

1. feladat: Fájlokból hallgatók szűrése

- Egy `.txt` fájlban soronként: név, Neptun-kód, átlag.
- Adj be egy input Neptun kódot, ha egy beolvasás során megegyezik a hallgatóéval akkor ne kerüljön bele a tömbbe.
- Olvasd be dinamikus tömbbe őket, szabj határt az egy tömbbe beolvasható hallgatóknak, ha kell több lépésben dolgozd fel.
- Szűrd ki azokat, akiknek az átlag > 4.0 , és írd ki őket egy másik fájlba.

2. feladat: Struktúrák másolása és módosítása

- Definiálj egy `Konyv` struktúrát:

```
typedef struct {
    char cim[128];
    char szerzo[64];
    int ev;
} Konyv;
```

- Olvass be egy könyvlistát `.txt`-ből.
- A szerzők neveiben számold meg a .
- Készíts új dinamikus tömböt csak azokról, melyek $ev > 2010$ és $ev < 2022$.
- A legkevesebb magánhangzót tartalmazó szerző nevet írasd ki konzolra.

3. feladat: Időpontok kezelése intervallummal

- Egy `.txt` fájlban minden sorban: időpont (óra:perc) és esemény.
- Kérj be egy időpontot.
- Listázd azokat az eseményeket, amelyek az adott időponttól ± 10 perc intervallumban vannak.
- Írd ki a részletes találati listát egy új fájlba.

4. feladat: Moláris tömeg számítás szénhidrogénekre

- Olvass be soronként képleteket (`C%uH%u` formátum).
- Számítsd ki mindegyik moláris tömegét ($C = 12.01$, $H = 1.008$).
- Írd ki a képletet és a moláris tömeget formázottan `fprintf`-vel egy új fájlba.

5. feladat: Dinamikus tantárgy lista kezelése

- Definiálj egy `Tantargy` struktúrát:

```
typedef struct {
    char nev[64];
    int kredit;
    int jegy;
} Tantargy;
```

- Kérd be `scanf`-vel, hány tárgyat ad meg a felhasználó (min. 5).
- Dinamikusan foglald a tömböt, olvasd be adatokat kézzel.
- Számítsd ki a tanulmányi átlagot és a KKI-t függvényekkel, (lehet összevonni is) írd ki `fprintf`-vel egy fájlba.

$$\bar{x} = \frac{\sum_{i=1}^n e_i k_i}{\sum_{i=1}^n k_i} \quad \text{KKI} = \frac{\sum_{i=1}^n e_i k_i}{30}$$

6. feladat: String transzformáció teljes szövegre

- Olvass be egy tetszőleges hosszú szöveget fájlból.
- Írj egy `void csere(char *s);` függvényt, amely karaktereket cserél számokra (o→0, i→1, stb.).
- Írd ki az átalakított szöveget és számold meg, hány karaktert cseréltél, `fprintf`-pel egy másik fájlba.

7. feladat: Hőmérséklet átváltás több mértékegység között

- Olvass be hőmérsékleti adatokat °C, °F vagy K formátumban. (sima C, F karakterekkel)
Minden sorban egy adat legyen tetszőleges mértékegységgel.
- Írj függvényeket: `CtoF`, `FtoC`, `CtoK`, `KtoC` stb.

$$F = \frac{9}{5}C + 32 \quad (\text{Celsius} \rightarrow \text{Fahrenheit}) \quad (1)$$

$$K = C + 273.15 \quad (\text{Celsius} \rightarrow \text{Kelvin}) \quad (2)$$

$$(3)$$

- Minden adatot írd ki átváltva Celsiusba, Fahrenheitbe és Kelvinbe egy új fájlba, soronként mindhárom szerepeljen.
- Mekkora volt a legnagyobb hőmérséklet változás két sor között? Írd ki konzolra.

8. feladat: RGB színek keverése és táblázat generálása

- Olvass be RGB komponenseket (`r`, `g`, `b`) formátumban soronként.
- Ellenőrizd, hogy minden színkomponens 0-tól 255-ig terjedjen.
Ellenkező esetben dobd el az adatot.
- 1. Keverd össze az összeset:

$$R = \frac{1}{n} \sum_{i=1}^n R_i, \quad G = \frac{1}{n} \sum_{i=1}^n G_i, \quad B = \frac{1}{n} \sum_{i=1}^n B_i.$$

- 2. Keverd össze őket kettesével is,
- Írd ki az eredményül kapott szín(ek)e(t) (`r`, `g`, `b`) egy .csv fájlba. (.txt is megfelel)

9. feladat: Soronkénti szócsere szövegben

- Olvass be egy szövegfájlt soronként.
- Cserélj le egy adott szót egy másikra (pl. minden “alma” → “körte”).
- Írd az eredményt egy másik fájlba `fprintf`-pel.

10. feladat: Logisztikus leképezés komplex iterációval

- Olvass be egy fájlból r értékeket és x_0 kezdőértékeket, soronként egy-egy pár legyen.
Csak azokat a sorokat vedd figyelembe ahol $|x_0| < 1$ és $0 < r < 4$
- Iterálj 10 lépést a logisztikus leképezéssel, az adatokat tárold el (nem muszáj):

$$x_{n+1} = r x_n (1 - x_n)$$

- Az iterációk eredményeit írd ki egy új fájlba.