

---

# C ISMÉTLÉS

---

Hallgatói segédlet

Levendovszky Tihamér

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Automatizálási és Alkalmazott Informatikai Tanszék  
Alkalmazott Informatika Csoport

*Ennek a rövid segédletnek a célja, hogy felidézze az elmúlt félévben tanultakból a C++-hoz legszükségesebb részeket, valamint segítsen az első gyakorlatok házi feladatainak megoldásában. Az anyag nagy része a laboron sorra kerül, de a kidolgozott példák segítségével otthon is át lehet ismételni ezeket a részeket. Kérek mindenkit, hogy ezeket az ismereteket „gyúrja át” rendszeren, mert nemcsak a C++-hoz szükségesek, hanem a villamosmérnöki szakma műveléséhez is nagyon fontosak. Az útmutatás során hivatkozásokat adunk az Automatizálási és Alkalmazott Informatikai Tanszék munkatársai által készített elektronikus AUT C példatárra.*

1. Címszerinti és érték szerinti paraméterátadás. Tanulmányozzuk a [Változó értékének megnövelése függvényből](#) című példát és annak megoldását! A C-ben csak pointeren keresztül lehet megváltoztatni egy függvényen kívül levő értéket. C++-ban a pointernek ezt a funkcióját a referencia veszi át.
2. A dinamikus memóriakezeléshez nézzük meg a [Számok kiírása fordított sorrendben](#) című példát és a hozzá tartozó megoldást! Különösen figyeljünk oda a tömbök és mutatók közötti összefüggésre. A példában mindenképpen dinamikus memóriahasználatra vagyunk kárhóztatva, hiszen el kell tárolnunk a számokat. Csak akkor használjunk dinamikus memóriakezelést, ha más megoldás nem hatékony (itt a rekurzió szóba jöhetne, de egyáltalán nem hatékony). A C++-ban nem elég a *malloc*, hiszen amikor osztályokat példányosítunk, akkor meg kell hívni a konstruktort, amire a *malloc* alkalmatlan. Ezért vezették be a C++ -ban a *new* operátort. A destruktort miatt a *free* helyett C++-ban a *delete* operátort használjuk.
3. A memóriakezelést és a sztringek használatának nagyon sok hibájára és azok elkerülésére mutat példát a [Sztring megfordítása](#) című feladat és annak megoldása. C-ben a sztringeknek ‘\0’ karakterre kell végződnie. A C++-ban a sztringek osztályok, azokban tárolhatjuk a méretet és a kezdőpointert, akkor nincs szükségünk a lezárókarakterre. Ugyanakkor ha C stílusú, nullterminált sztringekkel szeretnénk együttműködni, például konverzióval, sose felejtjük, hogy a sztringkezelő C könyvtári függvényeknek mindig szükségük van a lezárókarakterre, különben elkeverednek érvénytelen memóriaterületekre.