy tűnik, az *Excel* nem túl következetes az átalakítások illetve típusfelismerések terén se. Ha egy *Általános* formátumú cellába beírunk egy dátumot, a hagyományos *éééé.hh.nn.* formátumban, kivétel nélkül felismeri, és az ehhez tartozó számot tárolja el. Ugyanakkor, ha más formátummal adjuk meg, sokszor csak szöveggként tárolja el. Ami pedig még ennél is rosszabb, hogy időnként az olyan szöveget is dátumnak észleli, amit nem is annak szántunk, tipikusan, ha pont van benne. Ha ilyenkor nem vesszük észre

---

<sup>10</sup> Létezik „0” sorszám is, melyhez az egyébként nem létező „1900. január 0.” dátum tartozik.

időben a hibát, akár végleg elveszthetünk számunkra fontos adatokat. Importálást követően ezért mindig át kell nézni az adatokat, és olykor felülbírálni az *Excel* döntéseit.

Mint írtam, az „1900.01.01.” az a dátum, amit „1”-gyel azonosítunk. Fejtörést okozhat, ha ennél korábbi dátumokkal kell dolgoznunk, hiszen azok kezelésére az *Excel* sajnos nem képes. Erre a problémára és egy lehetséges megoldására mutatok egy példát a következőkben.

#### 6. példa: Nemes 2010-2011 1. forduló, 5/A, B (A szabadságharc kronológiája)

Ez a feladat is az adatok importálásával kezdődik, itt azonban dátumok importálásáról van szó, melyekkel tovább is kell majd dolgozni, tehát ügyelni kell rá, hogy a dátumok tényleg helyes értékkel rendelkezzenek, és ne csak jónak látszódnak.

Csak úgy, mint a 3. példában, a forrás egy weblap.

	A	B	C
1	Év	Hónap, nap	Esemény
2	1848.	március 15.	Az országgyűlés határozata kimondja a nemzetőrség felállításának szükségességét.
3	1848.	március 22.	Az országgyűlés elfogadja a nemzetőrség megalakításáról szóló törvényt.
4	1848.	április 15.	A minisztertanács Batthyány Lajosra bízta a nemzetőrség szervezését.
5	1848.	május 7.	V. Ferdinánd a magyar kormánynak való engedelmségre szólítja fel a főhadparancsnokok
6	1848.	május 15.	A minisztertanács elrendeli a délvidéki várak megerősítését.
7	1848.	május 16.	Felhívás önkéntesek (honvédek) toborzására.
8	1848.	június 12.	A pétervárad helyőrség támadása a karlócai szerb tábor ellen; az akció kudarccal végződik.
9	1848.	június 15.	A nádor Szeth Jánost nevezi ki a honvédség főparancsnokának.

#### *Az elkészítendő táblázat*

- Válasszuk ismét a vágólapos módszert, és illesszük be az adatokat *Word*-be! Mivel a forrás egy táblázat volt, a *Word* nagy eséllyel érzékelné ezt, így ott szintén egy táblázatot kapunk. Ezt szöveggé alakítva (*Elrendezés* fül/ *Adatok*) a forrást menthetjük *txt*-be, melyből importálhatjuk az adatokat *Excel*-be, a *B* oszloptól. Ügyeljünk a kódolásra, és az adatok importálásakor a *dátum* típus kiválasztására (HNÉ)!
- Az *A* oszlopba jöhet, hogy 1848 vagy 1849, cellaformázása *Egyéni/ Normál*, a *B* oszlopnak pedig állítsuk be, hogy a formátuma „hhhh n.”.
- Látszatra kész, de a neheze még csak most jön. Az kellene, hogy ezek a dátumok 1848/1849–esek legyenek, de jól tudjuk, hogy ilyen korai dátumokat az *Excel* nem tud tárolni. Ugyanakkor a *B* feladatban ki kellene számolni, hogy melyik

dátum milyen napra esett, **feltételezve, hogy „1848. márc. 15.” szerda volt.** Érdekeséggéppen meg lehet nézni a cellaformázás átállításával, hogy most milyen év érték tartozik ezekhez a dátumokhoz: nagy eséllyel az éppen aktuális évé.

- A trükk a következő: válasszunk 1848 és 1849 helyett két jóval későbbi évet, melyet már tud kezelni az *Excel*, és a hét napjainak tekintetében ugyanolyan eredményt ad majd. Ahhoz, hogy ne „csússzunk el” a napokkal, csak az kell, hogy a kérdéses évek kisebbike akkor és csak akkor legyen szökőév, ha 1848 is az volt. Ez azért elegendő feltétel, mert a kezdő napot megadtuk egy cellában, és csak két év adatival dolgozunk.
- Derítsük ki, hogy 1848 szökőév volt-e, és ennek függvényében keressünk egy 1900-nál későbbi dátumot! Mivel 1848 szökőév, 1948 megfelelő lesz, hiszen ez is osztható négygyel, de százszal nem.<sup>11</sup>
- Most már nincs nehéz dolgunk: vonjuk össze egy segédoszlopba az *A* oszlopot, plusz még a 100-at, illetve a *B* oszlop *hónap*, *nap* részét, hogy egy dátumot alkossanak:

```
=DÁTUM(
    A3+100 ;
    HÓNAP ( B3 ) ;
    NAP ( B3 )
)
```

- Az előbbi oszlop **értékeinek** másolásával kaphatjuk az igazi *B* oszlopot, az előbbi oszlopokat törölhetjük!
- A napok meghatározása már nem nehéz, használjuk hozzá például az *INDEX ( )*, *MARADÉK ( )*, *HOL.VAN ( )* függvényeket, és egy alkalmas segédtablát a napok nevééről illetve sorrendjéről.

Aki a *Libre Office Calc*-ját használja, annak jóval könnyebb a dolga, ha 1900.01.01. előtti dátumokkal kell dolgoznia: ezeket a „korai dátumokat” ugyanis egyszerűen negatív számokkal azonosítja a program. Lehetőség van továbbá a „0. nap” dátumának kiválasztására, ami kompatibilitási problémákat oldhat meg.

---

<sup>11</sup> Bővebben a szökő évekről: <http://hu.wikipedia.org/wiki/Sz%C3%B6k%C5%91%C3%A9v>

Az eddig leírtak után talán senkinek nem okozok meglepetést azzal, hogy elárulom: **az időpontokat is számként tárolja el az Excel.** Az viszont már lehet, hogy néhányaknak meglepetést okoz, hogy valójában **minden dátum egyben időpont is, és minden időpont egyben dátum is.** Akkor vajon miért beszélünk róluk külön? Hiszen igazából csak az a fontos, hogy az adott „dátum+időt” *dátum*, vagy *idő* formátumban jelenítjük meg. Gondoljunk csak bele: adott két esemény időpontja, például „2012.10.31. 17:36” és „2012.11.20 16:00”. Az *Excel*-nek természetesen képesnek kell lennie válaszolnia arra a kérdésre, hogy mennyi idő telt el a kettő közt. Ha a napokra vagyunk kíváncsiak, a válasz „19”. Ha az órákra is, akkor „19 nap és 22 óra”, vagy esetleg egyszerűen csak „478 óra”. Így már talán érthető, hogy a feladatok sokfélesége miatt nem megengedhető, hogy egy ilyen jellegű program külön kezelje a dátumokat és az időket.

Kérdés, hogy akkor vajon hogy oldja meg, hogy egy eltárolt számból ne csak a hozzá tartozó dátumot, de az időt is megkapjuk. A megoldás nagyon egyszerű: valós számokkal. **A valós szám egész része tartalmazza a dátumra vonatkozó információt, míg a törtrésze az időpontra vonatkozót.** Így megoldhatóak a fentihez hasonló műveletek, melyek eredménye egy újabb szám. Amennyiben csak a napokra vagyunk kíváncsiak, olyan formátumot kell választani, ami csak a szám egész részét mutatja meg, megfelelően átalakítva. Ettől még ott van a tört rész is, csak nem látjuk.

Próbáljuk ki a következőt!

- Formázzunk meg egy cellát valamilyen *dátum* formátumra!
- Vigyünk be egy tetszőleges dátumot, természetesen a kiválasztott formátumnak megfelelően!
- Formázzuk a cellát *óó : pp : mm* típusúra!

A megjelenített érték 00:00:00. Ennek az oka, hogy amíg a szám egész részét adtuk meg, addig később annak törtrészét jelenítettük meg, ami nulla, mivel nem adtuk meg.

Ez természetesen fordítva is így van:

- Állítsuk egy cella formátumát *óó : pp : mm* típusúra!
- Adjunk meg egy tetszőleges időpontot, az előbb beállított formában!
- Állítsuk át a cella formátumát *éééé . hhhh n . -ra!*

Amit látunk: 1900. január **0.**, hiszen a tizedes tört egész része nulla maradt.

Mint ezt a 7. példában is tapasztaltuk, „dátum+idők” importálásakor nagyon résen kell lennünk, mert olykor furcsa dolgok kerülhetnek a munkafüzetbe. Importálást követően mindig válasszunk egy dátumot és időt is megjelenítő formátumot, mielőtt tovább dolgoznánk, hogy ellenőrizhessük, minden rendben van-e!

Az eddig leírtak megértését és a témakörhöz tartozó pár függvény gyakorlását segítő, **javaslom a 15. példa alapos áttanulmányozását.**

### 3. Cellaformázás

#### 3.1. A tartalom igazítása

Eredetileg nem akartam ezen beállítási lehetőségeknek helyet adni ebben az írományban, hiszen alapszintű tudásnak minősül, és az adott cellán jobb egérgombbal kattintva, a *Cellaformázás* ablak *Igazítás* fülén minden lehetőség megtalálható.

Olykor viszont „furcsán viselkedhet”, amit beállítottunk, erre szeretnék egy példát mutatni.

#### 7. példa: Nemes 2010-2011 1. forduló, 5/C (A szabadságharc kronológiája)

A feladat a következőket kéri:

- „*a C oszlop szélessége pontosan 50 egység,*
- *a C oszlopban minden cella pontosan olyan magas, hogy illeszkedik a cellatartalomhoz (azaz minden szöveg látszik, és a cellán belül nincs üres „szövegsor”)*”

A feladat teljesen egyértelmű, be is állíthatjuk előbb az oszlopszélességet 50-re, majd betesszük a pipát oda, hogy *Sortöréssel több sorba*. Nagyjából működik is a dolog, mint ahogy azt vártuk, alaposabban átnézve az oszlopot azonban észrevehető, hogy néhány helyen mégis akad üres „szövegsor”... hogy miért, ki tudja. Ahelyett, hogy elkezdenénk kézzel összehúzogatni ezeket a sorokat, ajánlatosabb a két beállítást fordított sorrendben végrehajtani! Tehát előbb egy tetszőleges, de 50-nél nagyobb oszlopszélességre állítsuk be a sortörést, és csak ezt követően csökkentsük az oszlopszélességet a kívánt méretűre!

Sok bosszankodástól, figyelmetlenség okozta pontvesztéstől menthet meg minket, ha ezt megjegyezzük, és a későbbiekben figyelünk rá.

### 3.2. Sortörés cellán belül

A fejlécek mintának megfelelő formázása során előfordulhat, hogy a *Sortöréssel több sorba* opció használata mellett cellán belüli sortörésre van szükségünk.

#### 8. példa: Nemes 2010-2011 2. forduló, 5/C (A légielő pilótáinak kiképzése)

A mintán jól látható, hogy habár kiferne még a sor végére, hogy „(fő)”, mégis külön sorban van.

	AO	AP	AQ	AR
1	<i>Megoszlás a földi képzési órák és a repülési órák száma szerint (fő)</i>			
2				
3				
4				
5				

*Cellán belüli sortörés.*

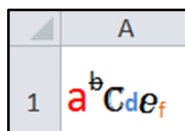
- Álljunk a szerkesztőlécen az új sorba teendő kifejezés elé, és nyomjunk *Alt+Enter* billentyűkombinációt!

Ezt megjegyezni nem nehéz, és nem csak a versenyeken, de a mindennapokban is hasznunkra válhat, tipikusan, amikor az oszlophoz tartozó mértékegységet is a fejlécben akarjuk feltüntetni, nem pedig mindegyik adat mellett egyesével. Én szeretem használni összetett képletek esetén is, hogy könnyebben áttekinthető legyen, mely argumentumok mely függvényekhez tartoznak. Hasznos lehet abban az esetben is, amikor a számítások eredményét egy szöveg részeként adjuk meg: jobb, ha nem bízunk a véletlenre a tördelést.

### 3.3. Karakterformázás

Habár egy nagyon triviális részét képezi a *karakterek formázása* az egyébként sem nehéz *cellaformázás* témakörének, egy dologra mégis érdemesnek találtam kitérni. Ritkán van rá ugyan szükség, de előfordulhat, hogy egy cellán belül többféle

karakterformázást kell beállítani. Például, szeretnénk használni az „m<sup>2</sup>” kifejezést valamely cellában. Erre természetesen van lehetőség: a szerkesztőlécen akár egyesével is ki lehet jelölni a karaktereket, és eltérően formázni őket. Tehát nem csak cellát lehet formázni, hanem annak szöveges tartalmát, részekre bontva is. Így például beleírhatjuk a fejlécbé, hogy az alatta lévő adatok mértékegysége „m<sup>2</sup>”, amennyiben nem szeretnénk az értékek mögött egyesével megjeleníteni.<sup>12</sup>



*Különböző karakterformázás, egy cellán belül.*

### 3.4. Oszlopszélesség, sormagasság

Még az *Excel*-t profi szinten használók közt is akadnak jó páran, akik nincsenek igazán tisztában, milyen mértékegység alapján történik az oszlopok szélességének, sorok magasságának beállítása. Érdekes ezzel tisztában lenni, hiszen bármikor szükségünk lehet rá, hogy például **az oszlopszélesség és sormagasság megegyezzen**, négyzet alakú cellát készítve ezzel.

A *Microsoft Office* honlapján megtaláljuk a megoldást, és mivel nagyon jól és érthetően össze van szedve a lényeg, szó szerint idézek:

- „Egy munkafüzet oszlopaihoz 0 és 255 közötti szélességet adhat meg. A megadott érték a normál betűtípussal formázott cellákban **megjeleníthető karakterek számát jelenti**. Az oszlopok alapértelmezés szerint 8,43 karakter szélesek. Ha az oszlopszélesség 0, az oszlop rejtett lesz.
- A sorokhoz 0 és 409 közötti magasság adható meg. Az érték **pont mértékegységben határozza meg a magasságot** (1 pont mintegy 1/72 hüvelyknek, azaz 0,35 mm-nek felel meg). A sorok alapértelmezés szerint 12,75 pont (tehát körülbelül 1/6 hüvelyk vagy 0,4 cm) magasak. Ha a sormagasság 0, a sor rejtett lesz.
- Ha a **Lap elrendezése nézetben** dolgozik (Nézet lap, Munkafüzetnézetek csoport, Lap elrendezése gomb), lehetősége van hüvelykben megadni az oszlopszélességet vagy a sormagasságot. Ebben a nézetben a hüvelyk az

<sup>12</sup> Lásd a *Formátumkódok/ Mértékegységek* című fejezetben.

*alapértelmezett mértékegység, de centiméterre vagy milliméterre is válthat (Beállítások, majd a Speciális kategória.”<sup>13</sup>*

### 3.5. Formátumkódok

A cellák formátumának megadására a *Cellaformázás* ablakban van lehetőség. Számos beépített lehetőség közül tudunk választani: pénznemek, dátumok, normál alak stb. Bármit is választunk, egy dologgal tisztában kell lennünk: **ezek a beállítások nincsenek kihatással a cella valódi tartalmára**, csupán a megjelenítési formájára.<sup>14</sup>

Például, számok esetén beállíthatjuk a tizedesjegyek számát: ekkor a cella valódi értékét a kívánt módon kerekíti a program, és ezt jeleníti meg, de a tartalma ettől még marad a régi. Így, ha tovább számolunk a cellával, akkor ez **a valódi értékkel történik, és nem a kerekítettel**. Ha mégis a kerekített értékkel kellene tovább dolgoznunk, akkor ehhez olyan függvényt kell használni, mint például a KEREKÍTÉS ( ).

B2		fx =2*A2	
	A	B	
1	a	2a	
2	2,4	4,7	
3			

$$2 \cdot 2,4 = 4,7?$$

Ha a megadott lehetőségeknél speciálisabb igényeink vannak, az *Egyéni* kategóriában magunk is adhatunk meg formátumot, a megfelelő kódokat használva. A továbbiakban ezeket a lehetőségeket tekintem át.

#### 3.5.1. Mértékegységek

Az *Excel* rengeteg beépített formátummal rendelkezik, melyek a begépett számok mögött valamilyen mértékegységet tüntetnek fel, mégsem ritka, hogy valami olyanra van szükségünk, ami nem szerepel a felkínált lehetőségek közt.

<sup>13</sup> A *Microsoft Office* honlapján található leírásban csupán a 2007-es verzióhoz kapcsolódó leírást aktualizáltam a 2010-es verzióhoz:

<http://office.microsoft.com/hu-hu/excel-help/oszlopszelesseg-es-sormagassag-modositasa-HP001216383.aspx>

<sup>14</sup> A cella valódi tartalmáról, illetve annak formátumtól való eltéréséről bővebben a *Dátum- és időkezelés* fejezetben olvashatunk.



Jól tudjuk, mi ilyenkor a teendő: kiválasztjuk az *Egyéni* kategóriát, majd a *Normál* kulcsszó után, dupla idéző jelek közt, megadjuk a kívánt szöveget, ügyelve az elválasztó szóközre.

Cellatartalom	Formátumkód	A formázott cella
1	Normál"kg"	1kg
2	Normál" kg"	2 kg
3	Normál" zsák"	3 zsák
4	Normál" pár"	4 pár
5	Normál" forint"	5 forint

*Formátumkódok és a formázott cellák.*

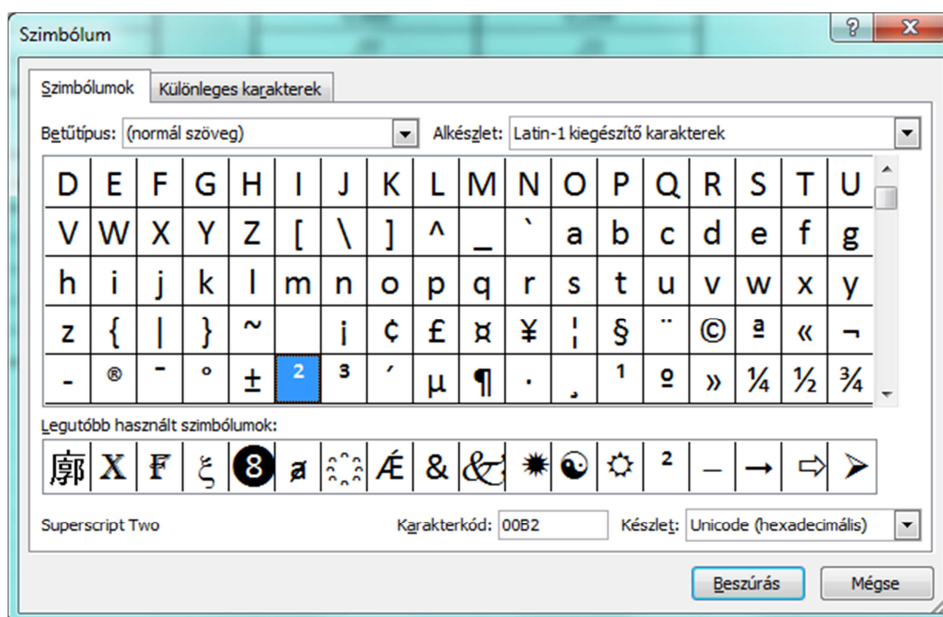
Ebben eddig nem volt semmi ördögösség, valószínűleg nem is mutattam semmi újat. Érdekesebb viszont az a kérdés, hogy vajon mit adjak meg a formátumkódban, ha valami speciális karaktert, esetleg felső vagy alsó indexet szeretnék használni.

Habár a *Cellaformázás* ablakban nincs lehetőség karakterek formázására vagy szimbólumok beszúrására, azért nem kell teljesen lemondani az ilyesfajta igényeinkről. Vágólapról ugyanis van lehetőség speciális karakterek, szimbólumok beszúrására. Tehát nincs más teendőnk, mint valamilyen cellába beszúrni a kinézett szimbólumot, vágólapra helyezni azt, majd a *Cellaformázás* ablakban a vágólapról beilleszteni.

Cellatartalom	Formátumkód	A formázott cella
1	Normál" m <sup>2</sup> "	1 m <sup>2</sup>
2	Normál" 2 <sup>2</sup> "	2 <sup>2</sup>
3	Normál" &"	3 &
4	Normál" 8 "	4 8
5	Normál" 廓"	5 廓

*Formátumkódok és a formázott cellák.*

Nem állítom, hogy minden esetben sikerrel járunk, de túl speciális karakterekre amúgy sincs gyakran szükségünk, sőt, leginkább csak a felső indexelésű *kettes* és *hármás* szokott kelleni. Ezeket a szimbólumokat a *normál szöveg* betűtípusnál találjuk.



*A felső indexelésű kettes beszúrása szimbólumként.*

### 3.5.2. Számok formázása

Már ejtettem pár szót arról, hogy lehetőség van megadni a megjelenített tizedesjegyek számát. Most azt fogom sorra venni, milyen egyéb beállításokra van még lehetőség számok megjelenítése esetén.

Ha a *Szám* kategóriát kiválasztva megadjuk a tizedesjegyek számát, majd a kategóriát átállítjuk *Egyénire*, megnézhetjük az előbbi beállításhoz tartozó formátumkódot. Ugyanitt egyéb beépített lehetőségek is vannak. Ha elkezdjük kipróbálni ezeket, esetleg változtatjuk a 0, # és ? helyőrzők számát a tizedesvessző jobb illetve bal oldalán, máris rengeteg beállítást tudunk kipróbálni.

A próbálgatásból rögtön le is szűrhetjük a tapasztalatainkat. Elsőként nézzük a 0 helyőrző használatát:

Cellatartalom	Formátumkód	A formázott cella
10,235	0000,0	0010,2
1000,235	0000,00	1000,24
1000,235	0 000,000	1 000,235
0,235	,00	,24
1000,235	00,0000	1000,2350

*Formátumkódok és a formázott cellák.*

A tizedesvesszőtől jobbra a tizedesjegyek számát korlátozhatjuk.

- Ha több tizedesjegy van, mint ahány 0, akkor a 0-k számával megegyező számú tizedesjegyre kerekített érték látszik.
- Ha több nulla van, mint ahány tizedesjeggyel az érték rendelkezik, a szám kiegészül a megfelelő számú, egyébként elhagyható szám végi nullákkal.

A tizedesvesszőtől balra az elhagyható nullák kiírását állíthatjuk be.

- Ha több számjegy van, mint ahány 0, nem történik semmi, vagyis a szám a megadott értéknek megfelelő számú számjegyet jelenít meg ezen az oldalon.
- Ha több nulla van, mint ahány számjeggyel az érték rendelkezik, a szám kiegészül a megfelelő számú, egyébként elhagyható szám eleji nullákkal.
- Szóköz elhelyezésével a megfelelő helyen, ezres tagolást adhatunk meg.

Ha nem kívánjuk megjeleníteni a szükségtelen nullákat, használjuk a 0 helyett a # helyőrzőt a kódban! Ugyanazokra a beállításokra ad lehetőséget, és ugyanúgy kell használni is, mint a 0-t. Nézzünk meg most pár érdekesebb példát is!

Cellatartalom	Formátumkód	A formázott cella
1000000,501	#, #	1000000,5
1000000,001	# ###,###	1 000 000,001
1000000,001	# ###,##	1 000 000,
0,35	,##	,35
307782531	"06 ("###") "###"- "###"- "###"	06 (30) 778-25-31
88012354	###"- "###"- "###"	880-123-545

*Formátumkódok és a formázott cellák.*

Egy újabb helyőrző bemutatása következik, mely a két előbbivel megegyező módon egy-egy számjegyet képvisel a kódban: ?. Az elhagyható nullákat nem jeleníti meg, hanem szóközzel tölti ki azok helyét. Erre akkor lehet szükségünk, amikor egy oszlop adatait szeretnénk helyi értékek szerint egymás alá rendezni, tehát szeretnénk, ha a tizedesvesszők egymás alatt helyezkednének el, **bármely igazítás esetén**. Arra kell csak

ügyelni, hogy mindegyik cellában ugyanazt a kódot adjuk meg, és hogy sehol ne legyen kevesebb ?, mint ahány számjegyből áll majd a szám.

Lehetőség van ezzel a módszerrel tört alakban írt kifejezéseket is egymás alá igazítani, ekkor a törtjelek kerülnek egymás alá.

Cellatartalom	Formátumkód	A formázott cella
-10,235	? ???,??	- 10,24
-1000,235	? ???,??	-1 000,24
155000,235	? ???,??	155 000,24
0,235	? ??? ???,???	,235
1000000,235	? ??? ???,???	1 000 000,235
2,25	# "??/??	2 1/4
3,42	# "??/??	3 21/50

*Formátumkódok és a formázott cellák.*

Habár ritkán van erre szükség, előfordulhat az is, hogy a szám elején vagy végén annyi elhagyható nullát szeretnénk megjeleníteni, amennyi csak kifer az oszlopban. Erre a \* kód használatával van lehetőség.

Cellatartalom	Formátumkód	A formázott cella
10,235	0000,*0	0010,00000000000000000000
1000,235	*0,00	000000000000000000001000,24
1000,235	*0,*0	1000,00000000000000000000

*Formátumkódok és a formázott cellák.*

Idetartozik még az \_ karakter bemutatása: a mögötte szereplő karakternek megfelelő térköz kihagyására használható. *Courier* betűtípus esetén ez semmivel sem jelent többet, mint egy szóköz, hiszen ott a karakterek egyforma szélesek. Más betűtípus esetén viszont, ha az adatokat egymás alá szeretnénk helyezni, a szóköz elhelyezése nem lesz elég. Erre a későbbiekben mutatok majd egy konkrét példát (10. példa).

Nagyon nagy vagy éppen kicsi számok megjelenítése esetén szükség lehet *normálalak* használatára, mely segíti az értékek áttekintését és összehasonlítását. Erre van egy beépített lehetőség, melyet a *Tudományos* kategóriában találunk meg. Meg lehet adni a tizedesjegyek számát, de az *Egyéni* kategóriára átkattintva (miután egyszer már elfogadtuk a *Tudományos* beállítást) egyéb változtatásokat is végezhetünk a formátumkód átírásával.

Cellatartalom	Formátumkód	A formázott cella
0,00056001	0,00E+000	5,60E-004
0,00056001	0,0000E+00	5,6001E-04
560100000	0,00E+00	5,60E+08
560100000	0,0000E+00	5,6010E+08

*Formátumkódok és a formázott cellák.*

Gyakran az értékeket százalékként szeretnénk megjeleníteni, erre szintén van egy beépített lehetőség, a *Cellaformázás* ablak *Százalék* kategóriájában. Mindig döntsük el előre, hogy előbb megadjuk az értékeket, és aztán formázzuk a cellákat, vagy fordítva, mert például ha 50%-ot szeretnénk megjeleníteni, akkor előbbi esetben 0,5-et kell beütnünk a cellába, míg a másodikban 50-et.

Olykor az is előfordul, hogy vegyes beállításra van szükségünk. Például szeretnénk, ha a nem egész számok kettő tizedesjegyre kerekítve jelennének meg, miközben az egész számok a tizedesvessző és az elhagyható nullák nélkül. Ilyenkor feltételes formázásra van szükség, mely feltételének kitalálása a *Feltételes formázás* című fejezet áttanulmányozása után nem lesz nehéz.

### 3.5.3. Dátumformátumok

Habár utaltam már rá, hogy nem túl szerencsés az összetartozó dátumot és időt elválasztani egymástól, számos esetben csak az egyikkel foglalkozunk, a másikkal nem. Ilyenkor természetes az igény, hogy olyan formátumot válasszunk, ami az eltárolt számból csak a *dátum*, vagy csak az *idő* részt mutatja meg. Ezért kaptak itt mégis két külön alfejezetet.

Számos beépített lehetőség áll rendelkezésünkre dátumok megjelenítésére, de érdemes tisztában lenni az egyes kódrészekkel, hogy teljesen speciális, saját igényeinknek megfelelő formában is megjeleníthessük őket. Alább egy összefoglaló táblázat látható, mely egy kicsit más formában az *Office* honlapján található:<sup>15</sup>

Kód	Leírás	Formátum
<b>h</b>	A hónapot számként, az első nulla nélkül jeleníti meg.	<i>1-12</i>
<b>hh</b>	A hónapot számként, a megfelelő esetben az első nullával együtt jeleníti meg.	<i>01-12</i>
<b>hhh</b>	A hónapot rövidítéssel jeleníti meg.	<i>jan-dec</i>
<b>hhhh</b>	A hónapot teljes nevén jeleníti meg.	<i>január-december</i>
<b>hhhhh</b>	A hónapot egyetlen betűvel jeleníti meg.	<i>j-d</i>
<b>n</b>	A napot számként, az első nulla nélkül jeleníti meg.	<i>1-31</i>
<b>nn</b>	A napot számként, a megfelelő esetben az első nullával együtt jeleníti meg.	<i>01-31</i>
<b>nnn</b>	A napot rövidítéssel jeleníti meg.	<i>H-V</i>
<b>nnnn</b>	A napot teljes nevén jeleníti meg.	<i>hétfő-vasárnap</i>
<b>éé</b>	Az évet két számjeggyel jeleníti meg.	<i>0-99</i>
<b>éééé</b>	Az évet négy számjeggyel jeleníti meg.	<i>1900-9999</i>

---

15

<http://office.microsoft.com/hu-hu/excel-help/egyeni-szamformatum-letrehozasa-es-torlese-HP010342372.aspx?CTT=1>

### 3.5.4. Időformátumok

Akár csak a dátumoknál, a kívánt formátumot itt is kódok összeépítésével állíthatjuk be: az *óra*, a *perc*, és a *másodperc* különböző kódjainak egymásutánjával.

Kód	Leírás	Formátum
<b>ó</b>	Az órát számként, az első nulla nélkül jeleníti meg.	0–23
<b>óó</b>	Az órát számként, a megfelelő esetben az első nullával együtt jeleníti meg. Ha a formátum tartalmazza az <i>AM</i> vagy a <i>PM</i> értéket, az órák száma a 12 órás rendszeren alapul. Egyébként az órák a 24 órás rendszeren alapulnak.	00–23
<b>p</b>	A percet számként, az első nulla nélkül jeleníti meg. Megjegyzés: A <i>p</i> vagy a <i>pp</i> kódot közvetlenül az <i>ó</i> vagy az <i>óó</i> kód után, illetve az <i>mm</i> kód előtt kell megadni.	0–59
<b>pp</b>	A percet számként, a megfelelő esetben az első nullával együtt jeleníti meg. Megjegyzés: A <i>p</i> vagy a <i>pp</i> kódot közvetlenül az <i>ó</i> vagy az <i>óó</i> kód után, illetve az <i>mm</i> kód előtt kell megadni.	00–59
<b>m</b>	A másodpercet számként, az első nulla nélkül jeleníti meg.	0–59
<b>mm</b>	A másodpercet számként, a megfelelő esetben az első nullával együtt jeleníti meg. Ha a másodperc törtrészét is meg szeretnénk jeleníteni, a következő formátumhoz hasonlóan kell használni: <i>ó : pp : mm , 00</i> .	00–59
<b>ó AM/PM , ó am/pm</b>	Az órát a 12 órás rendszer szerint jeleníti meg. Az Excel az éjfél és dél közötti időpontokhoz a <i>de.</i> , a dél és éjfél közötti időpontokhoz a <i>du.</i> karaktereket jeleníti meg.	0-12 <i>de./du.</i>
<i>A következő formátumok sokszor zavaróak vagy félrevezetőek lehetnek, ezért csak indokolt esetben használjuk őket!</i>		
<b>[ó] : pp</b>	Az eltelt időt jeleníti meg, órában. Akkor szokás használni, ha olyan képletet használunk, melyben a visszaadott órák száma meghaladja a 24-et.	26:02
<b>[pp] : mm</b>	Az eltelt időt jeleníti meg, percben. Akkor szokás használni, ha olyan képletet használunk, amelyben a visszaadott percek száma meghaladja a 60-at.	62:16
<b>[mm] , 00</b>	Az eltelt időt jeleníti meg másodpercben és századmásodpercben. Akkor szokás használni, ha olyan képletet használunk, amelyben a visszaadott másodpercek száma meghaladja a 60-at.	3735,80

### 3.5.5. *Feltételek használata*

Lehetőség van arra is, hogy a cella tartalmától függően más-más formában jelenítsük meg az adatot. A feltételt *szögletes zárójelek* közt kell megadni, ezt követi a formátumkód. Az egyes részeket pontosvessző választja el, és maximum négy rész adható meg, melyek valamilyen **relációs jellel megadott kifejezések**.

Ezzel a módszerrel lehetőség van például arra, hogy a túl kicsi vagy túl nagy adatokat normálalakban formázzuk, míg a többi számot a hagyományos formában jelenítsük meg. Azt is lehetőség van külön kezelni, ha a felhasználó véletlenül szöveget adott volna meg szám helyett, amolyan hibüzenet formájában.

Az egyes részek balról jobbra haladva történnek kiértékelésre, így a harmadik részhez nem is adunk meg feltételt, hanem amolyan „egyébként ágnak” tekintjük. A negyedik rész szöveges adat megadásának formátum-beállítására van fenntartva. Ebben a részben szokás a @ jelet is használni, mely a bevitt szöveges adatot adja vissza. Például:

...;@„: ez nem egy szám”

Cellatartalom	Formátumkód	A formázott cella
0,3	[<0,001]0,00E+00;>999999]0,00E+00;0,0000;"nem szám"	0,3000
-8	[<0,001]0,00E+00;>999999]0,00E+00;0,0000;"nem szám"	-8,00E+00
1500000	[<0,001]0,00E+00;>999999]0,00E+00;0,0000;"nem szám"	1,50E+06
0,00002	[<0,001]0,00E+00;>999999]0,00E+00;0,0000;"nem szám"	2,00E-05
zzz	[<0,001]0,00E+00;>999999]0,00E+00;0,0000;"nem szám"	nem szám

*Formátumkódok és a formázott cellák.*

### 9. példa: Egy saját példa: „többlet és hiány” típusú cellaformázás

Formázzuk a cellákat a következő feltételeknek megfelelően:

- *A pozitív értékek mögé írjuk oda a „Ft”-ot, és hogy „többlet”!*
- *A negatív értékek mögé azt, hogy „Ft hiány”!*
- *Az értékek mindkét esetben helyi értékek szerint egymás alatt legyenek!*
- *Nulla megadása esetén csak egy gondolatjelet jelenítsünk meg!*
- *Ha nem számot adtak meg, figyelmeztessük a felhasználót a „HIBA” kifejezés megjelenítésével!*



Adjuk meg egyelőre csak a feltételeket:

```
[>0]#" Ft többlet";
[<0]-#" Ft hiány";
"-";
"HIBA"
```

A megfelelő igazítást kell még elérni. Elsőként valószínűleg mindenki a ? karakter használatával próbálkozik, hiszen jól tudjuk, hogy az éppen erre való. Kivéve, ha a szám mögé többféle szöveget kell írni! Az alábbi képlet sajnos nem fog eredményre vezetni:

```
[>0]??? ???" Ft többlet";
[<0]??? ???" Ft hiány";
"-";
"HIBA"
```

Az első, amit észreveszünk, hogy negatív számok esetén nem írja ki a *mínusz* előjelet. Ennek oka a feltételek használata: ha szeretnénk, hogy az is megjelenjen (bár bizonyos esetekben ez felesleges), akkor a „??? ???” elé oda kell írunk azt is. Ekkor viszont tapasztalhatjuk a korábban már bemutatott jelenséget, hogy kis számok esetén a *mínusz* jel jóval az első megjelenített számjegy elé kerül. Igazából kár is ezzel tovább próbálkozni: a ? karakter használata semmiképp nem lesz eredményre vezető, hiszen a szám mögé kiírt két kifejezés különböző hosszú.

Ha *Courier* betűtípust adnánk meg, és balra igazítást, elég lenne megnézni, hány karakterrel tartalmaz többet a hosszabbik kifejezés, és kitoldani jobbról a rövidebbet ennyi szóközzel:

```
[>0]#" Ft·többlet";
[<0]#" Ft·hiány··";
"-";
"HIBA"
```

Tetszőleges betűtípus megadása esetén a \_ karakter használatára van szükség. Ha nem kellene kiírunk, hogy „hiány”, csak le kellene foglalnunk az igazításhoz annyi térközt, mint amennyit a „többlet” kifejezés elfoglal, a következő formátumkód (még mindig balra igazítás esetén) jól működne:

```
[>0]#" Ft többlet";
[<0]#" Ft "_t_ö_b_b_l_e_t;
"-";
"HIBA"
```

Hogy a „hiány”-t is ki tudjuk írni az igazítás megtartásával, egy kicsit tovább kell okoskodnunk: foglaljunk le mindkét esetben annyi helyet, mint amennyit a két kifejezés együttesen kiad! Ebből a következő képlet születik:

```
[>0]#" Ft többlet"_h_i_á_n_y;
[<0]#" Ft hiány"_t_ö_b_b_l_e_t;
"-";
"HIBA"
```

A kívánt igazítást csak akkor kapjuk meg, ha a cellákat **jobbra igazítjuk!**

Cellatartalom	A formázott cella
10000	10000 Ft többlet
-5800	5800 Ft hiány
0	-
a1200	HIBA

*A formátumkód hatása a cellákra, jobbra igazítás esetén.*

Persze lehet még szépíteni a formázást, például a „-” és „HIBA” igazításával, de ezt már az olvasóra bízom.

### 3.5.6. Színek megadása

Habár feltételtől függően színezni feltételes formázással is lehet, olykor egyszerűbb a formátumkódot használni erre a célra. Magyar nyelvi beállítások mellett a következő színeket adhatjuk meg:

- [fekete]
- [fehér]
- [zöld]
- [kék]
- [piros]
- [sárga]
- [bíbor]

### 10. példa: Nemes 2009-2010 3. forduló, 5/E (Holdak)

Egy holdak felfedezésének adatait tartalmazó táblázat alapján kell dolgozni. A felhasználó megad két holdnevet, nekünk pedig egyetlen szóval kell válaszolnunk arra a kérdésre, hogy az elsőként kiválasztott hold felfedezésének évvel *későbbi*, *korábbi*, vagy *ugyanaz*, mint a másodikként kiválasztott hold felfedezésének éve. Az eredménytől függően kell a kiírt szövegnek pirosnak, kéknek, vagy zöldnek lennie.

A feladat szövege segít abban, hogy rájövjünk, hogy formátumkódokkal kell dolgoznunk:

- „A középső (a lenti ábrákon „színes”) cellát töltsd ki egy olyan numerikus (tehát nem szöveges!) értéket adó képlettel, amely minden esetben a két hold felfedezési évének különbségét tartalmazza! Oldd meg, hogy – bármelyik két holdat is választjuk, – a cellában mégse ez az (egész) szám jelenjen meg, hanem a lenti három lehetőség közül a megfelelő, mégpedig éppen a minta szerinti színnel!”

Phoebe	felfedezésének éve	későbbi	mint	Triton	felfedezésének éve
Nereida	felfedezésének éve	korábbi	mint	Charon	felfedezésének éve
Larissa (89N2)	felfedezésének éve	ugyanaz	mint	Proteus (89N1)	felfedezésének éve

A három lehetőség, és megjelenítésének formája.

- Készítsük el a két legördülő listát, az E110 és K110 cellákhoz csatolva!<sup>16</sup>
- Számítsuk ki a kért cellába a két hold felfedezési évének különbségét!

$$=INDEX(D7:D99;E110)-INDEX(D7:D99;K110)$$

- Állítsuk be a következő formátumkódot a *Cellaformázás Egyéni* kategóriájában!

[Piros][>0]"későbbi";[Kék][<0]"korábbi";[Zöld]"ugyanaz"

Tehát ha a különbség pozitív, akkor a szám helyett azt írjuk csak ki, hogy *későbbi*, piros színnel. Ha a különbség negatív, kékkel írjuk ki, hogy *korábbi*. A „különböen ág” most azt az esetet takarja le, amikor a különbség éppen nulla, itt nem kell feltételt megadni, csak a zöld színt, és a kiírandó szöveget.

<sup>16</sup> Ennek mikéntjéről a *Vezérlők* fejezetben olvashatsz.

Abban a speciális esetben, amikor a feltételeink éppen ezek: *pozitív, negatív, nulla*, igazából el is hagyhatóak a feltételek, ezek ugyanis az alapértelmezett esetek. Az előző feltétel a megoldásfájlból ebben a rövidített alakban szerepel:

```
[Piros]"későbbi";[Kék]"korábbi";[Zöld]"ugyanaz".
```

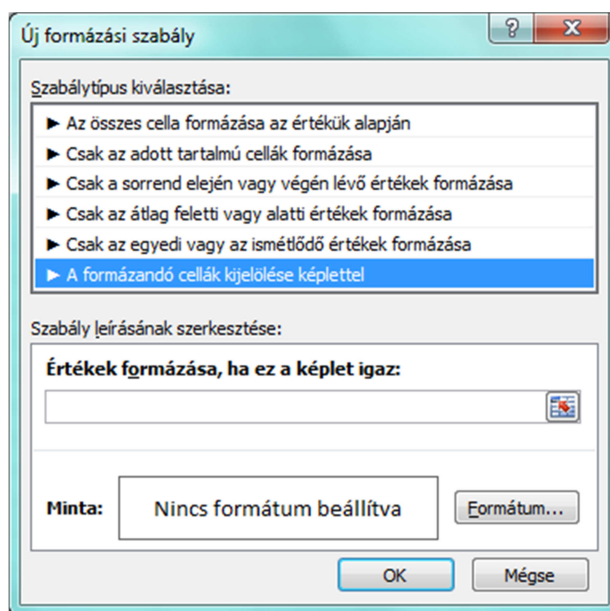
### 3.6. Feltételes formázás

Az adatok gyors áttekinthetőségét azzal is elősegítjük, hogy formázással kiemelünk bizonyos tulajdonságú adatokat, elősegítve ezzel azok könnyű összehasonlítását és a kirívó esetek gyors észrevételét. Ezeket a formázásokat beállíthatnánk éppen manuálisan is, de az egyrésztől sokkal több időt igényelne, másrésztől az adatok megváltozása esetén kezdenénk az egészet elölről.

Az ilyen esetekre a feltételes formázás való, ami nem csak a gyakorlati életben hasznos, de kedvelt témája a versenypéldáknak is: szinte nincs is olyan forduló, amikor nincs ehhez kapcsolódó feladat. Ritka azonban, hogy a példa szövege konkrétan a funkció használatát kéri, elvárás ugyanis a versenyzőktől, hogy maguktól is felismerjék, hogy erre van szükség. Aki ennek nem tesz eleget, és kézzel végzi el a formázást, biztosan nem kap egyetlen pontot sem a feladatra.

Az újabb *Excel* rengeteg beépített lehetőséget tartalmaz, úgy, mint *színskálák, ikonok* használata, értéktől függő *színátmenetes kitöltés* stb. Ezekről olvashatunk a súgóban, vagy felfedezhetjük a lehetőségeket magunk is.

A versenyeken leginkább arra szokott szükség lenni, hogy egy új szabályt hozzunk létre, ahol a formázandó cellákat képlettel jelöljük ki. A formátum megadása ekkor a már jól ismert *Cellák formázása* ablakban történik.



*Speciális (nem beépített) szabály megadásakor az utolsó szabálytípust kell választani.*

Tehát egy új szabály létrehozásakor a következőket kell megadni:

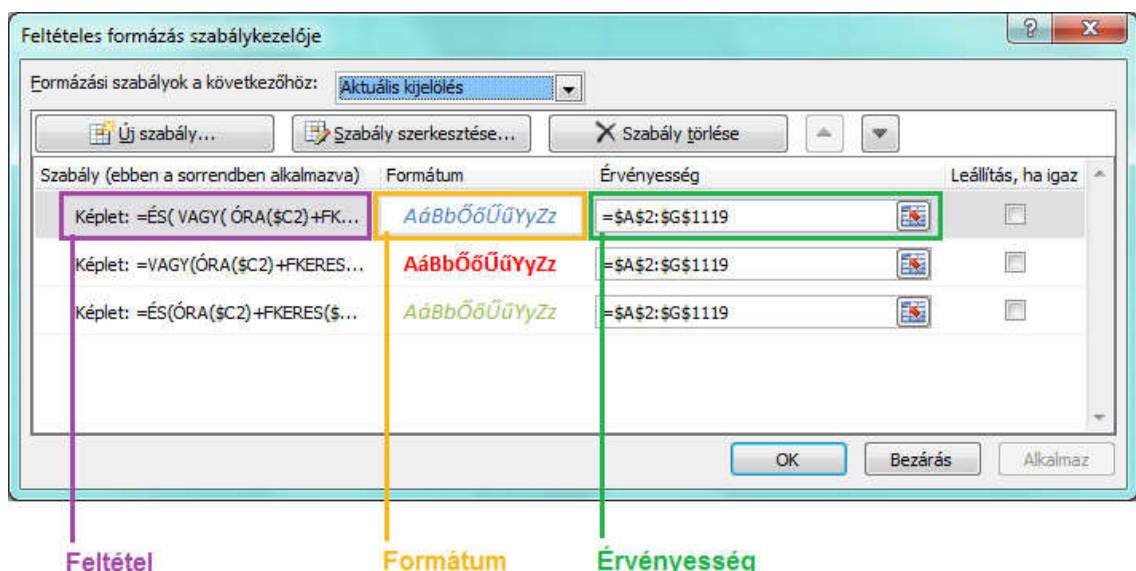
- *Érvényességet*, vagyis mely cellákat akarjuk formázni, amennyiben teljesül rájuk a feltétel. A legegyszerűbb szerintem, ha már az elején kijelöljük az egész érintett tartományt, így a szabály az egész kijelölésre vonatkozik majd.<sup>17</sup>
- *Formátumot*, vagyis hogyan módosuljanak a feltételnek eleget tevő cellák eredeti állapotukhoz képest. Ha például nem adunk meg betűméretet, akkor minden érintett cella karakterei maradnak akkorák, amekkorák voltak eredetileg, bármekkora is legyen az.
- *Feltételt*, mely egy IGAZ/HAMIS értéket tartalmazó kifejezés kell, legyen. Ezt mindig a kijelölt tartomány bal felső elemére kell megfogalmazni. Csakúgy, mint a képletek másolásánál, abszolút hivatkozás esetén a tartomány minden cellájára ugyanaz a képlet vonatkozik majd, míg relatív hivatkozás esetén annyit „csúsznak” a képletben szereplő sorszámok lefelé, mint amilyen messze van az érintett cella a tartomány első sorától, illetve annyit nőnek az oszlopszámok, mint ahány oszlopra van a kijelölés bal szélső oszlopától.

Nekem egyébként az a módszerem, hogy összetettebb képletek esetén, vagy ha nem tudom pontosan a használt függvény argumentumlistáját, a képletet előbb a munkalap egy cellájában készítem el. Zavaró számomra, hogy a feltételes formázás ablakában, a feltétel megadásakor nem működik a súgó, ráadásul a billentyűzet nyilait sem tudom a

<sup>17</sup> Néhányan jobban szeretik csak egy cellára/sorra létrehozni a formázást, majd a *Formátummásoló* ecsettel átvinni a többi cellára is. Ezt a módszert azonban veszélyesen találom, mert megnehezíti a megfelelő hivatkozások megadását.

megszokott módon használni. Azt is furcsának találom, hogy míg egy cellában elkészítve a képletet alapértelmezetten relatív hivatkozások keletkeznek, amikor a használandó cellákra kattintok, addig a feltételes formázás ablakában az abszolút hivatkozás a sztenderd.

Ha elkészültem, és úgy tűnik, hogy jól működik, kijelölöm a képletet a szerkesztőlécen, és a vágólap segítségével beillesztem a feltételes formázás ablakába. Ha meg valami gond van, akkor könnyebb megtalálni a hibát: másoljuk a képletet lefele, illetve jobbra, hogy megnézzük, a bal felső saroktól lefele, illetve jobbra eső cellákra milyen feltétel lesz érvényben. Gyakran a problémát ugyanis csak a nem megfelelő hivatkozás okozza.



*Kezdőlap/ Stílusok/ Feltételes formázás/ Szabályok kezelése. Az éppen aktuális kijelölésre három szabály vonatkozik, ezek mind ugyanarra a tartományra érvényesek.*<sup>18</sup>

A *Szabályok kezelése* ablakban lehetőség van megnézni, hogy az éppen aktuális kijelölésben, vagy akár az egész munkalapon milyen szabályok vannak érvényben. Ezek közül egyet kijelölve módosíthatjuk a paramétereit (*Szabály szerkesztése*), vagy akár törölhetjük is (*Szabály törlése*).

Az ablakban ez a számomra megtévesztő tájékoztatás olvasható: „Szabály (ebben a sorrendben alkalmazva)”. Talán jobb lett volna a „fontossági sorrend” kifejezést használni, a félreértések elkerülése végett. Ennek akkor van jelentősége, ha egy cellára

<sup>18</sup> Az *Érvényességnél* a *dollárjelek* automatikusan kerülnek oda, de egyébként semmi jelentőségük sincs.

több, egymásnak ellentmondó formázást tartalmazó szabály feltétele is teljesül. Például, ha pirosra akarjuk színezni az 5-nél nagyobb értékeket, zöldre pedig a párosakat, akkor mi legyen az 5-nél nagyobb páros számokkal? A kérdés ebben a formában nem megválaszolható, szükség van annak pontosítására. Az ablakban a fentebb megadott feltétel számít fontosabbnak, így elsősorban az érvényesül. Ha például az előbb említett piros színezést tartjuk fontosabbnak, annak kell az elsőnek lennie. Ekkor 6 piros lesz, és nem zöld. A nyilak segítségével lehetőség van a szabályok felcserélésre, ezzel változtatva prioritásukat. Például a zöld színezést az első helyre áthelyezve 6 már nem piros, hanem zöld színű lesz. Ha pedig a zöld betűszín helyett zöld háttérszín állítanánk be, akkor a mindkét feltételnek eleget tevő cellák mindegyike zöld háttérszínnel és piros betűszínnel rendelkezne, függetlenül a feltételek sorrendjétől.

Nekem sokkal testhezállobb a sorrend értelmezésének algoritmikus megközelítése. Eszerint a szabályok kiértékelése éppen ellenkező sorrendben történik, mint a felsorolás: alulról felfelé haladva teljesíti az *Excel* a szabályokban megadottakat, elrontva esetleg ezzel egy korábbi (lentebbi) szabály formázását. Természetesen ez a fajta megközelítés is ugyanazt az eredményt szolgáltatja, mint az *Excel* súgójában szereplő fontossági sorrend, amennyiben nem használjuk a *Leállítás, ha igaz* jelölőnégyzeteket.

A feltételekhez tartozik egy *Leállítás, ha igaz* jelölőnégyzet is. Ez egyrészt a korábbi verziókkal való kompatibilitás végett került ide, másrészt sok esetben megkönnyítheti a szabályok feltételeinek megfogalmazását is. Erre mutatok majd egy példát, a *Négyzetszámok kiemelése 10×10-es táblázatban* című feladatban, de előbb – hogy könnyebben megértsük az előbb leírtakat – nézzünk meg egy egyszerűbb példát!

### **11. példa: Egy tetszőleges táblázat minden második sorának árnyékolása**

Versenyen sem ritka<sup>19</sup>, hogy egy összetettebb feltételes formázási feladatnak egy része az, hogy a táblázat minden második sora legyen a többitől eltérő színű, de egyébként is gyakran alkalmazzák ezt a módszert, hiszen jó eséllyel megakadályozza a sorok közti „elcsúszást” olvasáskor.

Természetesen az nem opció, hogy minden második sort kijelöljük, majd beállítjuk rá a háttérszín, hiszen egy nagyobb táblázat esetén ez lehetetlen, és egy sor kitörlésénél vagy beszúrásánál kezdhethetnénk előlről a formázást. A megoldás a feltételes formázás.

---

<sup>19</sup> Például OKTV 2005-2006 3. forduló, 5/B (Távolságmутató).

- Készítsük el a képletet egy félreeső cellában (így könnyebb szerkeszteni a képletet)! Ne feledjük, a feltételt a kijelölt tartomány **bal felső cellájára** (A2) kell megadni:

=MARADÉK (SOR ( A2 ) ; 2 ) =0!

Mivel a SOR ( ) függvény az adott sor minden oszlopában ugyanazt az eredményt adja, így kivételesen nem számít, teszünk-e \$ jelet az A elé. A sorokra viszont muszáj relatívan hivatkozni, mert különben a lejjebb lévő sorokat is a szerint színezné, hogy a tartomány első sora páros sorszámú-e. Érdekes a képletet lefele végigmásolni, hogy lássuk a hivatkozások alakulását és a feltétel teljesülését minden sorra.

- Másoljuk az A2-re elkészült képletet a vágólapra!
- Jelöljük ki a formázandó táblázatot! (→ *Érvényesség*)
- Válasszuk a *Kezdőlap* megfelelő csoportjában a feltételes formázást, majd adjunk meg új szabályt: *A formázandó cellák kijelölése képlettel!*
- Illesszük be a képletet! (→ *Feltétel*)

A megadott feltétel minden egyes cellára IGAZ vagy HAMIS értéket ad. Mivel relatív hivatkozást használtunk, B2-re a feltétel valójában

=MARADÉK (SOR ( B2 ) ; 2 ) =0 lesz, A3-ra

=MARADÉK (SOR ( A3 ) ; 2 ) =0 stb.

- Állítsuk be a kívánt háttérszínt! (→ *Formátum*)

A szabályt alkalmazva az eredmény látható: a felváltott színezés akkor sem romlik el, ha beszúrunk vagy kitörlünk egy sort.

A példa megértését követően érdemesnek tartom a feltételes formázás leírásának újbóli elolvasását, a leírtak elmélyítése érdekében.

## 12. példa: Nemes 2010-2011 1. forduló, 5/C (A szabadságharc kronológiája)

Az 1848-1849-es szabadságharc eseményeit tartalmazza a táblázat, melynek sorait a következőképpen kell színezni:



- a) „sárga a háttere (az A..D oszlopokban egyaránt) minden olyan eseménynek, amelynél a C oszlopban szerepel az „országgyűlés” szó, és ez önmagától módosul, ha a feltétel megváltozik,
- b) piros a betűszíne (az A..D oszlopokban egyaránt) minden olyan eseménynek, amely vasárnapra esett, és ez önmagától módosul, ha a feltétel megváltozik,
- c) sárga háttéren piros a betűszíne (az A..D oszlopokban egyaránt) minden olyan eseménynek, amely egyszerre megfelel az előző két elvárásnak, és ez is önmagától módosul, ha a feltétel megváltozik.”

A harmadik pont igazából nem tartogat számunkra feladatot, hiszen a betűszín és a háttérszín nem kizáró formázások: ha megadjuk az első két pontban leírtaknak megfelelő szabályt, akkor azok a sorok, amelyekre mindkét feltétel teljesül, mindkét formázással rendelkezni fognak. Megjegyezném, hogy bizonyos korábbi *Excel* verzióknál ez még nem volt igaz, ezért is szerepel sok helyütt a pontozókban külön az és kapcsolatnak megfelelő feltétel megadása.

Sajnos feltételes formázásnál nem lehet helyettesítő karaktereket használni, így szövegkezelő függvényekre van szükség a feladat megoldásához.

- Készítsük el E2-ben a formázandó tartomány első sorára az a) feltételt!

=SZÖVEG.KERES("országgyűlés";\$C2)<HOSSZ(\$C2)

Ha szükségét érezzük, másoljuk a képletet lefele, a hivatkozások ellenőrzése végett!

- Másoljuk az E2-ben található képletet a vágólapra!
- Jelöljük ki az egész A1:D329 tartományt! (→ *Érvényesség*)
- A feltételes formázás ablakában illesszük be a feltételt, majd állítsuk be a sárga hátteret! (→ *Feltétel, Formátum*)
- Készítsük el F2-ben a formázandó tartomány első sorára a b) feltételt, majd másoljuk a vágólapra:

=\$D2="vasárnap"!

- Jelöljük ki az egész A1:D329 tartományt, majd a feltételes formázás ablakában illesszük be a feltételt, végül állítsuk be a piros betűszínt! (→ *Érvényesség, Feltétel, Formátum*)

Mindkét esetben, a teljes feltétel a sorokra vonatkozik, de annak egyes részei a megfelelő oszlopokra értendőek. Ezért használtunk **a soroknál relatív, az oszlopoknál pedig abszolút hivatkozást**.

### 13. példa: Egy saját példa: Négyzetszámok

Ezzel a példával először azt fogom bemutatni, hogyan alakulnak a hivatkozások, amennyiben nem soronként, hanem cellánként kell formáznunk, majd a *Leállítás, ha igaz* jelölőnégyzet működését.

Emeljük ki a négyzetszámokat narancssárga háttérzínnel egy 10×10-es, számokat tartalmazó táblázatban! A nem négyzetszámok közül a párosak legyenek kék, a páratlanok zöld betűszínnel formázva!

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

*Négyzetszámok kiemelése.*

- Készítsük el a táblázatot az *A1:J10* tartományba! A legegyszerűbb, ha feltöltjük 1-től 100-ig számokkal, de ennek most igazából nincs is jelentősége: olyan megoldást kell adnunk, ami a számok megváltozása esetén is jól működik.

Foglalkozzunk egyelőre csak a négyzetszámok kiemelésével!

Ellentétben mind az előbbi, mind az ezt követő példával, most nem sorokat, hanem egyesével **a cellákat kell színezn**i, ezért másképp alakulnak a hivatkozások.

- Adjuk meg a képletet *A1*-re, a *K1* cellában!

$$=INT(GYÖK(A1))=GYÖK(A1)$$

Mind az oszlopra, mind a sorszámra **relatíván kell hivatkoznunk**, hogy a cellák formátuma a saját tartalmuktól függjön. Érdeemes a képletet végigmásolni lefele illetve jobbra 9 soron illetve oszlopon át, hogy lássuk, az egyes cellákra milyen képlet adódik majd, és azok milyen értéket vesznek fel.

- Másoljuk a vágólapra az *A1*-re írt feltételt!
- Jelöljük ki az *A1:J10* tartományt! (→ *Érvényesség*)
- Válasszuk a szokásos menüpontokat a feltételes formázás ablakának megjelenéséhez, és illesszük be a képletet a megfelelő helyre! (→ *Feltétel*)
- Adjuk meg formátumnak a narancssárga kitöltést! (→ *Formátum*)

Fontos, hogy a kitöltésen kívül **más formázást ne adjunk meg**, így a betűszínt is hagyjuk *automatikusan*. Ekkor a formázás egész egyszerűen „békén hagyja” a cella eredeti betűszínét, legyen az bármilyen.

A négyzetszámok kiemelésével végeztünk, a páros és páratlan számok formázása következik. A szabály létrehozása az előbb leírt lépésekben történik, így részletezni nem fogom.

- Készítsük el a páros számok formázását biztosító szabályt! A feltétel:

$$=MARADÉK(A1;2)=0$$

- Készítsük el a páratlan számok formázását biztosító szabályt! A feltétel:

$$=MARADÉK(A1;2)=1$$

Látható, hogy ekkor nem az történik, mint amit a feladat kért: a négyzetszámokat is kiszíneztük kékre illetve zöldre, holott azoknak csak a narancssárga háttérrel kellett volna ellátni. Az ok az, hogy az egyik feltétel lefutását követően lefut egy másik is. **Sokan ilyenkor beállítják a négyzetszámokat kiemelő szabály formátumában a betűszínt feketére, helytelenül.** Látszatra ugyan megoldja a problémát, amennyiben az adataink eredetileg feketék voltak, de akkor is hibás a működése: az egész táblázat kijelölését követő betűszín-módosítás nem lesz hatással egyetlen cellára sem, holott a feladat szövege alapján a négyzetszámoknak a kézzel beállított színnel kellene rendelkezniük.

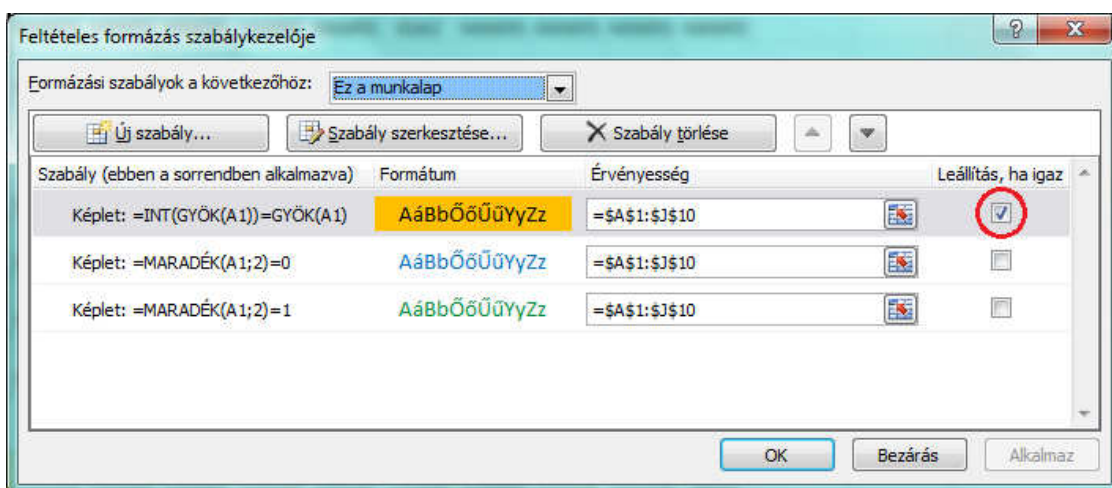
Szoftverhiba okozhatja azt a jelenséget, hogy ha egyszer megadjuk az *automatikus* helyett a fekete betűszínt, hiába is állítjuk később azt vissza *automatikusra*, a **fekete betűszín** „benragad”. Ilyenkor ki kell törölni a szabályt, és újra megadni.

Megoldja az előbbi problémát, ha a feltételeket kiegészítjük a következőképp:

=ÉS (NEM (INT (GYÖK (A1)) =GYÖK (A1)) ; MARADÉK (A1 ; 2) =0) és

=ÉS (NEM (INT (GYÖK (A1)) =GYÖK (A1)) ; MARADÉK (A1 ; 2) =1)

Ebben a konkrét példában ez még nem nagy munka, összetettebb szabályok esetén viszont könnyen azzá válhat. E helyett a módszer helyett bölcsebb inkább az ezt a célt szolgáló *Leállítás, ha igaz* funkciót használni: ha már egyszer formáztuk az egyik feltétel szerint, a többi szabályt ne vegye figyelembe. Ekkor a feltételek a következő egyszerű formában maradnak:



A *Leállítás, ha igaz* funkció használata esetén egyszerűsödnek a feltételek.

#### 14. példa: OKTV 2009-2010 2. forduló, 3/O (Afrimpex)

Adott egy cég telefonhívásainak adatait tartalmazó táblázat, ahol a hívások időtartama soha nem éri el az egy órát. A feladat a sorok formázása, a következőképp:

- „ha egy beszélgetés az afrikai helyi idő szerint még 9 óra előtt kezdődött, de pontban 9-kor vagy azután ért véget, akkor annak minden adata dőlt, zöld karakterekkel jelenjen meg,
- ha egy beszélgetés az afrikai helyi idő szerint még pontban 17 órakor vagy azelőtt kezdődött, de 17 óra után ért véget, akkor annak minden adata dőlt, kék karakterekkel jelenjen meg,

- c) *ha egy beszélgetés az afrikai helyi idő szerint teljes egészében a 9:00:00–17:00:00 (zárt) időtartományon kívül esik, akkor annak minden adata félkövér, piros karakterekkel jelenjen meg,*
- d) *végezetül: ha egy beszélgetés az afrikai helyi idő szerint teljes egészében a 9:00:00–17:00:00 (zárt) időtartományon belül van, akkor annak maradjon meg az eredeti (automatikus) színezése!”*

Az utolsó ponttal most sincs dolgunk, tehát csak három szabályt kell elkészítenünk. Elsőként értelmezzük a feladatot, hogy könnyebben megfogalmazzuk majd a feltételeket! A következő jelöléseket fogom használni: *K* – a hívás kezdete, *V* – a hívás vége.

- a)  $K < 09:00:00$  és  $V \geq 09:00:00$       *dőlt, zöld*
- b)  $K \leq 17:00:00$  és  $V > 17:00:00$       *dőlt, kék*
- c)  $K > 17:00:00$  vagy  $V < 09:00:00$       *félkövér, piros*

Ha valaki nincs tisztában azzal, miként kezeli az *Excel* az időt, szükség van az időpontok függvényekkel történő részekre bontására, mellyel a következő bonyolult feltételek szülehetnek:

```

b) ÉS(
    VAGY(
        ÓRA(K) < 17 ;
        ÉS(
            ÓRA(K) = 17 ;
            PERCEK(K) = 0 ;
            MPERC(K) = 0
        )
    ) ;
    ÉS(
        ÓRA(V) = 17 ;
        VAGY(
            PERCEK(V) <> 0 ;
            MPERC(V) <> 0
        )
    )
)

```

Ha valaki arról is elfeledkezik, hogy a hívások mindenképpen egy óra alattiak, akkor a képlet még tovább bonyolítható. Persze a feladat így is megoldható, de mivel összetettebbek a képletek, könnyebben hibázhatunk. A mellékelt megoldásban ez a módszer is megtalálható, a *Hívások (3)* munkalapon.

A másik módszerhez kitalálásához érdemes előbb átgondolni, miként is tárolódnak az *Excel*-ben az időpontok. Valójában minden időpont egyben dátum is, és fordítva. A két fogalom különválasztását csupán az adja, hogy a tizedes vesszőtől jobbra vagy balra lévő számjegyekkel foglalkozunk-e. Tehát ha csak az adott dátum *idő* részével szeretnénk foglalkozni (mivel most csak azok lényegesek), egész egyszerűen „le kell vágni” a tizedes vesszőtől jobbra eső részt, majd csak ezzel a „levágott” résszel végezni az összehasonlításokat. Erre kiváló a MARADÉK ( ) függvény, amennyiben osztónak 1-et adunk meg.

A kívánt feltételek tehát így is megfogalmazhatóak:<sup>20</sup>

- a) ÉS (
- $$\text{MARADÉK}(K;1) < \text{IDŐ}(9;0;0);$$
- $$\text{MARADÉK}(V;1) \geq \text{IDŐ}(9;0;0)$$
- )
- b) ÉS (
- $$\text{MARADÉK}(K;1) \leq \text{IDŐ}(17;0;0);$$
- $$\text{MARADÉK}(V;1) > \text{IDŐ}(17;0;0)$$
- )
- c) VAGY (
- $$\text{MARADÉK}(K;1) > \text{IDŐ}(17;0;0);$$
- $$\text{MARADÉK}(V;1) < \text{IDŐ}(9;0;0)$$
- )

Érdemes ezeket a táblázattól jobbra ténylegesen elkészíteni, hogy lássuk, működik-e a megfontolásunk. Mellesleg így sokkal átláthatóbb a képlet, könnyebb szerkeszteni, és működik a függvények sűgőja is, ha megfeledekneknk az argumentumlistáról.

Adjuk meg a helyes hivatkozásokat! Ne feledjük, az elkészített képlet jelen példánál A2-re vonatkozik. Mivel azt szeretnénk, hogy az A oszloptól jobbra is ugyanez legyen a feltétel – vagyis pontosan ezeket a cellákat vizsgálja – „le kell rögzíteni az oszlopokat egy dollárjellel” (abszolút hivatkozás). A sorokban lefele haladva már az aktuális sor értékeitől szeretnénk, hogy a formázás függjön, tehát a sorszámok elé ne tegyünk dollárjelet (relatív hivatkozás)!

---

<sup>20</sup> A MARADÉK ( ) függvény helyett használható a K-INT(K) módszer is.

Szükség van még  $K$  és  $V$  meghatározására, hiszen a  $C$  és  $D$  oszlopban a magyar időszámítás szerinti időpontok vannak, nem pedig a tárcsázott országoké. A dolog első nekifutásra egyszerűnek tűnik: kikeressük például `FKERES()` függvénnyel az időeltolódások értékeit, és hozzáadjuk a vizsgált időponthoz. Ez azonban csak akkor lenne jó, ha az adott időpont `ÓRA()` részéhez adnánk, mint ahogy először elkezdtünk dolgozni. A mostani módszerünk az, hogy időpontok helyett tizedes törteket képzelünk magunk elé, és annak is a vesszőtől jobbra eső részével foglalkozunk csak. Az 1 hozzáadása nem változtatna az értéken semmit (valójában egy nap eltolást jelent), ezért egy olyan érték sokszorosát kell hozzáadnunk a magyar időpontokhoz, ami az egy óra értékével megegyezik:

$$\frac{1}{24} \approx 0,04167$$

Így már meg tudjuk fogalmazni a feltételeket<sup>21</sup>, és nekiláthatunk a feltételes formázás beállításainak.

- Ha még nem tettük volna meg, készítsük el a végleges feltételeket, például a táblázat jobb szélén! Nyugodtan másoljuk is végig, ellenőrzésképp.
- Másoljuk a vágólapra az egyik feltétel képletét, természetesen a második sorból!
- Jelöljük ki az A2: *G1119* tartományt! Ezt úgy a leggyorsabb, hogy az A2-es cellába állunk, majd a *Ctrl* és *Shift* gombok nyomva tartása mellett előbb egy *jobbra*, majd pedig egy *lefele* nyilat nyomunk. (→ *Érvényesség*)
- Válasszuk a *Kezdőlap/ Stílusok/ Feltételes formázás/ Új szabály* parancsot! Ezen belül természetesen most is az utolsó szabálytípusra van szükségünk: válaszuk a *Formázandó cellák kijelölése képlettel* opciót!
- Illesszük be a beviteli mezőbe a korábban kimásolt képletet! (→ *Feltétel*)
- A *Formátum* gombra kattintva, állítsuk be a beillesztett szabályhoz tartozó formázást: a betűszínt, és hogy *dőlt* vagy *félkövér* stílusú! (→ *Formátum*)
- Ismételjük meg a fent leírtakat a másik két feltétellel is!

---

<sup>21</sup> Érdemes megnézni a *Hívások (2)* munkalapot, ahol szerepelnek az említett segédcellák is, bennük a képlettel.

## 4. Adatok áttekintését segítő eszközök

Ebben a fejezetben olyan eszközöket fogok bemutatni, melyek az adatok áttekintését segítik elő. Beépített funkciók lévén **használatuk nagyon egyszerű**, az eredmény pedig látványos. Sajnos azonban meg van a hátrányuk is: az utolsó alfejezetben bemutatott **részösszeget kivéve nincsenek tekintettel az adatok módosulására, vagyis nem adatkövetők.**<sup>22</sup>

### 4.1. Rendezés

Mivel a rendezés az egyik leglátványosabb és egyben legkönnyebben elvégezhető művelet a táblázatokkal, tanítására már a korai szakaszban sor kerül. Emiatt versenyen ritkán kérnek egy az egyben rendezést, inkább csak amolyan szükséges lépésként jelenik meg, egy komolyabb feladat részeként. Először egy egyszerű példán keresztül mutatom be használatát, majd a hibázási lehetőségeket, érdekességeket veszem sorra.

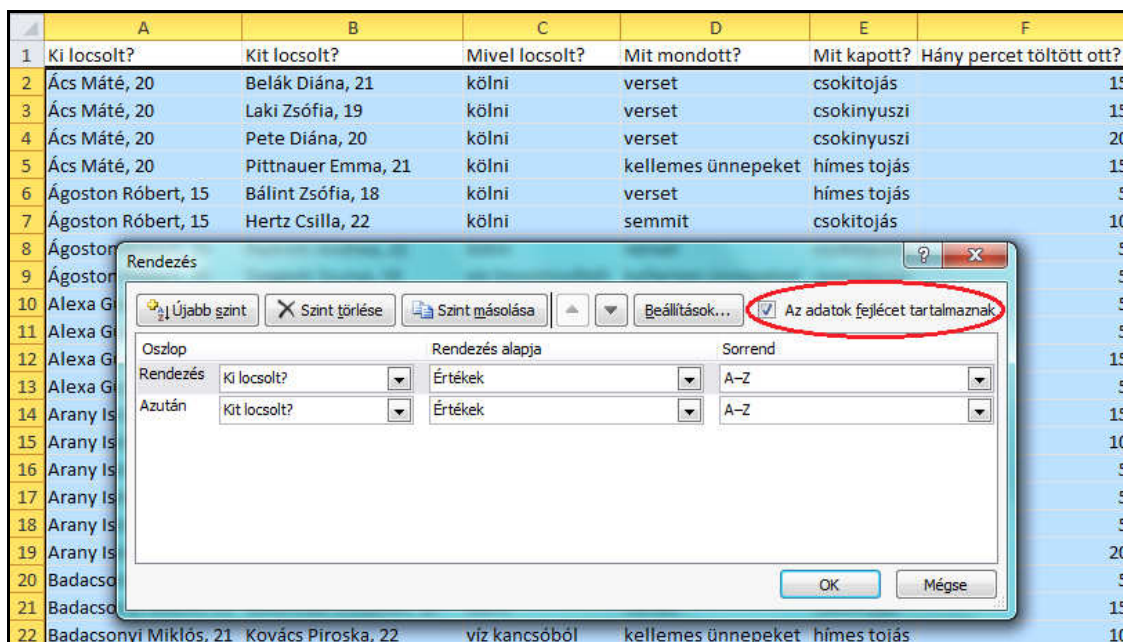
#### 15. példa: Nemes 2009-2010 1. forduló, 5/A (Locsolóverseny)

Ebben a példában nincs szükség importálásra, az adatokat *xls* formátumban kapjuk meg. A feladat a munkalap elnevezése, és a kapott táblázat rendezése, két szempont alapján. Először is a locsolók nevét (*Ki locsolt?*) kell abc szerint növekvően rendezni. Mivel egy locsoló általában több helyen is járt, ugyanazon név több, egymás alatti sorban is szerepelhet. Az ilyen esetekben van értelme annak, hogy még egy szempont alapján is rendezzük az adatokat: a feladat azt kéri, hogy a locsolókon belül a meglocsoltak nevei (*Kit locsolt?*) is legyenek abc rendben.

---

<sup>22</sup> Az „adatkövető” kifejezésen a továbbiakban is mindig azt fogom érteni, hogy a kiinduló adatok megváltozása automatikusan hatással van az eredményre. Ez egyébként egy általános elvárás táblázatkezelésnél, például ezért nem határozzuk meg „ránézésre” egy adathalmaz maximumát, és használunk inkább MAX ( ) függvényt.







A feladat megoldása: rendezés két szempont alapján, a fejlécek megfelelő kezelése mellett.

A legújabb Excel mellett már sokkal könnyebb dolgunk van, mint a régebbi verziók esetén. Nincs szükség a rendezendő táblázat kijelölésére, elég annak egyik celláján állni. A *Rendezés*re kattintva ugyanis automatikusan kijelölésre kerül az az összefüggő rész (táblázat), ahol állunk. Kézi kijelölésre csak akkor van szükség, ha valamilyen speciális helyzetből adódóan a szokásostól eltérő módon helyezkednek el rendezendő adataink.

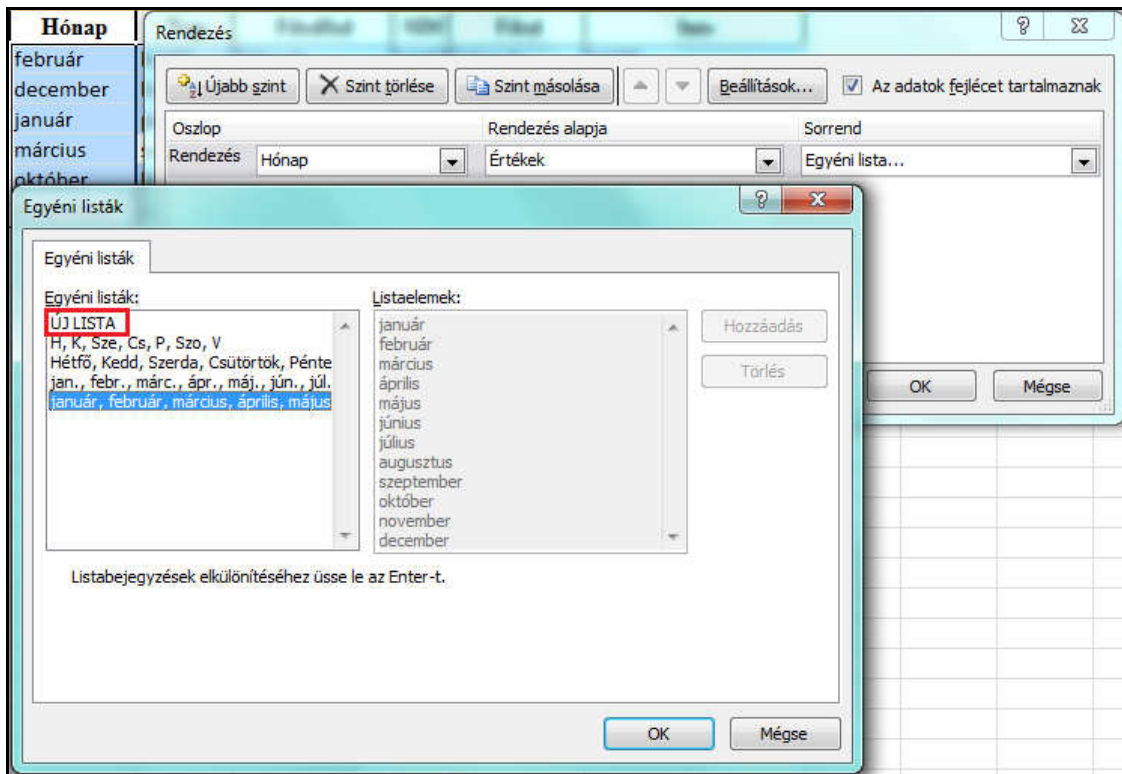
- A táblázatban állva kattintsunk az *Adatok* fülön a *Rendezés* gombra!
- A *Rendezés* ablakot töltsük ki az ábrának megfelelően, majd hajtsuk végre a rendezést!

A feladatot ezzel megoldottuk.

A gyakorlatlanabbak könnyen eshetnek abba a hiába, hogy **véletlenül a fejléct is beleveszik a rendezésbe**. Ennek a hibának az elkerülését egy jelölőnégyzet segíti, illetve árulkodó jel lehet a nem megfelelő beállításról az is, hogy az oszlopok kiválasztására szolgáló legördülő listában nem az oszlopok neve, hanem betűjele szerepel.

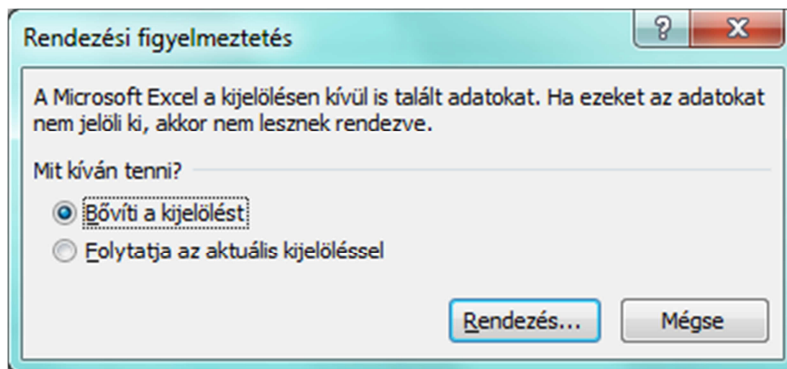
A *Rendezés* gomb mellett megtalálható két másik gomb is:  és . Ezek azt a célt szolgálják, hogy meggyorsítsák az előbb leírt folyamatot, amennyiben csak egy szintű rendezést szeretnénk.

Jó, ha tudjuk, hogy a hagyományos rendezési módszereken túl lehetőség úgynevezett *egyéni lista* szerinti rendezésre is. Például, ha egy oszlop hónapneveket tartalmaz, könnyen lehet, hogy azokat nem ábécé szerint szeretnénk sorba rakni, hanem aszerint, hogy hogyan követik egymást az évben. Konkrétan hónapok esetén van is egy beépített lista erre az esetre, speciális igények esetén pedig definiálhatunk mi magunk is „sorrendet”.



*Egyéni lista szerinti rendezés, új lista megadása.*

Ügyeljünk, hogy sose kövessük el azt a hibát, hogy úgy kattintunk a három *rendezés* gomb valamelyikére, hogy csak pár oszlop van kijelölve a táblázatból! Ha egyetlen oszlopot jelöltünk ki, még kapunk egy figyelmeztető üzenetet az *Excel*-től, de a nem megfelelő rádiógombot kiválasztva így is **végérvényesen elronthatjuk, összekeverhetjük adatainkat**. Sőt, ha több oszlopot is kijelölünk, de nem az összeset, akkor még csak **hibaiüzenetet sem kapunk**.



*Ha nem a kijelölés bővítését választjuk, az adatok összekeveredhetnek.*

Fontos azt is mindig szem előtt tartani, hogy a rendezés **nem adatkövető** funkció. Ha új sort adunk meg, vagy átírjuk egy korábbi sor tartalmát, ismét kérni kell az adatok rendezését, hiszen ez magától nem történik meg. Előfordulhat, hogy a versenypélda készítője pont azt akarja tesztelni, hogy a diákok észben tartják-e ezt. Elképzelhető, hogy rendezést követően olyan feladatot adnak fel, ami elrontja a korábban beállított rendezést. Ilyenkor bizony figyelmeztetés nélkül vissza kell lépni a rendezési feladatra, és **újra elvégezni** azt!

Egyedi adatok esetén – tehát amikor a rendezendő adatok közt nincs két egyforma – azért van módszer arra, hogy a rendezés az adatok változása esetén automatikusan aktualizálódjon, ekkor azonban nem ezt a beépített funkciót kell használni, hanem magunk kell, hogy elvégezzük a feladatot. A következő példa ezt mutatja be.

### **16. példa: Nemes 2008-2009 3. forduló, 4/F (ECDL)**

A feladat egy automatikusan frissülő vizsgarangsor készítése, amit ráadásul egy másik munkafüzet adataiból kell elkészíteni. Az eddig tárgyalt *Rendezés* funkció most nem lesz segítségünkre, hiszen azzal a kettő közül egyik feltétel sem valósítható meg. A feladatot képletekkel kell megoldanunk.

Nézzük előbb, hogy is néz ki ez a fájl, és az elkészítendő rangsor!

	A	L	M	N	O
1			2007	2008	2009
2	Bács-Kiskun		8 912	6 979	1 241
3	Baranya		6 312	4 721	682
4	Békés		8 137	7 164	1 260
5	Borsod-Abaúj-Zemplén		15 394	15 603	1 892
6	Budapest		32 866	33 584	4 832
7	Csongrád		4 068	5 278	952
8	Fejér		6 833	7 319	1 268
9	Győr-Moson-Sopron		7 211	7 991	1 085
10	Hajdú-Bihar		9 464	10 356	2 084
11	Heves		5 891	6 232	865
12	Jász-Nagykun-Szolnok	...	7 949	7 821	1 056
13	Komárom-Esztergom		4 825	4 723	489
14	Nógrád		4 240	3 619	585
15	Pest		12 297	11 049	1 950
16	Somogy		6 735	5 837	762
17	Szabolcs-Szatmár-Bereg		9 461	10 722	1 666
18	Tolna		4 071	4 006	547
19	Vas		5 940	6 381	622
20	Veszprém		4 779	5 691	706
21	Zala		7 561	6 695	957

2008-as vizsgarangsor:	
Budapest	
Borsod-Abaúj-Zemplén	
Pest	
Szabolcs-Szatmár-Bereg	
Hajdú-Bihar	
Győr-Moson-Sopron	
Jász-Nagykun-Szolnok	
Fejér	
Békés	
Bács-Kiskun	
Zala	
Vas	
Heves	
Somogy	
Veszprém	
Csongrád	
Komárom-Esztergom	
Baranya	
Tolna	
Nógrád	

A baloldalon az *ECDL.xls* adatai, a jobb oldalon az előbbi adatok alapján készült vizsgarangsor, az *NJSZT.xlsx* egy munkalapján.

Akik használták már a `NAGY()` függvényt, azoknak biztosan egyből beugrik, hogy ez lesz a megoldás kulcsa. A továbbiakban bemutatom a végeredményhez vezető okoskodás egy lehetséges lépéssorozatát, tehát a végleges megoldás csak a végén fog kijönni.

Mivel az *NJSZT.xlsx*-ben is szerepelnek ezek az adatok, először dolgozzunk ebből, elhagyva azt a feltételt, hogy az *ECDL.xls* adatait használjuk fel.

- Határozzuk meg a 2008-ban legtöbben vizsgázók **számát**, A30-ba, a `NAGY()` függvény használatával!

=NAGY(M2:M21;1)

- Módosítsuk az előbbi képletet, hogy tudjuk másolni egészen A49-ig!

=NAGY(\$M\$2:\$M\$21;1)

- Most azt kell kiokoskodni, hogyan érhető el, hogy a `NAGY()` függvény második argumentuma a rangsor első sorában 1, a másodikban 2 stb. értéket vegyen fel. Nyilván ez az érték attól függ, hányadik sorban állunk, az első sorhoz képest. Ebből kitalálható, hogy a `SOR()` függvényt kell használnunk, figyelembe véve, hogy a rangsor a 30. sorban kezdődik:

=NAGY(\$M\$2:\$M\$21;SOR()-29)

- Jöhet a már jól begyakorolt INDEX()–HOL.VAN() függvénytáras, hogy a vizsgázók száma helyett a megye nevét jelenítsük meg:

```
=INDEX(
  $A$2:$A$21;
  HOL.VAN(
    NAGY($M$2:$M$21;SOR()-29);
    $M$2:$M$21;
    0
  )
)
```

- Nincs más hátra, mint a helyi tartományhivatkozásokat lecserélni az *ECDL.xls* fájl megfelelő tartományaira. Ehhez nyissuk meg a fájlt **ugyanabban az Excel programban**, és az előbbi képlet szerkesztésekor - a lecserélendő hivatkozás kijelölését követően - egyszerűen csak kattintsunk át a másik munkafüzetre, és jelöljük ki a megfelelő munkalap szükséges tartományait!

Ismét hangsúlyoznám, hogy ez a módszer csak akkor működik, ha a vizsgázók száma mind különböző. Ha ez nem teljesül, és például Budapesten és Győrben is a 33584-en vizsgáztak, akkor a képlet mindkét esetben Budapestet fogja visszaadni, hiszen az szerepel előbb a táblázatban. Érdekes ezt kipróbálni, és azt is, hogy mi történik, ha megváltoztatunk egy eredményt az *ECDL.xls* fájl megfelelő tartományában, majd mentést követően megnyitjuk az elkészült *NJSZT.xlsx*-t.

Sajnos nem kizárt, hogy időnként nem működik helyesen az *Excel Rendezés* funkciója: az adatok átrendezése elronthatja a hivatkozásokat. A probléma egy kis odafigyeléssel megelőzhető, de akkor sem lenne szabad ennek előfordulnia.

### 17. példa: Nemes 2009-2010 2. forduló, 4 (Damimpex)

Nyissuk meg az elkészített munkafüzetet a *megoldas* mappából, a *Telefonkönyv\_rossz* munkalapon! A táblázatban egy cégnél dolgozók neve, beosztása és telefonmellékük van megadva, a hozzájuk tartozó országot illetve az összes hívás adatai pedig függvények adják.

Az ügyintézőkhöz tartozó országot így számoljuk:

```
=HA(
  B2="ügyintéző";
  INDEX(
    Hívások!$B$2:$B$1119;
    HOL.VAN( Telefonkönyv_rossz!C2;Hívások!$A$2:$A$111
  9;0)
  );
  ""
)
```

➤ Rendezzük a táblázatot *Mellékek* szerinti növekvő sorrendbe!

Elsőre talán fel sem tűnik, de jobban megnézve a képleteket, megdöbbenve tapasztalhatjuk, hogy a mellékekre történő hivatkozások összekeveredtek, és így az egész táblázat elromlott. Ha ezt nem vesszük észre időben, rengeteg munkánk vesztet kárba, vagy akár adatokat is veszthetünk.

Név	Beosztás	Mellék	Ügyintézőknél ország	Összes hívás száma	Összes hívás időtartama	Összes hívás költsége
Tóth Tímea	középvezető	110		22	7:48:58	37 025 Ft
Badacsonyi Krisztián	ügyintéző	111	Guyana	23	11:08:05	54 960 Ft
Tóth Vanda	ügyintéző	112	Venezuela	24	10:48:01	53 190 Ft
Toldi Tamás	ügyintéző	113	Guyana	6	2:53:57	14 350 Ft
Bakonyi Mátyás	ügyintéző	114	=HA(B15="ügyintéző";IND	10	3:53:46	20 745 Ft
Marosi István	ügyintéző	115	Uruguay	26	12:50:43	63 570 Ft
Mák Anna	ügyintéző	116	Kolumbia	30	13:46:18	64 650 Ft

#### *Összekeveredett hivatkozások.*

Látható, hogy az előbbi képletben a C2 előtt ott van a munkalap neve is. Ez teljesen felesleges, de gondot nem okozhat – gondolná az ember. Arról nem is beszélve, hogy a hivatkozásokban a munkalapok neveit ritkán adjuk meg szándékosan, begépeléssel, hiszen ezt az *Excel* maga is elvégzi a cellák kijelölésekor. Itt például annyi történt, hogy a *Hívások* munkalapról visszakattintottam az aktuális munkalapra, és a munkalap megjelölése máris odakerült a cella azonosítója elé. A probléma forrása azonban éppen ez a felesleges munkalap-hivatkozás, hogy miért, nem tudom. Mindenesetre, ha ezeket kitoröljük, már rendezhetjük a táblázatot, minden gond nélkül.

Nem árt tehát, ha rendezést követően mindig ellenőrizzük, minden rendben van-e. Ha ehhez hasonló hibát tapasztalunk, megoldás lehet a képlet átírása is: a cellákra történő hivatkozás megadható *SOR()* függvénnyel is, a hagyományos módszer helyett:

```
=HA(
  INDIREKT("B"&SOR())="ügyintéző";
  INDEX(
    Hívások!$B$2:$B$1119;
    HOL.VAN(INDIREKT("C"&SOR()));Hívások!$A$2:$A$1119;
  0)
);
"
```

Ez azonban nem egy univerzális módszer, és több munkát is igényel, mintha odafigyelnénk a kénytelen munkalap-hivatkozások feltűnésére, csupán érdekességképp mutattam meg.

## 4.2. Autoszűrő

Az *autoszűrő* szintén egy nagyon egyszerűen használható, ugyanakkor nagyon hasznos funkciója az *Excel*-nek. Egy összefüggő táblázat valamely mezőjén állva, majd az *Adatok* fül/ *Szűrő* gombjára kattintva az oszlopok fejlécében legördülő listát jelző nyíl jelenik meg. A gombra kattintva számos lehetőség kínálkozik, ezekről bővebben a *Microsoft* hivatalos honlapján olvashatunk<sup>23</sup>.

Amit viszont fontosnak tartok leírni, hogy hogyan is működik ez a funkció. Ezt is csak azért, hogy össze lehessen hasonlítani a következő alfejezetben bemutatott *irányított szűrő*vel, ami már kedvelt témája a versenyeknek.

Először is: egy általunk megadott feltétel a fejléc legördülő listájában ugyanazt eredményezi, mintha kézzel kijelölnénk az összes olyan sort, amire nem igaz a feltétel, majd a helyi menü *Elrejtés* parancsát használnánk. Így csak **azon sorok maradnak láthatóak, melyekre teljesül a megadott feltétel**. Tehát a sorok **helyben történő, adatvesztéssel nem járó** szűréséről van szó.<sup>24</sup>

Másodszor: a szolgáltatás **automatikusan nem frissül**, tehát az adatok módosítása esetén nem változik a sorok elrejtett, illetve megjelenített mivolta.

---

<sup>23</sup> Konkrétabban itt:

<http://office.microsoft.com/hu-hu/excel-help/adatok-szures-tartomanyban-vagy-tablazatban-HP010342517.aspx?CTT=1>.

<sup>24</sup> Szűrés alatt ebben a fejezetben azt értjük, hogy csak a feltételeknek megfelelő adatokkal foglalkozunk. Úgy, mint amikor téstát szűrünk (nem pedig, mint amikor teát).

Nagy hátránya, hogy **egy munkalapon egyszerre csak egy cellatartományon használható**, valamint a szűkítési lehetőségek is korlátozottabbak a következő alfejezetben leírt módszerrel szemben. **Az oszlopokra megadott feltételek közt és kapcsolat áll fenn**, vagy kapcsolat csak az egy oszlopra megadott feltételek közé helyezhető el, különböző oszlopok feltételei közé nem.

Különlegessége, hogy nem csak értékek szerint lehet szűrni, hanem például formátum alapján is.

### 4.3. Irányított szűrő

Ez az opció az előbbi *Szűrő* gomb mellett érhető el, és *Speciális* névre hallgat, tehát a *speciális szűrés* és az *irányított szűrés* kifejezések ugyanazt jelentik. Voltaképp ugyanarra való, mint az előbbi *autoszűrő*, de egy kicsit más elven működik, és sokkal több mindenre ad lehetőséget: lehetőség van bonyolultabb képletek megadására, és a szűrés helyének megadására. Ugyanakkor **ez sem adatkövető**, tehát az adatok módosítása esetén újból el kell végezni a szűrést.

Lehetőség van az adatokat oly módon szűrni, hogy a kiválogatott sorokat egy külön, **általunk megadott helyre másolja**, meghagyva így a táblázatot az eredeti állapotában is. Választhatjuk **helyben szűrést** is, ekkor hasonlóan működik, mint az előbb ismertetett módszer: **bizonyos sorokat elrejt**. Ha szeretnénk újból visszakapni az eredeti állapotot, kattintsunk a *Szűrők törlése* gombra, miközben az érintett tartomány egy celláján állunk! Ugyanígy lehetőség van a szűrőt újra igénybe venni, az *Újból alkalmaz* gombbal.

**A szűrés feltételeinek megadása egy táblázattal történik**, melynek egy sorába írt feltételeit az *és*, sorait pedig a *vagy* logikai művelet kapcsol össze. Ebből következik, hogy csak az ilyen formájú feltételeket tudjuk a táblázatba beírni:

$$\begin{array}{ll} A \wedge B \wedge C \dots & \vee \\ D \wedge E \dots & \vee \dots, \end{array}$$

ahol *A*, *B*, ... oszlopokra vonatkozó feltételek, beleértve ebbe azt az esetet is, hogy egy soron belül esetleg ugyanarra az oszlopra mondtunk ki több feltételt. Ha a kritériumok nem ilyen formában vannak megadva, akkor át kell alakítani ilyenre.



Fogjuk látni, hogy ez mindig lehetséges, és lesz példa arra is, hogyan. De előbb inkább kezdjük két egyszerűbb példával!

**18. példa: OKTV 2009-2010 1. forduló, 5/E (Karácsonyi ajándékok)**

A táblázatban egy karácsonyi adománygyűjtés adatai találhatók.

	A	B	C	D	E	F
1	Név	Szeretetszolgálat	Iskola	Önkormányzat	Vállalkozók	Külföld
2	Autó	0	220	66	27	0
3	Baba	0	120	118	143	30
4	Dominó	82	136	30	85	8
5	Édesség	0	0	182	77	144
6	Elektromos játék	4	28	191	135	35
7	Faéptő	0	82	9	17	0

*Adományok gyűjtése: az oszlopokban a gyűjtő csoportok, a sorokban az ajándéktípusok találhatóak.*

Azokat a sorokat kell kigyűjteni, melyekre az alábbi két feltétel egyike teljesül:

- „az Önkormányzat egyetlen ilyet sem gyűjtött, de a vállalkozók igen,
- az iskola 120-nál többet, de 150-nél kevesebbet gyűjtött ebből, a Szeretetszolgálat viszont nem kétszámjegyű darabot!”

Segítségünkre van, hogy a feladat szövege két pontba van szedve. Ezeket a pontokat vagy köti össze, tehát a szűrőtartomány egy-egy sorát képezik majd.

- Készítsük el a szűrőtartomány fejlécét: másoljuk az *Önkormányzat*, a *Vállalkozók*, az *Iskola*, és a *Szeretetszolgálat* feliratokat egy félreeső helyre, például I35-től!

Persze be is lehetne gépelni, de a másolás sokkal biztonságosabb. Könnyen előfordulhat ugyanis, hogy valamit elgépelünk, vagy nem veszünk észre egy szöveg végén lévő felesleges szóközt stb. Ekkor az *Excel* nem veszi figyelembe az adott oszlopra írt feltételeket, és így a szűrés hibás eredményt adhat.

- Adjuk meg a feltételeket a következő mintának megfelelően:

	Szeretetszolgálat	Iskola	Iskola	Önkormányzat	Vállalkozók
				0 >0	
	<10	>120	<150		
	>99	>120	<150		

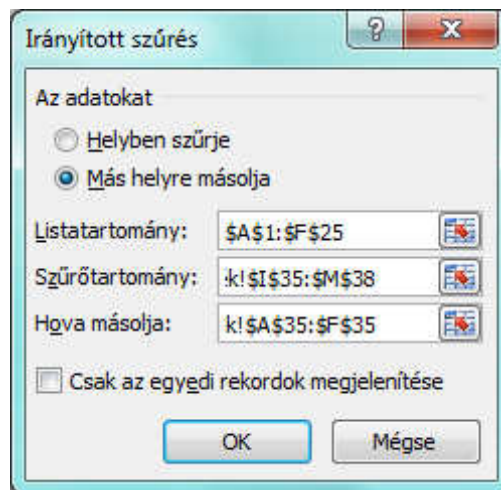
*Az irányított szűrés szűrőtartománya.*

Az első sorba az első feltétel került: „0” és „> 0”. Utóbbit szokás így is megadni:

= " > 0".

Erre azonban nincs szükség, bőven elég az is, hogy „> 0”. A második feltételt már nem lehet egy sorban megjeleníteni, hiszen egy szám kétféleképp lehet nem kétjegyű: tíznél kisebb vagy kilencvenkilencnél nagyobb. Az iskolára vonatkozó feltétel mindkét esetben megmarad!<sup>25</sup>

- Kattintsunk az *Adatok* fül/ *Speciális* gombjára!
- Elsőként meg kell adni a szűrendő táblázatot (*Listatartomány*), majd a feltételeket tartalmazó kritériumtáblát (*Szűrőtartomány*). Mivel az adatokat most nem helyben szűrjük, át kell állítani a rádiógombot az alapértelmezett beállításról, és megadnunk a helyet, ahova a szűrést kérjük (*Hova másolja*).



*Az irányított szűrés ablaka.*

Ahogy a képen is látszik, én úgy adtam meg azt, hogy hova másolja a megfelelő sorokat, hogy *A35*-től kezdődően kijelöltem annyi üres cellát, mint amennyi az eredeti táblázat szélessége. Igazából azonban elég csak az *A35*-re kattintani, mert akkor is ugyanez lesz az eredmény.

Ezzel a feladat készen van: az eredeti táblázat érintetlen maradt, a feltételeknek megfelelő sorokból pedig készült egy új táblázat. Mellette a kritériumtábla, mely a későbbiekben is segít áttekinteni, pontosan milyen szűrés eredménye is került oda.

### **19. példa: OKTV 2009-2010 1. forduló, 5/F (Karácsonyi ajándékok)**

- Az előbbi feladat újabb szűréssel folytatódik:

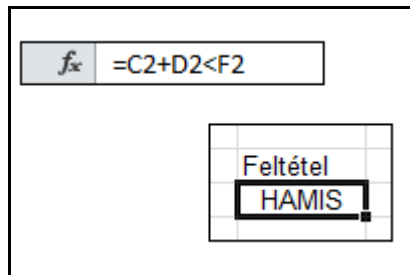
---

<sup>25</sup> Ennek a ténynek az elfelejtése típushibának számít. Mindig gondoljuk át alaposan, hogy mi mindennek kell egyszerre teljesülnie!

- „Gyűjtsd ki az A50-től kezdődően azokat a játékokat, amelyekből az iskola és az Önkormányzat együttesen is kevesebbet gyűjtött, mint amennyi a külföldi adomány volt ezekből! Jelenítsd meg az iskolai, az önkormányzati és a külföldi értékeket is!”

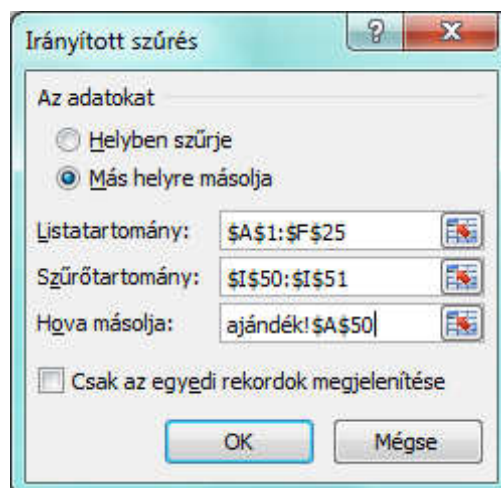
Most egy olyan feltételt kaptunk, ami több oszlop viszonyáról szól. Eddigi módszereink nincsenek segítségünkre, de szerencsére lehetőség van ilyesfajta feltételek megadására is. Oszlopnevet nem kell megadnunk, helyette odaírhatjuk például azt, hogy „Feltétel”. Alá egy olyan logikai vizsgálat jön, mely minden sorra IGAZ vagy HAMIS értéket ad eredményül. A kritériumtáblában ezt mindig a szűrendő táblázat első sorára kell megfogalmazni, de csak úgy, mint amikor képletmásolásnál relatív hivatkozást adunk meg, ez majd minden sorra mást fog jelenteni.

- Készítsük tehát el a következő kritériumtáblát:



A szűrőtartománya és az I51-es cella tartalma.

- Végezzük el a szűrést:



Az irányított szűrés ablaka.

Most csak egy cellát adtam meg a *Hova másolja* mezőben: így is ugyanaz történik, mint az előbbi példában. Ha azonban a feladat azt kéri, hogy csak bizonyos oszlopokat

jelenítsünk meg, netán az eredetitől eltérő sorrendben, akkor ez a módszer pontvesztést okoz.

Lehetőség van arra, hogy előre megadjuk a megjelenítendő oszlopokat, tetszőleges sorrendben. Ehhez csupán a megfelelő oszlopfejléceket kell a kívánt sorrendben bemásolni - jelen esetben *A50*-tól - és aztán ezeket a fejléceket megadni a *Hova másolja* mezőben.

Tehát a feltételek megadása, akárcsak az adatbázisfüggvények esetén, kritériumtáblával történik. Matematikailag bizonyítható, hogy így az értékek mindenfajta szűrésére lehetőség van<sup>26</sup>, ellentétben a korábban tárgyalt módszerrel. Ahhoz azonban, hogy a feltételeket a megfelelő formájúra hozzuk, kell egy kis jártasság a matematikai logikában. Erre mutatok most előbb egy egyszerűbb, majd egy nehezebb példát.

## 20. példa: Egy saját példa: Jedlik-díj<sup>27</sup>

Ez a példa elsősorban azoknak készült, akiknek van egy kis jártasságuk a matematikai logikában, vagy kíváncsiak arra, hogy mi az a matematikai háttér, amire az irányított szűrő működési elve épül. Elolvasnia azért mindenkinek érdemes, hiszen az itt leírtak adnak egy olyan szemléletet, mellyel a típushibák sokkal könnyebben elkerülhetőek. Eljjeszteni azonban senkit nem szeretnék: e példa alapos megértése nélkül is sikeresen szerepelhetünk a versenyen!

Adott egy iskola tanulóinak eredményeit tartalmazó *tanulok.xls*: tanulónként minden tantárgyból az év végi érdemjegy, valamint az év végi átlag. A tanévzárón Jedlik-díjat kapnak azon tanulók, akik fizikából ötösek, az év végi átlaguk nem kevesebb, mint 4,5, és a matematika valamint informatika tantárgyak közül legalább az egyikből szintén ötös érdemjegyet szereztek. Szűrjük ki a Jedlik-díjat kapó tanulók névsorát!

➤ A könnyebb értelmezés végett jelöljük az állításokat egy-egy betűjellel:

- **A:** *fizikából ötös*
- **B:** *az év végi átlag nem kevesebb, mint 4,5*
- **C:** *a matematika és informatika tantárgyak közül legalább az egyikből ötös*

---

<sup>26</sup> Irányított szűrő esetén lehetőség van az oszlopokra szabott feltételeket és illetve vagy kapcsolattal összekötni. A *Boole algebra alaptétele* szerint bármilyen bonyolult logikai kapcsolat kifejezhető az *és*, a *vagy*, és a *tagadás* logikai alapműveletek segítségével, tehát bármely logikai állítás megadható a szűrés feltételének.

<sup>27</sup> Avagy a feltételek kapcsolatának átalakítása *teljes diszjunktív normálformára*.

➤ Rövidítsük az állításokat, és a **C** állítást szedjük két részre:

- **A: Fizika:** „=5”
- **B: Év végi átlag:** „>=4,5”
- **C<sub>1</sub>: Matematika:** „=5”
- **C<sub>2</sub>: Informatika:** „=5”

➤ A szokásos matematikai jelekkel és a bevezetett betűjelekkel a feltétel így írható le:

$$A \wedge B \wedge (C_1 \vee C_2)$$

➤ Ezzel már sokat tettünk annak érdekében, hogy a feltétel megadható legyen kritériumtáblával, egy lényegi lépés azonban még hátravan: <sup>28</sup>

$$A \wedge B \wedge (C_1 \vee C_2) = A \wedge B \wedge C_1 \vee A \wedge B \wedge C_2$$

➤ Ebből már könnyen készíthetünk kritériumtáblát.

Fizika	Év végi átlag	Matek	Informatika
5	>=4,5	5	
5	>=4,5		5

*A feltételek átalakítást követően már megadhatóak szűrőtartományban.*

Az egyik diák ezt a megfigyelést tette: „Ebben az évben, ha valaki matekból és informatikából is ötös lett, akkor kapott Jedlik-díjat”. Vajon helyes-e a megfigyelése? Szűrjük ki azon diákokat, akikre igaz a következtetés!

➤ Az állításokat ismét formalizálnunk kell, hogy aztán felírassuk őket a megfelelő alakban:

- **P: matekból ötös**
- **Q: informatikából ötös**
- **R: Jedlik-díjas**

<sup>28</sup> Ne feledjük, hogy az *és* (konjunkció) precedenciája nagyobb, mint a *vagy* (diszjunkció) műveleté.

- Az állítás a matematika jelöléseivel így írható le:<sup>29</sup>

$$(P \wedge Q) \Rightarrow R$$

- Ahhoz, hogy valaki Jedlik-díjas legyen, a matematika vagy informatika ötös mellett kell az év végi minimum 4,5-ös átlag, és a fizika ötös is.

- **P:** *Matek:* „=5”
- **Q:** *Informatika:* „=5”
- **R<sub>1</sub>:** *Év végi átlag:* „>=4,5”
- **R<sub>2</sub>:** *Fizika:* „=5”
- $(P \wedge Q) \Rightarrow (R_1 \wedge R_2 \wedge (P \vee Q))$

- Ha valaki matematikából és informatikából is ötös, akkor természetesen az is igaz, hogy valamelyikből ötös, így az állítást így is felírhatjuk:

$$(P \wedge Q) \Rightarrow (R_1 \wedge R_2)$$

- Ezt még mindig nem tudjuk szűrőtartományba írni, amíg át nem alakítjuk olyan alakra, amiben csak *és* illetve *vagy* kapcsolat van. Matematikából talán ismerős a következő azonosság<sup>30</sup>:

$$X \Rightarrow Y = \bar{X} \vee Y$$

- Legyen most:

- **X:**  $P \wedge Q$
- **Y:**  $R_1 \wedge R_2$

- Tehát:

$$X \Rightarrow Y = \bar{X} \vee Y = \overline{P \wedge Q} \vee (R_1 \wedge R_2)$$

- Kihasználva a *De-Morgan* azonosságot:

$$\overline{P \wedge Q} \vee (R_1 \wedge R_2) = \bar{P} \vee \bar{Q} \vee R_1 \wedge R_2$$

---

<sup>29</sup> A kettősnyíl a *következtetés* (implikáció) jele.

<sup>30</sup> A felülvonás a tagadás (negáció) műveletét jelöli.

➤ Ebből már el lehet készíteni a kritérium-táblázatot:

Fizika	Év végi átlag	Matek	Informatika
		<5	
			<5
	5 >=4,5		

A feltételek átalakítást követően már megadhatóak szűrőtartományban.

Így azok a diákok lettek kiszűrve, akikre teljesült az észrevétel. Ha csak egy diák is kimaradt ebből, akkor a megfigyelés téves volt, ellenpéldának jók a kimaradó diákok. Vagy készítünk egy szűrést azokra is, akikre nem teljesül a megfigyelés. Ehhez tagadni kell az előbbi feltételt:

$$\overline{P \vee Q \vee R_1 \wedge R_2} = P \wedge Q \wedge \overline{R_1 \wedge R_2} = P \wedge Q \wedge (\overline{R_1} \vee \overline{R_2}) = P \wedge Q \wedge \overline{R_1} \vee P \wedge Q \wedge \overline{R_2}$$

Félreértés ne essék: ha ez a szűrés nem is ad eredményt, akkor sem mondhatjuk azt, hogy a „ha valaki matekból és informatikából is ötös, akkor kap Jedlik-díjat” állítás igaz! A matematika és informatika ötös természetesen nem elégséges feltétele a Jedlik-díjnak, mert például fizikából is ötösnek kell lenni. Az irányított szűrő elkészítésekor csupán azt vizsgáltuk, hogy az adott helyzetben igaz-e a diák megfigyelése, vagyis kivételesen tényleg egybeesnek-e a fizikából és matematikából ötöst kapók a Jedlik-díjasokkal.

Természetesen meg lehetett volna az előbbi feladatot sokkal egyszerűbben is oldani. Például kiszűrjük a Jedlik-díjat kapókat, majd a matekból és informatikából ötöst kapókat, és a két eredményt összehasonlítjuk. Szerettem volna azonban megmutatni, miként lehet a feltételeket elemi részekre bontani, és a logikai kapcsolatokat oly módon átalakítani, hogy azok a szűrőtartományban megadható alakba kerüljenek. Ezekre a lépésekre egyébként nincs mindig szükség, mert egyrésztől könnyen lehet, hogy józan paraszti ésszel az ember gyorsabban megoldja a feladatot, másrésztől nem kötelező a feltételeket elemi részekre bontani, és így összetettebb formában is megadható a szűrőtartomány.<sup>31</sup>

<sup>31</sup> Például használhatóak az ÉS ( ), VAGY ( ), NEM ( ) függvények.

Az ezelőtt bemutatott példát is megoldhattuk volna „egyszerűbben”:

	Szeretetszolgálat	Iskola	Iskola	Önkormányzat	Vállalkozók
					0 >0
	<10	>120	<150		
	>99	>120	<150		

*Az eredeti megoldás.*

fx =VAGY(B2<10;B2>99)					
	Feltétel	Iskola	Iskola	Önkormányzat	Vállalkozók
					0 >0
	IGAZ	>120	<150		

*A módosított megoldás: két sor összevonása.*

A feltételek ilyesfajta összevonásától azonban jobb, ha óvakodunk, mert túlzásokba esve megnő a hibázási lehetőség, valamint a megoldás menetét is kevésbé áttekinthetővé teszi!

#### 4.4. Ismétlődések eltávolítása

Egyedi rekordoknak nevezzük a táblázat olyan sorait, melyekkel megegyező sor nincs még egy a táblázatban. Olykor szükség lehet arra, hogy csak ezeket jelenítsük meg, mert az ismétlődések zavarnák a munkánkat.

Az ismétlődéseket kétféleképp távolíthatjuk el: **véglegesen vagy ideiglenesen**. Az előbbihez az *Adatok* fül/ *Adateszközök* csoportból kell az *Ismétlődések eltávolítása* opciót választani, ahogy ezt tettük az importálás példánál, nem is egyszer. Az ideiglenes lehetőségről eddig nem esett szó, és habár feladatpéldát nem tudok hozzá mutatni, bemutatom ezt a módszert is, hátha jól jön még valamikor.

Talán mindenkinek feltűnt már, hogy az *Irányított szűrő* ablakában van egy kis jelölőnégyzet: *Csak az egyedi rekordok megjelenítése*. Ha ezt beállítjuk, a szűrés nem csak a kritériumtáblában megadott feltételeknek eleget nem tevő sorokat szűri ki, hanem az ismétlődőket is.



Amennyiben a helyben szűrést választjuk, bejelöljük a jelölőnégyzetet, és nem adunk meg kritériumtáblát, **csak az ismétlődések tűnnek el**, és ellentétben a másik módszerrel, nem véglegesen: csak elrejtí őket az *Excel*.

#### 4.5. Kimutatás

A *Kimutatás* - más néven *Pivot-tábla* - egy annyira könnyen használható funkciója az *Excel*-nek, hogy versenyen nem is gyakran szoktak ehhez kapcsolódó feladatot adni. Pár kattintással elkészíthető ugyanis egy adatainkat összegző, interaktív táblázat, ahol két, a felhasználó által választott szempont alapján kerülnek kiértékelésre az ide tartozó adatok. Az elkészült táblázatra tetszés szerint állíthatunk be különféle szűrőket, tovább konkretizálva ezzel, mely adatoknak szeretnénk az összegét/ átlagát/ mennyiségét stb. megtekinteni.

A másik ok, amiért mellőzik az ilyen feladatokat, hogy sajnos **ez a módszer sem adatkövető**: ha megváltoznak az értékek, **frissíteni kell** a kimutatást, hogy jó eredményt mutasson. Mindezek ellenére szeretném mégis bemutatni, hiszen a mindennapi életben így is a hasznunkra lehet.

#### 21. példa: Nemes 2009-2010 1. forduló, 5/D (Locsolóverseny)

Az eszköz használatát ezen a példán fogom bemutatni, annak ellenére, hogy a feladat ezt kéri:

- „*Olyan megoldást válassz, hogy ha az adatok munkalap adataiban bármilyen változás történik, akkor annak következményei itt automatikusan jelentkezzenek!*”

Erre ez az eszköz nem alkalmas, így csak csekély pontszám járna a *képletek* helyett *kimutatással* megoldott feladatért.

Egy húsvéti locsolás információit tartalmazó táblázat adatait kell megjeleníteni táblázatos elrendezésben oly módon, hogy könnyen leolvasható legyen, hányan választották az egyes módszereket (mit mondtak illetve mivel locsoltak), mindezt a kapott ajándékok típusa szerint külön számolva.

piros tojás	verset	rigmust	kellemes ünnepeket	semmit
kölni	12	2	18	3
víz pohárból	2	2	1	0
víz kancsóból	5	1	4	0
víz vödörből	4	4	1	0
víz locsolócsőből	3	1	2	0
hímes tojás	verset	rigmust	kellemes ünnepeket	semmit
kölni	36	2	29	6
víz pohárból	5	2	7	5
víz kancsóból	11	1	4	1
víz vödörből	5	3	3	0
víz locsolócsőből	2	0	0	1
csokitojás	verset	rigmust	kellemes ünnepeket	semmit
kölni	46	7	18	8
víz pohárból	16	4	6	3
víz kancsóból	7	2	5	0
víz vödörből	6	2	3	1
víz locsolócsőből	2	0	1	1
csokinyuszi	verset	rigmust	kellemes ünnepeket	semmit
kölni	108	15	54	20
víz pohárból	23	4	13	7
víz kancsóból	14	4	15	2
víz vödörből	19	7	9	1
víz locsolócsőből	10	1	8	0

A megoldás függvénnyel: a locsolók számának összegzése a kapott ajándék, a választott locsolási forma, illetve annak függvényében, hogy mit mondtak a locsolást megelőzően.

Kimutatást használva a feladat megoldható egyetlen „táblázattal”, ahol az ajándékok szerinti szűrés éppen ezt a négy esetet adja ki. Szűkíthetjük a megjelenítendő adatokat a sorok és oszlopok kiválasztásával is, de erre most nem lesz szükség.

	A	B	C	D	E
1	Mit kapott?	csokinyuszi			
2					
3	Mennyiség / Ki locsolt?	Oszlopcímkék			
4	Sorcímkék	kellemes ünnepeket	rigmust	semmit	verset
5	kölni	54	15	20	108
6	víz kancsóból	15	4	2	14
7	víz locsolócsőből	8	1		10
8	víz pohárból	13	4	7	23
9	víz vödörből	9	7	1	19

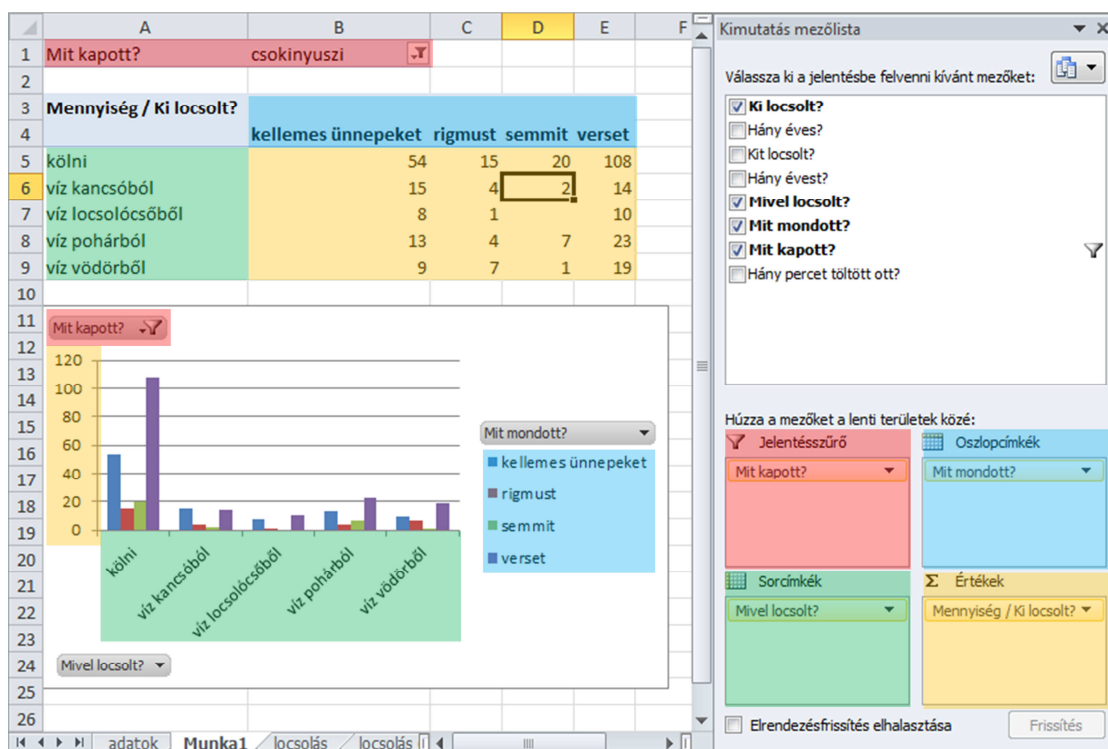
Az elkészült kimutatás: a B5-ös cellában látható, hogy hány olyan locsoló volt, aki kellemes ünnepeket kívánt, kölnivel locsolt, és csoki nyuszt kapott ajándékba.

Nézzük, hogyan készíthető el egy ilyen kimutatás!

- Álljunk abba a táblázatba, melynek adataiból a kimutatást el szeretnénk készíteni: a *locsolás* munkalap *AI:H661* tartomány valamely cellájára!
- A *Beszúrás* fül/ *Táblázatok* részében található a *Kimutatás* parancs, itt kérhetjük csupán egy kimutatás, vagy **egy kimutatás és egy hozzá tartozó diagram** elkészítését. Válasszuk az utóbbi lehetőséget!
- A megjelenő ablakban újra ellenőrizhetjük a forrást, és megadhatjuk a kimutatás helyét. Kérjük azt egy új munkalapra!

Egy új munkalapon jelenik meg a még üres kimutatás és diagramja, illetve egy olyan felület, ahol meg kell adni, hogy milyen adatot, hogyan kívánunk felvenni a kimutatásba. A *mezőlistában* az eredeti táblázat oszlopfeliratai láthatóak, ezeket lehet **vonszolásos módszerrel** a négy területre húzni:

- *Jelentésszűrő*: ide kell húznunk a *Mit kapott?* mezőt, ugyanis ez szolgál a táblázatban megjelenített értékek szűkítésére.
- *Oszlopcímkek/Adatsorok*: a mintának megfelelően, a *Mit mondott?* mező kerül ide.
- *Sorcímkek/Kategóriák*: a *Mivel locsolt?* mező helye.
- *Értékek*: az összeszámolandó adatok. Most megfelel például a *Ki locsolt?* mező. Minden esetben tartozik hozzá egy szempont, úgymint: *Összeg, Átlag, Darab* stb. Ezt az *Értékmező-beállításoknál* lehet megadni.



A kimutatás elkészítése, azonos színnel jelölve az összetartozó részeket.

➤ Adjuk meg a mezőket a kép szerint!

Miután elkészültünk, érdemes végigtanulmányozni a lehetőségeket, különböző szűréseket végezni, kipróbálni a *Kimutatáskereső* füleken található beállításokat. Az eszköz használata egyszerű, látványos és sok lehetőséget rejt magában. A forrás megváltozása esetén azonban nem automatikus az adatfrissítés, erről magunknak kell gondoskodnunk (*Kimutatáskereső/ Beállítások/ Adatok/ Frissítés*).

#### 4.6. Részösszeg

A mindennapi életben gyakran felmerülő igény, hogy elemezzük egy táblázat adatait, majd a kapott eredményeket összesítve jelenítsük meg. Például egy bolt napi számláján érdemes lehet áttekinteni, hogy az egyes árucikkekből összesen mennyi fogyott aznap, átlagosan mennyit vett belőle egy vásárló, hányszor vettek belőle stb. Ehhez készíteni kellene egy külön tétellistát (amiben minden árucikk csak egyszer szerepel), és aztán az egyes árucikkekre szűkített képletekkel megadni az eredeti táblázatban számolt értékeket. Egy nagyobb tétellista esetén ez a megoldás sok időnkbe kerülhet.

Szerencsére azonban van az *Excel*-nek erre egy beépített lehetősége, mellyel rengeteg munkától megkímélhetjük magunkat: az *Adatok* fül/ *Tagolás* csoportjában található a *Részösszeg* funkció.

## 22. példa: Nemes 2009-2010 1. forduló, 5/C (Locsolóverseny)

A *locsolás* munkalapon található a táblázat, melyben azt rögzítették, kik, kiket locsoltak meg, illetve Húsvét alkalmával, illetve ennek a részletei. Egy fiú annyiszor szerepel, ahány alkalommal meglocsolt valakit. Szeretnénk megtudni minden egyes fiúról: hány éves, hány lányt locsolt meg, a meglocsolt lányoknak mi az átlagéletkora, hány percet töltött összesen locsolással, és átlagosan mennyi ideig volt egy lánynál vendégségben.

A feladat megoldása a *Részösszeg* funkcióval a legegyszerűbb, és habár az eszközre jellemző bal oldali nézetsávot levágták a mintáról, azért vannak árulkodó jelek: a fiúk neve után szereplő kifejezések.

A	B	C	D
<i>Ki locsolt?</i>	<i>Hány éves?</i>	<i>Hány évest?</i>	<i>Hány percet töltött ott?</i>
Ács Máté Összesen			65
Ács Máté Darab		4	
Ács Máté Átlag	20	20	16
Ágoston Róbert Összesen			25
Ágoston Róbert Darab		4	
Ágoston Róbert Átlag	15	20	6
Alexa Gusztáv Összesen			30
Alexa Gusztáv Darab		4	

*Az elkészítendő összegzés, az árulkodó jelekkel.*

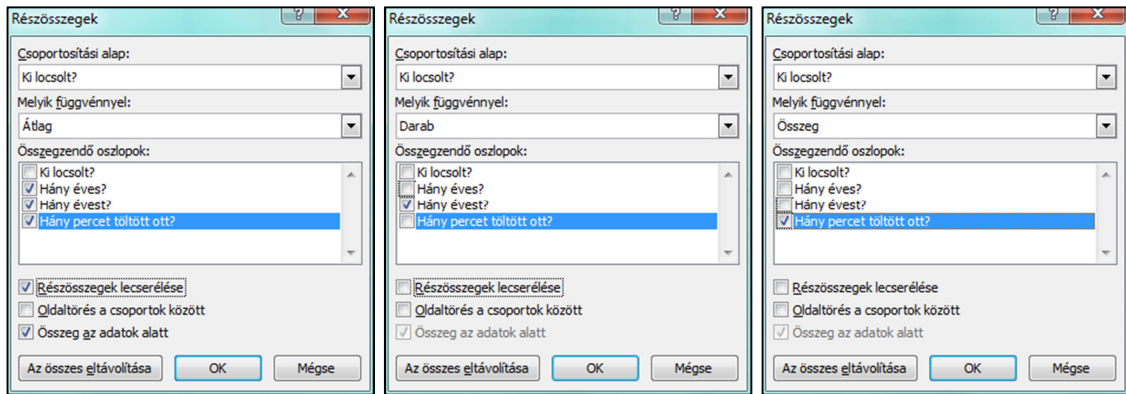
Ha már rájöttünk, mivel is van dolgunk, egy-két perc alatt összekattintgathatjuk a megoldást, szerevezve ezzel hét pontot.

- Mivel az eredeti táblázatunkat nem lenne jó átalakítani, a feladat szövege is külön munkalapra kéri a megoldást. A *Részösszeg* funkciónak viszont kell az egész táblázat, amit majd „összecsukogathat”. A legegyszerűbb az, ha lemásoljuk a *locsolás* munkalapot: kattintsunk a munkalap nevére, és a *ctrl* gomb lenyomása mellett vonszoljuk a kívánt helyre. Így a munkalapot nem áthelyeztük, hanem lemásoltuk.
- Természetesen továbbra is kell a kapcsolat az eredeti munkalappal. Töröljük az összes adatot, majd pótoljuk azokat az *adatok* munkalapról történő hivatkozással!

- Adjuk meg a kívánt *átlagolás* nevet az új munkalapnak (ami így kapcsolatban van az eredeti munkalappal)!
- Töröljük a felesleges oszlopokat, a feltételes és egyéb formázásokat is!
- Jelöljük ki az egész táblázatot, majd kattintsunk a *Részösszeg* gombra!
- **Csoportosítási alaphoz** mindegy, mit választunk, a függvény elsőnek legyen az **Átlag**.
- Szeretnénk tudni, hogy az egyes fiúk által meglocsolt lányok átlagosan hány évesek, és a locsoló átlagosan hány percet töltött egy lánynál, úgyhogy jelöljük be a megfelelő oszlopokat. Érdekel továbbá minket a locsolók életkora is, ehhez elég, ha ugyanitt bepipáljuk a *Hány éves?* oszlopát is.<sup>32</sup>
- A többi beállításon most nem kell változtatni, kattintsunk bátran az *OK*-ra!
- Fél percen belül meg kell jelenjen az eredmény, melynek többféle nézete közül is választhatunk, a bal oldali számokra kattintva.
- A feladatnak ezzel még nincs vége, szükségünk van arra is, hogy melyik fiú hány lányt locsolt meg. Jelöljük ki ismét a táblázatot (bár ha meg nem szüntetjük a kijelölést, akkor megmarad), és töltsük ki az ablakot értelemszerűen, a **Darab** függvényt választva. Amire ilyenkor nagyon vigyázni kell, hogy elfogadás előtt vegyük ki a pipát onnan, hogy *Részösszegek lecserélése*, hiszen az előbb képzett részösszegeket is meg szeretnénk tartani.
- Hasonlóan járjunk el harmadjára is, az **Összeg** függvényt használva!

---

<sup>32</sup> Ugyanazon számok átlaga egyenlő a számokkal, tehát az egyes locsolók életkorának önmagukkal vett átlaga éppen az életkoruk.



### A Részösszegek megadása.

- A feladat máris készen van, már csak ki kell választani a megfelelő nézetet, a bal oldali számok közül választva.

Részösszegek képzésénél mindig ügyeljünk azok megfelelő sorrendben történő generálására, hogy az eredmény a mintának megfelelően helyezkedjen el. Tehát mindig azzal kezdjük, ami legalul fog elhelyezkedni!

A feladat megoldható egyébként képletekkel is, de garantáltan nem a verseny ideje alatt. Érdekeséggépp azért érdemes lehet megnézni erre egy példát, az előző feladat megoldásában.

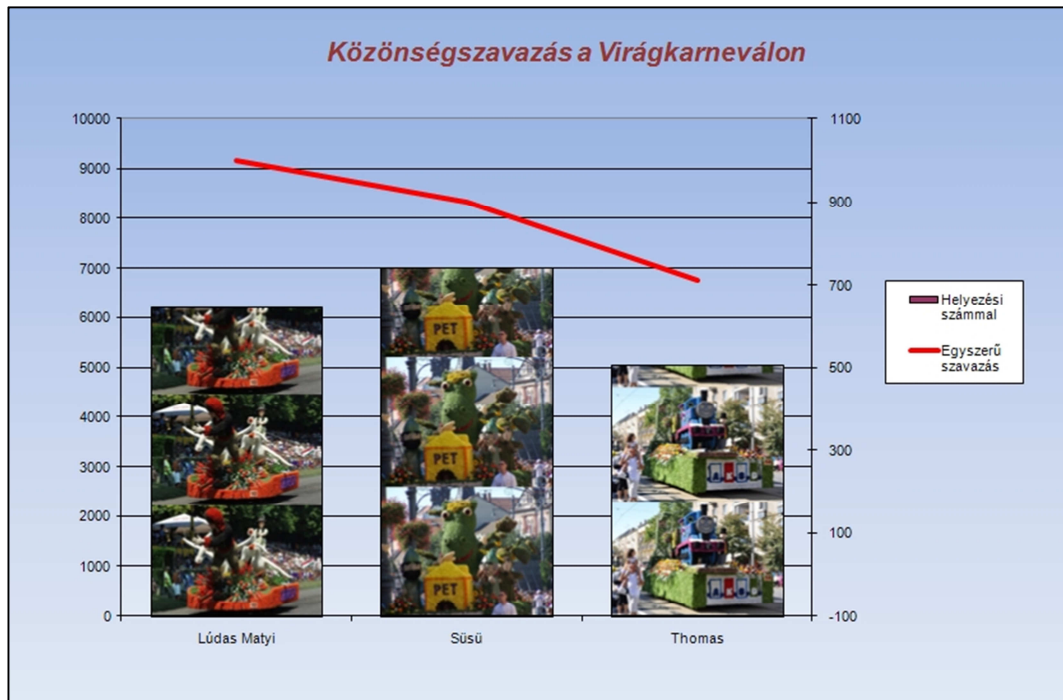
## 5. Diagram és formázása

A diagramok készítése a táblázatkezelés alapvető részét képezi, hiszen nagy segítséget nyújtanak az adatok áttekintésében. ECDL vizsgán vagy középszintű érettségien az adatok megfelelő kijelölése után már nincs nehéz dolgunk, a feladat csak pár egyszerűbb formázási beállítás szokott lenni. Versenyeken ennél jóval magasabbak az elvárások, és gyakran olyan beállításokat kérnek, melyeket a megfelelő gyakorlat nélkül csak hosszas keresgélés után talál meg az ember, főleg, ha még azt sem tudja, mit is keres igazán.

### 23. példa: OKTV 2010-2011 1. forduló, 4 (Virágkarnevál)

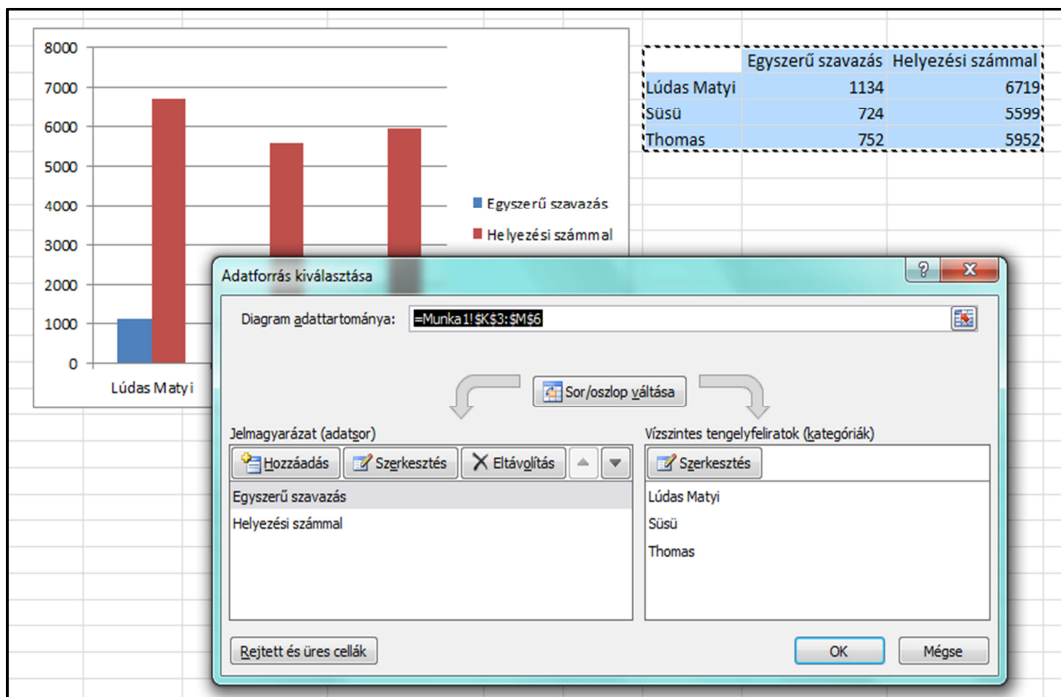
Az adatok importálását követően pár egyszerű képlet segítségével meg kell határozni, hogy a két számítási módszer szerint melyik alkotás hány pontot kapna a közönségszavazáson, majd ezeket az eredményeket egyetlen diagramon ábrázolni.<sup>33</sup>

<sup>33</sup> DARABTELI ( ) függvénnyel.



Az elkészítendő diagram

- Jelöljük ki az adatokat, és szűrjük be egy egyszerű **oszlopdigramot**. Megjegyezném, hogy az újabb *Excel* verziók már automatikusan felismerik (vagy legalábbis próbálják), hogy milyen adatot milyen formában érdemes a diagramon feltüntetni, ezért a kijelölést egyetlen téglalappal megoldhatjuk!



Az adatok kijelölése az **Adatforrás kiválasztása** ablakban ellenőrizhető, szükség esetén módosítható.



Ne legyen zavaró, hogy ha a mintán más adatok látszanak! A minta nem mindig a tényleges adatok alapján készül<sup>34</sup>, erre a feladat szövege mindig fel is hívja a figyelmet. Diagramoknál ez még zavaróbb lehet, mint a többi feladatnál...

Tehát: egy diagramon két adatsorunk is szerepel, a minta szerint ezek grafikonjának eltérő stílusúnak kell lenniük, és egy-egy saját függőleges tengellyel kell rendelkezniük.<sup>35</sup>

Állítsuk be először az *Egyszerű szavazás* adatsorának és tengelyének tulajdonságait!

- Kattintsunk az *Egyszerű szavazás* egyik oszlopára (az előbbi képen kék kitöltéssel), majd a *Tervezés* fülön, a *Típus*-nál válasszunk egyszerű **vonalt** stílust!
- Állítsuk be a **körvonal**-nak a megfelelő színt és vastagságot! Ezt megtehetjük a *Formátum* fülön, vagy a helyi menü (*jobb klikk*) *Adatsor formázása* ablakában, itt viszont a **szegélyszín** megnevezést kell keresni.
- Szükség van egy adatsorhoz tartozó, saját függőleges tengelyre is. Ezt úgy érhetjük el, hogy az adatsort az eredeti (**elsődleges**) tengely helyett a **másodlagos függőleges tengely**hez rendeljük, ami (mivel eddig nem használtuk) a beállítás hatására fog megjelenni, még pedig a jobb oldalon. A beállításra szintén a helyi menü *Adatsor formázása* ablakban van lehetőség.

A *Helyezési számmal* adatsor, illetve tengelyének beállításai következnek.

- Növeljük az **oszlopok vastagságát**! Ehhez az adatsor helyi menüjében kell az oszlopok közti térközt csökkenteni.
  - Nincs más hátra, mint **az oszlopok képpel való kitöltése**. Erre is a helyi menü *Adatsor formázása* ablakában van lehetőség, de ha az előbbieknél megfelelő kijelölés mellett hajtjuk végre a beállítást, akkor az összes oszlopra ugyanazt a képet állítjuk be. Ahhoz, hogy **oszloponként eltérő** legyen a kitöltés, még egyszer kattintani kell az egyik oszlopon, hogy csak az az egy legyen kijelölve.<sup>36</sup>
- Végezzük el a szükséges beállításokat mindhárom oszlopon, kitallózva a

---

<sup>34</sup> Habár ez sem kizárt!

<sup>35</sup> Kicsit talán zavaró, hogy adatsornak hívjuk az adatok azon halmazát, melyet a diagramban valahogyan ábrázolunk, és az adatsorhoz rendelt grafikont is.

<sup>36</sup> Ez a módszer általában is működik: első kattintással az egész adatsort jelöljük ki, másodikkal csak az adott adatpontot. Így a formázási beállításokat elvégezhetjük csak egy adatponton is.

biztosított képeket! A kép megnyújtásának elkerülése érdekében az elhelyezést állítsuk **halmozottra!**

- Adjunk az oszlopoknak megfelelő **szegélyt** is!

A mintán látható, hogy a két tengely *minimuma* és *maximuma* eltér az automatikustól. Nem olyan nehéz rájönni, hogy ennek a beállításnak valójában az a célja, hogy a két adatsor grafikonja ne takarja ki egymást.

- A mintán például a *másodlagos tengely minimuma*  $-100$ , *maximuma*  $1100$ , a *lépték*  $200$ . Ne feledjük, hogy nem kell pontosan ezekhez az értékekhez ragaszkodnunk, hiszen adatsorunk eltér a mintán lévőtől! A lényeg, hogy az *egyszerű szavazás* grafikonja elférjen az oszlopok felett.
- Válasszuk először az *elsődleges függőleges tengelyt*, és nyissuk meg a hozzá tartozó *Tengely formázása* ablakot! Ennek elég a *maximumát* nagyobbra állítani, helyet biztosítva így majd a vonaldiagramnak. Legyen például  $10000$ ! Így nem lesznek túl kicsik az oszlopok, de azért marad felettük hely.
- A *másodlagos tengely* beállítása következik.<sup>37</sup> *Minimumnak*  $-200$ -at, *maximumnak*  $1200$ -at érdemes beállítani, még ha nem is fordul soha elő, hogy a szavazatok száma negatív. Ezzel ugyanis „eltoltuk” a grafikont felfele, melynek következtében nem takarja az oszlopdigramot illetve a mintához hasonlóan, megfelelő távolságra helyezkedik el tőle.

Most már csak pár egyszerű beállításra van szükség.

- Helyezzünk el **címet** a diagram felett, a megfelelő szöveggel!<sup>38</sup> A betű formázása legyen a mintának megfelelő!
- A diagramterületnek és a rajzterületnek állítsuk be a kékes, színátmenetes **kitöltést!**
- Ahol szükséges, egérrel állítsuk az egyes részek méretét, elhelyezését, hogy minél inkább hasonlítson a mintához!

---

<sup>37</sup> Elrendezés fül/ Tengelyek/ Másodlagos függőleges tengely/ Másodlagos függőleges tengely további beállításai

<sup>38</sup> Elrendezés fül/ Címkék/Diagramcím

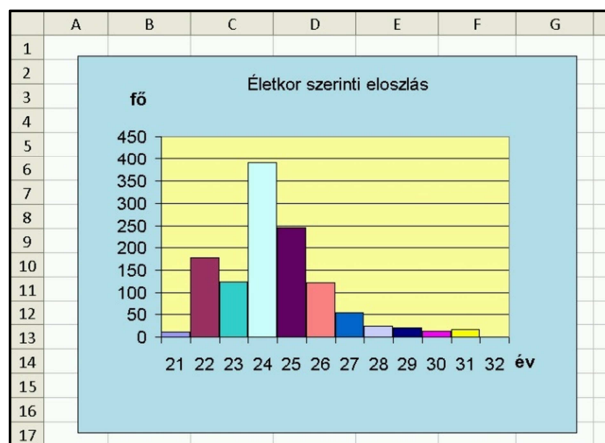
- A jelmagyarázat területe legyen fehér kitöltésű, szegélyes!

Érdeemes összevetni az általam leírt pontokat az értékelési útmutatóval. Vannak részek, amikért nem is jár pont, de mivel verseny alatt nem tudjuk, mi ér pontot, és mi nem, munkánk során igyekezzünk minél inkább a mintához hasonló diagramot készíteni! Gyakorlás során egyébként már pár feladat megoldásával, lepontozásával szerezhethetünk egy kis tapasztalatot arról, hogy általában miért jár pont, és mire nem. Az elhelyezésekkel, méretezéssel például nem érdemes sok időt tölteni, általában nem követelik meg a pontos egyezést.

#### 24. példa: OKTV 2010-2011 1. forduló, 5/G (Tisztavatás)

A feladat az, hogy diagramon ábrázoljuk a tisztek életkor szerinti eloszlását. Van hozzá minta, melyről minden szükséges adat leolvasható, a feladatszövegben csupán ezt emelték ki:

- „Az oszlopok formázásánál nem muszáj pont ezeket a színeket választanod, de ügyelj arra, hogy különböző színeket használj!”



*Az elkészítendő diagram.*

Természetesen először az adatforrást kell elkészíteni, melyhez a GYAKORISÁG( ) függvényt érdemes használni.

- Jelöljük ki a GYAKORISÁG( ) függvény eredménytömbjét, majd szűrjük be egy csoportosított oszlopdiagramot!
- Most az  $x$  tengelyen a következő kategóriák olvashatóak: 1, 2, .... Ez nekünk nem jó, cseréljük le az életkorokra: *Tervezés* fül/ *Adatok kijelölése*/ *Vízszintes*

*tengelyfelirat, Szerkesztés*, majd jelöljük ki a megfelelő tartományt, ügyelve, hogy az esetleges oszlopcímet ebbe a kijelölésbe ne vegyük bele!

Az adatsor grafikonjának formázása következik.

- Kattintsunk egyet az oszlopokra, és válasszuk a helyi menüből az *Adatsorok formázása* parancsot!
- Elsőként vegyük le az egyes oszlopok közti térközt nullára, az *Adatsor beállításai* részen!

Ha a feladatszövegben nem lenne ott az idézett mondat, akkor az egyes oszlopokat egyesével kellene színeznünk: az egész adatsor kijelölése után egyet kattintunk az egyik oszlopon (hogy csak az legyen kijelölve), majd a helyi menü *Adatpont formázása* parancsot választjuk. Így az előbbihez hasonló ablak nyílik meg, melyben a végrehajtott beállítások csak a kijelölt adatpontot módosítják: itt tudjuk megváltoztatni a kitöltés színét, minden egyes adatpont esetén. Van azonban egy gyorsabb módszer is, amennyiben nem ragaszkodunk konkrét színekhez.

- Az *Adatsorok formázása* ablakban, a *Kitöltés* résznél, tegyük pipát a következő jelölőnégyzetbe: *Adatpontonként változó színek!* Így egyszerre beállítottuk a kívánt, eltérő színezést az összes adatpontra.
- Végezzük el a még hiányzó formázási beállításokat! Ezek leírására - egyszerűségük miatt - nem térek ki bővebben.

Érdeemes észrevenni, hogy a kapott diagram nem ad túl sok információt azon kívül, hogy a legtöbb tiszt 23 éves. A többi korosztálynál szinte nem is látszik az oszlop, olyan kicsi. Versenyzőként ezzel nem kell törődnünk, hiszen a feladatot megoldottuk, de egyébként érdemes elgondolkodni azon, hogy vajon hogyan lehetne ezen javítani. Például másik diagramtípus választásával, adatfeliratok elhelyezésével lehet, hogy informatívabbá válik a diagram.

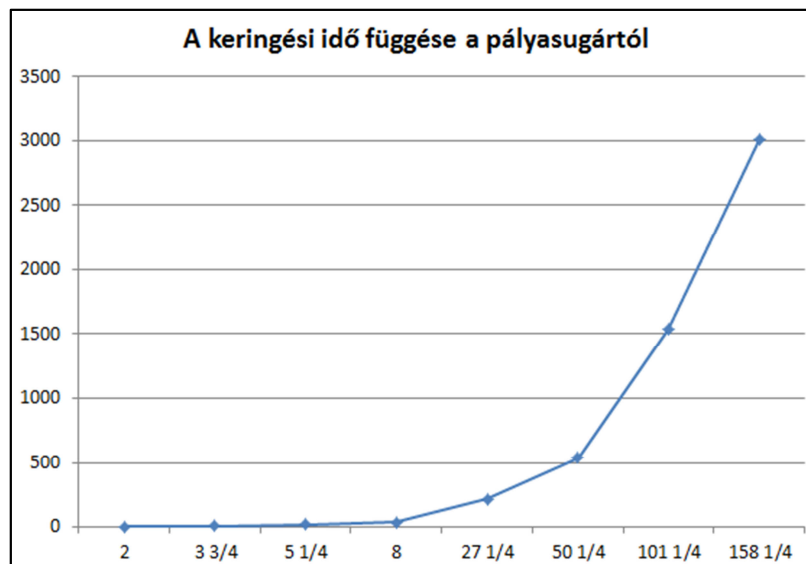
## **25. példa: OKTV 2009-2010 3. forduló, 4 (Szimuláció)**

Csakúgy, mint a virágkarneválos feladatnál, most is a diagram elkészítéséről szól az egész feladat. Mintát most nem mellékeltek, így egy kicsit nehezebb lehet a feladat értelmezése és a megoldás megtervezése, ugyanakkor jó pár beállítást illetően szabadkezet kaptunk.


A feladat egy naprendszerünk bolygóinak mozgását megjelenítő szimuláció adatainak kiemlézése. Konkrétabban: a bolygók keringési sebességét kell szemléltetni, a pályasugár függvényében.

- Elsőként pár átlag kiszámítására van szükség, az erre a célra fenntartott, összevont cellákban.

Ezután jön a neheze: ki kell választani a megfelelő diagramtípust. A feladat külön kéri, hogy a közbülső értékekről is kapjunk információt, tehát nem elég csak a kiszámított értékeket jelölni egy-egy ponttal, hanem valamilyen folytonos vonalat kell megjeleníteni. Sokan ilyen esetben gondolkodás nélkül a **vonaldiagramot** választanák, helytelenül: az ábrázolandó grafikon egy matematikai függvény lesz, így nem megengedett, hogy a tengelyeken ne arányosan legyenek az értékek feltüntetve (ekkor a közbülső értékekről is hamis információt kapnánk).



Az értékek torzulása **Vonal** stílusú diagramnál.

- Jelöljük ki a két ábrázolni kívánt oszloptartományt (B2:B41 és D2:D41), és válasszuk a **Beszúrás** fül/ **Diagramok** rész/ **Diagram létrehozása** gombot (  )! A megfelelő típuscsoport tehát nem a **Vonal** lesz, hanem a **Pont (X Y)**, válasszuk ebből például ezt: **Pont görbített vonalakkal és jelölőkkel!**

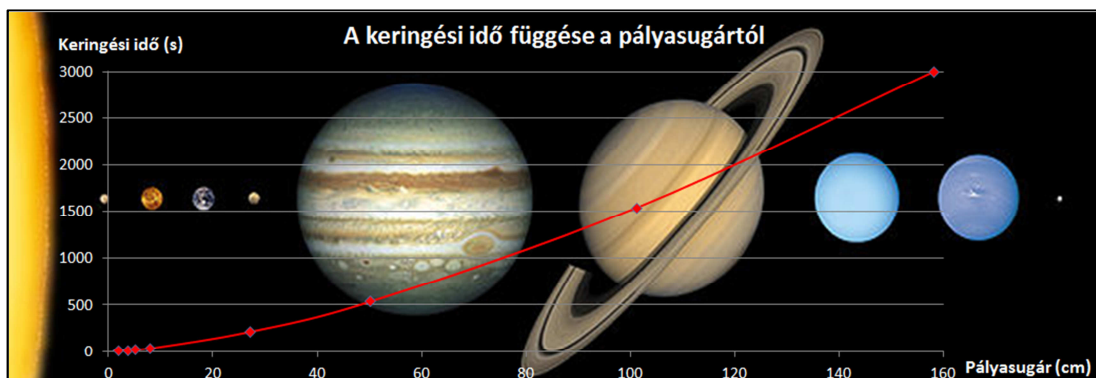
A gyakorlatlanabbak ezt követően megdöbbenve tapasztalják, hogy „nincs vonal”. Egy pár perces próbálgatással már megsejthető, hogy a problémát az **összevont cellák** okozzák. No de akkor mit tegyünk? Az értékek osztott cellákba történő másolása még csak megoldaná ezt a problémát, de pontveszteséggel járna.

- Nyissuk meg az *Adatforrás kiválasztása* ablakot! A meglévő forrást töröljük, ugyanis újra fel kell vennünk az adatokat!
- Válasszuk a *Hozzáadás* parancsot, és adjuk meg újból a megfelelő adatokat: most ne a tartományok folyamatos kijelölésével, hanem kattintgassunk egyesével az összevont cellákra, a *ctrl* gomb nyomva tartása közben!

			11,10	
			18,15	
			18,10	
Föld	5 1/4		18,05	18,11
			18,15	
			18,10	
			34,15	
Mars	8		34,05	
			34,00	34,10
			34,10	
			34,20	
			214,65	

*A megfelelő adatkijelölés összevont cellák esetén.*

- Formázzuk a diagramot a kértnek megfelelően! Érdekes lehet a tengelyek minimumát és maximumát ismét rögzítenünk, illetve a **tizedesjegyek számát** nullára állítani. Utóbbit a *Tengely formázása* ablak/ *Szám* paneljén tehetjük meg.



*Egy maximális pontszámú megoldás.*

Ezzel a feladatot meg is oldottuk.

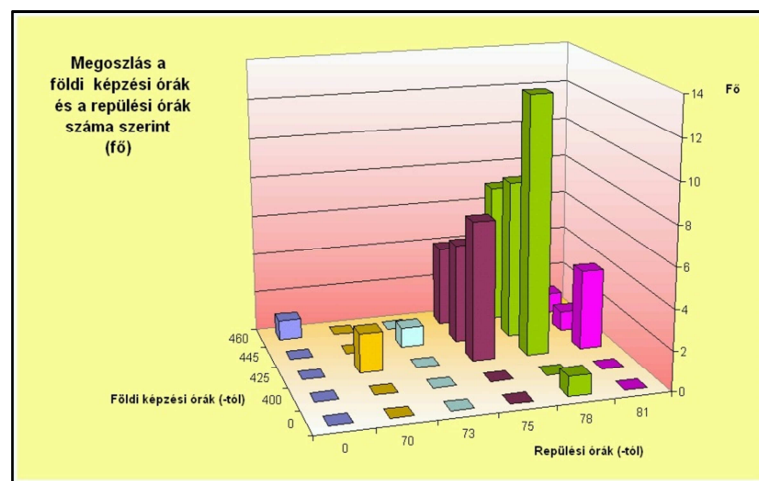
**26. példa: OKTV 2010-2011 2. forduló, 3/I (Űrhajósok kiképzése)**

A feladat egy korábban elkészített, megoszlásokat szemléltető tartomány ábrázolása.

	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX			
1	<p>Megoszlás a földi képzési órák és a repülési órák száma szerint (fő)</p>				Repülési óra								
2					0	70	73	75	78	81			
3					69	72	74	77	80				
4					között								vagy több
5													
6	Földi képzési óra	0	399	között	2	0	0	0	1	0			
7		400	424		1	0	0	0	0	1			
8		425	444		0	3	1	6	9	4			
9		445	459		0	0	4	5	6	1			
10		460			vagy több	1	0	0	4	5	1		

A diagram adatforrása.

A diagramot egy minta alapján kell elkészíteni, melyből rögtön látni, hogy melyik típust kell választani. Az eredmény olyan lesz, mintha egy háromdimenziós koordináta-rendszerben ábrázolnánk az adatokat: az  $x$  tengelyen a földi képzési órák, az  $y$  tengelyen a repülési órák, a függőleges ( $z$ ) tengelyen pedig az ezen értékekhez tartozó szám jelenik meg, vagyis, hogy hány olyan pilóta van, akik ezen óraszámokat érték el. Látni fogjuk, hogy az elkészítés elve azért eltér ettől a matematikai szemlélettől.



*Térhatású oszlopdiagram, hat adatsorral. Ennyi adatsor hagyományos oszlopdiagramon eléggé zsúfoltan helyezkedne el.*

- Az adatforrás kijelölése nélkül, szűrjünk be egy *Térhatású oszlopdiagramot!* Természetesen így semmi nem fog megjelenni a diagramterületen, de most jobban járunk, ha magunk adjuk meg az adatforrást.

Akkor sem kell kétségbe esni, ha egyébként nem tudjuk, melyik diagramra van szükségünk. A diagramtípus fölé helyezve az egérmutatót, rövid leírás olvasgató a diagramtípus sajátosságairól, és arról, hogy mikor érdemes használni. Ezek alapján, vagy esetleg egy kis próbálkozással már meg lehet találni a feladatnak megfelelőt. Persze még jobb, ha a versenyző ismeri mindegyik típust, hiszen akkor nem azzal megy el az ideje a versenyen, hogy keresgéli, mire is van szüksége.

	AQ	AR	AS	AT	AU	AV	AW	AX
<i>Repülési óra</i>								
0		73	75	78	81			
2		74	77	80				
		<i>között</i>					<i>vagy több</i>	
399		<i>között</i>	0	0	0	0	1	0
424		<i>között</i>	0	0	0	0	0	0
444		<i>között</i>	0	2	0	7	13	4
459		<i>között</i>	0	0	1	5	8	1
		<i>vagy több</i>	1	0	0	4	7	1

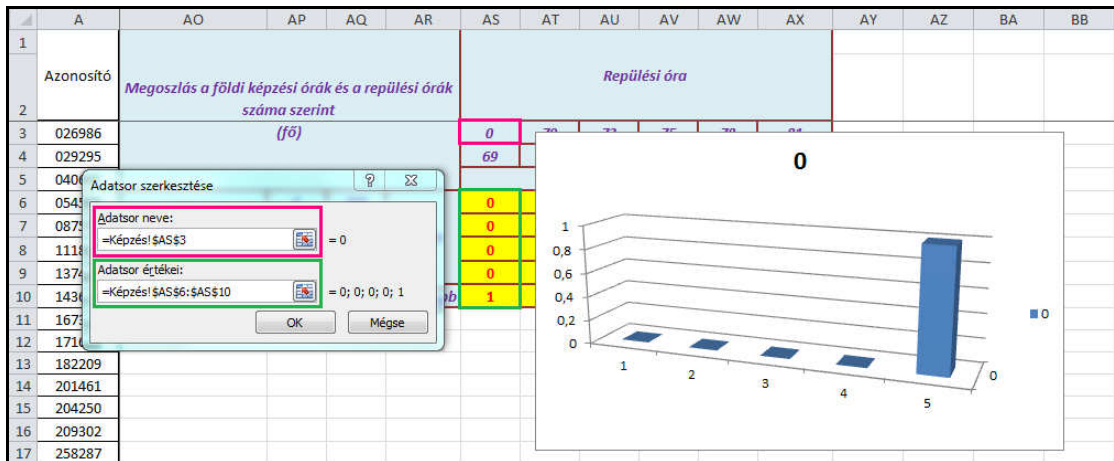
A megfelelő diagram kiválasztása és annak leírása.

- Válasszuk a diagram menüszalagján az *Adatok kijelölése* lehetőséget!

Mint már utaltam rá, az elkészítésének módja nem teljesen egyezik meg a matematikában tanultakkal. Itt nem  $(x,y)$  pontokhoz rendelünk egy  $z$  értéket, hanem különálló adatsorokat helyezünk el, csak éppen azonos függőleges tengelyhez rendelve őket. A két másik tengely a megjelenített értékek *kategóriáit* szemlélteti.

- Vegyük fel az első adatsort az *Adatforrás kiválasztása* ablak/ *Hozzáadás* gombjának segítségével: az adatsor neve az *AS3* cellából (0), az értékek pedig az eloszlásokat tartalmazó első oszlopból kerülnek ki, vagyis az *AS6:AS10* tartomány adatai (0, 0, 0, 0, 1) lesznek!





*Az adatokat oszloponként kell megadni.*

- Ennek mintájára vegyük fel a többi adatsort is, a *Hozzáadás* gomb újból és újból történő használatával!<sup>39</sup>

Az eljárás akkor is működne, ha az adatsorokat tényleg soronként adnánk meg. Például az első adatsor nevének megadhatnánk, hogy *AP6*, értékeinek pedig az *AS6:AX6* tartományt. Ekkor azonban nem egyezne meg a kapott diagram a mintával, mert a két nem-függőleges tengely felcserélt állapotban lenne.

Az *Excel* ilyenkor automatikusan különböző színekkel tölti ki az egyes adatsorokat, ezzel is szemléltetve, hogy mik tartoznak össze. Így az oszlopok színezésével a későbbiekben már nem lesz dolgunk.

A kapott diagram tengelyei, az *Excel* kifejezéseit használva a következők:<sup>40</sup>

- *elsődleges vízszintes tengely:* *a földi képzési órák*
- *Z tengely:* *a repülési órák*
- *elsődleges függőleges tengely:* *az adott kategóriákba eső pilóták száma*

A *Z* tengelyen a repülési órák száma helyett az 1,2,3,4,5 kategórianevek szerepelnek.

- Szintén az *Adatforrás kiválasztása* ablakban, adjuk meg a *vízszintes tengelyfeliratokat* is: *=Képzés!\$AP\$6:\$AP\$10!*

A nehezen már túl vagyunk, most már csak formázni kell.

<sup>39</sup> Az *adatsor* most valójában inkább „adatoszlop”. Szerencsésebb lenne talán az *adatsorozat* kifejezést használni, a félreértések elkerülése végett.

<sup>40</sup> Ezek a kifejezések a matematikában jártasak számára kicsit zavaróak lehetnek.

- Látható, hogy a mintához képest az *elsődleges vízszintes tengely* ellenkező irányba mutat. Fordítsuk meg az irányát!

Ezt kétféleképp is megtehetjük. Egyrészt, ha a tengely helyi menüjében a *Tengely formázása* opcióra kattintunk, a megjelenő ablakban van egy jelölőnégyzet erre a célra: *Kategóriák fordított sorrendben*. Másrészt a *Diagrameszközök/ Elrendezés* fülön, a *Tengelyek/ Elsődleges vízszintes tengely* menüben is beállíthatjuk, hogy *Jobbról balra mutató tengely megjelenítése*. Mint sok más helyen, a helyi menüben itt is más néven érjük el ugyanazt a beállítási lehetőséget, mint a menüszalagon.

- Adjuk meg a mintának megfelelően a tengelycímetek (*Elrendezés* fül)!
- Töröljük a jelmagyarázatot!

A színek és az elrendezés beállításán kívül már csak a diagramcím megadására van szükség. Először természetesen el kell helyezni hozzá a címkét, utána pedig megadni a nevet: ehhez be kell gépelnünk a megfelelő szöveget, vagy vágólap használatával átmásolni az *AOI* cella tartalmát. Általában azonban sokkal jobb, ha **a tengely címét is hivatkozással adjuk** meg: kevesebb munka, és az előbb említett cella tartalmának átírásakor is garantált az összhang, mert a tengelycím így automatikusan változik a cellával együtt. Ezt a módszert választva a következőket kellene tennünk:

- Állítsunk be *diagramcím* címkét a diagramnak!
- Kattintsunk a címkére, majd a szerkesztőlécre! Írjunk be egy *egyenlőség* jelet, majd kattintsunk az *AOI* cellára! Üssünk *Enter*!

Amennyiben azt is szeretnénk, hogy a cím a mintának megfelelően legyen tördelve, egyszerűbb, ha mégsem ezt a módszert választjuk, hanem kézzel írjuk be. Ekkor a kívánt eredményt a sortörések *Alt+Enter*rel történő elhelyezésével érhetjük el.

Az elrendezés kialakítása előtt helyezzük a diagramot egy új, *Megoszlás* nevű munkalapra:

- A diagram helyi menüjében válasszuk a *Diagram áthelyezése* parancsot, és adjuk meg a szükséges beállításokat!

Így sokkal könnyebb dolgozni, mellesleg a feladat egyébként is külön munkalapra kérte a diagramot.

- Alakítsuk ki az elrendezést a mintához hasonlóan! Ehhez szükség lehet a rajzterület térbeli forgatására (helyi menü/ *Rajzterület formázása*).<sup>41</sup>
- Állítsuk be a még hiányzó formázásokat!

## 6. Helyettesítő karakterek használata

A *Word Keresés és csere* funkciója révén, gondolom, mindenkinek egyértelmű, mik is azok a helyettesítő karakterek: a \* azt jelenti, hogy „tetszőleges számú, *tetszőleges karakter*”, míg a ? jelentése „*egy darab tetszőleges karakter*”.

Ezeket *Excel*-ben is használhatjuk, még hozzá nem csak a *Keresés és csere* ablakban. Sajnos azonban mutat egy kis következtelenséget, hogy pontosan hol használható, és hol nem, ezért gondoltam, hogy összeszedem pár mondatban a tapasztalataimat.

A példákat a Nemes 2010-2011-es évad 5. feladatából merítettem:

Év	Hónap, nap	Esemény	Milyen napra esett?
1848	március 15.	Az országgyűlés határozata kimondja a nemzetőrség felállításának szükségességét.	szerda
1848	március 22.	Az országgyűlés elfogadja a nemzetőrség megalakításáról szóló törvényt.	szerda
1848	április 15.	A minisztertanács Batthyány Lajosra bizza a nemzetőrség szervezését.	szombat
1848	május 7.	V. Ferdinánd a magyar kormánynak való engedelmségre szólítja fel a főhadparancsnokokat.	vasárnap
1848	május 15.	A minisztertanács elrendeli a délvidéki várak megerősítését.	hétfő
1848	május 16.	Felhívás önkéntesek (honvédek) toborzására.	kedd

*Az 1848.xlsx tartalma: a szabadságharc eseményei.*

### 6.1. A *Keresés és csere* ablakban

A helyettesítő karakterek a *Word* ugyanezen funkciója szerint *Excel*-ben is használhatóak.

<sup>41</sup> Ha nem sikerülne pont a kívánt elemre kattintanunk (például a rajzterületre), akkor használjuk az *Elrendezés* fülön az *Aktuális kijelölés* részen található legördülő listát is.

## 6.2. Függvények feltételt megadó argumentumában

A helyettesítő karakterek itt is használhatóak. Például ha szeretnénk megszámolni az „országgyűlés” kifejezést tartalmazó cellákat egy oszlopban, akkor használható a következő képlet:

```
=DARABTELI(C2:C330;"*országgyűlés*")
```

## 6.3. Cella tartalmát vizsgáló, IGAZ/HAMIS értéket visszaadó képletben

Sajnos itt már nem használhatóak a helyettesítő karakterek. A következő képlet csak akkor ad IGAZ értéket, ha a cella tartalma pontosan „\*országgyűlés\*”:

```
=C2="*országgyűlés*"
```

Hasonlóan rossz a helyzet HA( ) esetén is:

```
=HA(C2="*országgyűlés*";"talált";"nincs")
```

## 6.4. Feltételes formázásnál

Az előbbi megállapítás után talán mindenki sejtette, hogy helyettesítő karakterek feltételes formázásnál sem használhatóak. Ha az „országgyűlés” kifejezést tartalmazó cellákat szeretnénk például sárga háttérre színeznünk, a következő kifejezés nem fog működni:

```
=&C2="*országgyűlés*"
```

Ehelyett szövegkezelő függvényeket kell használnunk a feladat megoldásához:

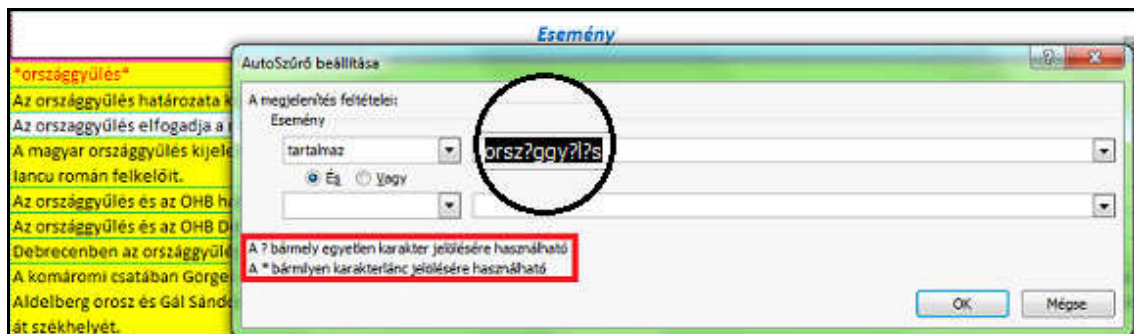
```
=SZÖVEG.KERES("országgyűlés";&C2)<HOSSZ(&C2)
```

## 6.5. Szűrők feltételeként

Szűrők esetén szerencsénk van: nem kell szövegkezeléssel foglalkoznunk, ugyanis nyugodtan használhatjuk a helyettesítő karaktereket.

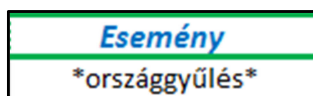
*Autoszűrő* esetén, ennél a konkrét példánál éppenséggel nincs sok értelme használni őket, hiszen a *Szövegszűrők* közt megtalálható a *Tartalmazza* opció is, mellyel pont ez a feladat oldható meg. Azonban ha például nem lennénk biztosak abban, hogy mindegyik „országgyűlés” szó ékezzel lett írva, akkor a következő szűrés eredménye egy olyan

lista lenne, melynek megfelelő oszlopában az „országgyűlés” szó ékezethiányos változatai szereplnének:



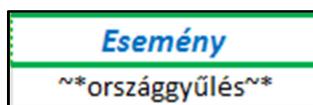
*Helyettesítő karakterek használata AutoSzűrőben.*

Írányított szűrő esetén így néz ki a kritériumtábla, amennyiben az „országgyűlés” kifejezéseket tartalmazó rekordokat akarjuk kilistázni:



*Helyettesítő karakterek használata irányított szűrő szűrőtartományában.*

Ehhez a témakörhöz kapcsolódik a ~ karakter használata is: ha tényleg \* illetve ? karaktereket akarunk használni, akkor ezeket a ~ karakterrel óvhatjuk meg attól, hogy helyettesítő karakterként értelmezze őket az Excel. A **pontosan** „\*országgyűlés\*” kifejezéssel megegyező eseményű sorok kiszűréséhez a következő feltételt kell megadni:

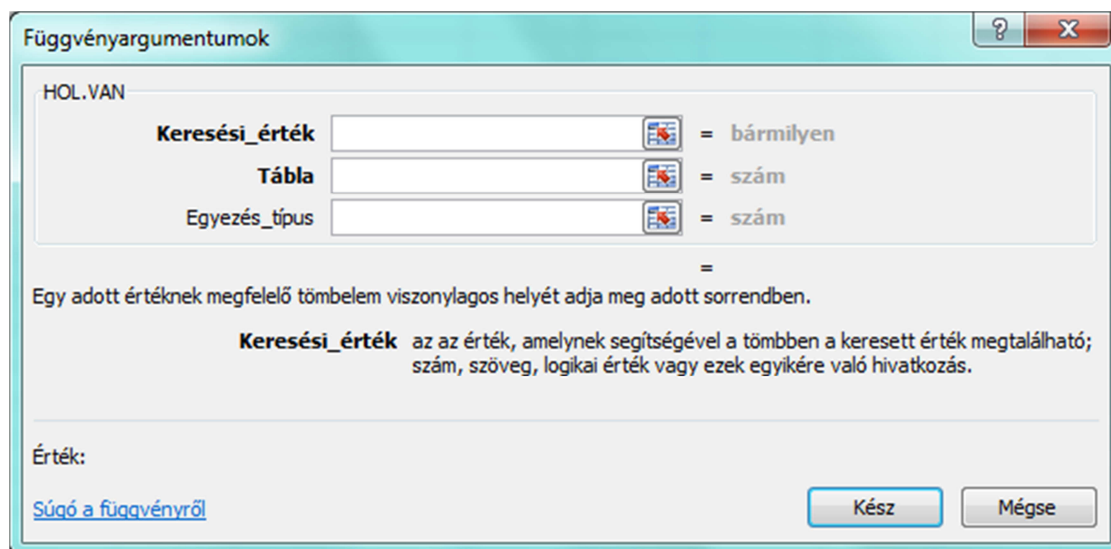


*A ~ karakter használata irányított szűrő szűrőtartományában.*

## 7. Függvények

Eredetileg nem állt szándékomban függvényekről külön is írni, hiszen az Excel nagyon segítőkész e téren. A *Függvény beszúrása* ablakban például lehetőség van a függvények közti tematikus keresésre, amire elsősorban akkor jön jól, ha van sejtésünk arról, hogy mire van szükségünk, de nem tudjuk, pontosan melyik függvénnyel érhető az el. A másik eset, amikor emlékszünk a függvény nevére, de a paramétereire nem.

Ekkor első körben segítségünkre van az argumentumlista, ami a szerkesztőléc alatt jelenik meg a függvény nevének megadását követően, másrészt - ha ez nem lenne elég - elindíthatjuk a függvényvarázslót is. Ha pedig még csak most ismerkedünk egy függvénnyel, érdemes megnézni a súgót, mert ott sokkal részletesebb a leírás, és gyakran példák is segítik a megértést.



*A függvényvarázsló, a HOL.VAN() függvény szerkesztésekor. A harmadik argumentum azért nem félkövér, mert elhagyható.<sup>42</sup>*

Nem látom tehát szükségét annak, hogy külön is bemutassam a versenyeken legtöbbet használt függvényeket. A megoldásfájlokban meg lehet nézni, milyen függvényekkel oldható meg az adott feladat, és akinek szüksége van rá, utánaolvashat a súgóban, vagy az *Office* honlapján.

Az ok, amiért mégis készült egy ilyen fejezet, az, hogy érdekesnek találtam pár módszer és észrevétel ismertetését. Talán ezekkel is elősegítem a megoldások minőségének jobbítását, vagy a ráfordított idő csökkentését. Az általam ismert típushibákat pedig konkrét példákon keresztül mutatom be.

## 7.1. Összetett képletek szerkesztése

Középfokú érettségien még csak 2-3 képlet egymásba ágyazása az elvárt, versenyeken ennél jóval magasabb az igény. Ha az ember nem elég trükkösen fog hozzá a feladathoz,

---

<sup>42</sup> Bölcsebb dolognak tartom, ha a harmadik argumentumot is mindig megadjuk, mert így tudatosul bennünk, hogy éppen milyen eset áll fenn. Gyakori hiba ugyanis, hogy elfelejtjük, mi az alapértelmezett érték, és rosszul használva a függvényt, hibás eredményt kapunk.

könnyen elveszhet a különböző függvények argumentumainak sokaságában, vagy belekeveredhet a zárójelezésbe.

**27. példa: Egy saját példa: Cukrászda**

A függvények egymásba ágyazásának általam használt technikáját ezen a nagyon egyszerű példán mutatom be, hogy a feladat nehézsége ne terelje el a figyelmet. Egy ilyen szintű feladat akár középfokú érettségien is előfordulhat, de mivel a technika alkalmazható ennél sokkalta összetettebb képletek esetén is, nem érzem szükségét egyből nehéz példával kezdeni.

Adott egy cukrászda megrendelhető tortáinak listája. Az egyszerűség kedvéért, most más oszloppal nem is foglalkozunk, mint a torták nevét és árát tartalmazókkal.

- Adjuk meg a legdrágább torta nevét!<sup>43</sup>

A módszer a következő: ahelyett, hogy nekiesnénk az egész feladat megoldásának, részeire szedjük azt, és azokat sorban oldjuk meg, külön segédcellákban, kihasználva mindig egy korábbi lépés eredményét. Ha végeztünk, a segédcellák eredményeit egybeépítjük, hogy aztán ki is törölhessük őket.

E12		fx =INDEX(D3:D10;HOL.VAN(MAX(E3:E10);E3:E10;0))				
	A	B	C	D	E	F
1						
2				<b>Név</b>	<b>Ár</b>	
3				Csokitorta	4 800 Ft	
4				Dobos torta	5 300 Ft	
5				Feketeerdő	7 200 Ft	
6				Gyümölcstorta	6 000 Ft	
7				Kávékrémes	4 900 Ft	
8				Lúdláb	6 400 Ft	
9				Oroszkrém	7 000 Ft	
10	1. lépés (MAX):	7 200 Ft		Puncs torta	5 200 Ft	
11	2. lépés (HOL.VAN):	3				
12	3. lépés (INDEX):	Feketeerdő		A legdrágább torta:	Feketeerdő	
13						

E12-ben a kész képlet, B oszlopban a részfeladatok megoldása.

<sup>43</sup> Az ilyen feladatokhoz illik mindig hozzáfűzni, hogy mi történjen, amennyiben az árak közt több maximum is van. Most például meglegszünk azzal, ha az elsőt adjuk meg közülük.

- Számítsuk ki *B10*-ben, hogy mennyibe kerül a legdrágább torta!

=MAX(E3:E10)

- Határozzuk meg *B11*-ben, hogy az előbbi eredmény az oszlop hányadik eleme!

=HOL.VAN(B10;E3:E10;0)

- *B12*-ben adjuk vissza a torták közül az előbbi cella értékével megegyező sorszámút!

=INDEX(D3:D10;B11)

*E12*-ben készítjük el az előbbiekből összeépítéséből adódó, segédcellák nélkül is működő képletet.

- Álljunk *B12*-re, és másoljuk a szerkesztőléc képletét a vágólapra!
- Illesszük be *E12*-be! *B12*-re már nincs szükség (sőt eredetileg se lett volna, csupán az átláthatóság kedvéért különítettem el ezt a két cellát).
- Most másoljuk *B11* képletét a vágólapra!
- Kattintsunk *E12*-re, és a szerkesztőléccen jelöljük ki a képletben a *B11*-re történő hivatkozást! Illesszük a helyére a vágólapra másolt képletet! Így *B11*-re sincs már szükség.
- Az előbb leírtakhoz hasonlóan, illesszük az *E12* képletében található, *B10*-re történő hivatkozás helyére az ottani képletet!

Ezzel a három függvényt összeépítettük, a segédcellák törölhetőek.

## 28. példa: Nemes 2009-2010 2. forduló, 4 (Damimpex)

Még egy példát mutatnék a módszer használatára. A képlet sokkal összetettebb lesz, ami nehezíti a feladatot, ugyanakkor a kiszámított részeredmények kisebb mértékben épülnek egymásra.

Egy cég telefonhívásainak adatai alapján kell egy cellában a következő szöveget megjeleníteni:

- „Az ügyintézők beszélgetései alkalmanként átlagosan (ennyi) Ft-ba, a közép- és felsővezetők beszélgetései pedig átlagosan (ennyi) Ft-ba kerülnek. Ennek



*alapján megállapíthatjuk, hogy a vezetők átlagosan (drágábban/olcsóbban) beszélnek, mint a beosztottaik.”<sup>44</sup>*

- Számoljuk ki az ügyintézők hívásainak átlagos díját például A72-ben! Az ÁTLAGHA( ) függvény most nem lesz segítségünkre, mert a táblázatban szereplő hívások különböző súllyal számolandók. Az osztást tehát magunknak kell elvégezni, hogy ne csak a táblázat megfelelő sorainak számával, hanem a hívások számának összegével osszuk:

```
=INT(
    SZUMHA($B$2:$B$63;"ügyintéző";$G$2:$G$63)/
    SZUMHA($B$2:$B$63;"ügyintéző";$E$2:$E$63)
)
```

- A73-ba, az előző képlet mintájára, számoljuk ki a nem ügyintézők átlagos hívásdíját!

```
=INT(
    SZUM(
        SZUMHA($B$2:$B$63;"felsővezető";$G$2:$G$63);
        SZUMHA($B$2:$B$63;"középvezető";$G$2:$G$63)
    )
    /
    SZUM(
        SZUMHA($B$2:$B$63;"felsővezető";$E$2:$E$63);
        SZUMHA($B$2:$B$63;"középvezető";$E$2:$E$63)
    )
)
```

A SZUMHA()-kat összeadó SZUM() helyett használhatunk összeadásjelet is, de akkor zárójellezni kell, a műveletek precedenciája miatt.

- Készítsünk egy olyan képletet, ami az előző két eredmény viszonya alapján a „drágábban”, illetve „olcsóbban” kifejezések egyikét jeleníti meg!

```
=HA(A73>A72;"drágábban";"olcsóbban")
```

- A feladat által megjelölt egyesített cellában a képlet az előbb készített cellákra fog hivatkozni:

---

<sup>44</sup> A feladat szövegében ez is szerepel: „Mivel nagyon kevésé valószínű, kivételesen tekints el attól az esettől, hogy a két átlag meg is egyezhetne.”.

```
=ÖSSZEFŰZ(
  "Az ügyintézők beszélgetései alkalmanként átlagosan ";
A72;
  " Ft-ba, a közép- és felsővezetők beszélgetései pedig
  átlagosan ";
A73;
  " Ft-ba kerülnek. Ennek alapján megállapíthatjuk, hogy
  a vezetők átlagosan ";
A74;
  " beszélnek, mint a beosztottaik."
)
```

Az ÖSSZEFŰZ() képlet helyett & jeleket is lehet használni.

Az eredmény így már helyes, de, hogy szebb legyen a megoldás, tüntessük el a segédcellákat! Ehhez megint a hivatkozásokat kell cserélni, a megfelelő képletekre. Vigyázzunk, a harmadik segédcella is hivatkozik A72-re és A73-ra!

- A vágólap segítségével végezzük el a cseréket, majd töröljük a feleslegessé vált cellákat!

A kapott eredményt inkább el sem helyeztem ebben a leírásban, olyan hosszú és átláthatatlan lett. A feladatkiírás most egyébként kijelölt egy külön tartományt is, ahol a részeredményeket kiszámíthatjuk, így nem is lett volna muszáj a képletet összeépíteni. Számtalan esetben azonban a segédcellák használata pontlevonással jár.

Elgondolkodtató, hogy a mindennapokban hogyan járunk jobban. Ha nincsenek segédcelláink, nyilván sokkal szebb lesz a munkafüzetünk, a képlet viszont sokkal kevésbé áttekinthető. Természetesen mindkét eljárás jó eredményhez vezet, versenyeken viszont sok esetben csak a segédcellák nélküli megoldásra adnak maximális pontot. Ugyanakkor, mivel a módszer visszafele nem működik, komoly problémát jelenthet, ha egy korábban készített, több soros képletet változtatni kell valamit: valószínűleg még a képlet készítője se fog rajta először kiigazodni. A mindennapokban megoldás lehet ilyenkor, hogy a segédcellákat „eltüntetjük” egy alkalmas cellaformázással, betűszínnel stb. A versenyzőket viszont óva inteném ettől: nem valószínű, hogy túljárnak a feladatot javító pedagógus eszén, így a segédcellák használata miatt ez esetben is jár a pontlevonás.

## 7.2. Adatbázisfüggvények

Nem áll szándékomban külön is bemutatni az adatbázisfüggvényeket, aki nem elég jártas e témában, nézzon utána a súgóban. Szeretnék azonban egy összefoglaló táblázatot adni, ami segít áttekinteni pár függvény rendeltetését és az azok közti összefüggéseket.

„alapfüggvény”	egy feltétellel	több feltétellel	adatbázisfüggvénnyel
SZUM ( )	SZUMHA ( )	SZUMHATÖBB ( )	AB.SZUM ( )
ÁTLAG ( )	ÁTLAGHA ( )	ÁTLAGHATÖBB ( )	AB.ÁTLAG ( )
DARAB ( )	DARABTELI ( ) <sup>45</sup>	DARABHATÖBB ( )	AB.DARAB ( )
DARAB2 ( )	DARABTELI ( )	DARABHATÖBB ( )	AB.DARAB2 ( )
MIN ( )	-	-	AB.MIN ( )
MAX ( )	-	-	AB.MAX ( ) <sup>46</sup>

Az utolsó két oszlop függvényei lényegében ugyanazt a célt szolgálják, az adatbázisfüggvényeknél azonban szükség van egy feltételeket megszabó segédtáblára is. Ennek elkészítése teljesen megegyezik az irányított szűrés szűrőtartományának létrehozásával, szokás is mindkettőt *kritériumtáblának* nevezni.

Egy konkrét példát szeretnék még megmutatni, amire majd a későbbiekben fogok visszautalni.

### 29. példa: Nemes 2009-2010 3. forduló, 5/D (Holdak)

A feladattal már többször is találkozhattunk: egy holdak felfedezésének adatait tartalmazó táblázattal dolgozunk. A feladat most az, hogy egy felhasználó által megadott névről megadjuk a következőket:

- „a táblázatban nem szerepel ilyen felfedező, vagy

<sup>45</sup> Furcsa, hogy a függvény neve ez, és nem „darabha”.

<sup>46</sup> Népszerű adatbázis-kezelő függvény az AB.MEZO ( ) is, funkciója az INDEX ( )-HOL.VAN ( ) párosításéhoz hasonló.

- *a megadott név szerepel a táblázatban, de csak egyszer találkozunk a névével, vagy ha többször is, csak egyazon naptári éven belül; ilyenkor a válasz tüntesse fel a felfedezett holdak számát és az adott évet, vagy*
- *a megadott név több évszámhoz kapcsolódóan is szerepel a táblázatban; ilyenkor a válasz tüntesse fel az általa felfedezett holdak számát és a felfedezéseikhez kapcsolódó legkorábbi és legkésőbbi évet.”*

Az első ponttal nincs sok dolgunk, hiszen egy egyszerű DARABTELI ( ) képlettel eldönthető, ez az eset áll-e fenn. Ezért ennek megoldását most nem mutatom be.

Ahhoz, hogy eldöntsük, hogy amennyiben van ilyen felfedező, a második, vagy harmadik állítás az igaz rá, azt kell megnézni, hogy a névhez tartozó évszámok közül a legkisebb és a legnagyobb megegyezik-e. A feladat kulcsa tehát ennek a két évszámnak a meghatározása, a feladat többi részével itt sem foglalkozom (a megoldásban úgymint megtalálható a teljes képlet).

Ezek után már triviális, hogy az AB.MIN ( ) és AB.MAX ( ) függvények használatára lesz szükség. Mivel a két függvény argumentumlistája megegyezik, a következő leírás mindkettőre értendő.

- Adjuk meg a függvény adatbázis argumentumát! Ez lehet az egész táblázat, vagy annak csak azon oszlopai, amik a feladat szempontjából érintettek: a nevek és az évszámok oszlopa.

D5:E99

- Adjuk meg a függvény mező argumentumát! Ez azt jelenti, hogy melyik az az oszlop az előbb megadott tartományból, aminek a minimumát, illetve maximumát kell megkeresni. Az oszlopot megadhatjuk fejlécével is, de ennél egyszerűbb, ha az előbb megadott tartománybeli sorszámmal adjuk meg:

1

- A kritérium argumentumban valójában egy táblázatot kell megadni, ami a feltételeket tartalmazza. Elkészítése a szűrőtartományéval megegyező.

felfedező(k) neve
Kuiper

*A feltételt biztosító kritériumtábla most ilyen egyszerű.*

A megoldás az *adatok (3)* munkalapon található meg.

Az adatbázisfüggvények hátránya tehát, hogy használatuk „segédcellákat” igényel, még ha ez nem is a szokásos módon értendő.<sup>47</sup> Ugyanebben a rejlik az előnyük is: a kritériumtábla révén a képlet általában sokkal átláthatóbb. Versenyeken azonban nem ez szokott a fő szempont lenni, sőt a segédcellák mellőzése a legtöbbször elvárt. Szerencsére az adatbázisfüggvények általában nélkülözhetőek, ez részben leolvasható a táblázatból is, és mutatok pár konkrét példát lecserélésükre a *Tömbképletek* című alfejezetben is.

Ajánlom további gyakorlásra az OKTV 2010-2011 1. fordulóját, 5/I (Tisztavetés) feladatot, melynek adatbázisos megoldása a *Válaszok* munkalapon található.

### 7.3. Tömbképletek

Tömb alatt általában egy téglalap alakú tartományt értünk. A tömbképletek – más néven *CSE típusú képletek* – olyan képletek, melyeknek minden „bemenete” egy-egy tömb. A „kimenet” lehet csak egy cella vagy akár egy újabb tömb is, de **a képlet jóváhagyásához mindkét esetben a *Ctrl+Shift+Enter* gombok egyszerre történő lenyomására van szükség.** Egy tömbképletet tartalmazó cellára kattintva a képlet minden esetben {}-k közt jelenik meg a szerkesztőlécen, ez segíti gyors felismerésüket.<sup>48</sup>

Használatukkal egyszerűsödhetnek bonyolult képleteink, és a segédcellák használata is kiküszöbölhetővé válik. Előnye továbbá, hogy a képlet átírására csak egyszer van szükség, míg egy hagyományos képlet megváltozása esetén újra kell másolni azt.

Nézzünk először pár egyszerű, hagyományos módszerrel is megoldható feladatot!

Adott egy 20 soros táblázat fejléccel, A és B oszlopában tetszőleges számokkal: az *A1:B21* tartomány.

---

<sup>47</sup> Nem ritka az ehhez hasonló feladatszöveg sem: „*Kritériumtartomány felvétele megengedett, de ha egyéb segédcellákat is használasz, számíts rá, hogy a megoldásod így kevesebb pontot ér!*”.

<sup>48</sup> A programozni tanulóknak számára a tömbképletek megértése valószínűleg sokkal könnyebben megy. Érdekesnek találom a feladatok vektorokkal/(mátrixokkal) történő „leprogramozását” a tömbképletek tanításakor, hogy a diákok meglássák az analógiát, és a már meglévő tudásukra építve, könnyebben elsajátítsák az anyagot.

- Szorozzuk össze az *A* és *B* oszlopok értékeit *C*-be!

A feladatot eddig egy alkalmas képlet előállításával, majd annak másolásával oldottuk meg:  $C2=A2*B2$ .

➤ Jelöljük ki a *C2:C21* tartományt!

➤ Írjuk a szerkesztőlécra:

**=A2:A21\*B2:B21!**

➤ Nyomjunk *Ctrl+Shift+Enter* billentyűkombinációt!

A feladat tehát tömbképlettel is könnyedén megoldható, csak arra kell vigyázni, hogy a bemeneti tömbök és a kijelölt tartomány **mind egyforma nagyságúak legyenek**, hiszen **tömbkimenetű** képletről van szó.

- Számoljuk össze az *A* oszlop 10-nél nagyobb elemeit!

A kimenet most nyilván nem tömb lesz, hanem **egyetlen cella**. Ezelőtt a feladatot így oldottuk meg: =DARABTELI(A2:A21;">10"), vagy esetleg AB.DARAB() függvénnyel.

➤ Adjuk meg a kiválasztott cellának a következő képletet:

**=SZUM(HA(A2:A21>10;1;0))**

➤ Nyomjunk *Ctrl+Shift+Enter* billentyűkombinációt!

A következő történik: a HA() függvény az *A* oszlop minden egyes eleméről eldönti, hogy igaz-e rá a feltétel. Így minden elemhez hozzárendeljük az 1 és 0 számok valamelyikét, a feltétel igazságtartalmától függően. Keletkezik tehát egy 20 nagyságú tömb, ennek elemeit a SZUM() függvény összegzi.

- Adjuk össze az *A* oszlop hárommal osztható elemeit!

Ha nem ismerjük a tömbképleteket, a feladat többnyire csak adatbázisfüggvénnyel oldható meg: =AB.SZUM(A1:A21;1;K6:K7), ahol *K6:K7* a =MARADÉK(A2;3)=0 feltételt tartalmazó kritériumtábla.

➤ Adjuk meg a következő képletet, majd fogadjuk el *Ctrl+Shift+Enter*-rel!

**=SZUM(HA(MARADÉK(A2:A21;3)=0;A2:A21;0))**

A módszer az előbbihez hasonló, csak most bonyolultabb a képlet, és mivel nem a hárommal osztható számok számára, hanem azok összegére vagyunk kíváncsiak, a megfelelő helyeken nem az 1-et, hanem magát az értéket adjuk meg a képzeletbeli tömbünknek. Mivel a nem megfelelő helyeken a HA ( ) függvény 0-t ad eredményül, a generált tömb elemeinek összege pont a hárommal osztható számok összege lesz.

Akinek nehezen megy a leírtak elképzelése, nyugodtan készítse el előbb ezeket a „képzeletbeli” tömböket, hogy lássa, miknek az összegét veszi a SZUM( ) függvény. Például a =HA(MARADÉK(A2:A21;3)=0;A2:A21;0) képlet, D1:D21-en állva, Ctrl+Shift+Enter-rel elfogadva egy ilyen tömböt generál.

### 30. példa: Nemes 2009-2010 3. forduló, 5/D (Holdak)

A feladatot az adatbázisfüggvényeknél ismertettem, ott az AB.MIN( ) és AB.MAX( ) függvényekkel oldottuk meg.<sup>49</sup> Most ezeket cseréljük tömbképletre, hogy ne kelljen segédcellákat elhelyezni.

A módszer talán már ismerős: készítünk egy tömböt, ami a megfelelő helyeken (tehát abban a sorban, ahol a megadott név szerepel mint felfedező) az ahhoz tartozó évszámot tartalmazza, a többi helyen pedig 0-t. Ennek a tömbnek kell a maximumát és a minimumát vennünk, hogy megkapjuk a kívánt két évszámot. Ehhez természetesen most nem SZUM( ), hanem MAX( ), illetve MIN( ) függvényeket fogunk használni.

➤ Egyelőre egy tetszőleges cellába állva, készítsük el a következő képletet:

=MAX(HA(E7:E99=C106;D7:D99;0))!

Ezzel megadtuk a felfedezőhöz tartozó évszámok legnagyobbikát, feltéve, hogy van ilyen nevű felfedező. Ez elsőre egyszerűnek tűnhet, de azért át kell gondolnunk, miért is adhattunk meg a HA( ) függvény harmadik argumentumában 0-t: mert ennél úgyis minden elem nagyobb. Ha nem tudnánk biztosan, hogy csak pozitív számok vannak a felfedezés événél megadva, már nem tehetnénk ezt meg.

Kérdés, hogy akkor mennyit adjunk meg a MIN( ) függvény esetén? Nyilván, a 0 nem lesz jó, hiszen akkor minden esetben az lenne a minimum. Egy olyan nagy számra van most szükségünk, aminél biztosan kisebb minden évszám: legyen például 10000!

---

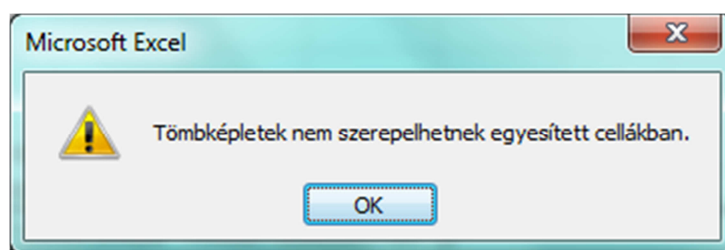
<sup>49</sup> MINHA( ) illetve MAXHA( ) függvények híján.

Kihasználva, hogy a HA ( ) függvény harmadik argumentuma nem kötelező, azt is megtehetjük, hogy egyszerűen elhagyjuk azt. Ilyenkor a képzeletbeli tömbünk azon soraiba, ahol nem teljesült a feltétel, HAMIS értékek kerülnek, és így nem zavarják a minimum- illetve maximumszámítást.

➤ Készítsük el a következő képletet egy cellába:

=MIN(HA(E7:E99=C106;D7:D99;10000))!

Ezzel meghatároztuk a két évszámot. Ezek összeépítése a teljes választ szolgáltató képlettel nem nehéz, de amikor az adatbázisfüggvényeket lecseréltük a tömbképletekre, és megpróbáljuk azt a szokásos módon érvényesíteni, a következő hibüzenetet kapjuk:



*A feladat egyesített cellába kéri az eredményt, de úgy tűnik, ilyet nem lehet.*

A megoldás a következő: előbb elkészítjük a teljes képletet az összevonandó cellák egyikébe (hiszen ezt engedi), és **csak ezt követően vonjuk össze a cellákat.**

### 31. példa: OKTV 2010-2011 1. forduló, 5/I (Tisztavetés)

Adott egy tisztek végzési adatait tartalmazó táblázat, többek közt a következő oszlopokkal: *Nem, Avatás éve, Kora az avatáskor, Szak, Tanulmányi átlaga.*

A feladat, hogy megadjuk az olyan tisztek tanulmányi eredményének átlagát, akikre igazak az alábbi állítások:

- a) *hölgy*
- b) *2007-ben végezett*
- c) *avatásakor 24 évnél idősebb volt*
- d) *valamelyik mérnökszakra járt*

A feladat könnyedén megoldható adatbázisfüggvénnyel, a következő kritériumtáblázattal:



Nem	Avatás éve	Kora az avatáskor	Szak
=n	=2007	>24	=*mérnök*

*AB.ÁTLAG() függvény esetén így néz ki a kritériumtáblázat.*

Ahhoz, hogy ne legyen segédcella, tömbképlet használatára van szükség. Ez már elég összetett lesz, és a feladat ki sem köti, hogy nem szabad segédcellákat használni, mégis megmutatnám a megoldást. A módszer egyébként az eddig ismertettek továbbfejlesztése.

- Először csak készítsünk minden feltételhez egy-egy tömböt, aminek egy sora akkor 1, ha arra a sorra igaz az állítás, különben meg legyen 0!

a) *hölgy:*

=HA(Nevék!B2:B933="n";1;0)

b) *2007-ben végzett:*

=HA(Nevék!D2:D933=2007;1;0)

c) *avatásakor 24 évnél idősebb volt:*

=HA(Nevék!G2:G933>24;1;0)

d) *valamelyik mérnökszakra járt:*

=HA(SZÁM(SZÖVEG.KERES("mérnök";Nevék!I2:I933));1;0)

Ne feledjük, HA() függvény esetén nem lehet helyettesítő karaktereket használni, ezért kellett a SZÁM()-SZÖVEG.KERES() egymásba ágyazás. Így a tömb egy eleme akkor lesz 1, ha az annyiadik sorban lévő szak megnevezése tartalmazza a „mérnök” kifejezést.

Azt akarjuk, hogy az összes feltétel teljesüljön. Ez pontosan akkor következik be egy sorra, ha az előbbi négy tömb mindegyikének 1 áll ebben a sorában. Másképpen: ahol a négy tömb szorzata egy, ott teljesült a feltétel.

- Készítsük el a négy tömb szorzattömbjét!

A tömb elemeinek összege így pont a darabszámot adja ki, vagyis hogy hány ilyen tiszt végzett.

Készítsük el a tömbképletet:

```
=SZUM(
  HA(Nevék!B2:B933="n";1;0)*
  HA(Nevék!D2:D933=2007;1;0)*
  HA(Nevék!G2:G933>24;1;0)*
  HA(SZÁM(SZÖVEG.KERES("mérnök";Nevék!I2:I933));1;0)
)!
```

Ezzel kell majd elosztani a megfelelő átlagok összegét. Utóbbi kiszámolására a módszer az eddigiekhez hasonló, de mivel szeretnénk az egyes átlagokat is megkapni, az azokat tartalmazó tömböt is bevesszük a szorzatba. Végül ennek a tömbnek is össze kell adni az elemeit.

- Készítsük el a most már végleges képletet! Ehhez az előbb kapott összeget el kell osztani az előfordulások számával.

```
=SZUM(
  Nevék!E2:E933*
  HA(Nevék!B2:B933="n";1;0)*
  HA(Nevék!D2:D933=2007;1;0)*
  HA(Nevék!G2:G933>24;1;0)*
  HA(SZÁM(SZÖVEG.KERES("mérnök";Nevék!I2:I933));1;0)
)
/
SZUM(HA(Nevék!B2:B933="n";1;0)*
  HA(Nevék!D2:D933=2007;1;0)*
  HA(Nevék!G2:G933>24;1;0)*
  HA(SZÁM(SZÖVEG.KERES("mérnök";Nevék!I2:I933));1;0)
)
```

Ez már egy eléggé elgondolkodtató példa volt. A nehézséget az is okozta, hogy **tömbképleteknél nincs lehetőség logikai függvények használatára**, mert azok nem tömböt, hanem egyetlen IGAZ/HAMIS értéket adnak vissza. Helyettük matematikai műveleteket kell használni. Előbb például szorzással váltottuk ki az és kapcsolatot, vagy kapcsolatot pedig összeadással lehet létrehozni, de ekkor figyelni kell a többször is megszámlolt értékek utólagos kivonására.<sup>50</sup>

Az *Excel* tartalmaz pár olyan beépített függvényt is, ami tömbképletként működik, ezek közül a legismertebb a GYAKORISÁG( ).

---

<sup>50</sup> Ilyenkor valójában **logikai szitaformula** alkalmazására van szükség.

**32. példa: Nemes 2010-2011 3. forduló, 4/E (Eszmei érték)**

Védett növények adatait tartalmazó táblázattal dolgozunk.

- „Az egyes fajokat a becsült összérték alapján kategóriákba soroljuk. A kategóriák határai 100 000 €, 150 000 €, 200 000 €, 250 000 € és 500 000 €. A 705-ös számú sortól kezdődően határozd meg, hogy hány faj kerül az egyes kategóriákba!”

A példaválasztás nem véletlen, GYAKORISÁG ( ) függvényre lesz szükségünk.

Ki lehetne egyébként DARABTELI ( ) függvényekkel is számolni az egyes példányszámokat, de ahhoz, hogy egy növényt csak egy kategória esetén számoljunk meg, mindig ki kellene vonni a már kiszámolt értékeket. A megoldás így időigényes lenne, hiszen függvénymásolást nem tudnánk alkalmazni.

Gyakoriság	
100 000 € és alatta	168
150 000 € és alatta	37
200 000 € és alatta	61
250 000 € és alatta	111
500 000 € és alatta	266
felette	53

*Az egyes értékhatárok, és a hozzájuk tartozó fajok száma.*

- Írjuk be az értékhatárokat a 705-ös sortól kezdődően! Ha szeretnénk, hogy az általam megadott minta szerint nézzen ki, használjunk *Egyéni* cellaformázást!
- Jelöljük ki egy, **az értékhatárok számánál eggyel hosszabb tömböt**, ide fogjuk megkapni rendre a példányszámokat. Azért kell itt eggyel több cella, mint értékhatár, mert azon példányok számát is meg kell jeleníteni, amik az utolsó kategória fölé esnek (500 000 € felett).
- Adjuk meg a következő tömbképletet:

`C705:C710=GYAKORISÁG(G3:G698;B705:B709)`

Itt az első argumentum a becsült összértékeket tartalmazó oszlop a táblázatban, a második pedig az imént készített tartomány. A függvényt természetesen *Ctrl+Shift+Enter*-rel kell jóváhagyni. Az első szám, amit kaptunk, azon példányok száma, melyek becsült összértéke  $\leq 100\,000$  €. A következő szám azon példányok mennyiségét jelöli, amik összértéke  $>100\,000$  € és  $\leq 150\,000$  €, és így tovább. A

kapott tömb utolsó eleme azon példányok számát jelöli, amik értéke még az utolsó megadott határt is meghaladta: >500 000 €.

A függvény minden esetben így működik, ezért ügyelni kell **a kijelölt cellák számára és a határok növekvő sorrendjére**. Az *Excel* egyébként **nem figyelmeztet**, ha véletlenül kevés cellát jelöltünk ki, hanem egész egyszerűen csak azokat az értékeket számolja ki. Érdemes ezért mindig ellenőrizni, reális adatokat kaptunk-e.

Elképzelhető olyan feladat is, amikor az előzővel eltérően **diszkrét eseteket** számolunk. Például, kigyűjtjük az összes létező becült eszmei értéket, és megnézzük, melyik érték hányszor fordult elő. Ekkor a kapott példányszámok azt jelentik, hány olyan egyed van, ahol az eszmei érték **pontosan annyi**, mint a határokat megadó tartomány aktuális eleme. A függvény ugyanúgy helyesen fog működni, mint eddig, az utolsó értéke viszont szükségképpen nulla lesz. Emiatt sokaknál bevett szokás, hogy pont annyi cellát jelölnek ki, mint ahány határ van, csak azért, hogy a nulla ne legyen zavaró. Ez diszkrét eseteknél általában még nem jelent problémát, de ez a „lespórolás” okozhatja azt a típushibát, hogy sokaknál **folytonos eseteknél is hiányzik az utolsó cella**. Ezért pedagógusként nem tartom tanácsosnak a diszkrét esetek megoldásának elkülönítését a folytonos esetekétől: nem olyan zavaró ott az a nulla, mint amekkora gondot okozhat az utolsó cella lefelejtése folytonos eseteknél.

#### 7.4. Szövegkezelés

Az *Excel* számos szövegkezelést megvalósító függvénnyel rendelkezik. A *Függvény beszúrása* ablak lehetővé teszi az ezek közti böngészést, amennyiben a *Szöveg* kategóriát választjuk. Mivel itt minden fontos információ megtalálható, a megoldásaim közt pedig jó pár idetartozó példa van, egyebet nem is írnék ehhez az alfejezethez.

Gyakorlásukhoz a következő példákat javaslom:<sup>51</sup>

- Nemes 2009-2010 1. forduló, 5/B (Locsolóverseny)
- Nemes 2009-2010 3. forduló, 5/C (Holdak)
- Nemes 2010-2011 3. forduló, 4/F (Eszmei érték)

---

<sup>51</sup> Érdemes újra átfutni a *Helyettesítő karakterek használata* fejezet *Cella tartalmát vizsgáló, IGAZ/HAMIS értéket visszaadó képletben* című részét.

- OKTV 2010-2011 3. forduló, 5/F (Eszmei érték)

## 8. Lehetőségelemzés

A következőkben olyan eszközöket mutatok be, amiket az *Excel* súgója nagyon találóan „**mi lenne, ha**” típusúaknak nevez.

### 8.1. Célérték keresése

Gyakran találkozhatunk a mindennapi életben olyan esettel, hogy tudjuk, hogy milyen értéket akarunk kapni, csak azt nem, hogy milyen kiinduló adatok szükségesek hozzá. Leggyakoribb példa, hogy ha kölcsönt veszünk fel: tudjuk, mekkora összeget kell majd visszafizetni, és azt is, hogy mennyi pénzt tudunk havonta megtakarítani. A kérdés, hogy így mennyi ideig tart majd visszafizetni. Ha a kölcsön kamatmentes lenne, könnyedén rendezhetnénk az egyenletet, és egy alkalmas osztással megkapnánk az eredményt. Általában viszont nem ilyen egyszerű a dolgunk (például az előbbi példát továbbgondolva: kamatos kölcsön esetén), mert a kívánt értéket tartalmazó cella képletét hosszadalmas lenne explicit alakra hozni, a hibázási lehetőségekről és a megoldhatóság kérdéséről nem is beszélve.

Jöhet egy másik stratégia: próbálgatás. Sajnos ez is elég hosszadalmas, és sokszor nem is elég pontos. Éppen ezért tartalmaz az *Excel* egy olyan funkciót, mely „helyettünk próbálgat”, egyre jobban közelítve a kívánt érték felé: az *Adatok* fül/*Adateszközök* panelén, a *Lehetőségelemzés* csoportban található a **Célértékkeresés**.

Lássunk a használatára egy egyszerű példát!

### 33. példa: Nemes 2010-2011 2. forduló, 4/J (A légierő pilótáinak kiképzése)

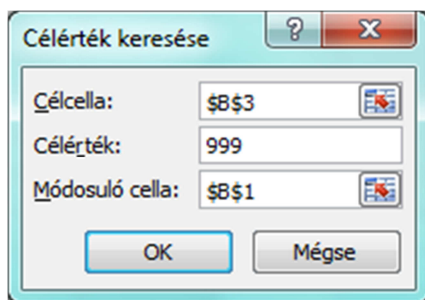
A feladat, hogy meghatározzuk, hány óra földi kiképzésen volt az a főhadnagy, aki azt állítja, hogy „*pontosan tizedannyi óra földi kiképzésben részesült, mint ahány percet töltött a levegőben, és éppen 999 pontja lett*”. A pontokat a rang, és a földi kiképzéseken, illetve levegőben töltött idő függvényében lehet kiszámolni, ennek pontos leírására a feladat korábbi részében került sor.

- Írjunk az *B1*-be egy tetszőleges számot, itt lesz a földi képzés óráinak száma, vagyis amit keresünk. Értéket nem is kötelező a cellába írni, de sokaknak így könnyebb dolgozniuk, mert felfogható ez az érték egy tippként.

- Az *B2*-be jön a levegőben töltött percek száma, ez a főhadnagy állítása szerint  $=A1 * 10$  perc, tehát ezt a képletet írjuk bele.
- *B3*-ba jön a kapott pont, vagyis aminek 999-nek kell majd lennie. Erre használjuk a korábbi képletet, javítva a hivatkozásokat, hogy az előbbi cellákat használja a képlet:

```
=INDEX(
    Képzés!AI3:AM7;
    HOL.VAN(B1;Képzés!AH3:AH7;1);
    HOL.VAN("Főhadnagy";Képzés!AI2:AM2;0)
)
+HA(B1>400;B1-400;0)
+KEREK.LE(B2/60;0)*10
```

- Válasszuk a *Célértékkeresés* funkciót!
- A *Célcella B3*, ennek kívánt értéke 999 (*Célérték*). A *Módosuló cella* az a cella, ami értéket tartalmaz (nem pedig képletet), és amitől függ a *Célcella* értéke, tehát *B1*.
- Az elvégzett számítás jóváhagyása előtt az ablakot le is kell fotózni, illetve képként beilleszteni a munkalapra. A legegyszerűbb módszer *Alt+PrintScreen* billentyűkombinációval vágólappra helyezni az ablak fotóját, majd *Ctrl+V*-vel beilleszteni, de *Win7* alatt rendelkezésünkre áll a *Képmetsző* is.



A *Célértékkeresés* ablaka.

Látható, hogy az eszköz használata milyen egyszerű, de nem szabad szem elől téveszteni, hogy a kívánt célértékhez tartozó bemenő adatot az *Excel* is csak próbálgatással számolja, így gyakran nem olyan pontos, mint azt várnánk. Például könnyen ellenőrizhető, hogy a  $x^2 - \frac{1}{2}x - \frac{1}{2} = 0$  egyenlet megoldásai az 1 és a  $-\frac{1}{2}$ . Ha azonban az egyenletet *Célértékkereséssel* oldjuk meg, akkor a gyök 1,0001679 lesz, és

a célérték sem a megadott nulla: 0,0002519. Ez egyrésztől egy kicsit pontatlan, másrésztől semmit nem tudunk arról, hogy van-e más megoldása az egyenletnek, és ha igen, az mennyi.

Tehát a módszer nagyon hasznos, de nem mindegy, mire akarjuk használni: például egy egyenlet összes gyökeinek megkeresésére nem való.

## 8.2. Solver

A *Solver* az *Excel* egyik legtöbbet használt **bővítménye**, tehát eredetileg csak passzív része a programnak. Így, ha használni szeretnénk, előbb be kell töltenünk. Ennek lépéseiről – sőt a funkció használatáról is, az itteninél bővebben – remek leírás olvasható a *Microsoft* honlapján.<sup>52</sup>

A funkció nagyon hasonlít az előbb tárgyalt *Célértékkeresés*hez, hiszen ugyanolyan típusú problémák megoldására való, csak éppen több beállítási lehetőséggel rendelkezik. Így minden *Célértékkereséssel* megoldható probléma megoldható *Solver*-rel is, egyes esetekben viszont csak a *Solver* nyújt segítséget.

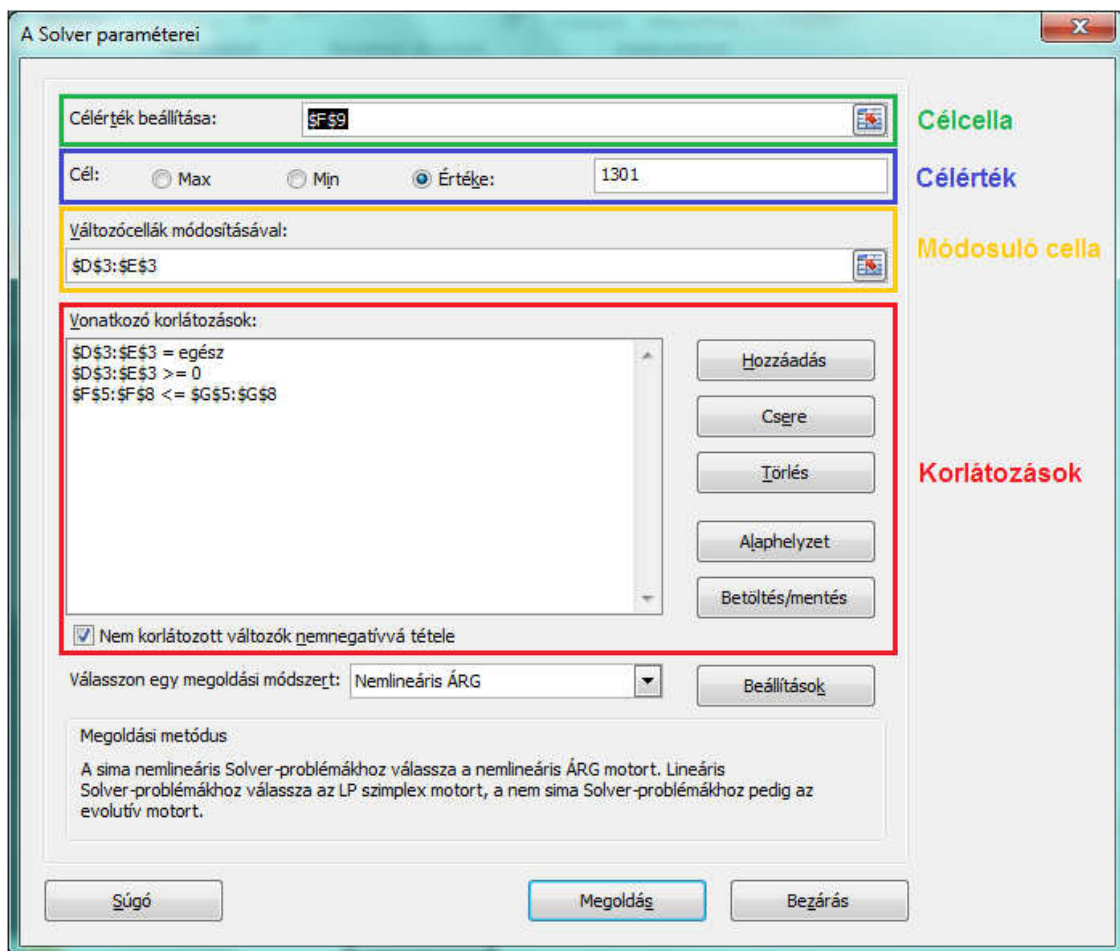
Nézzük, mely szempontok alapján egyezik meg a két funkció:

- Meg kell adni egy cellát, ahol az eredményt szeretnénk megkapni.  
Ennek *Célértékkeresés*nél **Célcella**, *Solver*-nél *Célkitűzőcella* nevet adtak. Természetesen képletet kell tartalmaznia.
- Meg kell adni az értéket, amit az előbbi cellában szeretnénk kapni.  
*Célértékkeresés*nél ezt hívjuk **Célérték**nek, a *Solver* pedig egyszerűen csak *Célként* hivatkozik rá.
- Meg kell adni, hogy mely cella tartalmát módosítsa ennek érdekében.  
*Célértékkeresés*nél **Módosuló cella**nak, *Solver*-nél *Változócella*nak hívjuk. Közvetve vagy közvetlenül kapcsolódnia kell a *Cél(kitűzés)cella*hoz.
- A megoldási módszer mindkét esetben a próbálgatás, így nem biztos, hogy pontos értéket kapunk, vagy, hogy egyáltalán van megoldás.
- Az eredmény **nem adatkövető**.

---

<sup>52</sup> <http://office.microsoft.com/hu-hu/excel-help/a-solver-bovitmeny-betoltese-HP010342660.aspx?CTT=1>

A következő képernyőkép az eszköz használatát mutatja be, megjelölve az előbb is említett paramétereket. Habár ez nem teljesen szakszerű, az összhang megőrzése érdekében **maradnék a Célértékkeresésnél megszokott kifejezéseknél**. A mellékelt feladatmegoldásoknál is ugyanezt a színezést használom, hogy könnyebb legyen összevetni azokat a leírással (persze kék színezést ne keressen senki). Gyakran előfordul, hogy a módosuló cellákra korlátozás is vonatkozik, ezért döntöttem úgy, hogy a narancssárga színt kitöltésre, a pirosat betűszínezésre használom, lehetőséget biztosítva így egy cella két színnel történő megjelölésére.



*A Solver ablaka, megjelölve benne a legfontosabb részek.*

Nézzük meg azt is, miben különbözik a Solver a Célértékkereséstől:

- Célértéknek nem csak egy konkrét szám adható meg, hanem az is, hogy a Célcella értéke legyen *minimális*, vagy éppen *maximális*.
- Akár több, de maximum 200 *Módosuló cellát* is megadhatunk, vesszővel elválasztva.



- Megadhatóak úgynevezett *korlátozások* is, vagyis, hogy mely kikötések mellett kívánjuk a célunkat elérni. Egy korlátozáshoz meg kell adni a cellatartományt és a feltételt. A feltételben a  $\leq$ ,  $=$ ,  $\geq$  relációs jelek, vagy az `int` (egész), `bin` (bináris) és `dif` (mind különböző) kulcsszavak valamelyike használható.

Látható, hogy a *Solver* használatának megértése nem is olyan nagy lépés a *Célértékkeresés* ismerete mellett. Mégsincs egyszerű dolgunk: a feltételek megfogalmazása, és annak olyan formába öntése, amit a *Solver* is megért, gyakran nem kis feladat. A funkció rutinszerű használatának képessége rengeteg gyakorlást, kreativitást igényel!

Nézzünk meg előbb egy egyszerűbb, majd egy nehezebb példát!

### 34. példa: Szendvicsek

Meghívtuk tizennégy ismerősünket vendégségbe. Kétféle szendvicset tudunk csinálni:

- *sonkásat* (3 dkg vaj, 3 dkg sonka, 2 dkg sajt, negyed tojás)
- *sajtosat* (2 dkg vaj, 1 dkg sonka, 5 dkg sajt, fél tojás).

Kenyérből bármennyi a rendelkezésünkre áll, de a többi alapanyagból csak korlátozottan: 120 dkg vaj, 100 dkg sonka, 200 dkg sajt, 20 db tojás.

- Készítsük el a lehető legtöbb szendvicset! Hány sonkásat és hány sajtosat tudunk majd felkínálni?*
- Úgy döntöttünk, hogy a szendvicseket inkább pénzért adjuk, hogy a bevétellel kompenzáljuk az egyéb kiadásokat. A sonkásan darabonként 40, a sajtoson darabonként 31 Ft-ot tudunk keresni. Mennyi lehet a maximális haszon?*
- Mennyi lesz a minimális haszon, ha legalább annyi szendvicset kell csinálnunk, mint ahányan leszünk?*
- Hány sajtos és hány sonkás szendvics készült, ha mindet eladtuk, és pontosan 1 301 Ft lett a nyereség?*

Az ilyen feladatoknál mindig az a legnehezebb, hogy kitaláljuk, milyen formában rögzítsük az adatokat, hogy aztán könnyedén megfogalmazhassuk a feltételeket. Szerencsére most pont nincs nehéz dolgunk, ráadásul a négy feladat erőteljesen hasonlít is egymásra, így elég egyszer kitalálnunk az elrendezést, és aztán munkalapmásolással előkészíteni a helyet mindegyik feladat megoldására.

- Rögzítsük az adatokat az alábbi formában!  $D5:F9$ , valamint  $C12$  számolt cellák, készítsük el a képleteket is! Ha végeztünk, másoljuk a munkalapot három példányban, hogy mind a négy feladat külön munkalapon legyen!

	A	B	C	D	E	F	G
1	<i>Hány szendvicset tudunk maximum készíteni?</i>						
2							
3	Összetevők	Darabja		0 db	0 db	Összesen	Maximum
4		Sonkás	Sajtos	Sonkás	Sajtos		
5	Vaj	3 dkg	2 dkg	0 dkg	0 dkg	0 dkg	120 dkg
6	Sonka	3 dkg	1 dkg	0 dkg	0 dkg	0 dkg	100 dkg
7	Sajt	2 dkg	5 dkg	0 dkg	0 dkg	0 dkg	200 dkg
8	Tojás	1/4	1/2	0	0	0	20
9	Bevétel	40 Ft	31 Ft	0 Ft	0 Ft	0 Ft	
10							
11	Meghívottak:	14 fő					
12	Szendvicsek száma:	0 db					

Az a) feladathoz tartozó munkalap. Szürke kitöltése azoknak a celláknak van, amikre ennél a feladatnál nincs szükség.

Az a) feladat megoldása következik.

Az ábra sokatmondó:  $C12$ -be szeretnénk kiszámolni, maximum hány szendvicset készíthető el. Ehhez  $D3$  és  $E3$  értékeit kell próbálgatni, miközben figyelünk arra, hogy ne lépjük túl a  $G5:G8$ -ban rögzített maximális mennyiségeket, és ne adjunk meg életszerűtlen adatokat.

- Válasszuk az *Adatok* fül/ *Solver* gombját<sup>53</sup> és töltsük ki az ablakot az alábbiak szerint:

<b>Célcella:</b>	$\$C\$12$
<b>Célérték:</b>	Max
<b>Módosuló cellák:</b>	$\$D\$3 : \$E\$3$
<b>Korlátozások:</b>	$\$D\$3 : \$E\$3 = \text{egész}$
	$\$F\$5 : \$F\$8 \leq \$G\$5 : \$G\$8$

<sup>53</sup> Ne feledjük, a *Solver* alapértelmezetten nem tartozik a programhoz, így amíg nem töltjük be, nem fogjuk a fülön találni!

Az eredményt gyorsan megkapjuk: 50 db szendvics készíthető ennyi alapanyagból. Ennél jóval többet is megtudhatunk, ha a megoldás megtalálásakor nem csupán elfogadjuk azt, hanem kérünk egy **jelentést** is:

➤ Az *OK*-ra kattintás előtt válasszuk a *Jelentések/ Eredmények* opciót!

Így egy generált munkalapon olvashatjuk a megoldás megtalálásának körülményeit, többek közt azzal a fontos információval, hogy nem a sonka és a sajt mennyisége a korlátozó.

Következzék a *b)* feladat megoldása!

Most nem a mennyiség optimalizálása a cél, hanem a bevétel, ez kerül *F9*-be. Ezt leszámítva azonban ugyanúgy kell megoldani a feladatot, mint előbb.

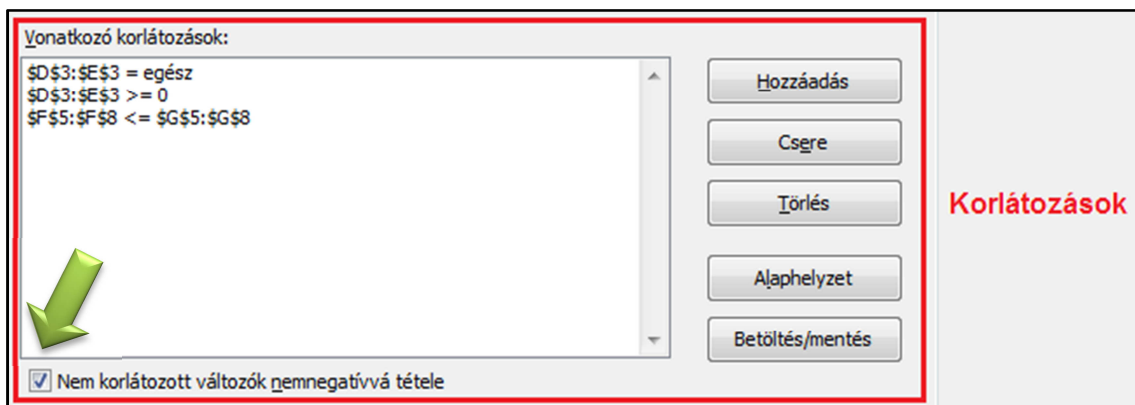
<b>Célcella:</b>	<b>\$F\$9</b>
<b>Célérték:</b>	Max
<b>Módosuló cellák:</b>	\$D\$3:\$E\$3
<b>Korlátozások:</b>	\$D\$3:\$E\$3 = egész
	\$F\$5:\$F\$8 <= \$G\$5:\$G\$8

A *c)*-ben egy kicsit már más jellegű a feladat, mint amilyen az előbbi volt. Egyrészt most *minimális* értéket keresünk, másrészt pedig egy új feltétel is adódik: mindenkinek jusson legalább egy szendvics.

<b>Célcella:</b>	\$F\$9
<b>Célérték:</b>	<b>Min</b>
<b>Módosuló cellák:</b>	\$D\$3:\$E\$3
<b>Korlátozások:</b>	\$D\$3:\$E\$3 = egész
	\$F\$5:\$F\$8 <= \$G\$5:\$G\$8
	<b>\$C\$12 &gt;= \$C\$11</b>

Valójában még egy dologra is figyelni kellene: ne lehessen negatív számú szendvicset megadni. Mivel a bevétel minimalizálásáról van szó, ennek figyelembevétele nélkül a megoldás a következő lenne:  $-43$  db sonkás szendvics,

57 db sajtos szendvics és így 47 Ft bevétel, ami nyilván nem egy életszerű megoldás. Az ok, amiért mégsem kell külön kikötni a szendvicsek számának nem negatív voltát, az egy kis jelölőnégyzet a Solver ablakában, ami alapértelmezetten ki van pipálva: *Nem korlátozott változók nem negatívvá tétele*.



*A jelölőnégyzet használata sok munkától megkímél minket.*

A d) feladatban csupán az az újdonság, hogy a cél most nem valamiféle minimum/ maximum optimalizálás, hanem egy konkrét érték. Igazából egy ugyanolyan visszakeresésről van szó, mint amit *Célértékkeresés*nél is szoktunk csinálni, csak éppen az eddigi feltételek megtartása mellett.

<b>Célcella:</b>	\$F\$9
<b>Célérték:</b>	1301
<b>Módosuló cellák:</b>	\$D\$3 : \$E\$3
<b>Korlátozások:</b>	\$D\$3 : \$E\$3 = egész
	\$F\$5 : \$F\$8 <= \$G\$5 : \$G\$8

Láttuk, hogy mekkora segítség lehet számunkra ez az eszköz, és azt is, hogy a táblázat elrendezésének kitalálását követően már egyszerű dolgunk van. Ennél bonyolultabb feladatoknál azonban jó résen lenni, és olykor esetleg felülbírálni a program működését, mert az sok esetben nem elég determinisztikus. A következő példa ezt mutatja be.

**35. példa: Nemes 2008-2009 3. forduló, 5 (Regionális forduló)**

Egy verseny regionális fordulójának beosztását kell elkészíteni, az utazási költség minimalizálása mellett.

Adott:

- az iskolák neve és az onnan továbbjutók száma,
- a vizsgaközpontok (A, B, C és D) és maximális befogadóképességük,
- az egyes iskolák és a négy vizsgaközpont közti utazások költsége

Kell:

- a beosztás (melyik iskola tanulói melyik vizsgahelyre menjenek, figyelembe véve azok befogadóképességét),
- az összes utazási költség.

Mindezt tehát úgy kell megadni, hogy „ne szegjük meg a szabályokat”, és **az összes útiköltség a lehető legkevesebb legyen.**

A feladat semmit nem ír arról, hogy ehhez a *Solver*-t kellene használni, így aki nem ismeri ezt a lehetőséget, nagy valószínűséggel neki sem tud fogni a feladatnak. Sok esetben lehetne kézzel is próbálgatni, ennél a konkrét példánál azonban  $4^{13}$  lehetőséget kellene megvizsgálni a férőhelyek figyelembevétele illetve a költség minimalizálása szempontjából, amire nem hinném, hogy bárki szívesen vállalkozik (arról nem is beszélve, hogy még ha meg is találná a jó megoldást, akkor se sok pontot kapna a feladatra).

Elsőként importálni kell az adatokat, és valamilyen formában elrendezni őket a munkalapon. Utána át kell gondolni, milyen feltételeknek is kell megfelelni:

- Minden vizsgahelyre maximum a férőhellyel megegyező számú diákot osszunk be.*
- Ugyanazon iskola tanulói ugyanazon vizsgahelyre legyenek besorolva.*

Ezt követően arra van szükség, hogy a táblázat elrendezésével, képletek használatával ezeket olyan formában fogalmazzuk meg, amit „a *Solver* is megért”. Ne feledjük, hogy csak a következő típusú feltételek adhatóak meg:

- *Egy tartomány minden cellája legyen  $\leq$ ,  $=$ , vagy  $\geq$ , mint valamilyen általunk megadott értékek, vagy cellák értékei.*

- Egy tartomány minden cellája legyen egész.
- Egy tartomány minden cellája legyen 0 vagy 1.
- Egy tartomány minden cellája legyen egyedi.

Most jön egy kis okoskodás: „hogyan lehetne az előbb felsorolt feltételekkel megfogalmazni az a)-ban és b)-ben leírtakat?”. Ezt kitalálni nem egyszerű. A következő képen a táblázat egy olyan elrendezése található, mely lehetővé teszi a feltételek megfogalmazását a következőképpen:

a)  $\sum_{i=1}^{16} C_i \leq \sum_{i=1}^{16} D_i$

b)  $\sum_{i=1}^{16} C_i = \text{bináris}$   
 $\sum_{i=1}^{16} D_i = 1$

1	A	B	C	D	E	F	G	Kijelölt vizsgahely				M	N	Versenyzők száma				T	
								A	B	C	D			Σ	A	B	C		D
2	Iskola neve	Tovább jutottak	Útiköltség																Kijelölt vizsgahely
3	Andrássy Gyula Gimnázium	5	140	200	130	150					0			0	0	0	0		
4	Arany János Szakközépiskola	7	190	370	330	260					0			0	0	0	0		
5	Deák Ferenc Gimnázium	7	370	250	260	120					0			0	0	0	0		
6	Hajós Alfréd Gimnázium	7	260	370	100	360					0			0	0	0	0		
7	Illyés Gyula Szakközépiskola	3	120	240	310	240					0			0	0	0	0		
8	Jékely Zoltán Gimnázium	3	220	120	320	210					0			0	0	0	0		
9	Kodály Zoltán Szakközépiskola	3	370	240	390	160					0			0	0	0	0		
10	Leőwey Klára Szakközépiskola	9	230	230	260	310					0			0	0	0	0		
11	Madách Imre Szakközépiskola	2	120	320	250	370					0			0	0	0	0		
12	Ónd Vezér Gimnázium	5	220	330	130	330					0			0	0	0	0		
13	Rákóczi Ferenc Gimnázium	6	130	330	280	180					0			0	0	0	0		
14	Szent István Gimnázium	5	180	350	200	250					0			0	0	0	0		
15	Zrínyi Miklós Szakközépiskola	3	110	160	120	130					0			0	0	0	0		
16														Σ	0	0	0	0	
17																			
18	Összes költség:	0 Ft																	
19																			
20																			
21																			
22																			

Az üres táblázat egy lehetséges elkészítése.

- Készítsük el a táblázatot a képnek megfelelően! A sárga tartományt kell a kívántaknak megfelelően kitölteni, a zöld cellában található az eredmény, a pirosakat pedig értelemszerűen, a megadott sárga mezők függvényében, képlettel számítjuk ki.
- Indítsuk el a Solver-t (Adatok fül), és állítsuk be a következőképp:

A Solver paramétereit

Célérték beállítása:

Cél:  Max  Min  Értéke:

Változócellák módosításával:

Vonatkozó korlátozások:

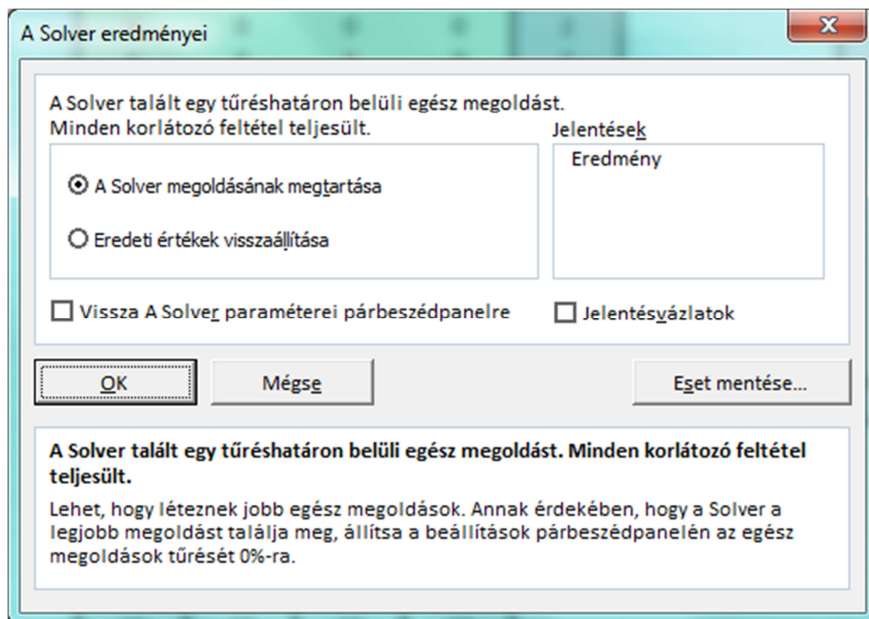
Nem korlátozott változók nemnegatív tétele

Válasszon egy megoldási módszert:

Megoldási módszer  
A sima nemlineáris Solver-problémákhoz válassza a nemlineáris ARG motort. Lineáris Solver-problémákhoz válassza az LP szimplex motort, a nem sima Solver-problémákhoz pedig az evolutív motort.

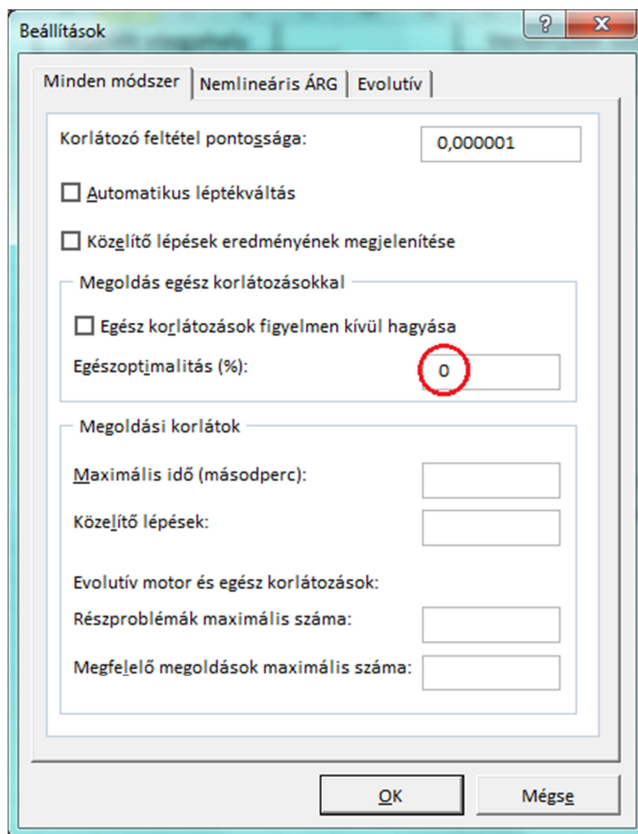
*A Célcella, a Célérték, a Módosuló cellák, valamint a korlátozások megadása.*

A keresés jó ideig is eltarthat, majd nagy eséllyel a következő ablakot kapjuk:



*Úgy tűnik, nem is olyan biztos, hogy jó megoldást találtunk.*

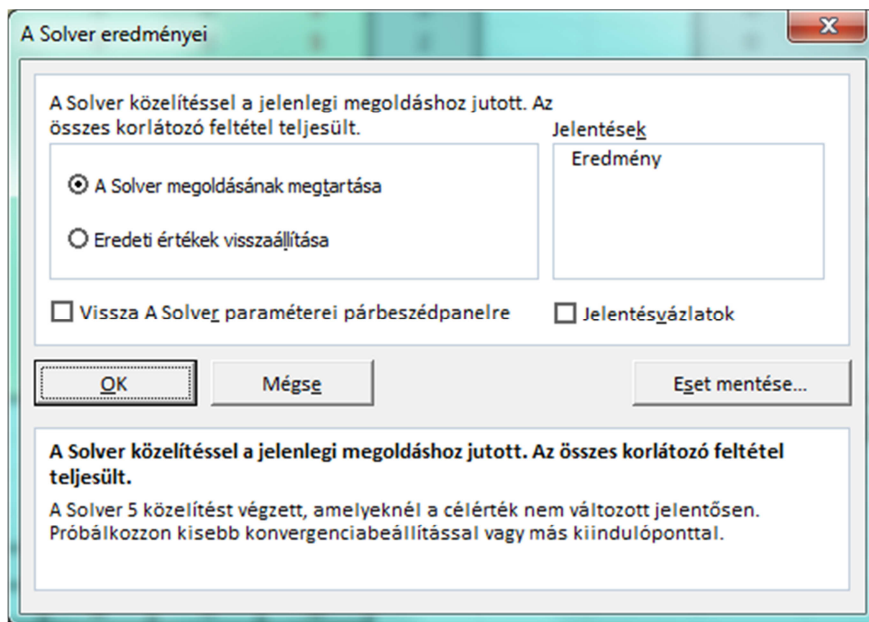
Az egész megoldások tűrését valóban célszerű nullára állítani, különben lehetséges, hogy egészek helyett csak egészekhez közeli értékekkel tölti fel a program a *Módosuló cellákat*.



*A Solver beállításai.*



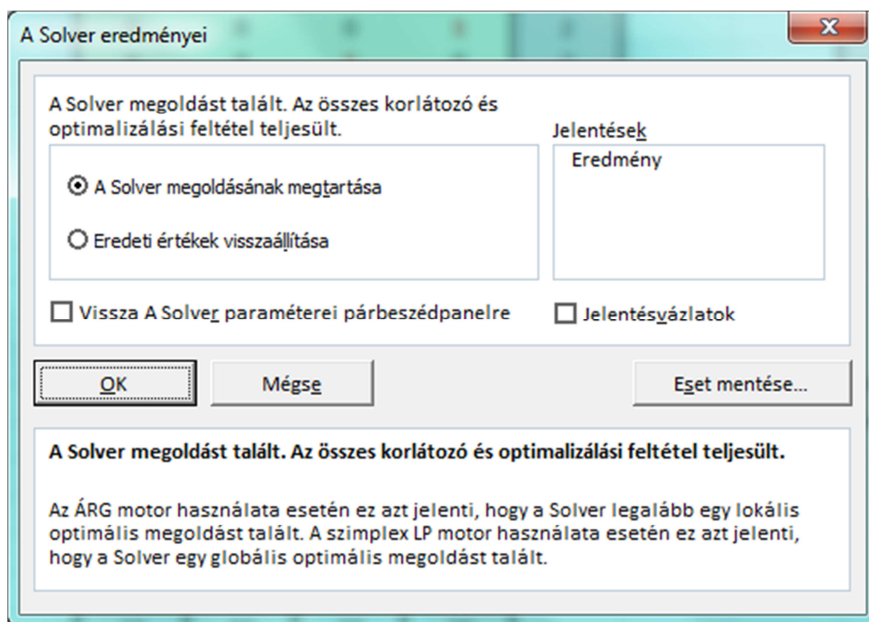
- Végezzük el az előbb említett beállítást, és futtassuk le újra a keresést!



*A beállítást követően se sokat javult a helyzet.*

Habár javult a helyzet, és jobb megoldást kaptunk, mint előbb, még ez sem az igazi. Ha azonban még egyszer lefuttatjuk, jó eséllyel megkapjuk végre az optimális beosztást, mellyel az utazási költség mindössze **10 530 Ft** lesz:

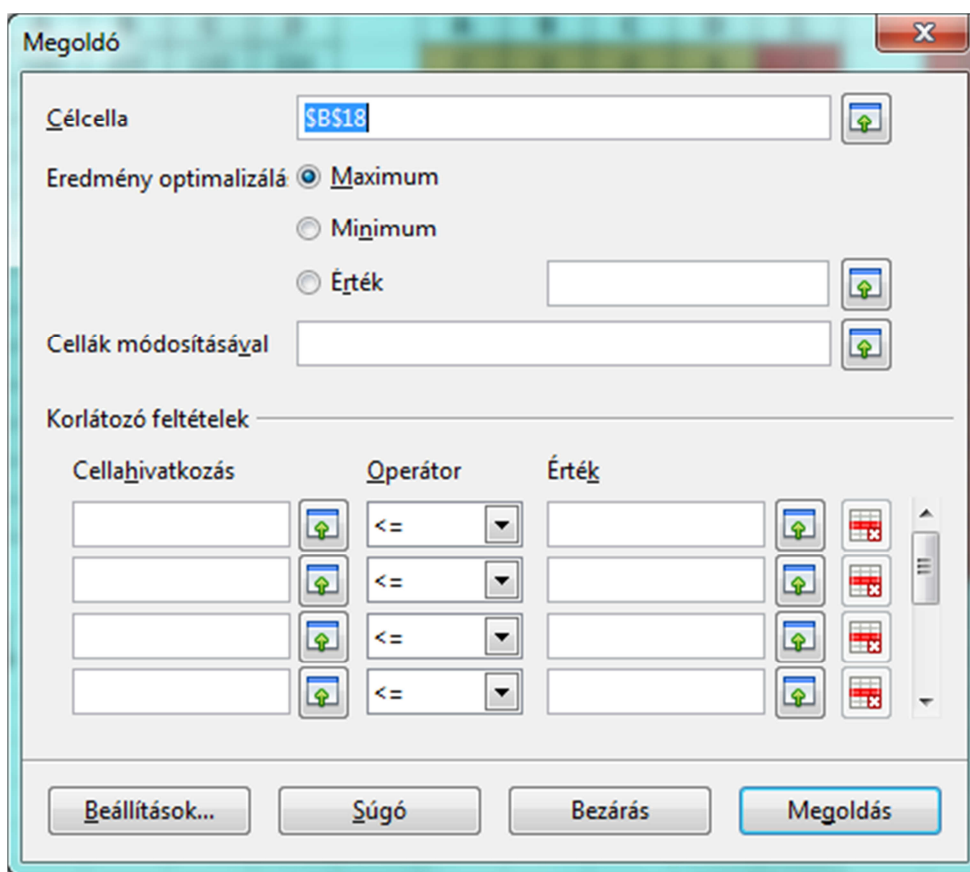
- Futtassuk ismét a *Solver*-t, még mindig az eddigi beállításokkal!



*A Solver megtalálta a megoldást.*

Az eddigi tortúrából egyértelműen megállapíthatjuk, hogy ez a feladat túl nagy falat a *Solver*-nek, hiszen már elsőre meg kellett volna találnia a megoldást. Tapasztalataim szerint az *Excel* bizonyos verziói akár még arra is hajlamosak, hogy egyáltalán nem adnak megoldást (hacsak nem adunk meg kézzel „pár” kezdő értéket, megkönnyítve ezzel a dolgát). Sőt, olykor az is előfordulhat, hogy hiába állítjuk az egészoptimalitást 0%-ra, mégsem egészeket kapunk.

Ugyanezt a feladatot a *Libre Office Calc*-ja kifogástalanul megoldja. A *Solver* funkciót teljes egészében megvalósító *Megoldó* pillanatok alatt megtalálja az optimális beosztást, amennyiben a *LibreOffice lineáris megoldóprogramot* választjuk a *Beállításokban* (nem ez az alapértelmezett). Ebben az esetben ezért érdekesebb lehet ezt használni.<sup>54</sup>



A *Calc* *Megoldó* nevű funkciójának ablaka.

<sup>54</sup> Az *xls* mellett megtalálható a feladat *Calc*-ban készült megoldása is.

### 8.3. Adattábla

Míg az előbb tárgyalt két eszköz a megadott eredményt adó bemeneti érték(ek)et keresi meg, addig az adattábla kimeneteket határoz meg, **különböző bemeneti érték(ek) függvényében**. Használatával egy olyan táblázatot kapunk, amiben könnyedén áttekinthető, milyen lehetséges eredmények szülehetnek, és azok hogyan változnak a különféle bemenetek esetén. Habár ezt általában képletmásolással is meg lehet oldani, egyszerűbb, ha inkább ezt az éppen erre tervezett eszközt használjuk.

Létezik egy- és kétváltozós adattábla is, de előbbit ritkán szoktuk használni. Ritkán fordul ugyanis elő, hogy a képlet annyira összetett, hogy a feladat megoldása függvénymásolással hosszabb lenne, mint adattábla használatával. A bemeneti adatok iránya ekkor lehet vízszintes vagy függőleges is, a képletet tartalmazó cella helye ettől függően változik.

	A	B	C	D	E	F	G
1	Heti óraszám	13					
2	Órabér	1 500 Ft					
3	Havi fizetés	78 000 Ft					
4	10						
5	11						
6	12						
7	13						
8	14						
9	15						
10	16						
11	17						
12	18						
13	19						
14	20						

**Adattábla**

Sorértékek bemeneti cellája:

Oszlopértékek bemeneti cellája:

OK      Mégse

*Mennyi lenne a fizetésünk, ha nem 13 órát tanítanánk hetente?*

*Az egyváltozós adattábla készítésekor a bemeneti cellák egyikének helye üresen marad.*

A példán képlet a sárga kitöltésű cellában van. Ha egyváltozós adattáblát akarunk használni, és a lehetséges értékek függőleges irányban helyezkednek el, a képletnek mindig azok tartományától eggyel jobbra és eggyel fentebb kell lennie. A funkció elindítása előtt szükség van az egész *A3:B14* tartomány kijelölésére, az ablakban pedig a sorértékek bemeneti cellájának üresen hagyására.

Ehhez hasonlóan működik a kétváltozós adattábla is, de itt – mint ahogy nevében is benne van – két bementtől függ az eredmény.

Hogy érthetőbb legyen a leírás, egy példán keresztül mutatom be, hogy hogyan kell használni.

### 36. példa: Nemes 2010-2011 2. forduló, 4/K (A légielő pilótáinak kiképzése)

Egy olyan táblázatot kell készíteni, ahonnan a főhadnagyok közvetlenül le tudják olvasni továbbképzési pontjaikat, a földi képzési óra és levegőben töltött órák függvényében.

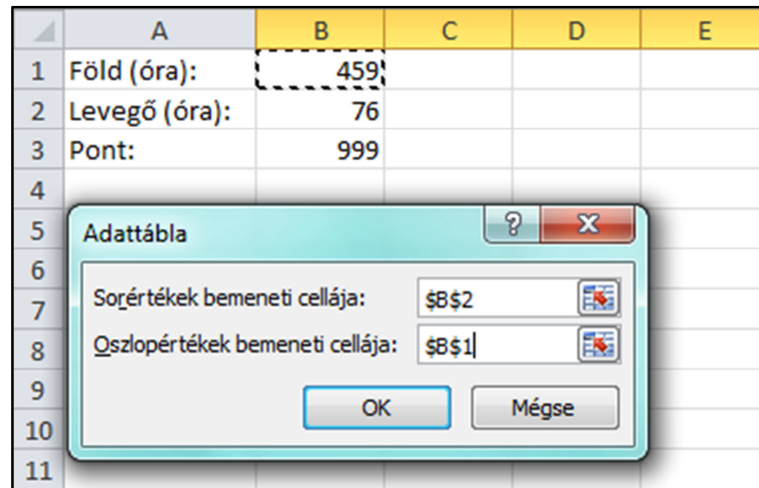
		Repülési órák										
		65	66	67	68	69	70	71	72	73	74	75
Földi képzési órák	350	650	660	670	680	690	700	710	720	730	740	750
	360	650	660	670	680	690	700	710	720	730	740	750
	370	650	660	670	680	690	700	710	720	730	740	750
	380	650	660	670	680	690	700	710	720	730	740	750
	390	650	660	670	680	690	700	710	720	730	740	750
	400	800	810	820	830	840	850	860	870	880	890	900
	410	810	820	830	840	850	860	870	880	890	900	910
	420	820	830	840	850	860	870	880	890	900	910	920
	430	830	840	850	860	870	880	890	900	910	920	930
	440	840	850	860	870	880	890	900	910	920	930	940
	450	880	890	900	910	920	930	940	950	960	970	980
	460	890	900	910	920	930	940	950	960	970	980	990
	470	900	910	920	930	940	950	960	970	980	990	1000
	480	930	940	950	960	970	980	990	1000	1010	1020	1030
	490	940	950	960	970	980	990	1000	1010	1020	1030	1040
500	960	970	980	990	1000	1010	1020	1030	1040	1050	1060	
510	970	980	990	1000	1010	1020	1030	1040	1050	1060	1070	

Egy részlet az elkészítendő kétváltozós adattáblából.

A feladat természetesen megoldható képletmásolással is, erre láthatunk példát a *Feladatok (2)* munkalapon. Egyszerűbb a dolgunk azonban, ha adattáblaként készítjük el, hiszen a vegyes hivatkozások elkészítésébe könnyen bele lehet keveredni.

- Készítsük el a táblázat keretét! A felsorolt földi és repülési órák lehetséges értékei a táblázat sor- illetve oszlopcímeiként funkcionálnak majd.
- Helyezzük el a korábban már elkészített, pontokat kiszámoló képletet **a táblázat bal felső sarkában (B14)**<sup>55</sup>
- Jelöljük ki **az egész táblázatot (B14:W31)**!
- Válasszuk az *Adatok fül/ Lehetőségelemzés/ Adattábla* parancsot!
- Az adattábla sor- és oszlopértékeinek bemeneti cellái a *Célértékkeresés*hez is használt két cella legyenek, a képek megfelelően!

<sup>55</sup> Kétváltozós adattábla esetén a képletnek mindig a táblázat bal felső cellájában kell lennie.



A kétváltozós adattábla bemeneti celláinak megadása.

A repülési órák lehetséges értékeit *sorértékek*nek nevezzük (vízszintesen, egy sorban helyezkednek el), míg a földi képzési óraszámokat *oszlopértékek*nek. Az adattábla ablakának kitöltésekor voltaképp azt adjuk meg, hogy **a képletet tartalmazó cella mely bemenete helyett helyettesítgesse be az általunk megadott sor- és oszlopértékeket.**

➤ Töltsük ki a mezőket, majd kattintsunk az *OK*-ra!

Időtakarékos megoldás tehát, hogy ha már egyszer valahol úgyis elvégeztük a számítást – legyen az bármilyen hosszú – akkor ezt újabb munka nélkül, nagyon egyszerűen felhasználjuk. Ez a módszer azért is praktikusabb, mert nem csak, hogy a bemenetek átírásakor frissíti az eredményeket, de még a képlet megváltozása esetén is automatikusan újraszámol mindent. Képletmásolás esetén az utóbbi nem teljesül: a friss képlet újbóli másolására van szükség.

Mivel a mintán nincs semmi a táblázat bal felső sarkában, vagy mégis a képletmásolást választjuk, vagy úgy állítjuk be a cella formázását, hogy semmilyen érték esetén se mutasson semmit. Például, ha a formátumkódnak ""-t adunk meg, vagy a betűszínt a háttérszínnel megegyezőre állítjuk.

## 9. Adatok érvényessége és lapvédelem

Nem fordult sokszor elő, hogy versenyen az *Excel* ezen funkcióit kérték számon, pedig a mindennapi életben gyakran lehet szükség rájuk. Ezért, és mert könnyedén előfordulhat, hogy ennek a két témának a bevonása lesz a versenyek következő újítása, szeretnék mutatni egy példát a használatukra.

Versenypélda híján egy saját példán keresztül fogom szemléltetni a lehetőségeket, melynek alapötletét egy feladatgyűjteményből merítettem.<sup>56</sup>

### 37. példa: Egy saját példa: Virágrendelés

Szerencsére a számla alapja biztosítva van a források közt, így nem kell a formázással bíbelődnünk, koncentrálhatunk a lényegre: a feladat egy „bolondbiztos” munkalap elkészítése.

- Nyissuk meg a forrásfájlt, és rögtön mentsük is a *Mentés másként* funkcióval, hogy a nyersfájlon ne történjenek módosítások!
- Az éppen aktuális dátum kitöltése függvénnyel történik: MOST( ). Formázzuk a cellát úgy, hogy a dátum mellett az idő is látszódjon (például éééé.hh.nn.óó:pp formátumkóddal)!
- Írjunk be egy előforduló terméknevet, hogy megkönnyítsük a munkánkat! A *cikkszám* megkeresésére most nem alkalmas az FKERES( ) függvény, mivel az mindig a kijelölt tábla bal szélső oszlopában keres, és ha a kijelölés emiatt a B oszloptól kezdődne, akkor az A oszlopban lévő értékeket nem tudná visszaadni. Szerencsére az INDEX( )-HOL.VAN( ) függvény együttes mindig működik:

```
=INDEX(
    cikkek!$A$2:$A$12;
    HOL.VAN(számla!A9;
    cikkek!$B$2:$B$12;0)
)
```

- Másoljuk a képletet az egész oszlopra (ezért kellettek a *dollár* jelek)!
- Így persze a kitöltetlen soroknál a #HIÁNYZIK üzenet jelenik meg. Ezen könnyedén javíthatunk, egy HA( ) függvény bevonásával.

A hibaüzenettel egyébként akkor is találkozhatunk, ha a képletben megadott tartomány összevont cellákat tartalmaz. Egérrel történő kijelöléskor ugyanis – a látszat ellenére – nem csak egy sor vagy oszlop kerül a képlet második argumentumába, ami hibát okoz. Javításhoz a tartomány kézzel történő begépelésére van szükség, ahol az összevont celláknak csak azon sora/oszlopa szerepel, ahol az adatok vannak.

---

<sup>56</sup> Reményi Zoltán - Siegler Gábor - Szalayné Tahi Zsuzsanna: Érettségire felkészítő feladatgyűjtemény, 86. oldal, 2. példa (Virágcsokor)

```
=HA(
    A9<>" ";
    INDEX(
        cikkek!$A$2:$A$12;
        HOL.VAN(számla!A9;cikkek!$B$2:$B$12;0)
    );
    " "
)
```

- A *egységár* megkeresése az előbbihez hasonlóan történik, töltsük ki ezt az oszlopot is!
- A *nettó ár* oszlopértékeinek kiszámításához csak egy egyszerű szorzásra van szükség, de most már figyelni kell arra is, hogy a *menyiség* is meg legyen adva. A két feltétel összekapcsolása **ÉS ( )** függvénnyel történik:

```
=HA(
    ÉS(A9<>" ";C9<>" ");
    C9*D9;
    " "
)
```

- Az *ÁFA értékének*, a *bruttó ár*nak a kitöltése az előbbihez hasonlóan, egyszerű szorzással illetve összeadással történik. Különbség csupán az, hogy az **ÉS ( )** függvénynek ezeknél már három argumentuma lesz.
- A 25. sor kitöltése szinte triviális, elég hozzá egy-egy **SZUM ( )** függvény. A megfelelő cellaformátum kiválasztásáról itt se feledkezzünk meg!

A feladat ezidáig igazából gyerekjáték volt, csupán a teljesség kedvéért írtam le a lépéseket. A lényeg még csak most jön: be kell állítanunk a számlakibocsátó által kitöltendő cellákra, hogy milyen értékeket vehetnek fel, és hogy amennyiben érvénytelen adatot adnak meg, akkor mit tegyen az *Excel*. Valójában ez is nagyon egyszerű, de az a tapasztalatom, hogy a diákok (vagy akár tanárok) valamiért kevésbé szokták ismerni ezt a lehetőséget. Pedig a mindennapi életben igencsak hasznos lehet.

- Jelöljük ki mindhárom, cím megadására fenntartott cellát, és válasszuk az adatok fölül **Érvényesítés** gombját!
- Először az **érvényességi feltételt** kell megadnunk. A feladat kérése alapján válasszuk a legördülő listából azt, hogy *Szöveghossz*, majd állítsuk be, hogy *Nagyobb vagy egyenlő*, minimumnak pedig adjuk meg a tizenötöt!

- Ezután kattintsunk át az ablak **hibajelzés** fülére! Itt lehet megadni, hogy miként járjon el a program, ha érvénytelen adatot adtak meg. Most nekünk **megállásra** van szükségünk, hiszen nem fordulhat elő, hogy tizenöt-nél kevesebb karaktert adjanak meg cím gyanánt. Ugyanitt adjuk meg a hibaüzenetet is, másoljuk be a feladatban megadott szöveget! Adhatunk neki címet is, például „Figyelem!”.
- A nevek érvényességének beállításánál hasonlóan járjunk el! Most azonban engedélyezett az „érvénytelen” adat megadása is, hiszen ha ritkán is, de előfordulhatnak nagyon rövid nevek. Amennyiben a **hibajelzés** fülön a **megállás** helyett a **figyelmeztetést** adjuk meg a hibajelzés stílusának, akkor lesz mód a figyelmeztető üzenet elolvasása után megerősíteni a programot abban, hogy direkt adtunk meg ilyen rövid nevet.
- A termékek nevének kiválasztásához könnyedén készíthetünk **legördülő listát**, ugyancsak az **Érvényesítés** ablakban. A feltétel legyen **lista**, a forrást pedig jelöljük ki egérrel!<sup>57</sup> Ettől még lehet a gépelést is választani a terméknév megadásához, de elgépelés esetén az **Excel** alapértelmezett hibaüzenete fog megjelenni, amennyiben nem állítottunk be a **Hibajelzés** fülön egyebet.
- Az **ÁFA** beállítása, mint az előbb, a hibajelzés stílusáról, vagy az üzenetről most sem mondott semmit a feladat, tehát maradhat minden alapértelmezett.
- A **menyiség** esetén fontos, hogy csak egész számot lehessen megadni. Az **érvényesítés** ablakban válasszuk az **egész szám** feltételt, és adjuk meg a minimumot és a maximumot is! A hibajelzés stílusa ismét **megállás** (hogy sehogy se lehessen mást megadni), az üzenetet másoljuk be!
- A dátum értékeinek felülbírálásához válasszuk a szokásos ablakban az **időt**, **jellegnek** pedig azt, hogy **nagyobb vagy egyenlő**. A **kezdő időpont** a kiállítás dátumától függ, ezért kattintsunk az ezt tartalmazó cellára! Ahhoz, hogy továbbmenjünk, emlékezni kell az **Excel** dátum- és időkezelésére: egy napot egy egész szám reprezentál, az adott napon belüli időt pedig az egész szám utáni tizedesjegyek. Mivel negyvennyolc óra az két nap, ezért adjunk a kiállítás dátumához kettőt, **kezdő időpont** gyanánt!

---

<sup>57</sup> Lehetőség van egyébként a listaelemek felsorolására is (pontosvesszővel elválasztva), ami akkor lehet hasznos, ha nem akarunk erre a célra a munkalapon segédcellákat elhelyezni.



- A kiszámolt értékek cellájában ne állítsunk be érvényességi feltételt, hanem csak a **Figyelmeztető üzenet** fülön adjuk meg a szöveget. Ki van pipálva, hogy már a cella kijelölésekor jelenjen meg az üzenet, ezen nem kell változtatni.

Ezzel végeztünk a feladatnak azon részével, ami a cellák tartalmának korlátozásával kapcsolatos. Szeretném megjegyezni, hogy vannak még egyéb módszerek is ebben a témában. Például olykor az is elég lehet, hogy beállítjuk az érvényességi feltételeket, mindenféle üzenet vagy megállítást nélkül. Ezt követően választhatjuk az *Érvényesítés* csoport azon funkcióját is, mely bekarikázza a rossz (érvénytelen) adatokat. Ez a mindennapi életben is hasznos lehet, de el tudnám képzelni versenyfeladatnak is: „*Karikázzuk be a táblázatban a tíznél nagyobb értékeket!*”. Feltételes formázásnál ilyen formátumot nem lehet megadni, igaz, ahhoz, hogy ez a módszer egy ilyen feladat esetén jól működjön, minden változtatást követően rá kellene kattintani a bekarikázásra.

Amiről még nem beszéltem, az a feltétel megadásánál lévő jelölőnégyzet: *üres cellák mellőzése*. Alapértelmezettként ki van pipálva, hiszen ritkán fordul elő, hogy nem akarjuk engedélyezni a cellák – akár ideiglenesen történő – üresen hagyását.

A kis kitérő után térjünk vissza a feladathoz, hiszen még hátra van az *F*, *G*, és *H* rész megoldása!

- Jelöljük ki az egész számlát (*A1:I30*), és válasszuk a helyi menü *Cellaformázás* parancsát! A *Védelem* fülön állítsuk be, hogy a cellák legyenek *zároltak!*

Figyelem: ez még önmagában semmit nem ér, ezzel csupán megadjuk az egyes cellákra, hogy hogyan viselkedjenek majd a **lap védelmének** beállítását követően. A **zárolt** cellákat a lap levédését követően nem lesznek szerkeszthetőek, a **rejtett celláknak** meg nem látni majd a tartalmát (képletét).

- Jelöljük ki a sárga cellákat, és állítsuk vissza, hogy ezek ne legyenek zároltak!

Ez a két lépés gyorsabb volt, mintha kijelöltük volna az összes nem sárga cellát.

- Jelöljük ki a képleteket tartalmazó cellákat, és ezekre állítsuk be azt is, hogy legyenek **rejtettek!**

Mint előbb írtam, ezek a beállítások most még nem érnek semmit.

- Válasszuk a *Korrektúra* fül/ *Lapvédelem* parancsát! Az alapértelmezett beállítások megfelelőek lesznek, jelszót ne adjunk meg!

A valós életben persze nincs sok értelme a jelszó nélküli lapvédelemnek, hiszen így bárki ki tudja kapcsolni azt, és ismét szerkeszthetővé válnak a zárolt cellák, olvashatóvá a rejtett cellák képlete. A feladattal viszont készen vagyunk!

## 10. Vezérlők

A *Fejlesztőeszközök* fül/ *Vezérlők* csoportjában lehetőség van különböző *Windows*-os elemek beszúrására, melyek egyik fő szerepe, hogy segítsék a felhasználókat a fejlesztők által elkészített munkafüzet kitöltésében. Használatukra csak keveseknek van szüksége, ezért ez a fül alapértelmezetten meg sem jelenik. Ha szükségünk van rá, az *Excel Beállítások* menüjében kell megjeleníteni.

Mivel versenyen ez idáig csak *legördülő listák* létrehozására volt szükség, csak ennek elkészítését fogom bemutatni. Ha valakit ennél jobban érdekel a téma, olvasson utána!<sup>58</sup>

### 10.1. Beviteli lista

Ez az űrlap-vezérlőelem lényegében ugyanazt a célt szolgálja, mint amikor cellák érvényesítésénél *lista* feltételt adunk meg. Könnyű is összekeverni, mikor melyiket kéri a feladat, hisz a mintául szolgáló képen hasonlóan néznek ki. Ügyeljünk tehát arra, hogy pontosan azzal készítsük el megoldásunkat, mint ami a mintán szerepel, mert a másikkal történő megoldás nem biztos, hogy teljes pontszámot ér, még ha ugyanúgy működik is!

#### 38. példa: Nemes 2009-2010 3. forduló, 5/E (Holdak)

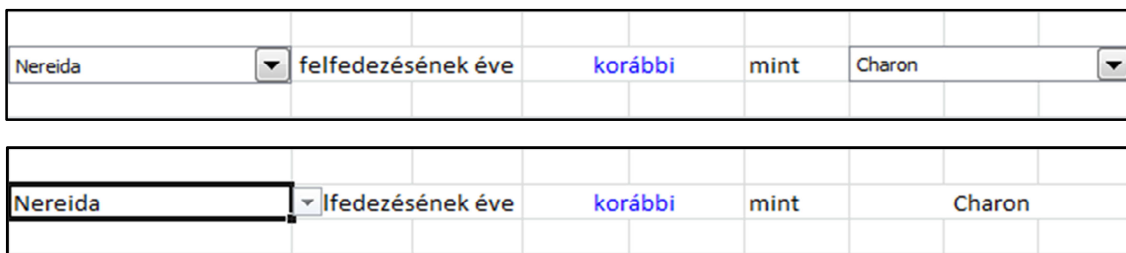
Egy táblázatban szereplő holdak nevének kiválasztására kell listát készíteni, a mintának megfelelően:

- „A két legördülő listából (a „mi” Holdunk kivételével) bármely holdat lehessen választani, akár ugyanazt is.”

Megoldható a feladat vezérlő használata helyett érvényesítéssel is, de mivel a mintán nem az szerepel, vezérlőket kell használnunk, ha teljes pontszámot akarunk kapni.

---

<sup>58</sup> Magyar nyelvű leírás: <http://support.microsoft.com/kb/291073/hu>, angol nyelvű, de az összes vezérlőt bemutató leírás: <http://office.microsoft.com/en-us/excel-help/overview-of-forms-form-controls-and-activex-controls-on-a-worksheet-HA010237663.aspx>.



Legördülő lista a minta szerinti vezérlővel, illetve cellaérvényességgel megoldva. A különbség kicsi, de a rossz módszer kiválasztása pontvesztést okoz.

- Szűrjük be a megfelelő elemet két példányban: *Fejlesztőeszközök* fül/ *Vezérlők/ Beszúrás/ Beviteli lista!* A vezérlő pozíciója és mérete az egérrel alakítható. Érdekes már most *E110*, illetve *K110:M110* fölé rajzolni őket, de később is lesz lehetőség a módosításra.
- A helyi menü/ *Vezérlő formázása* ablakban kell megadnunk a listák működésének mikéntjét. A lényeg a *Vezérlő* fülön van: a **Bemeneti tartomány** legyen  $C8:C99$ , kihagyva így a *C7*-ben lévő „Hold”-at, a **Cellacsatolás** pedig *E110*, valamint az egyesített *K110:M110* tartomány!

A csatolt cellában valójában a **kiválasztott elem bemeneti tartományban elfoglalt helyének sorszáma** kerül, mintha `HOL.VAN()` függvényt használna a vezérlő. A feladat megoldásának folytatásakor ezt a cellát kell használni, figyelembe véve, hogy ott mi szerepel.



A vezérlő, alatta a csatolt cella tartalma.

Így az évszámok különbségét kiszámító képlet ilyen egyszerű lesz:

$$=INDEX(D7:D99;E110) - INDEX(D7:D99;K110)$$

A feladat része még az előbbi képlet cellájának formázása. Ennek menetéről a *Cellaformázás* fejezetben olvashatunk, a 11. példában.

Ha a feladatot érvényességgel végeztük volna, a fenti képlet is bonyolultabb lenne. Azt a megoldást a *adatok\_érvényesítéssel* munkalapon találod.

## **Konklúzió**

Rengeteg téma van, amiről írhattam volna még, sőt, az elkészült fejezeteken is tudnék még bővíteni, legyen szó akár a leírás részletezéséről, akár további példák felsorakoztatásáról. Ugyanakkor nem hinném, hogy bármikor azt tudnám mondani irományomra, hogy „készen van”. Az újabb versenypéldák, a szoftverek változása ugyanis folyamatos újítását, bővítését igényli az anyagnak. Mindenesetre remélem, hogy így is sikerült egy olyan szakdolgozatot készítenem, ami nem csak a polcon fog porosodni, hanem tényleg hasznára lesz mind a kollégák, mind a versenyre önállóan készülő diákok számára.

Ezúton szeretném megköszönni konzulensem, Gudenus László lelkiismeretes munkáját, hogy mindig szakított rám időt, és munkám ellenőrzése mellett hasznos tanácsokkal látott el.

## **Tárgymutató**

### Billentyűparancsok

Minden adat kijelölése ..... 9

Tartomány kijelölése ..... 15

### Hibaüzenetek

#ÉRTÉK!..... 8

#HIÁNYZIK ..... 124

#SZÁM! ..... 18

### Módszerek

Másik munkafüzet tartományának megadása ..... 59

Munkalap másolása ..... 9, 75

Szöveggént tárolt szám számmá alakítása ..... 8

## **Irodalomjegyzék**

- *A Microsoft Office hivatalos honlapja:*  
<http://Office.Microsoft.com/hu-hu/Excel-help/Excel-sugo-es-utmutato-FX010064695.aspx?CTT=97>
- *A Libre Office internetes súgója:*  
[http://help.libreoffice.org/Calc/Welcome\\_to\\_the\\_Calc\\_Help](http://help.libreoffice.org/Calc/Welcome_to_the_Calc_Help)
- *Az OKTV és Nemes Tihamér versenyek honlapja:*  
<http://tehetseg.inf.elte.hu/>