

Programozói dokumentáció

A Legyen ön is milliomos című játékprogram a tévés játékhöz hasonlóan működik. A programban egymás utáni kérdések következnek, melyekre 4 válaszlehetőség van, ebből az egyik helyes, a többi helytelen. Rossz válasz esetén a játékos kiesik és elviheti az eddig összegyűjtött pénznyereményt, jó válasz esetén tovább lép a következő kérdésre. Összesen 15 kérdésre adhat választ, ha a 15.-et is helyesen megválaszolta, akkor a játékos nyert. A játék során lehet segítséget kérni, lehet felezní a válaszokat, illetve kérni a közönség szavazatát. A játékosnak összesen 30 perce van megválaszolni a 15 kérdést, ezután a játék automatikusan befejeződik. A játék során a játékos lementheti a játékállást, amit később betölthet, a játékménü elhagyása során a program automatikusan lementi a játékos nevét, a játékban töltött idejét, és a nyert pénzösszeget egy "dicsoseg.txt" nevű fájlba. A dicsőséglista menübe lépve kiírja ezeket a nyeremény szerint csökkenő sorrendbe.

Adatszerkezetek választása

A program működését a "kérdések.csv" szöveges fájlban elmentett kérdések információi teszik lehetővé. A program ezeket az információkat ki kell olvassa a szöveges fájlból és el kell tárolja egy adatszerkezetbe. A kérdések nagy száma miatt a leghatékonyabban ezt láncolt listával lehet megoldani, így a program egy láncolt listába helyezi be ezeket. A láncolt lista elemei egy sztringre mutató pointert, illetve a láncolt lista következő elemére mutató pointert tartalmazzák. A program soronként olvassa be a fájlból az adatokat, és csak azokat helyezi be a láncolt listába, amelyek a kiválasztott nehézségi szinthez tartoznak, így elkerülve azt, hogy azoknak a kérdéseknek az információit is eltároljuk, amelyeket nem használnánk. Mivel egy sor a szöveges fájlban, amely tartalmazza a kérdések információit 100-200 karakterből is állhat, ezért ezeket dinamikusan foglalt sztringekbe helyezi el a program, így elkerüljük a felesleges memóriefoglalást. A program nem bontja szét egyből a kérdések információit, hanem egy dinamikusan lefoglalt sztringben tárolja el a kérdés összes információját, majd csak a játék futása közben a láncolt listából random kiválasztott kérdés információit tárolja el egy keresinfo nevű struktúrában, ami tartalmazza a nehézségszintet, a kérdést, a válaszokat, a helyes válasz betűjelét és a kérdés kategóriáját.

A dicsőséglista előállítására számára a program szintén egy láncolt listát használ. A program szintén soronként olvassa ki a "dicsoseg.txt" fájlban elmentett információkat, majd nyeremény szerint csökkenő sorrendbe helyezi bele az elemeket a láncolt listába. A láncolt lista elemei dicsoseg nevű struktúrákat tárolnak, melyekben eltárolják a játékos nevét, a játékban töltött idejét percben és másodpercben, illetve a játékban nyert nyereményét. A szöveges fájlban a név, a perc, a másodperc illetve a nyeremény pontosvesszővel van elválasztva, a program ez alapján bontja szét őket és helyezi bele a dicsoseg nevű struktúra megfelelő elemébe.

A játékmentések szintén egy szöveges fájlban vannak eltárolva, ennek adatai pontosvesszővel vannak elválasztva, mindegyik játékmentés külön sorban található. A program ezeket egy duplán láncolt listában tárolja el, a lista végein egy-egy extra elemet helyez el a program ennek érdekében. A duplán láncolt lista azért alkalmas a játékmentések tárolására, mivel a mentések menüjében ezek fel vannak sorolva, és PgUp illetve PgDn billentyűkkel kell kiválasztani őket ahhoz, hogy a kiválasztott játékállást betöltsük, vagy esetleg töröljük. A játékállások adatait tároló struktúra a játékos nevét, a játékban töltött idejét percben és másodpercben, a kérdés számát, ahol a játékos abbahagyta a játékot, illetve a nehézségi szintet tárolja.

A játék indításához szükséges információkat egy jatekinfo nevű strukturában tároljuk, amely tartalmazza a játékos nevét, a játékban töltött időt percben és másodpercben, a kérdés számát, ahol a játékos abbahagyta a játékot, illetve a nehézségi szintet. Ez azért szükséges, ha egy elmentett játékállást akarunk betölteni, akkor betöltjük a jatekinfo nevű struktúrába a kiválasztott játékmentés információit, és így indítjuk el a játékot. A program indításakor, és minden játékból történő kilépés után inicializálja a jatekinfo nevű struktúrát, annak érdekében, hogy ha új játékot szeretnénk indítani, akkor a nevet, a nehézségi szintet mi adjuk meg, valamint a játékot az első kérdéstől kezdjük, és az idő a játék indítása pillanatától számolódjon.

A függvények dokumentációja, illetve a program felépítésének és moduljainak magyarázata

Az int main() függvény vezérli a programot. A status integer adja meg, hogy a program milyen funkciót végezzen el, kezdetben 0-ra van állítva, ekkor a játék belép a menübe, 1 esetén elindul a játék, ha a status 2 a program belép a dicsőséglista menüjébe, 3 esetén a játékmentések menüjébe, 4 esetén egy új nevet olvas be, -1 esetén leáll a program.

A program létrehoz egy jinfo nevű struktúrát, ahova a játék indításához szükséges információk kerülnek, ebbe a program az indításkor betölti az új játék kezdéséhez szükséges információkat.

A menu modulban a menü vezérléséhez és kiírásához szükséges függvények találhatók.

A void printmenuinfo() megrajzolja a menü ablakát és kiírja a menü információit a konzolra.

A int gomenu() függvény a menü vezérléséért felel. Meghívja a printmenuinfo() függvényt, hogy kiírja a menü információit, majd a char kivalasztmenu() függvény segítségével beolvassa a felhasználó által megadott parancsot, és az adott parancsnak megfelelően a függvény visszaad egy számot, amely betöltődik a status változóba. A menüben elindíthatjuk a játékot az 's' betűre kattintva, megnyithatjuk a dicsőséglistát a 'd' betűre kattintva, a játékmentés menüjét a 'b' betűre kattintva, új nevet adhatunk meg az 'u' betűre kattintva, illetve kiléphetünk a játékból az 'x' betűre kattintva.

Ebben a modulban található egy void nevolvas(char *str), amely segítségével beolvashatjuk a játékos nevét, amely maximum 15 karakterből állhat.

A képernyőfunkciók nevű modulban kiírást segítő függvények vannak.

A void gotoxy(int x,int y) függvény a kurzort a konzol x, y pozíciójába helyezi.

A void clrscr() függvény frissíti a képernyőt.

A void setcolor(int szin) beállítja a kiírandó betű színét.

A void hidecursor() függvény eltünteti a kurzort, ezzel szemben a void showcursor() megjeleníti a kurzort.

A rajzol modulban a void drawbox(int x, int y, int magassag, int hossz) segítségével egy téglalapot rajzolhatunk a konzolra, a void drawline(int x,int y,int hossz) függvény segítségével egy vonalat, illetve a void drawmilliomos() függvénnyel kirajzolhatjuk a LEGYEN ÖN IS MILLIOMOS címet.

A lista nevű modul a láncolt listák előállításához, felszabadításához, illetve a fájlokból való olvasáshoz és íráshoz szükséges függvényeket tárolja.

A sor *beolvaskerdes(int nehezseg, int *darab) beolvassa a fájlból és eltárolja a kiválasztott nehézséghez tartozó kérdésinformációkat egy láncolt listába amelyet a függvényen belül állít elő, majd a függvény visszaad egy, a lista első elemére mutató pointer-t, illetve *darab paraméterén keresztül visszaadja a beolvasott kérdések számát. A dicsoseg *beolvasdicsoseg() függvény hasonló, csak az elemeket rendezetten tárolja el a nyeremény szerint csökkenő sorrendben. A menteselem beolvasmentes() függvény az eddigi függvényekhez képest a játékmentések információit egy duplán láncolt listában tárolja el, majd egy menteselem nevű struktúrát ad vissza, amely tartalmazza a lista két strázsaelemére mutató pointerét.

A void beirdicsoseg(char *nev, int m, int s, int osszeg) függvény megnyitja a dicsoseg.txt nevű szöveges fájlt, a kurzort a legutolsó karakter után helyezi, és beírja a dicsőséglita új elemét a megadott paraméterek szerint. A void mentesfajlba(jatekinfo *info) függvény hasonló az előző függvényhez, azonban a mentes.txt fájlba írja bele az új játékmentést a megadott paraméterek szerint. A void ujrairmentesfajlba(menteselem *info) függvény teljesen újírja a mentes.txt szöveges fájlt a játékmentések duplán láncolt listájában található elemekkel, ezért paraméterként a lista két strázsaelemére mutató pointerét kapja egy struktúrában tárolva. Ez szükséges, mivel ha törölünk egy elemet, azt a szöveges fájlból is ki kell törölnünk.

A void felszabaditkerdes(sor *e), void felszabaditdicsoseg(dicsoseg *e), void felszabaditmentes(mentes *e) a láncolt listák által elfoglalt memóriaterület felszabadítására alkalmasak, paraméterként a lista elejére mutató pointer-t kapják.

A dicsoseglista modulban a dicsőséglista menüjének vezérléséhez és kiírásához szükséges függvények találhatók.

A void printdicsoseginfo(dicsoseg *e) kiírja a dicsőséglistát, paraméterként a dicsőséglista elemeit tartalmazó láncolt lista elejére mutató pointer-t kapja.

Az int dicsoseglista() meghívja a dicsoseg beolvasdicsoseg() függvényt, ezáltal létrehoz egy láncolt listát a dicsőséglista elemeivel a nyeremény szerint csökkenő sorrendbe rendezve. A függvény kilépéskor felszabadítja a láncolt listának lefoglalt memóriát és 0 értékkel tér vissza.

A mentes modulban a játékmentés menüjének vezérléséhez és kiírásához szükséges függvények találhatók.

A void printmentesinfo(int x, menteselem q) kiírja a játékmentéseket, paraméterként egy számot kap, amely megadja, hogy hanyadik elem van kijelölve a listában, és a lista két strázsaelemére mutató pointer-t.

Az int jatekmentes(jatekinfo *info) meghívja a menteselem beolvasmentes() függvényt, ezáltal létrehoz egy duplán láncolt listát. A függvényben a PgUp és PgDn billentyűk segítségével kiválaszthatunk egy adott játékmentést, amit betölthetünk, vagy kitörölhetünk. A függvény egy jatekinfo struktúrára mutató pointer-t kap paraméterként, amelybe beletöltődnek a kiválasztott játékmentés adatai, ha egy elmentett játékállást szeretnénk betölteni. Ha elmentett játékot szeretnénk betölteni a függvény 1-el, egyébként 0 értékkel tér vissza. A függvényből történő kilépés előtt felszabadul a duplán láncolt lista számára lefoglalt memória.

A jatek modulban a játék menüjének vezérléséhez és kiírásához szükséges függvények találhatók.

A void printjatekinfo(int x, char *nev) a játék menüjének információit, a void printkerdesinfo(kerdesinfo seged) a kérdést és a hozzá tartozó választ, a void printnehezseginfo() a kiválasztható nehézségeket írja ki, a void printkozonseginfo(int *t) a közönség szavazatait írja ki a négy kérdésre. A void printwin(int penz) függvény közli a játékkal, hogy megnyerte a játékot, a void printlose(int penz) függvény pedig azt, hogy nem nyert, és vége a játéknak.

A kerdesinfo atalakit(char *s) függvény szétbontja a láncolt listában egyik tagjában szereplő kérdés információit, és egy struktúrába helyezi, amit visszaad. Segítségül a void masol(char *s1, char *s2, int *i) függvény szolgál, ami pontosvesszőig vagy a sztring végéig bemásolja az egyik sztring tartalmát egy másikba.

A void felez(kerdesinfo *p) függvény megfelel a kérdésekre a választ a kapott struktúrából kitörölve két választ, először leellenőrizve, hogy nem ezek a helyes válaszok.

A void kozonsegszavaz(int *t, char helyesvalasz) függvény előállítja a közönség szavazatát.

A void elteltido(time_t k_ido, int *minute, int *sec) függvény kiszámítja, és paraméterein keresztül visszaadja a játék kezdete óta eltelt időt percben és másodpercben.

Az int gojatek(jatekinfo jatek) maga a játékért felelős függvény. A paraméterként kapott jatek struktúrában kapja a játék indításához szükséges információkat. Ha nem adtunk meg nevet, akkor a függvény 4-es értékkel tér vissza, ezáltal a program beolvas egy nevet, és visszatér a menübe. Ha a függvény paraméterében 0 nehézségi értéket kapott, beolvas a felhasználótól egy új nehézségi szintet. A megadott nehézségi szint alapján létrehoz egy láncolt listát az adott nehézségi szintnek megfelelő kérdésekkel a sor *beolvaskerdes(int nehezseg, int *darab) függvény segítségével. Random kiválaszt egy kérdést a láncolt listából, majd a kerdesinfo atalakit(char *s) függvény segítségével betölti a kérdés információit egy kinfo nevű struktúrába. A függvény kiírja a játék információit, és parancsokat olvas be. Az 'A', 'B', 'C', 'D' karaktereket beolvasva a kérdésekre adhatunk választ, ekkor a program leellenőrzi, ha jó választ adtunk-e meg, ha igen, akkor továbblép egy új kérdésre, ha nem, akkor kilép a függvényből. Az 'm' karaktert megadva a program beírja a mentes.txt szöveges fájlba a pillanatnyi játék adatait, az 'f' karaktert megadva felezi a kérdésekre megadható válaszok számát, a 'k' karaktert megadva előállítja a közönség szavazatát, majd kiírja a konzolra, az 'x' karaktert megadva kiléphetünk a játék menüjéből. Ha nem ezek közül adunk meg karaktert, a program nem csinál semmit. A program minden parancs után leellenőrzi, hogyha nem adtunk-e meg már 15 helyes választ. Ha nem, tovább lép, ha igen, akkor kilép a játékmenüből, és azzal az üzenettel tér vissza, hogy nyertünk. Ha helytelen válasz miatt, vagy csak azért lépünk ki a játékmenüből, mert a kilépés parancsot adtuk meg, a program kiírja, hogy ez alkalommal nem nyertünk. A program mielőtt kilépjen a függvényből, beírja a dicsoseg.txt szöveges fájlba a játék állását, ahol leálltunk, és felszabadítja a láncolt listánkat, majd 0-ás értékkel tér vissza, így a menübe ugrunk vissza.