

Fájlkezelés

A program a könyvtárban lévő „recepteskönyv” nevű txt fájlból olvassa be adatokat és ugyan ide írja ki őket a futása végén. A fájl elején, az első sorban az összes eddig használt hozzávalót írja ki a következő módon:

|név-sorszámmértékegység

Csak itt vannak elmentve, hogy mi a sorszámhoz tartozó név és mértékegység.

A fájl többi sora mind a recepteket tartalmazza a következő felépítésben:

-

recept neve

|1-2|3-100 //(első szám a „|” után a hozzávaló sorszámát jelöli, a második szám a „-” után a mennyiséget)

recept leírása //(egy sorban)

Saját fájlok

Felhasznált „.h” fájlok (könyvtári): stdio.h, stdlib.h, stdbool.h, string.h, time.h

Saját „.c” fájlok: main.c, fajliro.c, lefogjarni.c, meglevokBE.c, receptkezelo.c, szovegbekero.c

Hozzájuk tartozó saját „.h” fájlok: fajliro.h, lefogjarni.h, meglevokBE.h, receptkezelo.h, szovegbekero.h

Main felépítése

Itt a main-en kívül egy függvény található meg csak. A mainben legelőször létrehozok egy Recept pointert ami a recept lista elejére mutat majd, illetve egy Összetevő pointert ami ugyancsak az összetevők lista elejére fog mutatni, majd meghívom rájuk a hozzájuk tartozó beolvasó függvényt. Leellenőrzöm, hogy nem nullpointert adtak-e vissza.

A programban lévő függvények szinte mindegyike ezt a kettő pointert kapja meg, amelyik nem, arra kitérek külön.

Ez után, elindítja a fő menüt ami a ciklusából csak „0” bemenetre fog kilépni, elmenteni a kiolvasott adatokat és leállni a futással.

Hivatalosan 4 menüpont van a főmenüben: receptkezelő, új recept létrehozó, régi recept törlő és kilépés.

A mainben található függvény a program másik menüje, almenü amit a receptkezelő nyit meg a főmenüből. a felépítése hasonló. A válaszul kapott „0” esetén lép ki (vissza a főmenübe) és ezen kívül 4 menüpontja van: recept keresése, random recept kérése, „úgy ennék” funkció, ami 1 megadott összetevő alapján keres és „le fog járni” opció ami 1 vagy több összetevő alapján keres.

MeglevőBE

.h fájl: Itt van létrehozva a program 2 fő változója:

1. összetevő:

- char név pointer ami majd egy szövegre fog mutatni
- szorszám egész ami a fájlban lévő sorszámmal lesz egyenlő
- char mérték pointer ami az egy szövegre fog mutatni
- összetevő pointer a listába fűzés miatt

2. recept:

- char név pointer ami majd egy szövegre fog mutatni
- hozzávaló egészekből álló mátrix (2x30-as elegendő) amibe a hozzávaló sorszámát és a kellő mennyisége kerül
- char leírás pointer ami az egy szövegre fog mutatni
- recept pointer a listába fűzéshez

Kettő függvény található itt: meglevokBE és osszetevoBE

.c fájlban található a fentebb említett meglevoBE és osszetevoBE függvény. Az előbbi a main-ben található recept pointert kapja meg, az utóbbi az összetevő pointert.

MeglevoBE függvény

Megnyitja a fájlt, ha nem tudta null-lal tér vissza. Kihagyja az első sort, ami csak az összetevőkből áll. Ciklussal megkeresi a fájl végén álló entert. Ezen belül: létrehoz egy új recept pointert, amit dinamikusan foglalt (ezt tölti fel az adatokkal:), beolvassa a „-” jelet, ami elválasztja a recepteket. Meghívja a szovegbekérő.c –ben lévő fajlszoveg függvényt, hogy olvasson be egy szöveget (dinamikusan foglalt), ami elnyeli a recept neve utáni entert is (nem áll meg a fő ciklus). Maga olvassa be a recepthez tartozó hozzávalók sorszámát és mennyiségét. Az utolsó hozzávaló után rak egy -1-et a hozzávaló sorszámokhoz lezárásként. Újra meghívja a fajlszoveg függvényt, hogy a leírást is beolvassa. A kapott receptet felfűzi a kapott recept pointer lista végére. Bezárja a fájlt és visszatér a készített lista eleje pointerével.

OsszetevoBE függvény

Megnyitja a fájlt, ha nem tudta null-lal tér vissza. Az első sor végén lévő enterig ciklust indít (ezzel bekebelez egy betűt minden új körben). Új osszetevo pointert hoz létre, amit feltölt adattal, dinamikusan lefoglal neki helyet. Dinamikus szövegbeolvasást indít, ami a „-” jelig tart, ez a név. Beolvassa a sorszámát az összetevőnek. Dinamikus szövegbeolvasást indít, ami a „|” jelig tart, ez a mértékegység. A kapott összetevő pointer végére fűzi az új összetevőt, listát készít. A kapott eleje pointert adja vissza, amin lista lóg.

Fajliro

.c és a .h fájl is egy függvényt tartalmaz.

Függvény:

Megnyitja a txt fájlt, ha nem sikerül -1-el visszatér. Először az összetevőket írja ki a fájlba egy sorba mind: végig lépeget egy ciklussal az összetevő listán és minden fájlba írt adatot felszabadít (persze ami dinamikusan volt foglalva). Mindezt úgy, hogy a megfelelő formátumot tartja (bekebelezett betű).

Következő ciklusban a receptek listán megy végig, ahol ugyancsak minden kiírt adatot felszabadít és tartja a formátumot. Majd a végén, bezárja a fájlt és visszatér nullával.

Szovegbekero

Kettő függvényt tartalmaz. Szinte ugyan azok, egyszerű dinamikus szövegolvasók. A szovegbekero függvénynek nincs bemenete, csak egy pointert ad vissza, amibe dinamikusan foglalta a memóriát és beletöltötte a felhasználótól származó szöveget (gépet).

A fajlszoveg függvény viszont a txt fájlból olvas be, amit a bemenetén megkap a hívótól, illetve a pointert is megkapja, amit fel kell töltenie. Ugyancsak dinamikusan foglalt.

Le fog járni

.h fájlban csak egy függvény van, amit az almenüből lehet meghívni.

.c fájlban ezt a függvényt 4 másik egészíti ki: listamasolRecept, lefogjarniseged, listarolnevkiiro, felszabadito.

Recept *listamasolRecept(Recept *eleje, Recept *ujrec)

Létrehoz egy új recept pointert (dinamukusan foglalja) és belemásolja a kapott ujrec pointer tartalmát, majd felfűzi a kapott eleje pointert rá és visszaadja a létrehozottat (listát készít, elejére fűz).

Recept *lefogjarniseged(Recept *ujeleje, Recept *regieleje, Osszetevo *meglevoosszetevo, Osszetevo *mozgo)

Fealdata: végigmegy a kapott regieleje listán és megkeresi benne azokat a recepteket, amelyekben megvan a mozgo által mutatott sorszámú összetevő. Ezeket a megtalált recepteket odaadja a listamasolRecept függvénynek, hogy az ujeleje függvény elejére fűzzen receptet. Visszaadja az ujeleje pointert (a listát).

void listarolnevkiiro(Recept *eleje, int eng, Osszetevo *meglevoosszetevek)

Kettő funkciója van, a kapott eng egész alapján dől el, mit ír ki.

(eng ==0) A kapott listán lépeget végig és írja ki a benne tárolt receptek (itt) generált sorszámát és nevét.

(eng !=0) A kapott listán lépeget végig és ha annak a generált sorszáma megegyezik az eng értékével, meghívja erre a receptkiro függvényt. Nem ad vissza semmit.

void felszabadito(Recept* eleje)

A kapott listán végiglépeget és felszabadítja az összes elemét.

void lefoglarni(Recept *meglevorecept,Recept *ujkor, Osszetevo *meglevoosszetevek)

Ha ez az ujkor pointer nem nulla, az általánosan (általa) használt receptlista ez lesz. Bekér a felhasználótól egy összetevőt, meghívja a beolvasásra a szovegbekero függvényt. megkeresi, hogy létezik-e ez az összetevő. Ha nem, kilép. Ha megtalálta, felszabadítja az előbb bekért szöveget, létrehoz egy új recept pointert és meghívja rá a lefoglarniseged függvényt. Kiírja, a menüje alap lehetőségeit (reset, új hozzávaló adása, kilépés) és meghívja a receptkiro függvényt. Ciklust kezd (menüje). Ha resetelni akarja a hozzávalókat a felhasználó, meghívja magát újra, hogy a második recept pointere nulla legyen. Ha új hozzávalót szeretne a felhasználó a kereséshez adni, meghívja magát megint, de második recept pointernek a már kiválogatott receptek listát adja, így egy meghívás alatt csak 1 összetevő alapján kell keresni. Nem érvényes választ két helyen ellenőriznek a függvények: a menünél, ha -2-nél kisebb számot kapott, illetve a listarolnevkiiro ha nem nulla a kapott szám és nem is egyezik meg egy generált számmal sem. Ha érvényes számot adott meg a felhasználó, meghívja a listarolnevkiiro függvényt és odaadja neki a választ. Ha a felhasználó ki akar lépni, meghívja a felszabadító függvényt. Nem ad vissza semmit.

Receptkezelő

.h fájlban 8 függvény van: receptkiro, receptkereso, randomrecept, recepttorlo, hozzavalohozzaad, ujrecept, EE, ugyennek.

.c

void receptkiro(Osszetevo *meglevoosszetevek,Recept *kivalasztott)

Szépen elválasztva kiírja a kapott recept nevét, kikeresi a hozzávalókat, azoknak a nevét, mennyiségét és mértékegységét kiírja, végül az elkészítési módot is kiírja. Nincs visszatérési értéke.

void /*Z1*/receptkereso(Osszetevo *meglevoosszetevek,Recept *meglevorecept)

Bekéri a felhasználótól a keresett recept nevét, szovegbekero-t meghívva (dinamikusan foglalt), majd megkeresi a recept listán a kapott szöveget. ha egyezik. kiírja a receptkiroval. Ha nem, felajánlja (saját menüt használva - ciklussal-) a következő lehetőségeket: létrehozás – itt meghívja az ujrecept függvényt-, új keresés –itt meghívja magát újra-, vagy kilépés. Végül felszabadítja a bekért szöveget és nincs visszatérési értéke. Érvénytelen számot felismer.

Recept *recepttorlo(Osszetevo *meglevoosszetevek,Recept *meglevorecept)

Bekéri a felhasználótól a törölni kívánt receptet. Ehhez a szovegbekero-t hívja meg. Indít egy nagy ciklust ami végig lépeget a listán. Ha megtalálja, kiírja a receptet a receptkiroval majd menüt ír ki: biztosan törölni akarja, nem (kilépés). Ha törölni akarja, felszabadítja a recept összes elemét és a listát visszazárja. Érvénytelen választ kezel.

EE – triviális

void /*Z2*/ randomrecept(Osszetevo *meglevoosszetevek,Recept *meglevorecept)

Random sorsol egy számot 1 és 100 között, majd a kapott listán annyit lép. Ha a lista végére ér, az elejéről folytatja.

A kisorsolt receptet kiírja a receptkiroval. Saját menüt ír ki, ha újat kér a felhasználó, meghívja magát. Érvénytelen választ kezel.

Osszetevo *hozzavalohozzaad(Osszetevo *meglevoosszetevok)

Létrehoz egy új összetevő pointert és dinamikusan foglal neki helyet. Két szöveg elemére meghívja a szovegbekero függvényt. Végiglépked a listán, megkeresi az utolsó sorszámot, és hozzáadja az új összetevő sorszámához (+1). A lista végére fűzi az új hozzávalót és visszaadja az elejét.

Recept* ujrecept(Recept *meglevorecept, Osszetevo *meglevoosszetevok)

Foglal egy új receptet, szovegbekeroval beolvassa a nevét. Kiírja az összes eddig ismert hozzávalót. Ha innen választ a felhasználó, bekéri a mennyiségét, ha nem, meghívja a hozzavalohozzaad függvényt. Szovegbekeroval beolvassa a leírást. Minden adatot az új receptbe rak bele. Az új receptet a meglevoreceptek lista végére fűzi, majd visszaadja a pointert.

void ugyennekseged(Recept *meglevorecept, Osszetevo *meglevoosszetevok, Osszetevo *mozgo, int valasz)

Szinte ugyan úgy működik, mint a lefogjarniseged, csak itt nem listával dolgozik. Ha megvan a mozgo által mutatott összetevő sorszáma, kiírja a nevét a receptnek (+gsz), illetve ha a valasz nem nulla, akkor a megfelelő receptre ráhívja a receptkiro-t.

void ugyennek(Recept *meglevorecept, Osszetevo *meglevoosszetevok)

Majdnem úgy működik, mint a lefogjarni függvény csak lista nélkül. A megtalált összetevővel meghívja a lefogjarniseged-et és a megfelelő választ kapva a menüben: kiírat, kilép vagy érvénytelen-t ír ki.