

Beolvasás és annak ellenőrzése: // Hegysegek

```
class hegyek:
    def __init__(self, line):
        darabol=line.split(";")
        self.hegycsucsNeve=(darabol[0])
        self.Hegyseg=(darabol[1])
        self.magassag=int(darabol[2].rstrip("\n"))

lista=[]
with open("hegyekMo.txt", "rt", encoding="utf-8") as file:
    file.readline()
    lines=file.readlines()
    for i in range(0, len(lines)):
        lista.append(hegyek(lines[i]))

#beolvas.ell.
for i in range(0, len(lista)):
    print("{} {}, {}".format(lista[i].hegycsucsNeve, lista[i].Hegyseg,
                              lista[i].magassag))

"""Határozza meg és írja ki a képernyőre a minta szerint, hogy hány
hegy található az állományban! """
#3.feladat
print("3.feladat: Hegycsúcsok száma: {} db".format(len(lista)))

"""Határozza meg és írja ki a képernyőre a minta szerint az
állományban található hegyek átlagmagasságát!"""
#4.feladat
osszMag=0
for i in range(0, len(lista)):
    osszMag=osszMag+lista[i].magassag
atlag=round(osszMag/len(lista), 2)
print("4.feladat: Hegycsúcsok átlagos magassága: {} m".format(atlag))

"""Határozza meg és írja ki a képernyőre a minta szerint a legmagasabb
hegy adatait! Feltételezheti, hogy nem alakult ki holtverseny"""
#5.feladat
maxIndex=0
for i in range(1, len(lista)):
    if lista[i].magassag>lista[maxIndex].magassag:
        maxIndex=i
print("5. feladat: A legmagasabb hegycsúcs adatai: ")
print("\tNev: {}".format(lista[maxIndex].hegycsucsNeve))
print("\tHegyseg: {}".format(lista[maxIndex].Hegyseg))
print("\tMagassag: {}".format(lista[maxIndex].magassag))

"""Kérjen be a felhasználtól egy magasságértéket!
Döntse el, hogy a Börzsöny hegységben található-e a megadott értéknél
magasabb hegycsúcs! A keresést ne folytassa, ha a választ meg tudja adni!
A képernyőre írást a minta szerint végezze!"""
#6.feladat
print("6.feladat:")
m=int(input("Kérem adjon meg egy magasságértéket!: "))
for i in range(0, len(lista)):
    if lista[i].Hegyseg=="Börzsöny" and lista[i].magassag>m:
        print("Van magasabb hegycsúcs a Borzsonyban!")
        break
else:
    print("Nincs magasabb hegy a Borzsonyban!")

"""Határozza meg és írja ki a képernyőre a minta szerint
azoknak a hegycsúcsoknak a számát, amelyek 3000 lábnál magasabbak!
Az átváltáshoz az 1 m = 3.280839895 láb értékkel dolgozzon!"""
#7.feladat
db=0
for i in range(0, len(lista)):
    if lista[i].magassag*3.280839895>3000:
        db=db+1
print("7. feladat: 3000 labnal magasabb hegycsúcsok száma {}".format(db))
```

2, Nobel-díjasok

```
class nobel:
    def __init__(self, line):
        darabol = line.split(";")
        self.evszam = int(darabol[0]) #szam eseten kell int!!!
        self.tipus = darabol[1]
        self.keresztnev = darabol[2]
        self.vezeteknev = darabol[3].rstrip("\n")

lista = []
with open("nobel.csv", "rt", encoding="utf-8") as file:
    file.readline()
    lines = file.readlines()
    for i in range(0, len(lines)):
        lista.append(nobel (lines[i]))

#beolvasas ell.
for i in range(0, len(lista)):
    print("{} {} {} {}".format(lista[i].evszam, lista[i].tipus,
                                lista[i].keresztnev, lista[i].vezeteknev))

#3.feladat
"""Határozza meg és írja ki a képernyőre a minta szerint, hogy
Arthur B. McDonald milyen típusú díjat kapott! Feltételezheti,
hogy életében csak egyszer kapott Nobel-díjat"""

for i in range(0, len(lista)):
    if lista[i].keresztnev == "Arthur B." and lista[i].vezeteknev == "McDonald":
        print("3.feladat: {}".format(lista[i].tipus))
        break

#4.feladat
"""Határozza meg és írja ki a képernyőre a minta szerint, hogy
ki kapott 2017-ben irodalmi Nobel-díjat!"""

for i in range(0, len(lista)):
    if lista[i].evszam == 2017 and lista[i].tipus == "irodalmi":
        print("4. feladat: {} {}".format(lista[i].keresztnev, lista[i].vezeteknev))

#5.feladat
"""Határozza meg és írja ki a képernyőre a minta szerint, hogy mely
szervezetek kaptak béke Nobel-díjat 1990-től napjainkig!"""

print("5. feladat:")
for i in range(0, len(lista)):
    if lista[i].tipus == "béke" and lista[i].evszam >=1990 and lista[i].vezeteknev:
        print("\t{:} {} ".format(lista[i].evszam, lista[i].keresztnev))

#6.feladat
"""A Curie család több tagja is kapott díjat. Határozza meg és írja ki a
képernyőre a minta szerint, hogy melyik évben a család melyik tagja milyen
díjat kapott!"""

print("6. feladat:")
for i in range(0, len(lista)):
    if "Curie" in lista[i].vezeteknev:
        print("\t{:} {:} {} {} ({}).format(lista[i].evszam, lista[i].keresztnev,
                                            lista[i].vezeteknev, lista[i].tipus))

#7.feladat
"""Határozza meg és írja ki a képernyőre a minta szerint, hogy melyik típusú
díjból hány darabot osztottak ki a Nobel-díj történelme folyamán!"""

print("7. feladat:")
f = 0
k = 0
o = 0
i = 0
b = 0
g = 0
for i in range(0, len(lista)):
    if lista[i].tipus == "fizikai":
        f = f + 1
    elif lista[i].tipus == "kémiai":
        k = k + 1
    elif lista[i].tipus == "orvosi":
        o = o + 1
    elif lista[i].tipus == "irodalmi":
        i = i + 1
    elif lista[i].tipus == "béke":
        b = b + 1
    else:
        g = g + 1
print("\tfizikai: {0:>21} db".format(f))
print("\tkémiai: {0:>22} db".format(k))
print("\torvosi: {0:>22} db".format(o))
print("\tirodalmi: {0:>20} db".format(i))
print("\tbéke: {0:>24} db".format(b))
print("\tkozgazdasagtani: {0:>13} db".format(g))
```

„Balkezesek”

```
#2. feladat
class sportolo:
    def __init__(self, line):
        darabol = line.split(";")
        self.nev = darabol[0]
        self.elseo = darabol[1]
        self.utolso = darabol[2]
        self.suly = int(darabol[3])
        self.magassag = int(darabol[4].rstrip("\n"))

lista = []
with open("balkezesek.csv", "rt", encoding="utf-8") as file:
    file.readline()
    lines = file.readlines()
    for i in range(0, len(lines)):
        lista.append(sportolo(lines[i]))

#ell.
for i in range(0, len(lista)):
    print("{} {} {}".format(lista[i].nev, lista[i].elseo,
                             lista[i].utolso, lista[i].suly,
                             lista[i].magassag))

#3. feladat
"""Határozza meg és írja ki a képernyőre a minta szerint,
hogyan hány adatsor található a forrásállományban!"""

print("3.feladat: {}".format(len(lista)))

#4. feladat
"""Határozza meg, és írja ki a minta szerint, azoknak a játékosoknak
a nevét és testmagasságát centiméterben (1 inch = 2,54 cm), akik utoljára
1999 októberében léptek pályára! Az eredményt egy tizedesjegyre kerekítve
írja ki a képernyőre!"""

print("4. feladat:")
for i in range(0, len(lista)):
    if lista[i].utolso.startswith("1999-10"):
        cm = round(lista[i].magassag * 2.54, 1)
        print("\t{}, {} cm".format(lista[i].nev, cm))

#5. feladat
"""Kérjen be a felhasználótól egy évszámot a minta szerint! Az évszámra
teljesülni kell az 1990 ≤ évszám ≤ 1999 feltételnek, amennyiben a felhasználó
hibás évszámot adott meg, írja ki a „Hibás adat! Kérek egy 1990 és 1999
közötti évszámot!” hibaüzenetet és kérje be újra!
Feltételezheti, hogy az inputadat konvertálható pozitív egész számra!"""

print("5.feladat:")
ev = int(input("Kérek egy 1990 és 1999 közötti évszámot!"))
while ev < 1990 or ev > 1999:
    ev = int(input("Hibás adat! Kérek egy 1990 és 1999 közötti évszámot!"))

#6. feladat
"""Határozza meg és írja ki a minta szerint, mennyi az átlagsúlya a
játékosoknak, akik az előző feladatban bekért évben pályára léptek!
Az eredményt két tizedesjegyre kerekítve írja ki a képernyőre! Feltételezheti,
hogy az első és az utolsó pályára lépés dátuma között minden évben játszottak
a játékosok. Ha az előző feladatot nem tudta megoldani, akkor dolgozzon
az 1995-ös évvel """

suly = 0
sulydb = 0
for i in range(0, len(lista)):
    if int(lista[i].elseo[0:4]) <= ev <= int(lista[i].utolso[0:4]):
        suly = suly + lista[i].suly
        sulydb = sulydb + 1
atlag = round(suly / sulydb, 2)
print("6. feladat: {} font".format(atlag))
```