

11

AWK
tömbök, függvények

Tömb - Asszociatív tömb

- Dinamikus indexelés, szöveges indexekkel:
 - nem kell előre megadni a méretet
 - tetszőleges szöveges indexszel használható
 - értékek típusában sem kell egyeznie (tömbön belül lehet szöveg és szám is)
 - ha olyan indexre hivatkozunk, ami nem létezik, akkor 0 vagy "" lesz az eredmény
- `NÉV[INDEX] = ÉRTÉK`
- `delete NÉV[INDEX]` vagy `delete NÉV`

Tömb - feladat

- Számoljuk ki bocsanat.txt-ben lévő szavak előfordulását, majd írjuk ki. Tehát hozzunk létre egy tömböt, amelyet a szavakkal fogunk indexszelni és a hozzá tartozó értéket pedig növeljük mindig ha újabb előfordulást találunk.
- Az üres szó előfordulása nem érdekel minket → töröljük a tömbünkből

Beépített szöveges függvények

- `getline` változó; – beolvas a STDIN-ről
- `tolower(str)`, `toupper()` – kisbetűs/nagybetűs változata az `str`-nek
- `length(str)` – szöveg hossza
- `index(str, substr)` – a `substr`-et keresi a `str`-ben és visszatér a pozíciójával
- `split(str, array, delim)` – a `delim` mentén feldarabolja `str`-t és az `array` tömbbe tárolja el a darabokat
- `substr(str, start_index)` – a `str` `start_index` pozíciójától kezdődő részt adja vissza
- `substr(str, start_index, len)` – a `str` `start_index` pozíciójától kezdődve `len` hosszú részt ad vissza

Feladatok

- Egészítsük ki az előző feladatot az alábbiakkal:
 - hagyja figyelmen kívül a kis/nagy betűket
- Írjuk ki azokat a sorokat amelyekben megtalálható a „nem” szó
 - reguláris kifejezés használatával
 - az index függvény segítségével
- A split.txt-ből határozzuk meg azokat a sorokat, amelyeknél az évszám 2014-nél újabb

Beépített numerikus függvények

- `int(x)` - egészre vágás (simán elhagyjuk a tizedes jegyeket)
- `sqrt(x)` - négyzetgyök
- `exp(x)` - (e^x)
- `log(x)` - természetes alapú logaritmus
- `sin(x)`, `cos(x)` - sin, cos radiánon értelmezve
- `rand()` - 0 és 1 közötti véletlen számot ad

Az intervallum $[0,1)$. Tehát ha pl.: indexet akarunk vele dobni, akkor hozzá kell adnunk 1-et!