

Szkriptnyelvek

- 1. minta zárthelyi számonkérés -

1. feladat - ZH. Pontszámok (1 pont)

Demonstrátor Úr kijavította a nem létező másik csoportjának első zárthelyi számonkérését és a Hallgatók pontjait rögzítette egy *pontok.txt* nevű szöveges állományban. A csoport kifejezetten magas létszámmal rendelkezik, ezért kézzel szinte lehetetlen lenne megszámolni, ki hanyast kapott. Demonstrátor Úr megkéri a Hallgatókat, hogy készítsenek egy statisztikát, melyet szkript segítségével állítanak elő.

A szöveges állomány soronként egy sorszámot tartalmaz, mely egy Hallgatót indikálna és mellé egy pontot tartalmaz, mely az adott sorszámú Hallgató ZH.-n elért pontja lesz. Az első ZH.-n 4 pontot lehetett szerezni, 4 feladat volt és a feladatok 1-1-1-1 pontot értek. Az elégséges jegy megszerzésére 2 ponttól volt lehetőség, majd félpontossával növekedtek (azaz 2.5 - hármas, 3.0 - négyes, 3.5 - ötös) a jegyek az ötös, azaz a maximum 4 pontig. A feladat az lesz, hogy határozzuk meg, hogy hány Hallgató kapott egyest, kettest, hármat, négyest és ötöst a zárthelyin és a kurzus átlagot két tizedesjegy pontossággal (a kapott jegyek alapján). A feladatot kötelező a *szótár (dictionary)* adatszerkezet használatával megoldani!

A program a következő kimenetet produkálja, mely egy szótár alapján készült:

```
$ python3 ./feladat_1.py
Egyes - 82
Kettes - 45
Hármas - 57
Négyes - 49
Ötös - 30
Kurzus átlag: 2.62
```

Figyelem! A fenti kimenet nem a helyes eredményeket tartalmazza!

2. feladat - Hangrendek (1 pont)

Demonstrátor Úr kapott egy feladatot, hogy döntse el egy szöveges állományban lévő sztringekről, hogy mély, magas vagy vegyes hangrendűek, de Ő nem képes erre, szóval inkább kiadja feladatnak a Hallgatóknak.

Adott egy szöveges állomány (*words.txt*) mely soronként egyetlen egy angol szót tartalmaz. A feladat meghatározni, hogy hány mély, magas és vegyes hangrendű szó található a szöveges állományban. Demonstrátor Úr tudja a következő nyelvtani szabályokat:

Mély hangrendű magánhangzók: a, á, o, ó, u, ú.

Magas hangrendű magánhangzók: e, é, i, í, ö, ő, ü, ú.

Vegyes hangrendűnek tekintünk egy szót, ha mindkét típusú magánhangzó megtalálható benne, ellenkező esetben, ha csak az egyik típusú magánhangzó található meg a szóban, akkor olyan hangrendűnek is ítéljük a szót.

Természetesen mivel angol szavakról van szó, ezért csak az angol ábécé betűit kell figyelni.

Sajnos az állományban vannak olyan sztringek, amik véletlen kerültek bele az állományba, pl.: 3.14. Az olyan sztringeket, amikről nem tudjuk eldönteni, hogy milyen hangrendűek (mert pl.: nem tartalmaznak magánhangzókat) azokat figyelmen kívül hagyjuk.

A program egy szöveges állományba írja ki a kimenetet, melynek legyen a neve *datas.txt* és a következő formátumban tartalmazza az adatokat:

Magas hangrendű szavak - 127

Mély hangrendű szavak - 234

Vegyes hangrendű szavak - 533

Figyelem! A fenti példában nem helyes értékek szerepelnek.

3. feladat - Karakter számlálás (1 pont)

Készítsünk egy függvényt, mely valamilyen sztring paraméter alapján megszámlolja, hogy egy adott karakter, hányszor fordul elő a sztringben. A függvényt a következő módon akarjuk használni:

```
eredmeny = karakter_szamlalo(sztring)
```

Az “eredmeny” egy szótár legyen, tehát a karakter_szamlalo() függvénynek egy szótárat kell visszatértenie, melyben kulcsként az adott karakter és értékként az előfordulásainak száma fog állni.

A sztring paramétert a függvénynek parancssori argumentummal adjuk meg. A program egyetlen egy parancssori argumentumot fogadjon el, máskülönben hibaüzenettel termináljon.

A program a következő kimenetet produkálja pl.:

```
$ python3 ./feladat_2.py Cica
C - 1
i - 1
c - 1
a - 1
```

```
$ python3 ./feladat_2.py Cica cica
Hiba! A program csak egyetlen egy parancssori argumentumot fogad el!
```

```
$ python3 ./feladat_2.py 3.14
3 - 1
. - 1
1 - 1
4 - 1
```

```
$ python3 ./feladat_2.py
Hiba! A programnak kötelező megadni egy parancssori argumentumot!
```

4. feladat - Hamis random számok (1 pont)

Demonstrátor Úr még nem tanította meg a Hallgatókat véletlenszámokat generálni. A Hallgatókat a magas fizetést kínáló munkahelyükön megkérlik, hogy aznap éjjelig küldjenek 10 darab egész véletlenszámot főnöküknek. A főnökök nem tudnak algoritmizálni, ezért nem fog feltűnni nekik, hogyha valójában nem szkript által generált véletlenszámokat küldenek, hanem csak pár nem egybefüggő számot. Ezért megkérlik Demonstrátor Urat, hogy segítsen ebben. A következő ötletet ajánlja:

Kérjenek be a felhasználótól egy darab parancssori argumentumot, mely egy 10 karakter hosszúságú sztring lesz, ellenkező esetben a program hibaüzenettel terminál. Vegyék a kapott sztring összes karakterének ASCII kódját és tárolják el azokat egy **listában**. A kapott lista elemein végezzék el az alábbiakat:

- Ha a kapott ASCII kód páros, vegyék annak másfélszeresét egy új listába, **list comprehension** segítségével!
- Ha a kapott ASCII kód páratlan, vegyék annak -2-szeresét egy új listába, **list comprehension** segítségével!

A programnak egy darab kimenetet kell produkálnia, mely a páros és páratlan lista **egybefűzve** lesz, és a hamis véletlen számokat fogja tartalmazni:

```
$ python3 ./feladat_4.py Fájlkezelő  
[105, 159, 162, 183, 162, -450, -214, -202, -202, -674]
```

```
$ python3 ./feladat_4.py Cica  
Hiba! A megadott parancssori argumentum túl rövid!
```

```
$ python3 ./feladat_4.py CicaCicaCicaCicaCica  
Hiba! A megadott parancssori argumentum túl hosszú!
```

```
$ python3 ./feladat_4.py  
Hiba! Meg kell adni egy(!) darab parancssori argumentumot a működéshez!
```

```
$ python3 ./feladat_4.py Cicacicaci Cicacicaci  
Hiba! Meg kell adni egy(!) darab parancssori argumentumot a működéshez!
```