

1. Feladat – Tömb minimum és maximum keresése referenciával

Készíts egy függvényt, amely egy `int` típusú dinamikus tömb minimumát és maximumát két referencián keresztül visszaadja!

A `main`-ben töltsd fel a tömböt véletlenszámokkal, hívd meg a függvényt, majd írasd ki a minimumot és maximumot!

2. Feladat – Két tömb összefűzése sablonfüggvénnyel

Írj `template` függvényt, amely két azonos típusú dinamikus tömböt (pl. `int*`, `double*`) és méretüket kapja paraméterül,

és visszaad egy új, dinamikusan lefoglalt tömböt, amely az első és második tömb elemeit fűzi össze!

Az új tömb méretét referencia paraméterben adja vissza.

`Main`-ben próbáld ki `int` és `double` típusra is, színes kiírással!

3. Feladat – Medián számítása sablonfüggvénnyel

Készíts `template` függvényt, amely egy T^* típusú tömb mediánját számolja ki (a tömb és méret a paraméter)!

`Main`-ben próbáld ki `int` és `double` tömbre is, legalább kétféle esettel (páros/páratlan elemszám).

Használj sablont, és másold el a tömböt, ha módosítod (pl. rendezés miatt).

4. Feladat – Keresés `int` tömbben, pointerrel, index visszaadással

Írj függvényt, amely egy dinamikus `int` tömbben (amit `new`-val hozol létre), megkeresi az első, felhasználó által megadott számot,

és visszaadja az elem indexét (vagy `-1`-et, ha nincs ilyen).

Bónusz: készíts `template` változatot is, hogy `double` tömbre is működjön!

5. Feladat – Konzoltörlés animáció több színnel és animációval

Írj programot, amely egy szöveget karakterenként ír ki (gépelés animáció, `sleep`),

majd egy billentyűre törli a konzolt (`system("cls")` vagy `clear`).

Utána háromféle színnel és legalább kétféle animációval (pl. soronként, karakterenként) újra kiírja a szöveget!

Használj saját színező makrókat/funkciókat!

Tipp:

C++11-től használhatod a `<chrono>` és `<thread>` könyvtárakat.

Például 50 ms várakozás:

```
#include <chrono>
```

```
#include <thread>
```

```
// ...
```

```
std::this_thread::sleep_for(std::chrono::milliseconds(50));
```