

Feladatok

Az első feladat kivételével mindenhol Python függvényt kell írni. A függvényeket teszteljük 2-3 bemenettel. A teljes zh 20 pontos, minden feladat maximum 5 pontot ér, részfeladatok sikeres megoldására is pont kapható.

1. Készítsünk egy dictionary-t (Python dict) ami két kulcs értékkel rendelkezik: „Névkonvenciók”, „Kódformázás”. Mindkét kulcshoz tartozó érték legyen egy lista (Python list). Idézzük fel, mit tanultunk a PEP8-as szabályokról. A „Névkonvenciók”-hoz kapcsolódóan írjunk legalább két szabályt, míg a „Kódformázás” listához adjunk hozzá 3 string-et, amelyek mindegyike egy kódformázással kapcsolatos szabályt említ meg. (Másképpen fogalmazva a „Névkonvenciók”-oz legalább 2 szabályt, a „Kódformázás”-hoz kapcsolódóan legalább 3 szabályt kell említeni.)

Az így kapott struktúrán iteráljunk végig, és írjuk ki a tartalmát az alábbi formában: „PEP 8 szabály [kulcs]: [lista elem]”.

Total: 5 pont. A struktúra létrehozása, iterálás és kiíratás 3 pontot ér, a PEP8 szabályok ismerete 2 pontot.

2. Készítsünk egy függvényt, amely pozitív egész számok egy listáját kapja bemenetként. Számoljuk ki a kapott páros számok összegének és a kapott páratlan számok összegének hányadosát. A kapott összegekhez adjunk hozzá 1-et.

Így, ha nincs páros szám a listában, a páros számok összege 1. Hasonlóan a páratlan számok esetén, ha nincs páratlan szám a listában, akkor az összeg 1. A függvény kimenete a páros és páratlan számok összegének hányadosa. (Páros összeg / páratlan összeg.)

Total: 5 pont.

3. Írjunk egy függvényt, ami egy mondatot kap bemenetként. (Tehát a bemenet egy darab string.) A szavakat szóközzel választjuk el egymástól. Gyűjtsük ki a 4 karakter vagy annál rövidebb szavakat egy listába. Az így kapott lista legyen a függvény egyik kimenete. Legyen egy második kimenet is, ami a kapott lista elemeit nagybetűkre transzformálva összefűzi a „&” jellel.

Példa:

```
sentence = "Ez egy egyszerű példa a feladathoz"
```

```
1. Várt kimenet: [„Ez”, „egy”, „a”]
```

```
2. Várt kimenet: EZ&EGY&A
```

Total: 5 pont. Az 1. kimenet helyessége 3 pontot ér, ha a második kimenet is korrekt, akkor 2 pont kapható.

4. Írjunk egy függvényt, ami két listát vár bemenetként, a listákban egész számok szerepelnek. A függvény kimenete legyen azoknak az egész számoknak a halmaza, amik mindkét listában szerepelnek, a kimenetben ne legyenek ismétlődő számok. Amennyiben a két lista hosszúsága azonos, írjuk a két lista tartalmát egy dictionary-be és az legyen a függvény kimenete. (Ez utóbbi esetben a sorrend ne változzon, az első lista elemei legyenek a kulcsok.)

Azonos hosszúságú listák esetén problémát jelent, hogy ha van ismétlődés az első listában, akkor információ vesztes lehetséges a dictionary-be rendezés során. Például:

```
list1 = [1, 2, 3, 3, 5]
```

```
list2 = [3, 4, 5, 6, 7]
```

```
kimenet: {1: 3, 2: 4, 3: 6, 5: 7} # a 3-as ismétlődik
```

Találjunk valami megoldást, arra, hogy megtartsunk minden értéket. (Másképpen fogalmazva a dictionary value-jai között megmaradjon a példában az 5-ös érték.)

Total: 5 pont. 2 pont kapható az első részfeladatra (nem azonos hosszúságú listák), 2 pont kapható a második részfeladatra (azonos hosszúságú listák). 1 pont kapható az azonos hosszúságú listás eset javítására.