

A programom felépítése:

.c és .h fájlok:

- main.c
 - A main függvény beállítja a konzolt utf-8ra és meghívja a főmenüt.
- osszetevok.c + osszetevok.h
 - Ebben a fájlban találhatóak az összetevőket kezelő függvények.
- keres.c + keres.h
 - Ez a fájl tartalmazza a program 4 kereső függvényét.
- uf.c + uf.h
 - Az „univerzális függvénykönyvtár” . Ebben a fájlban olyan kicsi, pár soros függvények vannak, amit több modulban is használok.
- recept.c + recept.h
 - Ebben a fájlban találhatóak a recepteket kezelő függvények.
- menu.c + menu.h
 - A menü modul hívja meg a többi modul függvényeit és teszi lehetővé a köztük való navigálást.

Kiegészítő fájlok:

- receptek.txt
 - Ez a text fájl tárolja a recepteket a következőképpen:
 - Név;összetevő1,menyiség?összetevő2,menyiség;leírás1?leírás2?;
 - Megjegyzés: A program nem engedi, hogy ez a fájl üres legyen. Az üres fájl a program összeomlását okozza.
- osszetevok.txt
 - Ez a fájl tárolja az összetevőket ilyen formában:
 - Összetevő1\nÖsszetevő2\n
 - Megjegyzés: A program nem engedi, hogy ez a fájl üres legyen. Az üres fájl a program összeomlását okozza.

Kiegészítő könyvtárak:

A saját moduljaimon kívül a következő standard C könyvtárakat használom.

- ❖ <time.h>
- ❖ <windows.h>
- ❖ <stdio.h>
- ❖ <stdlib.h>
- ❖ <stdbool.h>
- ❖ <string.h>

Függvények:

A programomban az alábbi függvények vannak:

➤ *Menü függvények:*

- void fomenue(void);
 - **Bemenet:** Egész számok a függvényen belül
 - **Kimenet:** Nincs
 - **Célja:** Navigálás az almenük között.
- void almenu_osszetevo();
 - **Bemenet:** Egész számok a függvényen belül
 - **Kimenet:** Nincs
 - **Célja:** Navigálás az összetevő függvények között.
- void almenu_uj();
 - **Bemenet:** Egész számok a függvényen belül
 - **Kimenet:** Nincs
 - **Célja:** Navigálás az új recepthez tartozó funkciók között.
- void almenu_recept(void);
 - **Bemenet:** Egész számok a függvényen belül
 - **Kimenet:** Nincs
 - **Célja:** Navigálás a recept függvények között.
- void almenu_keres(void);
 - **Bemenet:** Egész számok a függvényen belül
 - **Kimenet:** Nincs
 - **Célja:** Navigálás a kereső függvények között.

➤ *Összetevő függvények és adatszerkezetek:*

- typedef struct Osszetevo {char **o_lista;int meret;} Osszetevo;
 - **Változók:** o_lista = Az összetevők listája, kétdimenziós sztringeket tároló tömb. meret = a lista mérete
 - **Célja:** Egy dinamikus tömb ami az össszetevőket és a lista méretét tárolja
- void uj_osszetevo(Osszetevo *lista,int meret);
 - **Bemenet:** Az osszetevo_lista által létrehozott lista memóriacíme és a hozzátartozó méret.
 - **Kimenet:** Nincs
 - **Célja:** Egy új összetevő hozzáadása a listához.
- Osszetevo osszetevo_lista();
 - **Bemenet:** Nincs
 - **Kimenet:** Egy Osszetevo típusú dinamikus tömb.
 - **Célja:** Kiolvassa az „osszetevok.txt” fájlban tárolt adatokat és egy dinamikus tömböt csinál belőle.
- void osszetevot_felszabadit(char **osszetevok, int meret);
 - **Bemenet:** Egy kétdimenziós tömb és a mérete.
 - **Kimenet:** Nincs
 - **Célja:** Felszabadít egy kétdimenziós tömböt.
- void osszetevot_torol(Osszetevo *lista,int meret);
 - **Bemenet:** Az osszetevo_lista által létrehozott lista memóriacíme és a hozzátartozó méret.

- **Kimenet:** Nincs
- **Célja:** Egy összetevő törlése a listából.
- void osszetevo_t_fajlba_ir(Osszetevo lista);
 - **Bemenet:** Az osszetevo_lista által létrehozott lista másolata.
 - **Kimenet:** Nincs
 - **Célja:** A dinamikus tömböt beleírja az „osszetevok.txt” fájlba.
- **Recept függvények és adatszerkezetek:**
- typedef struct Recept{char nev[52];char **o_lista;int o_meret;char **ml;char **el_lista;int el_meret;struct Recept *kov;} Recept;
 - ❖ **Változók:**
 - ❖ nev = A recept neve.
 - ❖ o_lista = Az összetevők listája, kétdimenziós sztringeket tároló tömb.
 - ❖ o_meret = Az o_lista mérete.
 - ❖ ml = A mértékegységek listája, kétdimenziós sztringeket tároló tömb.
 - ❖ el_lista= Az elkészítési lépések listája, kétdimenziós sztringeket tároló tömb.
 - ❖ el_meret = Az el_lista mérete.
 - ❖ kov = A láncolt lista következő elemére mutató pointer.
 - **Célja:** Egy láncolt lista ami a receptekhez tartozó adatokat tárolja.
- void uj_recept(Recept **eleje);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** Nincs
 - **Célja:** Egy új recept hozzáadása a láncolt listához.
- char** elvalaszt(const char* string, const char* stop, int* hossz);
 - **Bemenet:** Egy string, egy delimiter karakter és a string hossza.
 - **Kimenet:** Az elválasztott stringek.
 - **Célja:** A „receptek.txt” fájlban tárolt adatok szétválasztása.
- void receptet_kiir(Recept *recept);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** Egy recept stdout-ra.
 - **Célja:** Egy recept kiírása a láncolt listából.
- void receptet_felszabadit(Recept **eleje);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** Nincs
 - **Célja:** Felszabadítja a láncolt listát.
- void recept_lista(Recept **eleje);
 - **Bemenet:** A láncolt lista elejének memóriacíme.
 - **Kimenet:** Közvetetten: Egy láncolt lista.
 - **Célja:** A „receptek.txt” fájlban levő adatokból egy láncolt listát készít.
- void receptet_fajlba_ir(Recept **eleje);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** A „receptek.txt” fájlba írt adatok
 - **Célja:** Beírja a láncolt listát a „receptek.txt” fájlba.
- void receptet_listaz(Recept **eleje);

- **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
- **Kimenet:** A láncolt listában levő receptek stdout-ra
- **Célja:** Kiírja az összes receptet.
- void recept_torol(Recept **eleje,int hanyadik);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme. A kiválasztott recept sorszáma.
 - **Kimenet:** Nincs
 - **Célja:** Kitöröl egy receptet a recepteket tároló láncolt listából.
- void recept_modosit(Recept **eleje,int mennyi);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme. A kiválasztott recept sorszáma.
 - **Kimenet:** Nincs
 - **Célja:** Egy receptet módosít a láncolt listában.
- int recept_szamolo(Recept **eleje);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** A receptek száma
 - **Célja:** Megszámolja a recepteket.
- **UF függvények:**
- int sorokat_szamol(char *fajl);
 - **Bemenet:** Egy fájl neve.
 - **Kimenet:** Az összetevők száma.
 - **Célja:** Megszámolja az összetevőket.
- void sztringet_nagybetusit(char *sztring);
 - **Bemenet:** Egy sztring.
 - **Kimenet:** Nincs
 - **Célja:** Minden betűt nagybetűsít.
- int valaszt_tesztel(char *valasz);
 - **Bemenet:** A felhasználó által megadott válasz string.
 - **Kimenet:** 1. Igent kapott, 0 Nemet kapott, -1 Hibás bemenet
 - **Célja:** User inputot tesztel.
- void listat_kiir(char **lista, int meret);
 - **Bemenet:** Egy kétdimenziós tömv és a mérete.
 - **Kimenet:** A lista elemei kiírva stdout-ra.
 - **Célja:** Kiírja egy lista elemeit.
- **Kereső függvények:**
- bool van_e_benne(char** o_lista,int o_meret,char *keres);
 - **Bemenet:** Egy kétdimenziós tömb és a mérete, illetve a keresett sztring.
 - **Kimenet:** true, ha megtalálta az elemet, false, ha nincs benne ilyen elem.
 - **Célja:** Megkeres egy sztringet egy kétdimenziós, sztringeket tároló tömbben.
- void innel_egy_kis(Recept **eleje,int recept_szam);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme és a receptek száma
 - **Kimenet:** Nincs.

- **Célja:** Megkeres egy összetevő sztringet a receptek között.
- void el_kell_hasznalni(Recept **eleje,int recept_szam);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme és a receptek száma.
 - **Kimenet:** Nincs.
 - **Célja:** Megkeres több összetevő sztringet a receptek között és azt listázza amelyikben mindegyik megvan.
- void nincs_otlet(Recept **eleje,int receptek_szama);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme és a receptek száma.
 - **Kimenet:** Nincs.
 - **Célja:** Random sorsol egy receptet.
- void nev_szerint(Recept **eleje);
 - **Bemenet:** Az recept_lista által létrehozott láncolt lista elejének memóriacíme.
 - **Kimenet:** Nincs.
 - **Célja:** Megkeres egy receptet nev szerint.

A projektem itt is elérhető: github.com/peterbalog4/NHF