Programozási feladatok

* Az Ön feladata az alábbiakban olvasható leírás alapján három program elkészítése.
* A feladatok megoldásához szükséges fájlokat az  
  **Info(hálózati meghajtó)/\_HTML\_/Python23**   
  mappában találja.
* 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!
* 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!
* Figyeljen a változó nevekre! A változó neve beszédes legyen a benne található érték alapján nevezze el!
* **Minden program indulásakor a konzol előzményeit törölje**!

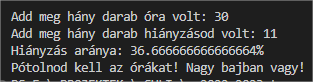
1. **feladat - 8 pont**

Fájl neve: 8\_pontos.py   
Ebben a feladatban egy tanóra hiányzási arányát kell kiszámolnia százalékban. Kérje be a felhasználótól (konzolon) keresztül, hogy hány darab óra volt, majd utána a hiányzások darabszámát. A két adatból számolja ki a százalékot az alábbi képlet szerint:

***százalék = hiányzások száma/hány darab óra volt\*100***

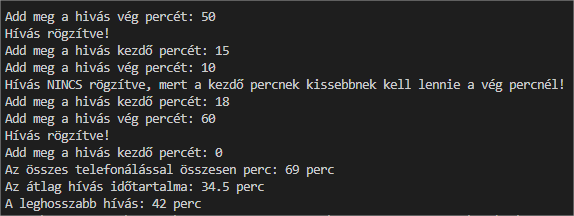
Írja ki a konzolra az eredményt (pl.: Hiányzás aránya: 20%)! A kiszámolt érték alapján írjon ki figyelmeztetést a program:   
- Ha 20% vagy a feletti akkor: *pótolnod kell az órákat! Bajban vagy!*  
- Ellenkező esetben ha 10 és 19% közötti az érték: *Ne hiányozz többet mert pótolnod kell!*   
- Minden más esetben: *Minden rendben van!*

Az alábbi mintában a kód egy futási lehetőségét láthatja:



1. **feladat – 14 pontos**

Fájl neve: 14\_pontos.py   
Ebben a feladatban telefonhívásokat kell elemeznie.

* + - 1. Készítsen egy *telefonHivasokRogzitese* függvényt (nem vár paramétert). A függvény egészen addig kérje be a kezdő és vég-perceket amíg, a kezdő perc értékének 0-át nem adunk meg. Ha a kezdő perc 0, akkor a vég-percet már ne kérje be a felhasználótól. Ha mind a két adat rendelkezésre áll, akkor egy szótár típusú változóba (*egyHivas*) mentsük le az adatokat.   
         A szótár kulcsai: *start, stop, ido*. A start és a stop-hoz értelemszerűen a bekért adatokat mentsük, míg az időhöz a start és a stop érték különbségét (ido=start-stop). Fontos a szótár csak akkor jöjjön létre és csak akkor mentsük el az adatokat, hogyha a stop idő nagyobb, mint a start idő.   
         A szótárakat egy lista típusú változóba (*hivasok*) mentsük, és ezzel a listával térjen vissza a függvény. Ha a szótár létrejött, akkor írjuk ki a képernyőre, hogy „*Hívás rögzítve*”, ellenkező esetben pedig: „*Hívás NINCS rögzítve, mert a kezdő percnek kissebbnek kell lennie a vég-percnél!*”
      2. Készítsen egy másik függvényt *telefonHivasokKiertekelese* néven, legyen egy paramétere melynek neve *hivasok*. A paraméter típusa megegyezik az előző függvényben lévő visszatérési értékkel (vagyis lista, aminek az elemei szótárak). A függvény feladata, hogy kiszámolja és kiírja a képernyőre az alábbi információkat:   
         - Az összes telefonhívás összesen perce   
         - Az átlaghívás időtartalma   
         - A leghosszabb hívás.
      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 focicsapatok edzését és játékát kell szimulálnia.

* + - 1. Írjon egy osztályt **fociCsapat** néven, ami egy objektumot valósít meg.
      2. A **fociCsapat** adattagjai a következőek legyenek: *név, bajnoki helyezés, edzettségi szint, energia szint*. Ezeket az adattagokat az osztály, a konstruktorából kapja paraméterként. A paramétereken kívül legyen még három darab tulajdonság. Az első az edzések száma, értéke: 0 (int). A második a pihenések száma értéke: 0 (int). És a harmadik az eredmény, értéke egy üres string.
      3. Hozzon létre az osztálynak, egy ***pihen*** nevű függvényt (nincs paramétere). A függvény növelje a pihenések számát 1-el. Generáljon egy véletlen számot 0 és 3 között. Az energiaszintet növelje a véletlen szám 50%-val.
      4. Hozzon létre az osztálynak egy, ***edz*** nevű függvényt (nincs paramétere), amely egy boolean típusú értékkel fog visszatérni. A függvény generáljon egy véletlen számot 1 és 3 között. Az energia szint a véletlen szám 150%-val növekedjen, az edzettségi szint pedig a véletlen szám 70%-val csökkenjen. A edzések száma nőjön, 1-el.   
         Ha az energia szintje 3 alá csökken, akkor a csapat pihenjen.  
         Ha az edzettségi szint eléri a 9-et vagy annál nagyobbat akkor térjen vissza True-val ellenkező esetben False értékkel.
      5. Hozzon létre az osztálynak egy ***jatszik*** nevű függvényt. Generáljon ki egy véletlen számot, amelyet elment egy *dontoPont* nevű változóba. A véletlen szám 1.000 és 10.000 között lehet. Ezután az alábbiak szerint döntse el, hogy:  
         - Ha a csapat neve M betűvel kezdődik VAGY a *dontoPont* osztható 13-al akkor: A bajnoki helyezés 2 értékkel csökkenjen, az eredmény értéke pedig NYERT legyen.   
         - Ellenkező esetben HA a névben megtalálható az „a” betű ÉS a döntőpont páros szám akkor: A bajnoki helyezés csökkenjen 1-el és az eredmény értéke DÖNTETLEN.   
         - Minden más esetben a bajnoki helyezés növekedjen 3-al és az eredmény értéke pedig VESZTETT legyen.

* + - 1. A *fociCsapatok.txt* fájlban az alábbi adatokat találja soronként, „|” karakterrel tagolva:   
         Csapat neve|bajnoki helyezés|edzettségi szint|Energia szint   
         Olvassa be a fájlt, tagolja, és mindegyik sorból készítsen egy-egy példányt a korábban megírt osztály segítségével. A létrehozás után addig edzen a csapat ameddig az értéke az edzésnek hamis (False).
      2. Az edzések után a csapat játsszon egyetlen egyszer.
      3. A létrehozott objektumokat egy változóban melynek neve *csapatok* (lista típus) gyűjtse össze, hogy a későbbiekben fel tudja használni.
      4. A *csapatok* nevű változót felhasználva írja ki a fájlba az eredményeket az alábbi minta szerint:   
         