

## 5. hét: vegyes feladatok, gyakorlás

 Czirkos Zoltán, Frey Balázs ·  2023.10.01.

### Tartalom

- [Prímtényezősz felbontás](#)
- [A Fisher-Yates keverés](#)
- [Útiterv](#)
- [Kő, papír, olló](#)
- [Memórijáték](#)
- [További feladatok](#)

### 1. Prímtényezősz felbontás

Írj egy programot, amelyik kér egy számot a felhasználótól, és kiírja a prímtényezősz felbontását!

```
Melyik számot? 75

75 | 3
25 | 5
5  | 5
1  |
```

- [Tipp](#)
- [Megoldás](#)

### 2. A Fisher-Yates keverés

Előadáson szerepelt Fisher és Yates algoritmus, amellyel egy listát lehet megkeverni. Az ott bemutatott verzió helyben keveri meg a listát (egyetlen listát módosít az elemek cseréjével), és az elejétől a végéig halad. A működése:

```
CIKLUS i = 0-tól n-2-ig:
    j = véletlenszám i ≤ j < n között
    csere: lista[i] ↔ lista[j]
```

Implementáld újra az algoritmust a pszeudokód alapján! (Csak akkor nézd meg az [előadásanyagot](#), ha elakadtál.) Utána írd meg két további változatban:

- Az első verzióban dolgozz egy listával, de keverd azt a végétől haladva az eleje felé! Vagyis cseréld meg az utolsó elemet egy véletlenszerűen választottal, az utolsó előttiit egy véletlenszerűen választottal és így tovább.
- A második verzióban dolgozz két listával! Az eredeti lista véltelenszerűen választott elemeit tedd át egy másik listába, törölve azt folyamatosan az előbbiből!

Mindkét változatban írd ki a listát (listákat) folyamatosan, iterációnként, hogy látszódjon, mi történik!

- [Megoldás](#)

### 3. Útiterv

Egy kamionsofőr autópályán vezet, 100 km/h-val haladva. A benzinkutak egymástól való távolságát előre ismeri a térképről. Minden másfél órányi vezetés után pihenőt kell tartania. Szeretne egy úttervet készíteni: előre eldönteni azt, hogy melyik benzinkúton kell majd pihennie. A szabály egyszerű: minden olyan benzinkútnál megáll, amelynél ha nem tartana pihenőt, a következő kúthoz már csak úgy érne el, ha túllépné a másfél órás egyfolytában vezetést.

Írj programot, amely soronként beolvassa az egyes benzinkutak közötti távolságot km-ben! Előbb megkapja a távolságok számát, majd magukat a távolságokat (soronként egy darabot). Végül pedig írja ki az úttervet. Például ha bemenetként a 45.3, 30.7, 64, 35, 72 számsort kapja (km-ek), akkor a kimenet az alábbi kell legyen:

```
45.3 km, 30.7 km, 64 km, szünet.
35 km, 72 km, vége.
```

- [Megoldás](#)

### 4. Kő, papír, olló

Írj programot, amelyik „kő, papír, olló” játékot játszik! A program először kérje el a felhasználó tippjét (k, p, o betűk, mint kő, papír, olló). Ezután válasszon ő maga is egyet, és hasonlítsa a kettőt össze! A kő erősebb, mint az olló, mert kicsorbítja. A papír erősebb, mint a kő, mert becsomagolja. Az olló erősebb, mint a papír, mert elvágja. Ezek alapján a gép vagy a játékos kapjon egy pontot! Ha egyformát tippeltek, akkor semelyikük nem kap. A v beírása után írja ki a program, hogy ki nyert!

- [Tipp](#)

Kő (k), papír (p), olló (o) vagy vége (v)?

Szerinted: k
Szerintem: k.
Senki nem kap pontot.

Szerinted: p
Szerintem: k.
p>k, ezt te vitted!

Szerinted: v

Te nyertél, 1>0 ponttal.

### 5. Memórijáték

6×6 kártya van lefordítva a játékosok előtt, 18 pár, amelyek egyformák. A kártyákon betűk vannak, A, B, C, ... Írj egy programot, amelyik generál egy véletlenszerű leosztást! Természetesen egy betűnek pontosan kétszer kell szerepelnie (egy pár)!

- [Tipp](#)

### 6. További feladatok

Érdemes gyakorolni a [statisztikával](#), [indirekt adateléréssel](#) illetve az [adatok kezelésével és beolvasással](#) kapcsolatos feladatokat, mert a 2. kis zárhelyin ilyen feladat lesz.

A feladatgyűjteményben találsz még [különféle játékokat](#). Írj programot ahhoz, amelyik tetszik, vagy fejleszd az eddigieket a saját ötleteid alapján!

