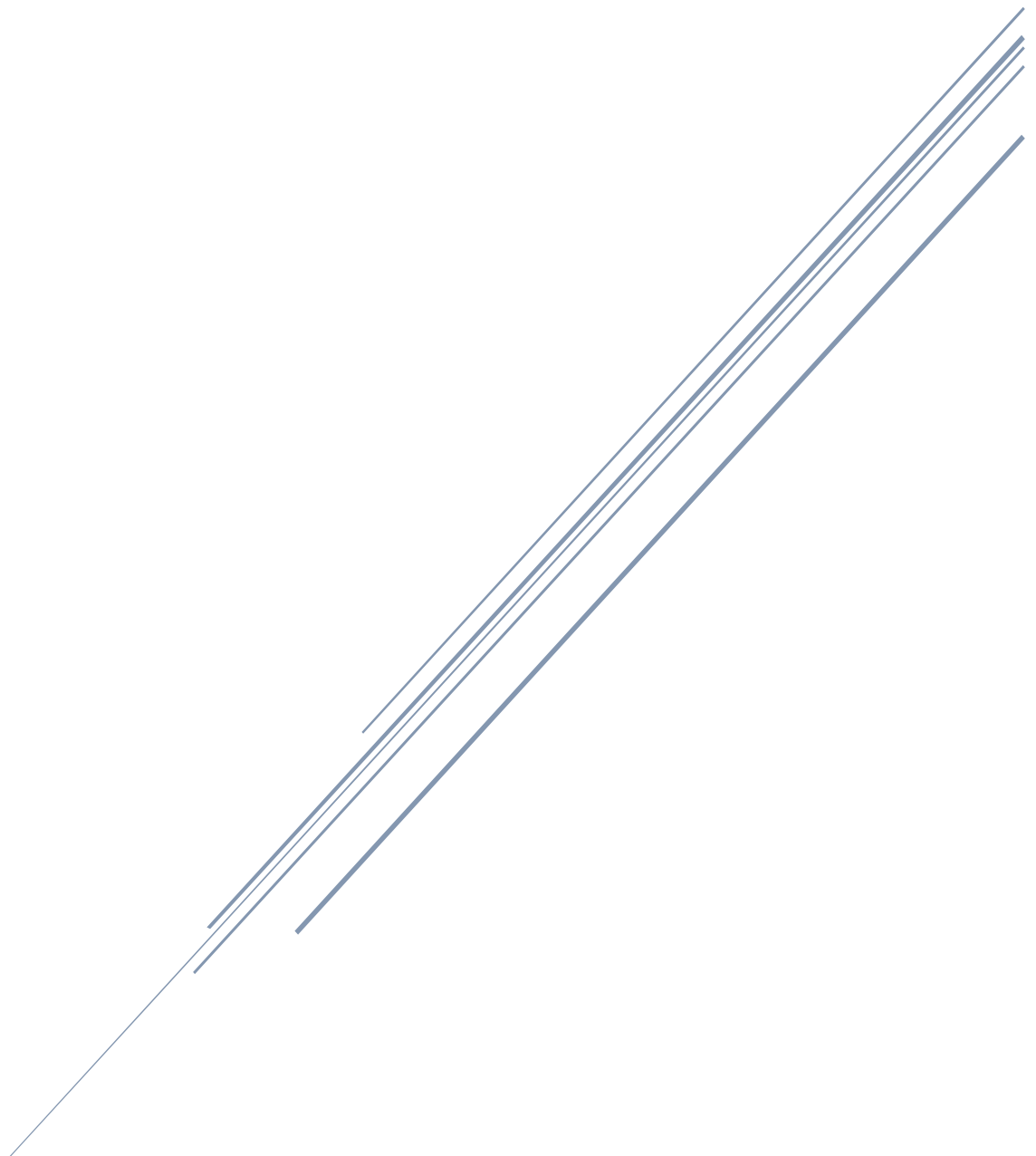


# PROGRAMOZÓI DOKUMENTÁCIÓ

Könyvtár – Nagy Házi Feladat

Ujhelyi Bence



Budapesti Műszaki és Gazdaságtudományi Egyetem  
Programozás alapjai 1

# A program működése

Ez a program egy könyvtár olvasókról és könyvekről tárolt adatainak kezelésére készült. A program képes a bevitt adatok elmentésére, tárolására és későbbi visszatöltésére. A program képes új adatok felvételére, régiak törlésére és az elemek közötti keresésre. Képes továbbá a könyvek kölcsönzésének nyilvántartására és kezelésére. Az adatokat dinamikusan foglalt memóriaterületre foglalja a program, összetett adatstruktúrában.

## Forráskódok és függvényeik

### main.c

A program fő modulja, az összes többi fájl és függvény innen kerül meghívásra (közvetlen vagy közvetett módon)

### menu.c és menu.h

Ez a fájl felel a menü megjelenítéséért, illetve az összes adatmanipulációs függvény itt kerül meghívásra. Néhány függvény itt kerül létrehozásra, elkerülve a fájlok ciklikus hívását.

**menu()** – menü megjelenítéséért felelős függvény

**olv\_kereses()** – Olvasók kereséséért felelős függvény. Paraméterei az olvasók és könyvek listáira mutató pointerok. Nincs visszatérési értéke, a függvényben megadott paraméterek szerint tudunk keresni

**foglalas()** – A könyvek kölcsönzéséért felelős függvény. A könyvek listára mutató pointer az egyetlen paramétere, visszatérési értéke nincs. A függvényben meg kell adni a kikölcsönzendő könyv adatait és annak a személynek az ID-ját, akinek a nevére írjuk a könyvet

**konyv\_kereses()** – Könyvek keresése történik itt. Paraméterei a könyvek, olvasók listákra mutató pointerok, illetve a kiválasztott keresési szempont. A szempont alapján végigmegy a listán a függvény és ha egyezést talál, akkor kiírja az adatait, illetve, ha „kolcs” struktúra változóba eltárolt érték nem nulla, akkor annak a nevét és adatait, aki kikölcsönözte

**UjSorki()** – A beolvasott sztringek végéről eltávolítja az új sor ('\n') karaktert

### konyvek.c és konyvek.h

Ezekben a fájlokban kerül definiálásra a „konyvek” adatstruktúra, továbbá itt találhatóak a könyvekkel kapcsolatos függvények

#### Adatstruktúra:

- |          |   |   |
|----------|---|---|
| • cim    | – | A könyv címe (char tömb)  |
| • szerzo | – | A könyv szerzője (char tömb)  |
| • kiadas | – | Kiadás éve (int)  |
| • tema   | – | Téma (char tömb)  |
| • kolcs  | – | Ha ki van kölcsönözve, akkor a kölcsönző ID-ja, ha nincs, akkor 0 (int) |
| • kov    | - | Következő elemre mutató pointer   |

#### Függvények:

**konyv\_hozza()** – Ezzel a függvénnyel tudunk új könyveket hozzáfűzni a listához. Egyetlen paramétere a lista kezdőelemére mutató pointer. Visszatérési értéke az új listába fűzött elemre mutató pointer

**konyv\_torl()** – Ezzel a függvénnyel lehet a listából elemeket törölni. Megkapja a lista első elemére mutató pointert és a függvényben megadott paraméterek alapján megkeresi a listában az adott elemet és felszabadítja. Nincs visszatérési értéke

**kisbetu()** – A beolvasott sztringeket kisbetűssé alakítja, annak érdekében, hogy az elemekre bármilyen formában hivatkozni lehessen

**konyv\_felsz()** – A könyvek listát szabadítja fel a program futásának végén. Paramétere a lista első eleme

## olvasok.c és olvasok.h

Ezekben a fájlokban történnek az olvasókkal kapcsolatos műveletek többsége. Itt található az „olvasok” adatstruktúra is

### Adatstruktúra:

- |        |   |                                    |
|--------|---|------------------------------------|
| • nev  | - | Olaszó neve (char tömb)            |
| • szul | - | Születési éve (int)                |
| • lakc | - | Lakcím (char tömb)                 |
| • ID   | - | Olaszó személyes azonosítója (int) |
| • kov  | - | Következő elemre mutató pointer    |

### Függvények:

**olv\_hozza()** - Ezzel a függvénnyel tudunk új olvasókat hozzáfűzni a listához. Egyetlen paramétere a lista kezdőelemére mutató pointer. Visszatérési értéke az új listába fűzött elemre mutató pointer

**olv\_torl()** - Ezzel a függvénnyel lehet a listából elemeket törölni. Megkapja a lista első elemére mutató pointert és a függvényben megadott paraméterek alapján megkeresi a listában az adott elemet és felszabadítja. Nincs visszatérési értéke

**olv\_felsz()** – Az olvasok listát szabadítja fel a program futásának végén. Paramétere a lista első eleme

**ujSorKi()** – A beolvasott sztringek végéről eltávolítja az új sor ('\\n') karaktert

**KisBetu()** – A beolvasott sztringeket kisbetűssé alakítja, annak érdekében, hogy az elemekre bármilyen formában hivatkozni lehessen

## file.c és file.h

Itt történik az összes fájlkezeléssel kapcsolatos művelet. Itt találhatóak a ki- és beolvasással kapcsolatos függvények

**beo\_konyv()** – Ez a függvény olvassa be a „konyvek.txt” fájlból az adatokat és elmenti egy dinamikus láncolt listába. Nincs bemenő paramétere, visszatérési értéke az új lista első elemére mutató pointer

**beo\_olv()** - Ez a függvény olvassa be az „olvasok.txt” fájlból az adatokat és elmenti egy dinamikus láncolt listába. Nincs bemenő paramétere, visszatérési értéke az új lista első elemére mutató pointer

**konyv\_ki()** – A program futásának végén a bemenő paraméterként kapott listát kiírja a „konyvek.txt” fájlba, pontosvesszővel elválasztva az adatokat. Nincs visszatérési érték

**olv\_ki()** - A program futásának végén a bemenő paraméterként kapott listát kiírja az „olvasok.txt” fájlba, pontosvesszővel elválasztva az adatokat. Nincs visszatérési érték

**reverse\_print()** – Ez a függvény azért felelős, hogy a listát fordított sorrendben (tehát az első beolvasottól az utolsóig) kiírja az adott fájlba

**reverse\_print2()** – Ez a függvény azért felelős, hogy a listát fordított sorrendben (tehát az első beolvasottól az utolsóig) kiírja az adott fájlba

[debugmalloc.h](#)

Ez a header fájl felel az esetleges memóriakezelési hibák észleléséért és kiírásáért