

Programozás HF

Programozói dokumentáció

Adatstruktúrák:

```
typedef struct Etel {  
    char neve[64];  
    int ara;  
}Etel;
```

```
typedef struct Menu{  
    Etel *eloetel;  
    Etel *foetel;  
    Etel *desszert;  
    Etel *ital;  
    int eloetel_hossz;  
    int foetel_hossz;  
    int desszert_hossz;  
    int ital_hossz;  
}Menu;
```

```
typedef struct Rendeles {  
    Etel *ertek;  
    struct Rendeles *kov;  
} Rendeles;
```

```
typedef struct egy_Asztal {  
    int szama;  
    int ferohelye;  
    char helye[64];  
    Rendeles *rendelese;  
} egy_Asztal;
```

egy_Asztal – típusban definiálva van, egy asztal száma, férőhelyének száma, elhelyezkedése és rendelése, amely egy pointer, és rámutat az ahhoz az asztalhoz tartozó rendelések listájára.

Rendeles – struktúra a rendelések láncolt listaként tárolásához kell. Egy rendelés tartalmaz egy értéket és egy a következő rendelésre mutató pointert.

Etel – ez a struktúra azt írja le, hogy az egyes ételeknek van nevük és áruk is.

Menu – struktúrában definiálva van a négy fajta ételtípus, amelyek mind Etel típusú pointerok, hiszen ezek egy egy dinamikus tömbre mutatnak amelyben ételek vannak. Emellett pedig ezeknek a dinamikus tömböknek a hosszát is tárolja.

Függvények:

A program felépítése menükből és almenükből áll, amelyek függvények meghívásával van megoldva. A main függvényen belül a két legfőbb függvény a beallitas_mod() és a felhasznalo_mod().

Beállítás mód:

Az asztalokhoz kapcsolódó függvények:

```
void beallitas_mod(egy_Asztal **asztaltomb, int *hossz, Menu *menutomb)
```

Ezzel a függvénnyel a beállítás módba tudunk lépni. Paraméterként kap egy az asztaltombbre mutató pointerre mutató pointert, ennek az asztaltombnek a hosszára mutató pointert és az ételekre mutató pointert. Nincs visszatérési értéke hiszen csak arra használjuk, hogy egy menüpontba lépjünk át.

Ezen a módon belül lehet az asztalokat megadni az asztalok_megadasa() függvénnyel.

```
egy_Asztal *asztalok_megadasa(int *hossz)
```

Ez a függvény egy dinamikus tömbben tárolja az egy_Asztal típusú asztalok beolvasott adatait, majd ki is írja ezeket az asztal.dat nevű fájlba. A beolvasás végét az „end” szó beírásával tudjuk jelezni.

Fontos, hogy csak ékezet nélküli adatokat tud beolvasni és elmenteni fájlba a program. A visszatérési értéke egy asztaltömbbre mutató pointer.

```
void fajlbol_beolvas(egy_Asztal **asztaltomb, int *hossz, Menu *menutomb)
```

Az asztalok és a menü beolvasását külön függvények végzik el amelyeket ezen a függvényen belül tudunk meghívni. Tehát ez a függvény arra van, hogy egy almenün keresztül beolvashassuk az asztalokat és a menüt.

```
egy_Asztal *asztalok_beolvas(int *hossz)
```

Ez a függvény olvassa be az asztal.dat fájlból az asztalok adatait, ha azok már egyszer meg lettek adva.

A menühöz kapcsolódó függvények:

```
Menu menu_megadasa()
```

Ezen a függvényen keresztül olvassuk be a menüben szereplő ételeket és mentjük el fájlba ételtípusonként (előételek, főételek, desszertek és italok)

```
Etel *etel_beolvas(char *eteltipus, int *db)
```

Beolvassa a felhasználó által megadott ételeket az egyes ételtípusok tömbjébe, és tárolja a darabszámát is. Csak ékezet nélküli ételeket tud beolvasni. A beolvasás végét az „end” szó beírásával lehet jelezni. A menu_megadasa függvényben hívjuk meg. Visszatérési értéke egy ételtömbbre mutató pointer.

```
void mentes_fajlba(FILE *file, char *tipus, int db, Etel *lista)
```

Elmenti a menu.dat nevű fájlba az egyes eteltomboket. A menu_megadasa függvényben hívjuk meg.

```
Menu menu_beolvas()
```

Fájlból beolvassa menutípusonként az ételek tömbjét ha azt már egyszer megadta a felhasználó. Ezt a fajlbol_beolvas függvényben hívjuk be.

Felhasználói mód:

A program csak, akkor engedi a felhasználó számára, hogy belépjen a ebbe a módba ha már megadott vagy beolvasott adatokat.

Asztalfoglalásokhoz kapcsolódó függvények:

```
typedef struct Foglalas{
```

```
    char datum[12];
```

```
    char nev[64];
```

```
    char idopont[6];
```

```
}Foglalas;
```

Ez a struktúra nem függ a többitől, csupán definiálja, hogy egy asztalfoglalásnak van neve, dátuma és időpontja. A dátumot és az időpontot azért kezeli szöveggént mert nem használja fel a program ezeket máshol. Fontos, hogy a dátumot szóközök nélkül ÉÉÉÉ.HH.NN formában kell megadni az időpontot meg ÓÓ:PP-ben

```
void foglalas_megadassa()
```

Beolvassa a felhasználó által megadott adatokat és elmenti az asztalfoglalások.dat fájlba úgy, hogy mindig hozzáírja a megadott asztalfoglalást a fájlban eddig lévőkhöz.

```
void foglalasok_listazasa()
```

Kilistázza az eddigi foglalásokat.

Rendeléshez kapcsolódó függvények:

void egy_rendeles_megadasa(egy_Asztal *asztaltomb, int *hossz, Menu menu)

Eltárolja az egyes rendeléseit egy asztalnak láncolt listában. Tehát először azt kell megadni, hogy hányas asztalról van szó. Ha nincs ilyen asztal vagy nincs olyan étel a menüben amit beír a felhasználó akkor azt jelzi. Ha egy asztal mást is szeretne kérni, akkor könnyen hozzáadható az eddigi rendelésekhez az új a láncolt lista miatt.

Etel *etel_leker(Menu menu, char *szoveg)

Ez a függvény mondja meg, hogy az adott rendelés megegyezik-e a menüben lévő egyik fogással.

Számlához kapcsolódó függvény:

void szamla_kiallitasa(egy_Asztal *asztaltomb, int *hossz)

Kiírja egy adott asztal összes rendelésének nevét és árát és végül ezek összegét. Ha egy asztalnak kiállítottuk a számláját, akkor annak az asztalnak a rendelései automatikusan törlődnek.