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 programot kell írnia, mely kiszámolja az edzés közbeni megfelelő pulzusértéket a zsírégetéshez.

Kérje be a konzolon keresztül, a felhasználótól az életkorát, valamint kérdezzük meg tőle, hogy kezdő-e (i/n)?, majd az alábbi képletet felhasználva számolja ki az alsó és felső pulzus határértéket:

A felső szorzó értéke kezdő esetén 0,8 minden más esetben 0,7

*Alsó határérték = (220-életkor)\*0,65*

*Felső határérték = (220-életkor)\*felső szorzó*

Az eredményekről tájékoztassa a felhasználót!

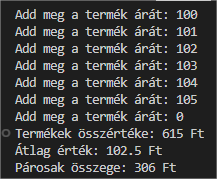
(A felhasználó arra a kérdésre, hogy kezdő-e feltételezzük, hogy egy i vagy egy n betűt ad válaszként)

Az alábbi mintában a kód egy futási lehetőségét láthatja:



1. **feladat – 14 pontos**

Fájl neve: 14\_pontos.py   
Ebben a feladatban termékek árát kell bekérnie és ezen értékek alapján számolnia.

* + - 1. Készítsen egy **termekBekeres** nevű függvényt, aminek nincs paramétere. A függvény addig kérjen be a felhasználótól árakat, amíg 0-át nem ír.  
         Az árakat tárolja el egy *termekek* nevű lista típusú változóban, és ezzel a változóval is térjen vissza a függvény, ha a felhasználó 0-át ír árnak.  
         FONTOS, ha a felhasználó negatív értéket ír, akkor is pozitív értékként tároljuk el!
      2. Készítsen egy másik függvényt **kiszamolas** néven, legyen egy lista típusú bemenő paramétere (formátuma megegyezik az előző függvény visszatérési értékével). A függvény az alábbi értékeket számolja ki és írja ki a képernyőre:   
         - Termékek árának összértéke.   
         - Termékek árának átlag értéke.   
         - Csak a páros termék árak összege.
      3. 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 termékekre való licitálását kell megvalósítania.

* + - 1. Írjon egy osztályt **licitTermek** néven. Hozzon létre egy konstruktort, amely bemenő paraméterként az alábbi értékeket kapja: *név, bid, licit lépcső*.  
         A bemenő paramétereken kívül, hozzon létre még három darab osztályváltozót a konstruktorban. Zárójelben a beállítandó értékét adtuk meg:  
         - licitek száma (érték:0)   
         - ár (érték: 0)   
         - termék megnyerve (érték: NEM)
      2. Az osztálynak legyen egy **bidVasarlas** nevű függvénye, amely egy bemenő paramétert vár *osszeg* néven (egész szám). A függvény növelje a bid változót a bemenő paraméter 1000-el osztott értékének 90%-val.  
         Képlet a számoláshoz:   
           
         *(összeg/1000)\*0.9*
      3. Az osztálynak legyen egy **arSzamolas** nevű függvénye (nem vár paramétert), amely az ar osztályváltozó értéket állítja be. Az érték a bid osztályváltozó értékének 1000-el való szorzatának a 110%-a.   
         Képlet a számoláshoz:   
           
         (bid\*1000)\*1.1
      4. Az osztálynak legyen egy **licitalas** nevű függvénye is. A függvény generáljon egy véletlen számot egy és ezer között. A licitek számát növelje 1-el. A bidek számát csökkentse a licit lépcsővel.  
            
         Ha a kigenerált véletlen szám osztható 12-vel, akkor a *termekMegnyerve* osztályváltozó értékét állítsa Igen-re és térjen vissza False értékkel.  
           
         Ellenkező esetben, ha a névben megtalálható az „a” betű és a név 0-ra végződik, akkor hívjuk meg a bidVasarlas osztály függvényét. A függvény bemenő paraméternek szintén egy véletlen számot kapjon (egy másikat) amely 1 és 8 között lehet, majd térjen vissza True értékkel.  
           
         Ellenkező esetben, ha a bidek száma 0 vagy az alá csökken, akkor a bidek számát állítsa 0-ra és térjen vissza False értékkel.  
           
         Minden más esetben térjünk vissza True értékkel.
      5. Készítsen egy *licitTermekek* nevű lista típusú üres változót.
      6. A *licitTermekek.txt* fájlban az alábbi adatokat találja soronként, pontosvesszővel tagolva:   
           
         *Név (termék neve);Bid (bidek száma);licit lépcső*   
           
         Olvassa be a fájlt és mindegyik sorból készítsen egy-egy példányt a korábban megírt osztályt felhasználva.
      7. A létrehozás után mindegyik termék egészen addig licitáljon amíg False értéket nem kap a függvény hívására.
      8. A licitálás után számoljuk ki az arSzamolas függvényt felhasználva a termék árát.
      9. Az elkészült példányt adjuk hozzá a korábban létrehozott licitTermekek nevű listához.
      10. Az a licitTermekek változót felhasználva írjuk ki az termékeket az *erdemenyek.txt* nevű fájlba az alábbi minta szerint:   
            
          