

1. Az "Életkor" program: Készítsen egy Python programot **eletkor.py** néven, amely bekéri a felhasználótól az életkorát, majd kiírja, hogy milyen korosztálynak tartozik. A program üzeneteinek megfogalmazásában kövesse az alábbi példát!

Példa kimenet:

```
Kérem az életkorát: -2
Az életkor nem lehet negatív szám!

Kérem az életkorát: 16
Ön még kiskorú.

Kérem az életkorát: 32
Ön felnőtt korú.

Kérem az életkorát: 68
Ön idős korú.
```

Pontozás – minden teljesülő feltétel egy-egy pontot ér:

1. Létrehoz egy programot **eletkor.py** néven, a program hibaüzenet nélkül lefut.
2. Bekéri a felhasználótól az életkort és tárolja.
3. A bekért számot szám típusúvá alakítja.
4. Elágazást használ a különböző esetek kezelésére.
5. Helyesen állapítja meg a mintában megadott teszt életkorokat.
6. Helyesen kezeli, ha 0-nál kisebb számot ad meg a felhasználó.
7. Helyesen írja az eredményeket.
8. A kiírt üzenetek helyesek (pl. nincs benne elgépelés, helyesen jelennek meg a szóközők.)

2. Írj egy programot **maganhangzok.py** néven, amely meghatározza egy adott szövegben az összes magánhangzót! A program használjon függvényt a magánhangzók meghatározására.

Példa kimenet:

```
Adjon meg egy szöveget: Árvíztűrő tükörfúrógép
A szövegben 9 magánhangzó található.
```

Pontozás – minden teljesülő feltétel egy-egy pontot ér:

1. Létrehoz egy programot **maganhangzok.py** néven, a program hibaüzenet nélkül lefut.
2. Bekéri a szöveget és tárolja.
3. Függvényt hoz létre a magánhangzók számának eldöntésére.
4. Létrehozza a magánhangzók listáját.
5. Létrehoz egy változót a darabszámok tárolására és beállítja 0 kezdő értékre.
6. Ciklust szervez a bemenetként kapott szöveg bejárására.
7. Megvizsgálja, hogy az adott betű magánhangzó-e
8. A szövegbetűire alkalmazza a lower metódust.
9. Ha magánhangzó akkor növeli a darab számláló változó értékét eggyel.

10. A függvény a darabszámmal tér vissza.
11. Helyesen kér be és tárol a felhasználótól szöveget.
12. Helyesen hívja meg a függvényét és tárolja az eredményt.
13. A mintának megfelelően kiírja az eredményt.
14. A kiírt üzenetek helyesek (pl. nincs benne elgépelés, helyesen jelennek meg a szóközök).

3. Készíts programot **allatkert.py** néven, amelyben hozz létre egy osztályt **Allatkert** néven. Az Allatkert attribútumai legyenek: **novenyek**, **allatok**. Legyen két metódusa. Az egyik írja ki a konzolra a növények és a állatok számát minta szerint. A másik metódus gondoskodjon arról, hogy az előző eredményeket írja ki egy allatkert.txt nevű fájlba. Példányosítson egy **allatkert** nevű példányt, majd hívja meg rajta külön-külön az osztály két metódusát.

Példa kimenet:

```
Növények: 45
Állatok: 67
```

Pontozás – minden teljesülő feltétel egy-egy pontot ér:

1. Létrehoz egy programot allatkert.py néven, a program hibaüzenet nélkül lefut.
2. Létrehoz egy Allatkert nevű osztályt.
3. Az osztály nevét nagybetűvel írja.
4. Szintaktikailag helyesen létrehoz egy inicializáló metódust.
5. Bemeneti paramétereket ad meg **noveny** és **allat** néven.
6. A **noveny** paramétert átadja a példánynak.
7. Az **allat** paramétert átadja a példánynak.
8. Szintaktikailag helyesen létrehoz egy kiíró metódust.
9. A metódusban kiírja a példány **noveny** attribútumát.
10. A metódusban kiírja a példány **allat** attribútumát.
11. Szintaktikailag helyesen létrehoz egy kiírás fájlba metódust.
12. A metódusban context managert használ.
13. Helyesen határozza meg a célfájlt, valamint a kiírást, valamint az utf-8 kódolást.
14. Szintaktikailag helyesen írja fájlba a **noveny** és a **allat** attribútumokat.
15. Helyesen példányosít egy osztályt.
16. Helyesen használja az osztályon a konzolra író metódust.
17. Helyesen használja az osztályon a fájlba író metódust.
18. A kiírt üzenetek helyesek (p. nincs benne elgépelés, helyesen jelennek meg a szóközök.)