

Receptes könyv – programozói dokumentáció

Project felépítése, fájlok:

A program összesen 4 .c típusú és 4 .h típusú fájlból áll össze.

- **C fájlok:**
 - *main.c*
 - *kislista.c*
 - *nagylista.c*
 - *fuggvények.c*
- **Header fájlok:**
 - *debugmalloc.h*
 - *kislista.h*
 - *nagylista.h*
 - *fuggvények.h*

A *Receptes könyv* egy konzolos alkalmazás, nem igényel semmilyen grafikus könyvtárat. A szükséges C könyvtárak a következők: *stdlib.h*, *stdio.h*, *string.h*, *ctype.h*. Az indításhoz elégséges az .exe fájl megnyitása, nem igényel a program előre létrehozott .txt fájlokat.

Adatszerkezet:

Az adatokat láncolt listákban tárolja a program. Van egy nagyobb lista (*NagyLista*), amely tárolja az összes beolvasott és hozzáadott receptet. Ennek egy eleme lesz egy teljes recept minden adatával. Ez a lista egy duplán láncolt, strázsás lista, ami megkönnyíti a receptek (*ListaElem*) törlését és hozzáadását. Mindegyik listaelem tartalmaz egy kisebb, egyszerűen láncolt listát (*KlistaElem* – továbbiakban *KisLista*), amely a hozzávalókat és annak mennyiségeit tárolja.

- ***ListaElem* struktúra:**
 - sorszam: a recept sorszáma
 - nev: a recept neve, egy pointer, ami egy dinamikusan foglalt karaktertömbre mutat majd
 - hozz_menny: egy *KisLista* elejére mutató pointer, a hozzávalókat és mennyiségeket tárolja
 - elkeszites: a recept elkészítése, egy pointer, ami egy dinamikusan foglalt karaktertömbre mutat majd
 - elozo és kov: pointerok, amik mutatnak majd az előző és következő *ListaElem*-re
- ***NagyLista* struktúra:**
 - elso és utolso: két *ListaElem* típusú pointer, ezek lesznek a strázsák
- ***KlistaElem* struktúra:**
 - hozzavalo: egy hozzávaló, egy pointer, ami egy dinamikusan foglalt karaktertömbre mutat majd
 - mennyiseg: a hozzávalóhoz tartozó mennyiség, egy pointer, ami egy dinamikusan foglalt karaktertömbre mutat majd
 - kov: egy *KlistaElem* típusú pointer, ez mutat majd a következő *KlistaElem*-re

A main.c függvényei:

Mindegyik menüpont tartalmaz egy almenüt, amely az azonnali visszalépést a főmenübe, vagy az adott menüpont megismétlését teszi lehetővé.

void kereses_recept_neve(NagyLista *rec_konyv):

Ez a függvény a *NagyLista* neveiben keres. Ha a begépelt karaktersorozat bármelyik recept nevében előfordul, azt a receptet kilistázza névvel és sorszámmal.

- Használt függvények a függvényben belül:
 - o *hosszu_sort_olvas*
 - o *reszletek*

void nincs_otletem(NagyLista *rec_konyv):

Három darab random recept nevet és sorszámot listáz ki ajánlatképpen. Ha az adatbázisban még nincs legalább 3 recept, akkor figyelmeztet erre és visszalép az almenübe.

- Használt függvények a függvényben belül:
 - o *reszletek*

void de_ennek_egy_kis(NagyLista *rec_konyv):

Ez a függvény egy bekért hozzávalót keres a *NagyLista* elemeinek hozzávalóiban, és ha egyezést talál az adott receptet névvel és sorszámmal kiírja.

- Használt függvények a függvényben belül:
 - o *hosszu_sort_olvas*
 - o *hozzavalo_keres*
 - o *reszletek*

void el_kell_hasznalni(NagyLista *rec_konyv):

Itt több bekért hozzávaló alapján keres a függvény a *NagyLista* elemeinek hozzávalóiban, és ha mindegyik hozzávaló megtalálható egy receptben, akkor az adott receptet kiírja névvel és sorszámmal ellátva.

- Használt függvények a függvényben belül:
 - o *el_kellHozzavalok*
 - o *hozzavalo_keresTobb*
 - o *reszletek*
 - o *felszabadit_kisLista*

NagyLista *recept_hozzaad(NagyLista *rec_konyv, KlistaElem *beolv_hozzEleje):

Ez a függvény intézi az új receptek felvételét, egy új *ListaElemet* hoz létre, melyet a bekérések végén majd az átadott *NagyListához* fűz, és visszaadja azt.

- Használt függvények a függvényben belül:
 - o *hosszu_sort_olvas*
 - o *hozzafuz*
 - o *mentes*

NagyLista *recept_torlese(NagyLista *rec_konyv):

Az átadott *NagyListából* majd a módosított listát visszaadja. Névvvel és sorszámmal kilistázza az összes mentett receptet, majd a megadott sorszámút eltávolítja.

- Használt függvények a függvényben belül:
 - *torles*
 - *mentes*

A kislista.h:

Ez a header fájl tartalmazza a típusdefinícióját a *KlistaElemnek* és az összes *kislista.c*-beli függvény deklarációját.

A kislista.c függvényei:

KlistaElem *beolvasKisLista():

1. Megvizsgálja létezik-e már a „hozzavalok.txt”, ha igen a 2. pontban leírtak alapján megy tovább, ha nem létrehozza a fájlt, és beleír egy vég jelet („##”).
2. Ha már létezik a „hozzavalok.txt”, akkor a *hosszu_sort_olvas_FILE* függvénnyel soronként beolvassuk azt a vég jelig. Minden beolvasást követően egy új *KlistaElem*-et hoz létre, amibe most mennyiségeket nem írunk, csak a sorszámot és a hozzávaló nevét használjuk. Majd az elemekből felépít egy egyszerűen láncolt *KisListát* az *uj_elem_hozzafuz* függvénnyel.

void felszabadit_kisLista(KlistaElem *lista):

Felszabadítja a paraméterként kapott *KisListát*.

void hozzavalok_mentes(KlistaElem *lista):

Kiírja a „hozzavalok.txt”-be a paraméterként kapott *KisListát*.

KlistaElem *megvizsgal(KlistaElem *lista, KlistaElem *szo):

Megvizsgálja, hogy a paraméterként kapott *KisListában* benne van-e a másik paraméterként kapott elem. Ha nincsen, akkor hozzáfűzi azt és visszatér a kapott *KisListával*.

KlistaElem *egy_elem():

Bekér a felhasználótól egy hozzávalót és egy mennyiséget, majd ezt visszaadja egy újonnan létrehozott *KlistaElem*-ben.

KlistaElem *uj_elem_hozzafuz(KlistaElem *hozz_menny, KlistaElem *uj):

A paraméterként kapott *KisListához* hozzáfűz egy szintén paraméterként kapott új *KlistaElem*-et. Visszatér a *KisListával*.

KlistaElem *fajlbol(KlistaElem(*beolv_hozz):

Kilistázza a paraméterként kapott *KisList*a hozzávalóit és sorszámát, majd a felhasználó által megadott sorszámú elemhez egy mennyiséget is bekérve a felhasználótól visszaadja azt.

A nagylista.h:

Ez a header fájl tartalmazza a típusdefinícióját a *NagyListának* és a *ListElemnek*, ezen felül a *nagylista.c* függvényeinek deklarációját.

A nagylista.c függvényei:

NagyLista *mentes_beolvas():

1. Létrehoz egy új *NagyListát*, amely később a vissza térési értéke lesz a függvénynek. Ez egy duplán láncolt, strázsás lista, *ListaElem* típusú elemekkel.
2. Először is megvizsgálja létezik-e már az adatbázisunk, azaz a „receptek.txt”, ha nem, máris létrehozza a fájlt és bele ír egy „###” vég jelet, ami a későbbiekben megállítja majd a visszaolvasást. Miután létrejött a fájl a 2. pontban leírtak alapján történnek a dolgok.
3. Ha már létezik a „receptek.txt”, akkor megnyitja azt, és beolvassa a **hosszu_sort_olvas_FILE** függvény segítségével, egy újonnan létrehozott *ListaElem*-be, amit hozzáfűz a létrehozott *NagyListához*, a **hozzafuz** függvény segítségével. Addig olvas a fájlból, amíg meg nem találja a vég jelet.
4. Egy *ListaElem* beolvasásánál mindig épít egy *KisListát* is. Amíg el nem éri a „#hozzv” végjelet, addig a hozzávalókat és a mennyiségeket olvassa be egy adott recepthez.

void felszabadit_egesz(NagyLista *rec_konyv):

Felszabadítja a teljes beolvasott adatbázist, paraméterként egy *NagyLista* típusú listát kell megadni.

void mentes(NagyLista *rec_konyv):

Kiírja a teljes adatbázist a „receptek.txt”-be, azaz elmenti az esetlegesen megváltoztatott *NagyListát*. Paraméterként egy ilyen típusú pointert kell megadni.

void hozzafuz(NagyLista *rec_konyv, ListaElem *uj):

Hozzáfűz a paraméterként kapott *ListaElemet*, a szintén paraméterként kapott *NagyLista* végére.

A fuggvenyek.h:

A *fuggvenyek.c* fájl függvényeinek deklarációját tartalmazza.

A fuggvenyek.c függvényei:

char *hosszu_sort_olvas() és char *hosszu_sort_olvas_FILE(FILE *file):

(Forrás: <https://infoc.eet.bme.hu/lab08/#4>)

Egy tetszőleges hosszú karaktersorozatot olvas be, majd egy dinamikusan foglalt karaktertömbre mutató pointert ad vissza. A **hosszu_sort_olvas** ezt a felhasználótól kéri, míg a **hosszu_sort_olvas_FILE** egy paraméterként megadott fájlból olvas.

void hozzavalok_keres(ListaElem *aktualis, char *keresett, char *seged):

Ez a függvény egy megadott *ListaElem*-ben a hozzávalók és mennyiségek listájában keresi a megadott szavakat. A *keresett* a keresett hozzávaló, a *seged* ugyan az a hozzávaló, csak nagy

kezdőbetűvel. Ha az adott *ListaElem*ben megtalálható a keresett hozzávaló kiírja a sorszámát és a nevét.

bool hozzavalokeresTobb(ListaElem *aktualis, char *keresett):

Az előző függvényhez hasonlóan ez is hozzávalókat keres egy megadott *ListElem*ben, de itt egy egész *KisListányi* keresett hozzávalót adunk át. A keresés úgy történik, hogy a *ListaElem* hozzávalóin megyünk egyesével, és minden hozzávalónál megvizsgáljuk, hogy a *keresett* lista valamelyik hozzávalója megegyezik-e vele. Ha igen egy számlálót növel és a végén megvizsgálja, hogy a találatok száma megegyezik-e a *keresett* elemszámával. Ha igen *true*, ha nem *false* a visszatérése a függvénynek.

void reszletek(NagyLista *rec_konyv):

Ez a függvény egy recept részleteinek megjelenítésére hívatott. A felhasználótól bekér egy sorszámot, és az azzal megegyező sorszámú recept részleteit kiírja a képernyőre.

KlistaElem *el_kellHozzavalok():

Folyamatosan hozzávalókat kér be a felhasználótól, mindaddig, amíg nem '0' karaktert nem talál. Ezekből egy *KisLitát* épít, és ezt adja vissza.

void torles(NagyLista *rec_konyv):

Bekért sorszám alapján kiveszi a *NagyListából* és felszabadítja az ezzel megegyező sorszámú *ListaElem*et.

KlistaElem *almenuFGV(KlistaElem *kisList_uj, KlistaElem *beolv_hozzEleje, int almenu):

Segédfüggvénye a *recept_hozzadnak*. Meghívott függvények: *uj* és a *fajl*.

KlistaElem *uj(KlistaElem *kisList_uj, KlistaElem *beolv_hozzEleje):

Egy új hozzávalót kér be a felhasználótól és adja hozzá kislistához. Meghívott függvények: *uj_elem_hozzafuz* és *megvizsgal*.

KlistaElem *fajl(KlistaElem *kisList_uj, KlistaElem *beolv_hozzEleje):

A hozzávalók text fájlból visszaolvasott alapanyagok közül a felhasználó által választottat adja hozzá a kislistához. Meghívott függvények: *fajlbol* és *uj_elem_hozzafuz*.