

## 4. hét: nevezetes algoritmusok, listák

✎ Czirkos Zoltán, Frey Balázs, Bulla Ádám · 🕒 2023.09.20.  
Listákkal végezhető műveletek. Egyszerűbb adatszerkezetek építése, indirekt adatelérés.

### Tartalom

- [Első kis ZH](#)
- [Sztring formázása: "A válasz: 42."](#)
- [Lista megfordítása](#)
- [Növekvő sorozat-e](#)
- [Legnagyobb](#)
- [Hol a legkisebb elem?](#)
- [Indexek tárolása](#)
- [Autópálya forgalmi statisztika](#)
- [Betűk gyakorisága](#)

### 1. Első kis ZH 🔒

A hétfői alkalmakon lesz az első kis ZH.

### 2. Sztring formázása: "A válasz: 42." 🔒

Írj programot, amely kér a felhasználótól két számot, és kiírja az összegüket! Méghozzá pontosan ebben a formában:

```
a = 30
b = 12

A válasz: 42.
```

Oldd meg a feladatot kétféleképpen:

- Állítsd elő sztringek összefűzésével és konverzióval a kiírandó szöveget!
- Alkalmazd a [sztringek formázási lehetőségeit](#), az előadáson tanult módon!

### 3. Lista megfordítása 🔒

A listák `.reverse()` függvénye megfordítja a listát, amelyre meghívják:

```
szamok = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
szamok.reverse()
print(szamok)
```

```
[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]
```

Készíts listákat megfordító algoritmusokat, amelyek a következőképpen működnek:

- Adott egy lista az **eredeti** nevű változóban. Járd be ezt visszafelé (végétől az elejéig), és **fűzd hozzá a fordított** nevű listához a benne tárolt adatokat!
- Adott egy újabb lista az **eredeti** nevű változóban. Minden **végéről kivett elemet** fűzz hozzá egy új lista végéhez!
- Fordítsd meg a listát helyben! Cseréld meg az első és az utolsó, a második és az utolsó előtti, ... elemét!

► **Megoldás**

### 4. Növekvő sorozat-e 🔒

Adott egy számokból álló lista. A programodnak azt kell eldöntenie, hogy a számok növekvő sorozatot alkotnak-e (rendezett-e a lista), és ezt kiírni a kimenetre!

Melyik **programozási tételt** kell ehhez használnod? Mit jelent az elempárokra nézve, hogy rendezett a lista? Mikor mondhatod azt, hogy nem alkotnak növekvő sorozatot?

Hozz létre tíz elemű listát, amin tesztelsz! Próbáld ki a programod olyan bemenetekre is, ahol a rendezettség a) több helyen elromlik, b) egy helyen romlik el, c) az elején van a hiba, d) a végén van a hiba!

► **Megoldás**

### 5. Legnagyobb 🔒

Adott az alábbi lista:

```
lista = [-25, 12, -54, 8, 77, 98, -29, 35, 3, 71]
```

Írj ciklust, amely meghatározza, melyik a legnagyobb szám a listában, és írd ki ezt a számot! (Melyik elemeket kell a ciklusnak vizsgálnia? Vigyázat: nem kell az összeset! Ha a listának csak egyetlen eleme van, az egyben maximum is.)

Módosítsd úgy a programot, hogy a ciklusba belépés előtt, és a ciklustörzs végén is kiírod mindig az épp megvizsgált elemet, és a maximum változó értékét! Vizsgáld meg az eredményt, figyeld meg, a maximum változó hogyan változik!

► **Megoldás**

### 6. Hol a legkisebb elem? 🔒

Írj programot, amely tartalmaz egy tíz elemű listát, az általad megadott kezdeti értékekkel inicializálva! (Tehát nem kell a programnak egyesével beolvasnia azokat a billentyűzetről.) Figyelj arra, hogy az elemek különbözőek legyenek, ez most fontos lesz. Írd ki ezt a listát!

```
A lista: 25 69 54 8 77 6 29 10 3 98
```

Írd át úgy a programot, hogy megjelenjen a listaelemek előtt a listaindex is!

```
A lista: [0]=25 [1]=69 [2]=54 [3]=8 [4]=77 [5]=6 [6]=29 [7]=10 [8]=3 [9]=98
```

Írj programrészt, amely megmondja, melyik a legkisebb szám a listából! Írd ki ezt a számot! (Próbáld ki a programod úgy is, hogy a legkisebb szám a lista legelején és legvégén van!)

```
A lista: [0]=25 [1]=69 [2]=54 [3]=8 [4]=77 [5]=6 [6]=29 [7]=10 [8]=3 [9]=98
A legkisebb szám: 3
```

Alakítsd át a minimumkeresést úgy, hogy ne csak a legkisebb szám értékét, hanem annak helyét, azaz listabeli indexét is meg tudd mondani! Ehhez szükséged lesz egy minhely változóra, amely azt fogja megjegyezni, hányadik indexen volt a legkisebb szám.

Az igazán jó megoldás az, ha továbbra is egy ciklusod lesz, amelyik ezt a keresést végzi. Tehát nem úgy kell működnie, hogy előbb megjegyzi a legkisebb számot (3), utána pedig újból végigszalad a listán, hogy rájöjjön, hol is volt az (8-as hely). A helyet már a keresés közben is meg lehet jegyezni. A futási eredmény legyen ilyen:

```
A lista: [0]=25 [1]=69 [2]=54 [3]=8 [4]=77 [5]=6 [6]=29 [7]=10 [8]=3 [9]=98
A legkisebb szám: 3
A legkisebb index: 8
```

Végül pedig írd úgy ki a listát, hogy a legkisebb elem mellé egy jelölést teszel:

```
Jelölve: 25 69 54 8 77 6 29 10 3[MIN] 98
```

► **Megoldás**

### 7. Indexek tárolása 🔒

Adott egy valós számokat tartalmazó lista, benne vegyesen mindenféle előjelű számokkal.

```
szamok = [2.5, -69, 5.4, -8, -7.7, 6, 2.9, -10, -3, 9.8]
```

Írj egy olyan programot, amelyik kilistázza a számokat az elemek indexeivel! (Emlékezz vissza, ezt a feladatot egyszer már megoldottad laboron.) Valahogy így:

```
Összesen 10 szám van.
[0] = 2.5
[1] = -69
[2] = 5.4
[3] = -8
...
```

A következő lépés kigyűjteni egy **másik listába** a negatív listaelemek **indexeit**. Hozd létre ezt a másik listát, töltsd fel az indexekkel, majd legvégül – ha már kész vagy a lista megépítésével – írd ki ezeket is!

```
Ebből 5 szám negatív.
Indexeik: 1 3 4 7 8
```

Ha ez is megvan, egy olyan programrészt kell írnod, amelyik az indexek ismeretében kiírja, hogy mik voltak a negatív számok. Fontos, hogy ne keresd meg újra a negatív számokat! Elvégre is, ha az indexeik megvannak, abból már lehet tudni, melyek voltak azok. A végeredmény ilyen formátumú legyen:

```
Ebből 5 szám negatív.
1. [1] = -69
2. [3] = -8
3. [4] = -7.7
...
```

Rajzold ábrát, amely a listákat, az elemekben tárolt számokat, és az azok közötti összefüggéseket ábrázolja!

► **Megoldás**

### 8. Autópálya forgalmi statisztika 🔒

Traffipax méri az autópályán az autók sebességét (0–199 km/h). A forgalom elemzése céljából statisztikát szeretnének készíteni, milyen sebességgel közlekednek az elhaladó autók, 10 km/h bontásban.

Olvasd be a programban üres sorig az autók sebességét, majd írd ki a megadott formátumban a statisztikát! Ügyelj arra, hogy fix méretű tárolóval oldd meg a feladatot, ne jegyezze meg a program a teljes bemeneti adatsort!

Bemenet:

```
125
97
149
128
117
...
```

Kimenet:

km/h	autó
0-9	0
10-19	3
...	
120-129	12
130-139	7

► **Megoldás**

### 9. Betűk gyakorisága 🔒

Szeretnének megvizsgálni a szavakban előforduló betűk gyakoriságát egy programmal.

A program egy angol nyelvű szöveget kap. A szöveg eleve nagybetűkből áll: „TO BE OR NOT TO BE: THAT IS THE QUESTION.” Ehhez kell statisztikát készíteni, hogy melyik betű hányszor fordult elő a szövegben, és az az összes betű hány százalékát adja. A kimenet legyen ehhez hasonló:

```
A: 1 db, 3.33%
B: 2 db, 6.67%
E: 4 db, 13.33%
...
```

Ha valamelyik betű nem szerepelt a szövegben, az ne jelenjen meg a statisztikában se! A nem betű karaktereket figyelmen kívül kell hagyni.

Milyen adatszerkezet kell az adatok tárolásához? Rajzold le a program megírása előtt!

► **Tipp – adatszerkezet**

► **Tipp – karakterek**

► **Megoldás**

