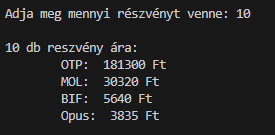
1. Feladat

Írjon programot **reszveny.py** néven, ahol a felhasználó meg tudja tekinteni, hogy a felhasználótól bekért részvény mennyiség mennyibe kerül a megadott vállalatoknál.

Kérje be a felhasználótól a kívánt mennyiséget és a bevitt adatok helyességének ellenőrzése nélkül nyomtassa ki az eredményt a minta szerint. Az előre megadott (nem valós) részvény értékek a reszveny.txt fájlban találhatóak.

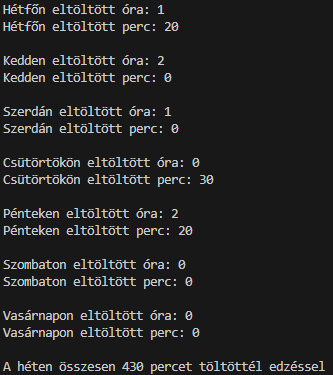


*Pontozás – minden feltétel egy-egy pont*

1. Létrehoz programot reszveny.py néven
2. A program hibaüzenet nélkül lefut
3. Bekéri a mennyiséget és tárolja
4. A felszorzást helyesen végzi el
5. Szükséges helyen kerekítést elvégzi
6. Minta alapján elrendezés
7. Bekért mennyiség számmá alakítása
8. A kiírt üzenetek helyesek (pl.: nincs benne elgépelés, helyesen jelenik meg a szöveg)

2. Feladat

A feladatban a program bekéri (adatok helyességének ellenőrzése nélkül), hogy a hét napjain mennyi órát és mennyi percet edzett a felhasználó. Bekérés után átszámolja csak percekre (például 3 óra 40 perc = 220 perc), majd ezt kiírja. A program üzeneteinek megfogalmazásában kövesse az alábbi példát! A program alapja az **edzesido.py** és a **percek.py** fájl. Ezek kiegészítésével írja meg a mintához igazítva a programot! Egészítse ki a megadott függvényt, illetve kérje be a napi adatokat a megadott lista segítségével.



A **percek.py** tartalma:



*Pontozás – minden feltétel egy-egy pont*

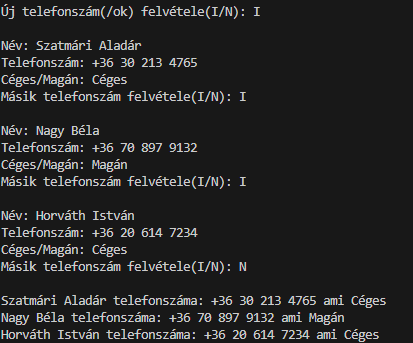
1. A program hibaüzenet nélkül lefut
2. Bekéri mind a 7 nap óra mennyiségét
3. Bekéri mind a 7 nap perc mennyiségét
4. A bekért számot szám típusúvá teszi
5. Egy óra adatról helyesen állapítja meg hány percet jelent
6. A függvény összeadását helyesen egészíti ki
7. A függvény paramétereit helyesen kitölti.
8. Ciklust szervez és a lista használatával a helyes sorrendben jeleníti meg a napokat, kéri be az adatokat
9. Felhasználja a függvényt a számításra
10. Helyesen meghívja a függvényt
11. A függvény visszatérési értéke alapján a fő program kiírja az összesen eltöltött időt
12. A függvényhívás a ciklusba kerül
13. A függvény visszatérési értéket megadja
14. A kiírt üzenetek helyesek

3. Feladat

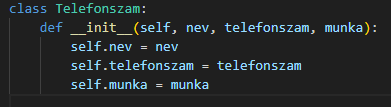
Az elkészítendő programban a felhasználó tetszőleges mennyiségű telefonszámot vehet fel. A program a felhasználótól addig kéri (az adatok helyességének ellenőrzése nélkül) az új telefonszámot, nevét és státuszát, amíg az „N” (nem) opciót nem választja. Ezután kiírja az összes bekért telefonszámot, majd külön fájlba kiírja a céges és a magán telefonszámokat.

1. Írjon programot **nevjegyzek.py** néven!
2. A telefonszámok és adatainak tárolására szolgáló objektumok alapját képező osztály rendelkezésre áll a **telefon.py.** A programjába töltse be ezt a modult, és használja a venne lévő osztályt!
3. Kérjen be a felhasználótól tetszőleges új névjegyzéket! Az adatok alapján hozzon létre Telefonszam osztályú objektumokat és tárolja őket!
4. Írja ki az összes bekért új névjegy nevét, telefonszámát és státuszát a minta alapján
5. Határozza meg a „Céges”, illetve „Magán” telefonszámokat, majd írja ki a céges telefonszámokat **cegesek.txt** szövegesfájlba, a magán telefonszámokat szöveges **magan.txt** fájlba.

A program üzeneteinek megfogalmazásában kövesse az alábbi példát. A feladat hibátlan elvégzéséért 18 pont jár.



A **telefon.py** tartalma:



*Pontozás – minden feltétel egy-egy pont*

1. Létrehoz egy programot nevjegyzek.py, a program hibaüzenet nélkül lefut.
2. Meghívja a **Telefonszám** nevű osztályt.