

10

AWK programozás
tömbök
beépített 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 10_feladat.txt-ben lévő szavak előfordulását, majd írjuk ki
 - figyeljünk oda arra, hogy a különböző írásjelek miatt egy szó ne szerepeljen többször → szűrjük le az írásjeleket
 - üres szó előfordulása nem érdekel → 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 használata nélkül).
Ötlet: index
- Az előbbi sorokat csak a „nem” szótól kezdve írjuk ki.
- Töltsünk be minden sort egy tömbbe, ahol minden karakter külön tömb-elemen helyezkedik el. Írjuk ki a „nem” szót, ezzel a karaktertömb segítségével!
- Előző feladat egyszerűen substr segítségével

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

Feladat

- Minden sorban kérjünk egy véletlen számot, amivel meghatározunk a sorban egy pozíciót:
 - a véletlen szám a sor hossza és 0 közötti legyen
 - kerekítsük egészre, hogy indexként tudjuk használni
- Írjuk ki a pozíción lévő karaktert