Programozási feladatok

* Az Ön feladata az alábbiakban olvasható leírás alapján három program elkészítése.
* Mind a három elkészült programot a saját mappáján belül mentse a Sajátnév\_python mappába, ahol Sajátnév helyére értelemszerűen a saját nevét írja! A mappát természetesen hozza létre az Asztalon!  
  **FONTOS: a vizsga végeztével másolja fel a mappáját a MEGOLDASOK meghajtóra!**
* A feladatok megoldásához szükséges fájlokat az  
  **FORRASOK** meghajtón találja. **A SZÜKSÉGES FÁJLOKAT MÁSOLJA ÁT A SAJÁT MAPPÁJÁBA**
* A programok elkészítése során a felhasználó által megadott adatok helyességét nem kell ellenőriznie! Feltételezheti, hogy ha például a program egy 1 és 5 közé eső szám megadását kéri a felhasználótól, akkor a felhasználó számot, és a megadott feltételeknek megfelelő számot ad meg.
* Törekedjen arra, hogy a tanult programozási elveknek megfelelő adatszerkezeteket, vezérlési szerkezeteket alkalmazzon!
* Figyeljen a változó nevekre! A változó neve beszédes legyen a benne található érték alapján nevezze el!
* **Minden program indulásakor a konzol előzményeit törölje**!

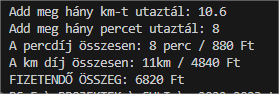
1. **feladat - 8 pont**

Fájl neve: 8\_pontos.py

Ebben a feladatban, egy taxiban való utazás költségszámító kalkulátorát kell elkészítenie.

1. Hozzon létre három darab változót, amelynek fix értéket ad. Az első változó az alapdíj legyen, értéke: 1100. A második változó a kilométer díj, értéke: 440, a harmadik pedig a percdíj, értéke: 110.
2. Kérje be a felhasználótól hány kilométert utazott. Figyeljen, mert a bekért adat lebegőpontos szám is lehet. A beírt adatot kerekítse egész számra.
3. Kérje be a felhasználótól, hogy mennyi percet utazott (csak egész szám lehet)
4. Számolja ki a meglévő adatok alapján az alábbi értékeket:  
   *perc ár = percek száma x percdíj   
   kilométer ár = megtett kilométer x kilométer díj   
   utazás díja = alap díj + perc ár + kilométer ár*
5. A kiszámolt értékeket a mintában látható módon írja ki a képernyőre

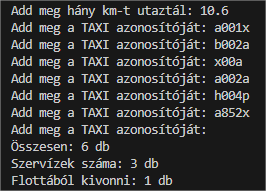
Az alábbi mintában a kód egy futási lehetőségét láthatja:



1. **feladat – 14 pontos**

Fájl neve: 14\_pontos.py   
  
Ebben a feladatban egy TAXI flottaállományáról kell adatokat rögzítenie és feldolgoznia.

* + - 1. Hozzon létre egy függvényt *taxiAzonositokRogzitese* néven, amelynek nincs bemenő paramétere.
      2. A létrehozott függvény térjen vissza egy listával, a listába folyamatosan rögzítse a felhasználótól bekért taxi azonosítókat, amíg üres értéket nem kap.
      3. FONTOS a beírt értéket, csupa nagybetűkkel tárolja le.
      4. Hozzon létre egy metódust *taxiElemzes* néven, legyen egy bemeneti paramétere, amelynek típusa az előző függvény visszatérési értékével egyezik meg.
      5. A metódus a minta alapján írja ki a képernyőre:   
         - Hány darab taxi lett rögzítve   
         - Hány darab taxi szorul szervizre   
         - Hány darab taxit kell kivonni a flottából
      6. Akkor kell szervizelni egy taxit ha „A” betűvel kezdődik.
      7. Akkor kell kivonni a flottából, ha „X” betűre végződik és megtalálható benne a „0”.
      8. A függvényeket használja is fel.

Az alábbi mintában a kód egy futási lehetőségét láthatja:   


1. **feladat – 18 pontos**

Fájl neve: 18\_pontos.py   
Ebben a feladatban a taxik fuvarozását kell feldolgoznia OOP felhasználva.

* + - 1. Írjon egy osztályt **taxiFuvar** néven. Hozzon létre egy konstruktort, amely bemenő paraméterként az alábbi értékeket kapja: *azonosito, utasok száma, km óra állás.* A bemenő paramétereken kívül, hozzon létre még két darab osztályváltozót a konstruktorban. Zárójelben a beállítandó értékét adtuk meg:   
         - kilométer díj (440)   
         - Szervízelés („NEM”)
      2. Az osztálynak legyen egy **fuvarozas** nevű függvénye, amely egy *megtettUt* nevű paramétert vár. A függvény a km óra állást növelje a megtett úttal, valamit az útdíj értékét állítsa be a megtett út és a kilométer díj szorzatára.   
         *útdíj = megtett út x kilométer díj*
      3. Az osztálynak legyen egy **szamlaNyomtatas** nevű függvénye, amely nem vár paramétert. A függvény állítson be egy osztályváltozót *szamlaSzam* néven, melynek értékét egy véletlen számra állítsa be 100-999 intervallumban. Valamint ha az *utasok száma* kettőnél nagyobb, akkor az *útdíj értékét* 5%-al emelje meg.
      4. Az osztálynak legyen egy **szervizCheck** nevű függvénye is, amely nem vár paramétert. A függvény állítsa be a *szervizelés* osztályváltozó értékét „IGEN” -re ha a km óra állás nagyobb mint 10000.
      5. Készítsen egy *fuvarok* nevű, üres, lista típusú változót.
      6. A taxiFuvarok.txt fájlban az alábbi adatokat találja soronként, „|” karakterrel elválasztva:   
         *Azonosító|utasok száma|km óra állás*   
         Olvassa be a fájlt és mindegyik sorból készítsen egy-egy példányt a korábban megírt osztályt felhasználva.
      7. A példányosítás után mindegyik példány nyomtassa ki a számlát, ellenőrizze a szervízelést, majd ezután adja hozzá a korábban létrehozott listához.
      8. A *fuvarok* lista változót felhasználva írjuk ki a taxis fuvar adatokat a *taxiFuvarok.txt* nevű fájlba az alábbi minta szerint:   
         