Ki lakik odalent? (6 pont)

- Írj egy `mocking_spongebob` nevű függvényt, amely egy adatot olvas be a felhasználótól, billentyűzetről.
- A függvény alakítsa át a felhasználótól kapott szöveget úgy, hogy a szövegben VÉletLEnszErŰeN VÁLtAkoZnAK a kis- és nagybetűk!
    - Pythonban véletlenszámot a random modul (import random) egyik függvényével tudsz létrehozni. Ennek neve: random.randint().
    - Paramétere egy alsó és egy felső határ; random.randint(0, 1) véletlenszerűen 0-t vagy 1-et ad vissza.
- A visszatérési érték az átalakított szöveg legyen!

Példa:

```
Input: 'Azért járunk az egyetemre, hogy tanuljunk.'
Return: 'azÉrt jáRuNk AZ EgYetEMRe, hogy tAnUljUNk.'
```

Leghosszabb szó (6 pont)

- Írj egy `leghosszabb_szo` nevű függvényt, amely egy szöveget kap paraméterül (a szöveg szóközzel elválasztott szavakat tartalmaz)!
- A függvény térjen vissza a szövegben található leghosszabb szóval!
- Amennyiben több szó is ugyanolyan hosszú, akkor az algoritmus a szövegben legkorábban előforduló szót adja vissza!
- Ha egy írásjel szerepel a szövegben, akkor az számítson bele a mellette lévő szó hosszába!

- Ha a paraméterül kapott szöveg egy üres string, akkor a visszatérési érték a HIBA! szöveg legyen!

Példa:

```
Input: 'Szia uram! Jeles zárthelyi dolgozat érdekel?'
Return: 'zárthelyi'

Input: ''
Return: 'HIBA!'
```

Borospince (30 pont)

Egy borospincében szekrények vannak, amelyek polcain borokat tárolnak. Készítsd el a Bor és Szekreny osztályokat a feladatleírás alapján! A feladatok megoldását egyetlen Python szkriptbe készítsd el!

1. A Bor osztály (11 pont)

Hozz létre egy Bor osztályt, amely a __fajta, __evjarat és __alkoholtartalom adattagokkal rendelkezik!

- Az osztály konstruktora a bor fajtáját, évjáratát és alkoholtartalmát várja paraméterben (ebben a sorrendben)!
  - Az alkoholtartalom paraméter értékét ne legyen kötelező megadni, alapértéke legyen 12.5!
  - Inicializáld a __fajta, __evjarat és __alkoholtartalom adattagokat a paraméterek alapján! **(3 pont)**
- Írj get és set property-t a __fajta adattaghoz, fajta néven! A getter adja vissza az adattag értékét, a setter pedig állítsa be azt a paraméterben kapott értékre! **(2 pont)**
- Írj get és set property-t az __evjarat adattaghoz, evjarat néven! A getter adja vissza az adattag értékét, a setter pedig állítsa be azt a paraméterben kapott értékre! **(2 pont)**
- Írj get és set property-t az __alkoholtartalom adattaghoz is, alkoholtartalom néven!
  - A setterben kezeld le, hogy csak 0 és 100 közötti valós szám lehessen az adattag értéke!
  - Nem megfelelő típusú vagy értékű paraméter esetén írd ki a konzolra: Nem megfelelo alkoholtartalom! **(1 pont)**
- Írd át az osztály konstruktorát úgy, hogy az alkoholtartalom értéke itt is ellenőrizve legyen! **(1 pont)**
- Definiáld felül az osztályban az objektum szöveggé alakításáért felelő metódust úgy, hogy az a következő szöveggel térjen vissza: {__fajta} (evjarat: {__evjarat}), melynek alkoholtartalma: {__alkoholtartalom}% (a kapcsos zárójelek helyére értelemszerűen a megfelelő értékek legyenek behelyettesítve)! **(2 pont)**

2. A Szekreny osztály (19 pont)

Hozz létre egy Szekreny osztályt, amelynek egyetlen adattagja egy borok nevű lista! Ebben a listában tároljuk a szekrény polcain lévő borokat.

- Az osztály konstruktora nem vár paramétert, és a borok adattagot egy üres listával inicializálja! **(2 pont)**
- Írj egy get_bor metódust, amely paraméterül egy n egész számot kap! A metódus térjen vissza a borok lista n-edik indexű elemével!
    - Az egyszerűség kedvéért csak a nemnegatív indexeket kezeljük, így, ha a paraméterben érkező n-érték negatív vagy nagyobb, mint a lista utolsó elemének indexe, akkor írja ki a konzolra: Nem letezo index! **(3 pont)**
- Írj egy atlag_alkoholtartalom metódust, amely térjen vissza a szekrényen lévő borok (borok adattag) alkoholtartalmának átlagával! Amennyiben nincs egyetlen bor sem a szekrényen, akkor írja ki a konzolra: Ures a szekreny! szöveggel inicializálj! **(3 pont)**
- Készíts egy paraméter nélküli statisztika metódust! A metódus számolja össze, hogy a különféle borfajtákból mennyi található a szekrényen, és az eredményt adja vissza egy dictionary-ben! **(5 pont)**
    - Ha a szekrényen nincs egyetlen bor sem, akkor a visszatérési érték egy üres dictionary.
    - Ha a szekrényen van bor, akkor számold össze az egyes borfajták darabszámát! A borfajták nevében ne különböztesd meg a kis- és nagybetűket (tehát pl. aszu és Aszu ugyanaz a fajtanév)!
- Írj egy megisszak metódust, amely egy Bor típusú objektumot vár paraméterben! A metódus törölje az adott bort a borok listából, amennyiben az szerepel a listában! Feltehetjük, hogy minden bor objektum egyszer fordul elő a szekrényen.
    - Ha a bor nem szerepel a listában, akkor írja ki a konzolra: a Bor nem talalhato!
    - Ha a metódus nem Bor típusú paramétert kap, akkor írja ki a konzolra: Nem bor! **(3 pont)**
- Definiáld felül az objektum szöveggé alakításáért felelő metódust az osztályban!
    - Amennyiben a borok lista üres, akkor a metódus térjen vissza az Ez egy ures szekreny. szöveggel!
    - Ellenkező esetben a metódus térjen vissza egy olyan szöveggel, amely 1 vesszővel és 1 szóközzel elválasztva tartalmazza a szekrényen lévő borfajták nevét és az azokból meglévő mennyiséget, {darab} {fajta} formátumban! Ehhez használd fel a statisztika metódus visszatérési értékét!
        - Egy dictionary adatszerkezet kulcsain és értékein az items() metódus segítségével tudsz végig iterálni:

```
for x, y in thisdict.items():
  print(x, y)
```

    - A szöveg végén ne legyen se vessző, se szóköz! **(3 pont)**