13

AWK
tömbök, függvények

zh

- felépítése
  - két bevezető könnyebb feladat (5-5) pontért (hasonló nehézségű, mint az extra feladatok voltak) – awk scriptek
  - egy nehezebb, összetettebb feladat 10 pontért – bash script
- anyagrész: minden, ami a félév során elhangzott, de főleg az első zh óta elhangzott anyag.
- amit használhatsz (ne bízd el magad, ezek zh-n való lapozgatása nem lesz elég, ahhoz hogy megold a feladatokat):
  - Rodek Lajos gyakorlati jegyzete
  - pub
- amit nem:
  - gyakorlati diasorjaim, honlapomon lévő anyagok
  - Erika jegyzete
  - internet

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 → reguláris kifejezés segítségével adjuk meg az elválasztó jeleket
  - üres szó előfordulása nem érdekel → töröljük a tömbünkből

Beépített szöveges függvények

- getline valtozo; – 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) az index függvény segítségével.
- Töltsünk be minden sort egy tömbbe, ahol minden karakter külön tömb-elemen helyezkedik el. A „nem” szótól kezdve írjuk ki a sorokat.
- 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

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

Feladat

- Minden sorban kérjünk egy véletlen számot, amivel meghatározunk a sorban egy pozíciót:
  - a véletlen szám a sor hossza és 1 közötti legyen
  - kerekítsük egészre, hogy indexként tudjuk használni
- Írjuk ki a pozíción lévő karaktert