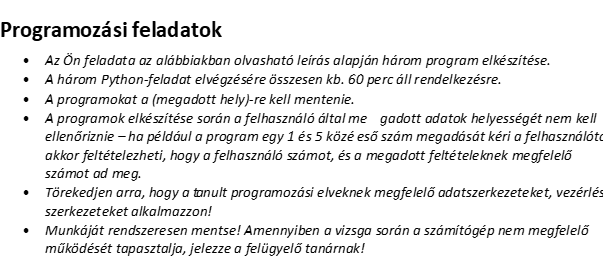
# Dolgozat



A projektet mentse: neve\_csoportja helyre. A modulokat a főprogramból (main.py) hívja meg eljáráshívásokkal.

## Első feladatt -8 pont

Feladat szövege Írjon modult jelszo.py néven, amely azt vizsgálja, hogy egy felhasználó helyesen adja-e meg a jelszavát! A program addig kérdezi újra a felhasználónév-jelszó párost, amíg a felhasználó mindkettőt hibátlanul meg nem adja.A program egyetlen felhasználó (bori99) jelszavát (Szivecske>3) ismeri, csak ezt a párost fogadja el helyesként. Mind a sikertelen, mind a sikeres bejelentkezési kísérletekről üzenetet ír a képernyőre. A program üzeneteinek megfogalmazásában kövesse az alábbi példát! Azokat a részeket, amiket a felhasználó gépel be, a mintában vastagított és döntött betűkkel emeltük ki.

Első feladat:

Adja meg a felhasználónevét! **bori99**

Adja meg a jelszavát! Szivecske<3

Belépés engedélyezve.

Első feladat:

Adja meg a felhasználónevét! Bagaméri

Adja meg a jelszavát! vanília fagylalt

Belépés megtagadva.

Első feladat:

Adja meg a felhasználónevét! bori99

Adja meg a jelszavát! hibásjelszó

Belépés megtagadva.

Első feladat:

Adja meg a felhasználónevét! hibásfelhasználó

Adja meg a jelszavát! Szivecske<3

Belépés megtagadva.

Első feladat:

Adja meg a felhasználónevét! bori99

Adja meg a jelszavát! Szivecske<3

Belépés engedélyezve.

### Pontozás – minden teljesülő feltétel egy-egy pontot ér

1. 1. Létrehoz modult jelszo.py néven, a program hibaüzenet nélkül lefut.
2. 2. Bekéri a felhasználónevet és tárolja.
3. 3. Bekéri a jelszót és tárolja.
4. 4. Helyesen állapítja meg és írja ki, amikor be kell engedni a felhasználót.
5. 5. Helyesen állapítja meg és írja ki, amikor meg kell tagadni a belépést.
6. 6. Helyesen működő ciklust szervez az ismétlődő tevékenység elvégzésére.
7. 7. Kilép a ciklusból, ha a felhasználónév és a jelszó is helyes volt.
8. 8. A kiírt üzenetek helyesek (pl.: nincs benne elgépelés, helyesen jelennek meg a szóközök).

## Második feladattípus -14 pont

A program egy listát vizsgál. Egy függvény segítségével megszámolja, hogy a listában hány darab héttel osztható szám található. Írjon modult **oszthato.py** néven!

Írja meg azt a **Ottel\_oszthato** nevű függvényt, amely megszámolja egy adott listában hány darab héttel osztható szám van! A függvény paramétere lista legyen, a visszatérési értéke a listában szereplő héttel osztható számok darabszáma legyen!

Hozzon létre egy **szamok** listát, melyet töltsön fel 50 darab 1 és 100 közé eső véletlen számokkal! Írja ki a lista elemeit!

Hívja meg a programban a megírt függvényt és írja ki, hogy hány darab héttel osztható szám van a listában!

A lista elemei:

72, 40, 8, 36, 29, 51, 38, 34, 89, 70, 77, 3, 37, 5, 47, 96, 41, 83, 31, 62, 75, 66, 9, 70, 61, 54, 57, 47, 55, 37, 9, 42, 5, 20, 49, 73, 70, 71, 27, 31, 82, 50, 70, 8, 81, 84, 33, 10, 37, 16

A listában  8  darab öttel osztható szám van.

### Pontozás – minden teljesülő feltétel egy-egy pontot ér

1. Létrehoz programot **oszthato.py** néven, a program hibaüzenet nélkül lefut.
2. Helyesen írja meg a **otte\_Oszthato()** függvényt.
3. Helyes paraméterrel írja meg a függvényt.
4. Helyesen adja meg a visszatérési értékeket.
5. Helyesen írja meg a függvény törzsét.
6. Létrehozza a **szamok** listát.
7. Helyesen állít elő egy véletlen számot.
8. Hozzáfűz egy véletlen számot a listához.
9. Ciklust szervez a véletlen számok listába írására.
10. Minden véletlen számot hozzáfűz a listához.
11. Helyesen írja ki a lista elemeit.
12. Helyesen hívja a **ottel\_oszthato** () függvényt.
13. A **ottel\_oszthato** () függvény hívásával helyesen írja ki a héttel osztható számok számát.
14. A kiírt üzenetek helyesek (pl.: nincs benne elgépelés, helyesen jelennek meg a szóközök, mondatvégi írásjelek).

Harmadik feladat – 18 pont

## Harmadik feladat 18 pont

Készítsen programot **autom.py** néven. Olvassa be **auto.txt** fájlból az auto adatait, tárolja **Auto** osztály típusú **kocsi** nevű listában a példányokat.

Készítsen metódust:

* **tabla** néven, amely a paraméterként kapott autó márkájának hosszát adja vissza a konzolra.
* **kiiras** metódust, amely kiírja **kiir.txt** fájlba az autók márkáját és korát az alábbi formában.

Készíts függvényt:

* **flotta** néven, amely kiírja hány autónk van.
* **ertekes** néven, ami kiírja a legfiatalabb autó nevét, és gyártási dátumát.

Minta:

III/Tabla:

Skoda Fabia: 11 hosszú.

III/Flotta:

Autók száma: 7.

III/Értékes

A legfiatalabb autó: Seat Ibiza(2011))

ki.txt

Skoda Fabia: 11 hosszú.

### Pontozás – minden teljesülő feltétel egy-egy pontot ér

1. Létrehoz programot autom.py néven, a program hibaüzenet nélkül lefut.
2. Listát hoz létre kocsi néven az autók Auto osztályú objektum tárolására.
3. A fejlécet nem olvassa be.
4. Beolvas egy adatsort.
5. Beolvas egy sornyi adatot a fájlból.
6. A beolvasott adatokból Auto osztályú objektumot hoz létre.
7. Egy Auto osztályú objektumot elhelyez a létrehozott kocsi adatszerkezetben.
8. Minden beolvasott alapján három objektumot hoz létre.
9. Minden objektumot elhelyezi az adatszerkezetben.
10. Az adatszerkezetben tárolt objektumok alapján megjelenít az objektumot a Név - kor formában, a konzolon.
11. A tabla() metódus megfelelően számolja ki az autó nevének hosszát..
12. A tabla() metódusnak van visszatérési értéke, és megfelelő.
13. A kiir() metódus létezik. A ki.txt kiir() metódus minden adatsort kiír megfelelően.
14. Létezik flotta() függvény és kiírja az autók számát.
15. Létezik ertekes() néven függvény.
16. Megkeresi a legöregebb autót nevét.
17. Megkeresi a legöregebb autót gyártási dátumát.
18. A kiírt üzenetek helyesek (pl.: nincs benne elgépelés, helyesen jelennek meg a szóközök, jók az írásjelek).