

Gyakorló feladatok

2025

1 További gyakorló feladatok

1. Feladat

Olvassa be a `tavolsag.txt` fájl tartalmát. Számolja ki a fájlban megadott sebesség és idő függvényében a megtett távot méterben. A sorokat a számítási eredménnyel együtt írjuk ki a képernyőre.

```
[ ]: try:
    with open("./file/tavolsag.txt") as f:
        for sor in f:
            db = sor.strip().split(" ")

            s = db[0].split("/")
            sebesseg = 0
            if "km" in s[0]:
                sebesseg = int(s[0].replace("km", ""))
                sebesseg *= 1000
            elif "m" in s[0]:
                sebesseg = int(s[0].replace("m", ""))

            if "h" in s[1]:
                sebesseg /= 3600
            elif "min" in s[1]:
                sebesseg /= 60

            ido = 0
            if "min" in db[1]:
                ido = int(db[1].replace("min", ""))
                ido *= 60
            elif "h" in db[1]:
                ido = int(db[1].replace("h", ""))
                ido *= 3600
            elif "s" in db[1]:
                ido = int(db[1].replace("s", ""))

            print(" * ".join(db), "=", ido * sebesseg, "m")
```

```
except FileNotFoundError:
    print("A megadott fájl nem található.")
```

2. Feladat

A `TanAtlag.txt` fájlban soronként szerepelnek a hallgatók nevei a felvett tárgy neve, kreditje és az elért eredmény. Határozza meg az egyes hallgatók tanulmányi átlagát. Határozza meg a minimum, maximum és az átlag értékeket.

```
[ ]: hallgatok = []
eredmenyek = []
darabszam = []
atlagok = []

with open("./file/TanAtlag.txt") as f:
    for sor in f:
        db = sor.strip().split("\t")
        if db[0] == "":
            continue
        if db[0] not in hallgatok:
            hallgatok.append(db[0])
            eredmenyek.append(int(db[3]))
            darabszam.append(0)
        else:
            index = hallgatok.index(db[0])
            eredmenyek[index] += int(db[3])
            darabszam[index] += 1

for i in range(len(hallgatok)):
    avg = eredmenyek[i] / darabszam[i]
    atlagok.append(avg)
    print(f"{hallgatok[i]}: {avg:.2f}")

print("Min: ", min(atlagok))
print("Max: ", max(atlagok))
print("Átlag: ", sum(atlagok) / len(hallgatok))
```

3. Feladat

A `TanAtlag.txt` fájlban soronként szerepelnek a hallgatók nevei a felvett tárgy neve, kreditje és az elért eredmény. Határozza meg az egyes hallgatók korrigált tanulmányi átlagát. Határozza meg az ösztöndíjak értékét, ha a szakminimumKKI 3.73, a minimum összeg 6000 Ft, a maximum összeg 30000Ft. A képlet: $\text{minÖsszeg} + (\text{SajátKKI} - \text{szakminimumKKI}) * (\text{maxÖsszeg} - \text{minÖsszeg}) / (5 - \text{szakminimumKKI})$. A neveket és az ösztöndíj összegeket írassa ki.

```
[ ]: hallgatok = []
eredmenyek = []

with open("./file/TanAtlag.txt") as f:
    for sor in f:
        db = sor.strip().split("\t")
        if db[0] == "":
            continue
        if db[0] not in hallgatok:
            hallgatok.append(db[0])
            eredmények.append(int(db[3]))
        else:
            index = hallgatok.index(db[0])
            eredmények[index] += int(db[3]) * int(db[2])

for i in range(len(hallgatok)):
    avg = eredmények[i] / 30
    osszeg = 6000 + (avg - 3.73) * (30000 - 6000) / (5 - 3.73)
    if osszeg < 0:
        osszeg = 0
    print(f"{hallgatok[i]}: {int(osszeg)}Ft")
```

4. Feladat

A `Csuszda.txt` fájlban gyerekek nevei és testmagasságuk van feljegyezve. A gyerekek egy kalandparkba mennek, ahol az egyes csúszdákra elő van írva egy minimális magasság (155 cm), az alacsonyabb gyermekek nem használhatják. Névsorrendbe listázza ki azoknak a gyerekeknek a nevét, akik lecsúszhatnak a csúszdán.

```
[ ]: gyerekek = []
with open("./file/Csuszda.txt") as f:
    for sor in f:
        db = sor.strip().split("\t")
        if int(db[1]) >= 155:
            gyerekek.append(db[0])

gyerekek.sort()
for nev in gyerekek:
    print(nev)
```

5. Feladat

A `Csuszda.txt` fájlban gyerekek nevei és testmagasságuk van feljegyezve. A gyerekek egy kalandparkba mennek, ahol az egyes csúszdákra elő van írva egy minimális magasság (155 cm), az alacsonyabb gyermekek nem használhatják. Listázza ki a gyerekek nevét a magassági sorrendjük alapján.

```
[ ]: gyerekek = []

with open("./file/Csuszda.txt") as f:
    for sor in f:
        db = sor.strip().split("\t")
        gyerekek.append((db[0], int(db[1])))

db = len(gyerekek)
for i in range(db):
    for j in range(db - 1 - i):
        if gyerekek[j][1] > gyerekek[j + 1][1]:
            gyerekek[j], gyerekek[j + 1] = gyerekek[j + 1], gyerekek[j]

for gyerek in gyerekek:
    print(*gyerek)
```

6. Feladat

Írjon egy Python függvényt, amely bekér egy dátumot három pozitív egész számként (év, hó, nap)! A függvény határozza meg, hogy az adott dátum az év hányadik napja!

```
[ ]: def szokoev(ev):
    if ev % 400 == 0:
        return True
    if ev % 4 == 0 and ev % 100 != 0:
        return True
    return False

def napok_szama(ev, honap, nap):
    napok = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    eltel_t_napok = nap
    if szokoev(ev):
        napok[1] = 29
    if honap > 1:
        eltel_t_napok += sum(napok[: honap - 1])
    return eltel_t_napok

print(napok_szama(2024, 4, 29))
```

7. Feladat

Írjon egy Python függvényt, amely bekér három számot a felhasználótól és kiírja a képernyőre, hogy a számok közül bármelyik kettőnek az összege egyenlő-e a harmadik számmal!

```
[ ]: def osszeg(a, b, c):
    if a + b == c:
```

```

    print(a, "+", b, "=", c)
elif a + c == b:
    print(a, "+", c, "=", b)
elif b + c == a:
    print(b, "+", c, "=", a)

```

```
osszeg(2, 4, 2)
```

8. Feladat

Írjon egy Python függvényt, amely bekér egy dátumot szöveggént (pl.: "2025.12.01.")! Majd vissza adja, hogy melyik hónap van a dátumban (hónap neve) és az melyik évszakban van!

```

[ ]: def honap_evszak(datum: str):
    honapok = [
        "Január",
        "Február",
        "Március",
        "Április",
        "Május",
        "Június",
        "Július",
        "Augusztus",
        "Szeptember",
        "Október",
        "November",
        "December",
    ]
    reszek = datum.split(".")
    honap = int(reszek[1]) - 1
    honap_nev = honapok[honap]
    evszak = "Tél"
    if 2 <= honap <= 4:
        evszak = "Tavas"
    elif 5 <= honap <= 7:
        evszak = "Nyár"
    elif 8 <= honap <= 10:
        evszak = "Ősz"
    return (honap_nev, evszak)

print(honap_evszak("2025.12.01."))

```

9. Feladat

Írjon egy függvényt, amely bekéri az időt (óra perc formátumban), és kiírja a képernyőre, hogy a nap hányadik percében vagyunk.

```
[ ]: def percek(ora, perc):
    return perc + ora * 60

print(percek(9, 32))
```

10. Feladat

Írjon egy programot, ami bekéri a sakktábla egy mezőjének koordinátáját, és ha jó a koordináta, akkor kiírja a mező színét (“világos” vagy “sötét”, az A1 mező sötét). Ha a koordináta nem jó, nem ír ki semmit.

```
[ ]: def sakk(mezo: str):
    mezo = mezo.upper()
    betu = ord(mezo[0]) - ord("A")
    szam = int(mezo[1]) - 1

    if 0 > betu or betu > 7 or 0 > szam or szam > 7:
        return

    # print(betu, szam)
    if (betu + szam) % 2 == 0:
        print("sötét")
    else:
        print("világos")

sakk("E5")
```

11. Feladat

Írjon egy programot, ami bekéri a sakktábla két mezőjének koordinátáját (honnan, hová lépünk). Határozza meg, hogy melyik bábuval lehet végrehajtani a lépést, ha a lépés üres mezőre történik (nem ütünk le bábút).

```
[ ]: def sakk(honnan: str, hova: str):
    honnan = honnan.upper()
    hova = hova.upper()
    x = ord(hova[0]) - ord(honnan[0])
    y = int(hova[1]) - int(honnan[1])

    if 0 <= abs(x) < 2 and 0 <= abs(y) < 2 and abs(x) + abs(y) > 0:
        print("Király")
    if abs(x) == 0 and abs(y) == 1:
        print("Gyalog")
    if abs(x) == 2 and abs(y) == 1:
        print("Huszár")
    if abs(x) == 1 and abs(y) == 2:
        print("Huszár")
```

```
if x == 0 and y != 0:
    print("Vezér")
    print("Bástya")
if y == 0 and x != 0:
    print("Vezér")
    print("Bástya")
if abs(x) == abs(y) and x != 0:
    print("Vezér")
    print("Futó")

sakk("D4", "E6")
```