

# Gyümölcsök és Kézilabdások

Rostagni Csaba

A feladatok elvégzése során figyeljen a helyes, hiba- és figyelmeztetés mentes kód írására! A bementekről feltételezheti, hogy a felhasználó létező adatot ad meg, ezt a feladatoknál nem kell vizsgálni, hacsak a feladat külön nem kéri!

Törekedjen arra, hogy a programkód minél általánosabb legyen, **a tömbök módosítása esetén is működőképes legyen!** Feltételezheti, hogy minden tömbben legalább két adat rendelkezésre áll.

A feladatokat pontosan a **feltüntetett néven mentse el!**

A létrehozott **függvények** esetében **határozza meg a bemenet és a kimenet típusát is!**

Mind a két feladatban egy-egy tömböt kell létrehozni. Míg az elsőben egy egyszerű sorszámozott tömb szükséges, a második feladatban egy összetettebb tömböt kell elkészíteni.

Minden fájl elején szerepeljenek az alábbi beállítások:

```
declare(strict_types=1);
error_reporting(E_ALL);
ini_set("display_errors",1);
```

## 1. feladat: Gyümölcsök

\$gyumolcsok

alma	piros	citrom	citrom	narancs	narancs	kiwi	zöld
0	1	2	3	4	5	6	7

- Hozza létre az `adatok.php` fájlt. Készítsen benne egy `$gyumolcsok` nevű tömböt a fenti minta alapján, ahol a gyümölcsök neveit követő elem az adott gyümölcs színe lesz.
- Hozzon létre egy fájlt `gyumolcsok.php` néven. Ez lesz a feladat belépési pontja.
- Kösse össze a két fájlt úgy, hogy a `$gyumolcsok` tömb elérhető legyen a `gyumolcsok.php` fájlból.
- A szkript ellenőrizze, hogy legfeljebb 2 paramétert kapott. Amennyiben többet kapna, úgy jelezze a mintán látható hibaüzenettel, majd lépjen ki 4-es hibakóddal.

```
$> php gyumolcsok.php egy kettő három négy
Túl sok paraméter!
```

5. Amennyiben a felhasználó nem ad meg paramétert, úgy az első sorba írja a minta szerint hány gyümölcs van, majd ezek után jelenítse meg minden gyümölcstről, hogy milyen színű szintén a minta szerint.

```
$> php gyumolcs.php
Gyümölcsök száma: 4
- a(z) alma piros színű
- a(z) citrom citrom színű
- a(z) narancs narancs színű
- a(z) kiwi zöld színű
```

6. Amennyiben a felhasználó egy paramétert ad meg, és annak értéke **megegyezik**, akkor jelenítse meg egymás alatt azoknak a gyümölcsöknek a nevét, melyeknek a neve egyben a színe is!

```
$> php gyumolcs.php megegyezik
citrom
narancs
```

7. Amennyiben a felhasználó egy paramétert ad meg, de az **NEM** “megegyezik”, akkor egy színnek kell lennie! Ennek létezését nem kell ellenőriznie. Jelenítse meg a megadott színű gyümölcsöket egymás után vesszővel és egy szóközzel elválasztva. Figyeljen rá, hogy az utolsó után ne legyen fölösleges vessző!

```
$> php gyumolcs.php piros
alma, eper
```

8. Amennyiben a felhasználó két paramétert ad meg, és az első paraméter értéke **hozzafuz** úgy egy új gyümölcsöt kell felvinni a rendszerbe úgy, hogy a jelenlegi rendszernek megfeleljen.

- A második paraméter az új érték az alábbi formátumban:
  - a gyümölcs neve, amit egy per jel (/) követ, végül a szín következik.
- Az adatok bevitel után jelenítse meg a a tömb tartalmát egymás alatt, az indexekkel együtt a minta szerint!

```
$> php gyumolcs.php hozzafuz "áfonya/kék"
0: alma
1: piros
2: narancs
3: narancs
4: citrom
5: citrom
6: kiwi
7: zöld
8: áfonya
9: kék
```

9. Az elkészített feladatot dockerizálja. Az Dockerfile neve legyen **gyumolcsok.Dockerfile**, míg az image neve legyen **monogram/gyumolcsok**.

```
$> docker run rcs/gyumolcsok piros
alma, eper
```

## 2. feladat: Kézilabdások

A Telekom Veszprém csapat tagjai:

mezsza	nev	poszt	allampolgarsag	magassag
19	Nagy László	jobblövő	magyar	207
15	Tønnesen Kent Robin	jobblövő	norvég	195
63	Ilic Momir	ballövő	szerb/magyar	198

A Balatonfüredi KSE csapat tagjai:

mezsza	nev	poszt	allampolgarsag	magassag
67	Ligetvári Patrik	ballövő	magyar	200

1. Bővítse ki az `adatok.php` fájlt egy `$jatekosok` tömbbel, ami a fenti adatokat tartalmazza megfelelő adatstruktúrában.
2. Hozzon létre egy `kezi.php` nevű fájlt.
3. Kösse össze a két fájlt úgy, hogy a `$jatekosok` változó elérhető legyen az `kezi.php` fájlból.
4. Készítse el az alábbi függvényeket
  - Készítsen egy függvényt `osszmagassag` néven, ami megadja a játékosok összmagasságát!
  - Készítsen egy függvényt `allampolgarsag` néven, megadja hány játékos rendelkezik (legalább részben) a megadott állampolgársággal.
5. A program legfeljebb egy paramétert kaphat, ez alapján futtassa le a megfelelő parancsot
  - Amennyiben ez `osszmagassag`, úgy hívja meg a `osszmagassag` függvényt megfelelően felparaméterezve és írja ki az eredményt!
  - Amennyiben egy paramétert kap feltételezheti, hogy az egy állampolgárság, így azt nem kell ellenőrizni. Ebben az esetben hívja meg az `allampolgarsag` függvényt megfelelően felparaméterezve és írja ki az eredményt!
  - Amennyiben nem kapna paramétert jelenítse meg minden játékos néhány adatát a megadott formátumban!

```
#mezsza{TAB}JÁTEKOS_NEVE{TAB}csapat_neve{TAB}(poszt).
```

A {TAB} helyén tabulátor szerepeljen!

Minta

```
$> php kezi.php 2
3
```

```
$> php kezi.php magyar
3
```

```
$> php kezi.php
#19  NAGY LÁSZLÓ    Telekom Veszprém    (jobblövő)
...
```

6. Az elkészített feladatot dockerizálja. Az Dockerfile neve legyen `kezilabdasok.Dockerfile`, míg az image neve legyen `monogram/kezilabdasok`.

```
$> docker run rcs/kezilabdasok magyar  
3
```