

# Legyen ön is milliomos

NEPTUN: IKAR61; Név: Szász Dávid.

## Programozói Dokumentáció

### Adatok a programról

- **Modularitás:** A program moduláris felépítése elősegíti a kód karbantarthatóságát. Az egyes funkciók és logikák elkülönített fájlokban és funkciókban való implementálása megkönnyíti a hibák izolálását és a kód frissítését.
- **Memória-kezelés:** A programban a dinamikus memória-kezelés jelentős szerepet játszik. Fontos, hogy a program minden allokált memóriaterületet megfelelően felszabadítson a futás végén, hogy elkerülje a memóriaszivárgást.
- **Konzol Alapú Grafika:** Mivel a program konzol alapú, a grafikai megjelenítés korlátozott. A felhasználói felület egyszerű szöveges kiírásokon és menükön alapul.
- A program nem függ külső **könyvtáraktól**, csak a standard C könyvtáraktól.

### Main.c

A main.c a "Legyen Ön is Milliomos" játék belépési pontja, amely az alapvető inicializálást és a játékmenet indítását végzi. A fájl tartalmazza a main() függvényt, ami a program futásának kezdetét és a szükséges modulok betöltését irányítja.

### Kód Magyarázat

#### Include-ok:

- A program kezdete előtt több, a C standard könyvtárhoz és a játékhoz szükséges saját fejlesztésű könyvtárat (#include) importál. Ezek biztosítják a szükséges függvényeket és adattípusokat a játékhoz.
- #include <stdio.h>
- #include <stdlib.h>
- #include <stdbool.h>
- #include <windows.h>: Windows-specifikus fejléc, amely a Windows operációs rendszer API-jának hívásait tartalmazza. Szükséges lehet a konzolkezelés és más rendszer-specifikus funkciók miatt.
- #include <time.h>: Időkezelő funkciókat biztosít, mint például az aktuális idő lekérdezése és a srand() függvény a véletlenszám-generáláshoz.
- #include "lista.h": Egy saját fejlesztésű könyvtár. A dinamikus adatstruktúrák kezeléséhez szükséges függvényeket és típusdefiníciókat tartalmazza.
- #include "debugmalloc.h": A memóriaszivárgás és más memóriakezelési hibák felderítése.
- #include "file\_kezeles.h": Egy saját fejlesztésű könyvtár, amely a fájlkezeléssel kapcsolatos funkciókat tartalmazza.
- #include "jatek.h": Egy további saját fejlesztésű könyvtár, ami a játéklógikához kapcsolódó funkciókat és típusdefiníciókat tartalmazza.

### main() Függvény:

- **Inicializálás:**
- Beállítja a konzol karakterkódolását, hogy megfelelően kezelje az UTF-8 karaktereket.
- Inicializálja a véletlenszám-generátort a srand() függvény használatával, ami a játék során fellépő véletlenszerű eseményekhez szükséges.
- **Adatok Betöltése:**
- A file\_beolvas() függvény segítségével betölti a ranglistát és a kérdéseket a megfelelő fájlokból.
- **Játékmenet Indítása:**
- A jatekIndit() függvény meghívásával elindítja a játékot, ami a játékmenetet és a felhasználói interakciókat kezeli.
- **Adatok Mentése és Memória Felszabadítása:**
- A játék végén menti a ranglistát a ranglista\_mentese\_fileba() függvény segítségével.
- Felszabadítja a dinamikusan foglalt memóriaterületeket a lista\_free\_ranglista() és a lista\_free\_kerdesek() függvényekkel, hogy elkerülje a memóriaszivárgást.

## Lista.c és .h

A lista.c a "Legyen Ön is Milliomos" program egyik alapvető komponense, amely a dinamikus adatstruktúrák kezeléséért felelős. Ebben a modulban találhatók azok a függvények, amelyek lehetővé teszik a listaelemek létrehozását, manipulálását, keresését, és az adatok felszabadítását. Itt vannak definiálva a Lista.h – belüli függvények.

### Adattípusok

#### adat\_ranglista:

- Struktúra a ranglista elemek tárolására.
- Mezők:
- nev: Egy karakterlánc, amely a játékos nevét tárolja (max. 10 karakter).
- pontszam: Egy egész szám, amely a játékos pontszámát tárolja.
- ido: Egy egész szám, amely a játék során eltöltött időt tárolja.

#### adat\_kerdes:

- Struktúra a kvízkérdések tárolására.
- Mezők:
- szint: Az adott kérdés nehézségi szintjét jelölő egész szám.
- kerdes: Egy karakterlánc, amely magát a kérdést tárolja.
- a, b, c, d: Karakterláncok, amelyek a válaszlehetőségeket tárolják.
- helyes: Egy karakter, amely a helyes válasz betűjelét tárolja.

#### ListaElem:

- Általános célú lista elem struktúra a dinamikus adatkezeléshez.
- Mezők:
- adat: Általános célú pointer, ami az adatokra mutat.
- kov: ListaElem típusú pointer, amely a következő listaelemre mutat.

## Főbb Funkciók

### 1.lista\_hossza (ListaElem \*eleje):

- Cél: Meghatározza egy adott lista hosszát.
- Paraméterek: ListaElem \*eleje - a lista első elemére mutató pointer.
- Visszatérési érték: A lista hossza egész számként.

### 2.lista\_free\_ranglista (ListaElem \*eleje):

- Cél: Felszabadítja a ranglista által használt memóriaterületet.
- Paraméterek: ListaElem \*eleje - a felszabadítandó ranglista első elemére mutató pointer.
- Visszatérési érték: Nincs (void).

### 3.lista\_free\_kerdesek (ListaElem \*eleje):

- Cél: Felszabadítja a kérdések listájának memóriáját.
- Paraméterek: ListaElem \*eleje - a kérdések listájának első elemére mutató pointer.
- Visszatérési érték: Nincs (void).

### 4.lista\_elejere\_beszur (ListaElem \*eleje, void \*data):

- Cél: Új elemet szúr be a lista elejére.
- Paraméterek: ListaElem \*eleje - a lista első elemére mutató pointer, void \*data - az új elem adata.
- Visszatérési érték: Az új lista elejére mutató ListaElem pointer.

### 5.lista\_vegere\_beszur (ListaElem \*eleje, void \*data):

- Cél: Új elemet szúr be a lista végére.
- Paraméterek: ListaElem \*eleje - a lista első elemére mutató pointer, void \*data - az új elem adata.
- Visszatérési érték: Az új lista elejére mutató ListaElem pointer.

### 6.lista\_n\_elem (ListaElem\* eleje, int n):

- Cél: Lekéri a lista n-edik elemét.
- Paraméterek: ListaElem\* eleje - a lista első elemére mutató pointer, int n - az elem sorszáma a listában.
- Visszatérési érték: A keresett elemre mutató ListaElem pointer.

### 7.ranglista\_keres (ListaElem\* eleje, char s[]):

- Cél: Megkeresi a megadott nevű játékost a ranglistán.
- Paraméterek: ListaElem\* eleje - a ranglista első elemére mutató pointer, char s[] - a keresett játékos neve.
- Visszatérési érték: A keresett játékosra mutató ListaElem pointer, ha létezik.

### 8.rendez\_ranglista (ListaElem \*l\_ranglista):

- Cél: Rendezett ranglistát készít a meglévő adatok alapján.

- Paraméterek: ListaElem \*l\_ranglista - a rendezendő ranglista első elemére mutató pointer.
- Visszatérési érték: Az új, rendezett ranglistára mutató ListaElem pointer.

### 9.lista\_kerdesek\_keres (ListaElem\* l\_kerdesek, ListaElem\* p):

- Cél: Ellenőrzi, hogy egy adott kérdés szerepel-e már a listában.
- Paraméterek: ListaElem\* l\_kerdesek - a kérdések listájára mutató pointer, ListaElem\* p - a vizsgálandó kérdésre mutató pointer.
- Visszatérési érték: int, ami = 1, ha a kérdés már szerepel a listában, különben 0.

## File\_kezeles.c és .h:

### 1.beolvas\_mondat (FILE\* file):

- Célja: Egy fájlból beolvas egy mondatot a tab vagy az új sor karakterig és azt tárolja egy dinamikusan foglalt karakterláncba.
- Paraméterek: FILE\* file - a fájlra mutató pointer ahonnan beolvasunk.
- Visszatérési érték: Dinamikusan foglalt karakterlánc, amely a beolvasott adatokat tartalmazza.

### 2.file\_beolvas (ListaElem\*\* l\_ranglista, ListaElem\* l\_kerdesek[]):

- Célja: Adatokat olvas be fájlokból, a ranglistát és a kvízkérdéseket és láncolt listába helyezi az adatokat.
- Paraméterek: Címmel átvett paraméterek. ListaElem\*\* l\_ranglista - ranglista pointerére mutató pointer, ListaElem\* l\_kerdesek[] - a kérdéseket tartalmazó tömb.
- Visszatérési érték: Nincs (void). A függvény közvetlenül a megadott adatstruktúrákba tölti az adatokat.

### 3.ranglista\_mentese\_fileba (ListaElem \*l\_ranglista):

- Célja: A ranglista adatait menti egy fájlba.
- Paraméterek: ListaElem \*l\_ranglista - a mentendő ranglista első elemére mutató pointer.
- Visszatérési érték: Nincs (void). A fájlba írja az aktuális ranglista adatokat.

## Fileok

### Ranglista.txt

- A program az elején beolvassa. A játékosokat tartalmazza, nevüket és játszott idejüket (percben). A program kilépéskor ide menti le a ranglista láncolt listában lévő adatokat.
- Formátum: Név\tPont\tIdo\n
- Pl: David            10000   30

### Kerdesek.txt

- A program az elején beolvassa és ez a fájl nem frissül a program menete alatt. Egy kérdés nehezégi szintjét, a kérdést, a 4 lehetséges választ és a helyes válasz karakteret tartalmazza minden soron \t – al elválasztva.
- Formátum: szint\tkerdes\ta\tb\tc\td\thelyes\n;

- pl: 1    Hol található a csarnokvíz?    a pályaudvaron a szemben    a tejcsarnokban  
barlangfürdőben    B

## Jatek.c és .h:

A jatek.c fájl a "Legyen Ön is Milliomos" kvízzjáték logikájának és felhasználói interakciójának központi részét képezi. Ez a fájl tartalmazza az összes alapvető függvényt, amelyek a játékmenetet irányítják, beleértve a menük kezelését, a kérdések megjelenítését, a válaszok kezelését, és a játékállapot változásainak nyomon követését. A jatek.c fő célja, hogy egy koherens, interaktív és felhasználóbarát játékelményt nyújtson, ahol a játékosok különböző menüpontok között navigálhatnak, válaszolhatnak a kérdésekre, és nyomon követhetik saját előrehaladásukat.

**include-ok:** "lista.h", mivel használja a láncolt listákat kezelő függvényeket.

## Adattípusok

**enum Menuk:** Felsorolja a lehetséges menüpontokat, mint például Fomenu, Start, User, Ranglista, stb.

```
enum Menuk
{
    Fomenu = 1,
    Start,
    User,
    Ranglista,
    Jatek,
    JatekVege,
    Kilep,
    UjKerdes,
    AdatokMentese
};
```

**struct menu\_valtozoi:** Tárolja a játék állapotával kapcsolatos változókat, mint például a ranglistát, kérdéseket, felhasználói nevet, játék állapotát, pontszámot, stb.

```
typedef struct menu_valtozoi
{
    ListaElem* l_ranglista;
    ListaElem* l_kerdesek[16];
    ListaElem* l_kerdesek_voltak;
    ListaElem* p;
    ListaElem* rendezett_ranglista;
    adat_kerdes* adat;
    char v;
    enum Menuk menu;
    char user[11];
    int hossz, szint, hanyadik_kerdes, pontszam, jatszott_ido;
    bool helyes, nyert, valaszolt, kozonseg_segitseget_hasznalt, gep_segitseget_hasznalt,
    kozonseg_segitseget_megjelenit, show[5];
    int szazalek[5];
```

```
time_t start_t, veg_t;  
} menu_valtozoi;
```

## Függvények

### 1. valasz()

- Célja: Bekér egy karaktert a felhasználótól.
- Paraméterek: Nincsenek.
- Visszatérési érték: A felhasználó által megadott karakter.
- Kódmagyarázat: A függvény kiírja a "Válaszod: " szöveget, majd a scanf segítségével beolvassa a felhasználó választát. A getchar ciklus biztosítja, hogy az extra karakterek (pl. enter) ne zavarják a további bemeneti olvasást.

### 2. print\_fejlec()

- Célja: Kiírja a játék fejlécét.
- Paraméterek: Nincsenek.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A printf függvények sorozata egy előre meghatározott fejléct jelenít meg a képernyőn, amely a játék nevét és stílusát tükrözi.

### 3. kiir\_top\_10()

- Célja: Kiírja a ranglista top 10 játékosát.
- Paraméterek: ListaElem \*l\_ranglista - a ranglista első elemére mutató pointer. Ahoz, hogy a meghatározott célt hajtsa végre a függvény, ez a lista rendezett kell hogy legyen (erről gondoskodik a rendez\_ranglista függvény).
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: Egy ciklus bejárja a ranglista első 10 elemét (vagy annyit, amennyi létezik), és kiírja a játékosok nevét, pontszámát és játszott idejét.

### 4. teendokFomenu()

- Célja: Kezeli a főmenü interakcióit és választásait.
- Paraméterek: menu\_valtozoi\* m\_v - a játék aktuális állapotát tartalmazó struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény megjeleníti a főmenü opcióit, majd a valasz függvény segítségével bekéri a felhasználó választását, és a switch szerkezet alapján átvált a megfelelő menüpontra.

### 5. teendokStart()

- Célja: A játék indítása és a nehézségi szint kiválasztása.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A felhasználó választja ki a nehézségi szintet. A függvény inicializálja a játékkal kapcsolatos változókat, mint például a kérdések száma és a használt segítségék. Ezután a menüt az UjKerdes – re állítja.

### 6. teendokUser()

- Célja: Felhasználónév beállítása és a felhasználói adatok kezelése.
- Paraméterek: menu\_valtozoi \*m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A felhasználótól bekéri a becenevét, majd megnézi ha már létezik e a ranglistában. Frissíti a játékos aktuális pontszámát és játszott idejét.

## **7. teendokRanglista()**

- Célja: A ranglista kezelése és megjelenítése.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény kiírja a ranglistát (csak a 10 legjobb játékost), és lehetőséget biztosít a visszalépésre a főmenübe.

## **8. teendokUjKerdes()**

- Célja: Új kérdés kiválasztása a játék során.
- Paraméterek: menu\_valtozoi \*m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény véletlenszerűen választ egy új kérdést a megadott nehézségi szinten, figyelve arra, hogy a kérdés ne ismétlődjön. Inicializálja a változókat melyek kérdések között inicializálni kell (Ilyen pl. a bool Helyes, mely azt tárolja hogy helyesen válaszolt- e a játékos vagy nem).

## **9. teendokJatek()**

- Célja: A játék folyamatának kezelése.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény megjeleníti a jelenlegi kérdést és a válaszlehetőségeket, valamint kezeli a segítség használata és a válaszadás folyamatát.

## **10. teendokJatekVege()**

- Célja: A játék végének kezelése.
- Paraméterek: menu\_valtozoi\* m\_v, int prize[] - a játék állapotát tároló struktúra és a díjak tömbje.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény kiszámítja és kiírja a játékos végső pontszámát és játszott idejét, majd lehetőséget ad a főmenübe való visszalépésre.

## **11. teendokAdatokMentese()**

- Célja: A játék állapotának és a felhasználói adatoknak a mentése.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény frissíti vagy újonnan tárolja a felhasználó adatait a ranglistában, és menti azokat.

## 12. menuKezelo()

- Célja: A menürendszer kezelése.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: A függvény kezeli a játék különböző menüpontjait és átvált a megfelelő függvényekre az aktuális menüpont alapján.

## 13. jatekIndit()

- Célja: A játék inicializálása és indítása.
- Paraméterek: menu\_valtozoi\* m\_v - a játék állapotát tároló struktúra.
- Visszatérési érték: Nincs (void).
- Kódmagyarázat: Inicializálja a játékhoz szükséges változókat, és elindítja a menükezelő függvényt.

# Felhasználói Dokumentáció:

Ez a dokumentáció a "Legyen Ön Is Milliomos!" játék használatához készült. A játék célja, hogy a résztvevők kérdésekre válaszolva haladjanak előre a nyereménylistán, reménykedve a fődíj elnyerésében.

A játék konzolos alkalmazásként fut, így elindítása a futtatható állomány megnyitásával történik. A játék UTF-8 karakterkódolást használ, ezért bizonyosodjon meg róla, hogy a rendszer támogatja ezt a kódolást.

**Szükséges** a működéshez a ranglista.txt és a kérdések.txt file - ok.

**Főmenü:** A játék indításakor a főmenü jelenik meg, ahol több opció közül választhat:

- Start: Új játék kezdése.
- Ranglista: A legjobb 10 pontszám megtekintése, becenévvel és játékban töltött idővel együtt.
- User: Felhasználói beállítások (becenév megadása).
- Adatok Mentése: Menti a felhasználó adatait, ha már választott becenevet.
- Kilép: Kilépés a programból.

A menüpontok kiválasztása a billentyűzet használatával történik, az egyes menüpontok mellett látható számok megnyomásával.

## Játékmenet

- Nehézségi Szint Választása: Az új játék kezdésekor válassza ki a kívánt nehézségi szintet. A szintek 1-től 12-ig terjednek, ahol a magasabb számok nehezebb kérdéseket jelentenek.
- Kérdések és Válaszok: Minden kérdésnél négy válaszlehetőség közül választhat. A választ az a, b, c, d billentyűk megnyomásával adhatja meg.
- Segítségek: Két segítséget használhat a játék során:
  - 50/50: Két helytelen válasz eltávolítása.
  - Közönség segítsége: A közönség szavazatainak megjelenítése.



- Pontszámok és Haladás: Minden helyes válasz pontokat ér, és segít a játékosnak előrehaladni a ranglistán.
- Minden kérdés után a 'T' karakter beírásával léphet tovább. Ezt a program is kimutatja és látni fogja.

### **Ranglista**

A ranglistában a legjobb játékosok nevei, pontszámai és a játékban eltöltött idejük látható, csökkenő sorrendben. Itt nyomon követheti a legjobb teljesítményeket.

### **Játékos Beállítások**

A "User" menüpontban adhat meg vagy módosíthat becenevet, amely alatt a játékban szerepelni fog. Ha név nélkül játszik, a játék nem menti az eredményét.

### **Adatok Mentése és Kilépés**

A játékállás mentéséhez válassza az "Adatok Mentése" opciót. Kilépés előtt ajánlott ezt megtenni, hogy a jövőben folytathassa a játékot a megszerzett eredményekkel. A "Kilép" opcióval léphet ki a programból. Ha nem a kilép opciót válassza hanem eloltja a konzolt, a játék nem fogja lementeni adatait.