

Gonosz akasztófa fejlesztői dokumentáció

Bevezetés

A feladat egy gonosz akasztófa játék megírása volt. A gonosz jelző arra utal, hogy nem egyetlen szóra gondol, hanem a felhasználó által adott tippek alapján úgy szűkíti a szavak halmazát, hogy minél nehezebb legyen azt kitalálni.

A struktúra:

A függvények.h-ban van deklarálva. Tartalmaz egy dinamikus tömbre mutató pointert, amelyben a szó van tárolva, a szó hosszát, amelyet már a beolvasásnál megszámol a program, és, hogy a szó hányszor tartalmazza a tippelt betűt, ez minden tippnél változik.

```
typedef struct adat_elem
{
    char* szo;
    int hossza;
    int szoban_elofordulo_betu_szama;
} adat_elem;
```

Függvények modul:

Tartalmazza a program általános függvényeit.

fuggvények.c:

A dintomb_atmeretez függvény:

Paraméterként kap egy char típusú tömböt és egy int típusú méretet. A lefoglal a méretnek megfelelő nagyságú dinamikus tömböt a memóriába és a régi tömb tartalmát átmásolja az újba. Felszabadítja a régi tömböt. Az új tömbre mutató karakter típusú pointerrel tér vissza.

A hosszu_sort_olvas függvény:

Paraméterként nem kap semmilyen értéket. '\n'-ig beolvas a bemenetről a dintomb_atmeretez függvény segítségével egy folyamatosan növekvő dinamikus tömbbe. Visszatérési értéke char típusú tömbre mutató pointer.

A beker függvény:

Paraméterként nem kap semmilyen értéket. Visszatérési értéke integer. A hosszu_sort_olvas függvény segítségével beolvas a bemenetről ENTER leütésig. Ha 1 és 20 között van a határokat is beleértve akkor kilép a do while ciklusból, ha nincs, akkor vagy „Ervenytelen bemenet.\nKerem adja meg ujra:” vagy „Igy mar tul konnyu lenne.\nKerem adja meg ujra:”, figyelmeztetést ír ki. Végül visszatér a számmal.

A kivalogat függvény:

Paraméterként megkapja a szavakat tartalmazó dinamikus tömböt, azt, hogy hány szó szerepel a tömbben, a megfelelő szavak számát, amely kezdetben nulla és ebben a függvényben kap értéket és a kitalálendő szó hosszát. A függvény megszámolja, hogy hány olyan szó van a tömbben, amelynek a hossza egyezik a kitalálendő szó hosszával. Ennek megfelelő nagyságú dinamikusan foglalt tömböt hoz létre, amelybe átmásolja a kitalálendő szóval megegyező hosszúságú szavakat, a többi szót felszabadítja.

Az uj_jatek függvény:

Paraméterként kapja a kilép változót. A játék legvégén van lehetőség van annak újraindítására. A felhasználó 1-es vagy 2-es számot adhat annak megfelelően, hogy szeretne-e újra játszani. Ezt az értéket kapja meg az int típusú kilép változó a beker_vege függvényről.

A nehezseg függvény:

Paraméterként kapja a tippek számát. Kiírja a menüvezérlést: "A mozaik szavak is kis betuvel vannak megadva.\nFeladas: 0 \nJatek vegen kilepes: 1\nJatek vegen uj jatek inditasa: 2\n\nKerem adja meg, hogy maximum hany tippelési lehetoseggel szeretne játszani(maximum: 20): ". A tippek száma a beker függvényről kap értéket.

A felad függvény:

Paraméterként kapja a szavakat tartalmazó tömböt, azt, hogy hány szót tartalmaz a tömb, a nehezség függvényben értéket kapott kilep változót és a játék vége változót. Generál random egy számot egy és a szavak száma között. Ha a játék vége változó értéke egy, akkor kiírja, hogy "Megoldas: " és a még lehetséges szavakból egy random szót, majd meghívja az `uj_jatek` függvényt és átadja neki a kilep változót. Ha a játék vége változó értéke nulla, akkor pedig, hogy "Feladta a jatekot. Megoldas: „, és ugyanúgy egy random szót a lehetséges szavakból, majd meghívja az `uj_jatek` függvényt és átadja neki a kilep változót.

A tippeles függvény:

Paraméterként kapja a szavakat tartalmazó tömböt, a tömb által tartalmazott szavak számát, a kitalálandó szó hosszát, a tippek számát, a kilép változó értékét, a nem talált tippeket tartalmazó tömböt, a nem talált tippek számát, a kitalálandó szó tömböt(ebben tárolódnak a már eltalált betűk a megfelelő helyen), az eltalált tippeket tartalmazó tömböt és az eltalált betűk számát és a játék vége változó értékét. Feltölti a kitalálandó szó tömböt ' _ ' karakterekkel. Kiírja, hogy „A szó: „, majd annyi ' _ ' karaktert, amilyen hosszú a kitalálandó szó hossza. Kiírja, hogy "\nKerem adjon egy tippet: ". A kitalálta változó értékét nullára állítja, mert a while ciklus kilépési feltétele a kitalálta == 1 lesz. Bekér a felhasználótól egy tippet, amely, ha nagy betű, akkor kis betűvé teszi. A tippek számát csökkenti eggyel. Ha a tipp értéke 0, akkor meghívja a felad függvényt átadva neki a szavakat tartalmazó tömböt, a szavak számát, a kilep értékét és a játék vége változó értékét. Felszabadítja a megmaradt szavakat és végül a tömböt. Ha a tipp változó értéke egyenlő EOF-fal vagy nem esik a 'a' és 'z' közé, akkor kiírja, hogy "Ervenytelen karakter. Kerem probálja újra: " és a tippek számát megnöveli eggyel, hogy ilyenkor ne csökkenjen, mivel lehet félrenyomott a felhasználó egy karaktert. Ha érvényes a tipp, akkor megvizsgálja a függvény, hogy már volt-e valamikor az a betű, ha igen akkor figyelmeztet, hogy "Ez a betu mar volt.\n\n" és a tippek számát növeli eggyel, hiszen azért a betűért már csökkent a tippek száma. Ha tipp érvényes és még nem volt egyszersem, akkor pedig meghívja a `betut_beir` függvényt. Ha a tippek száma nagyobb egyenlő nullánál, akkor lehet, hogy a felhasználó kitalálta a szót, emiatt ilyenkor a program megszámolja, hogy a kitalálandó szó tömbben hány ' _ '-től eltérő karakter van és ha egyenlő a kitalálandó szó hosszával, akkor kitalálta, ha nem akkor pedig kiírja a megmaradt tippek számát, és hogy, "\n\nKerem adjon egy tippet: ". Ha a tippek száma eléri a nullát és nincs kitalálva a szó, akkor a játékos veszít, kiírja a "Vesztett. Elfogytak a tippeles lehetosegek.\n", a játék vége változó értékét egyre állítja, meghívja a felad függvényt és felszabadítja a szavakat és a tömböt. Ha játékos kitalálta a szót még azelőtt, hogy a tippjei elfogytak volna, akkor kiírja a program, hogy "Gratulalok, On kitalalta a szot es nyert.\n\n", meghívja az `uj_jatek` függvényt és felszabadítja a szavakat.

Betut_beir modul:

A program leglényegesebb része. Ez tartalmazza azokat a függvényeket, amelyek a tipp alapján elvégzik a megfelelő szűréseket a szavakon és megjelenítik a betűket a megfelelő helyen(helyeken).

betut_beir.c:

A nincs_benne függvény:

Paraméterként kapja a szavakat tartalmazó dinamikus tömböt, a szavak számát, a nem talált tippeket tartalmazó tömböt, a nem talált tippek számát, a tippelt betűt, hogy hány darab szót tartalmazza legkevesebbszer a betűt, hogy ez a legkevesebbszer pontosan mennyi, a kitalálandó szó hosszát és a kitalálandó szó tömb által tartalmazott eltalált betűket a megfelelő helyen. Mivel ez a függvény sehova nem fog beírni betűt, mivel a legkevesebb betű nulla lesz, így kiírja, hogy a menüt, azt, hogy nem talált a tipp, a kitalálandó szó tömb tartalmát a „A szo:” felirat után, majd a nem talált tippeket is megjeleníti. Megszámolja, majd kiválogatja azokat a szavakat, amelyek egyszersem tartalmazzák a tippelt betűt, ezeket átmásolja egy új tömbbe, a régi tömböt és azokat a szavakat, amelyek tartalmazzák a betűt felszabadítja.

Az egyszer_fordul_elo függvény:

Paraméterként kapja a szavakat tartalmazó dinamikus tömböt, a szavak számát, a nem talált tippeket tartalmazó tömböt, a nem talált tippek számát, a tippelt betűt, hogy a szavak közül hányszor fordul elő legkevesebbszer a tippelt betű, a kitalálandó szó tömb által tartalmazott eltalált betűket a megfelelő helyen, az eltalált tippeket tartalmazó tömböt, az eltalált tippek számát és a kitalálandó szó hosszát. A függvény megnézi, hogy mely szavakban fordul elő

csak egyszer a tippelt betű, és, hogy az melyik indexen a leggyakoribb. A leggyakoribb indexre beírja a betűt és kiválogatja azokat a szavakat, amelyek csak egyszer és ott tartalmazzák a betűt, a többit felszabadítja. Megjeleníti még a nem talált tippeket is és a kitalált betűket a megfelelő helyen.

A `tobbszor_fordul_elo` függvény:

Paraméterként kapja a szavakat tartalmazó dinamikus tömböt, a szavak számát, a nem talált tippeket tartalmazó tömböt, a nem talált tippek számát, a tippelt betűt, kitalálendő szó tömb által tartalmazott eltalált betűket a megfelelő helyen, az eltalált tippeket tartalmazó tömböt, az eltalált tippek számát és a kitalálendő szó hosszát és a szavak közül hányszor fordul elő legkevesebbszer a tippelt betű. Külön vizsgálja, ha már csak egy szó jöhet szóba, hiszen ilyenkor nem kell válogatni a tömbből, csak az egy szóban vizsgálni a tippet. Ha még több szó szerepel a tömbben, akkor azok közül, amelyek a legkevesebbszer tartalmazzák a betűt kiválaszt random módon egyet és megvizsgálja, hogy az hol tartalmazza a betűt, majd ennek megfelelően kiválogatja a többi szót, amelyik pedig nem felel meg azt felszabadítja. Megjeleníti a nem talált betűket és a talált betűket a megfelelő helyen.

A `betuket_beir` függvény:

Paraméterként kapja a szavakat tartalmazó dinamikus tömböt, a szavak számát, a tippelt betűt, a nem talált tippeket tartalmazó tömböt, a kitalálendő szó hosszát, a kitalált betűk számát, a kitalálendő szó által tartalmazó betűket a megfelelő helyen, a tippek számát, a talált tippeket tartalmazó tömböt és a nem talált tippek számát. Ez a függvény választja ki a tippnek megfelelően, hogy melyik függvényt kell meghívni, az alapján, hogy mi a tippelt betű előfordulásának egy szóban a legkisebb értéke.

Beolvas modul:

Tartalmazza a beolvasáshoz szükséges függvényt.

beolvas.c:

Paraméterként kapja a szavakat tartalmazó dinamikus tömböt, a szavak számát, amely kezdetben nulla, és, hogy mennyi a leghosszabb szó karakter száma, amely kezdetben szintén nulla és majd ebben a függvényben kap értéket. Itt valósul meg a fájlkezelés. Megnyitja a program a `magyar_ascii.txt` forrásfájlt olvasásra, megszámlolja a benne lévő sorok számát, majd ekkora területet foglal le a memóriába. Bezárja, majd megnyitja újra a fájlt és a `dintomb_atmeretez` függvény segítségével beolvassa a dinamikus tömbbe a szavakat betűnként egészen a fájl végéig. Vizsgálja, hogy sikerült-e megnyitni a fájlt és, hogy sikerült-e memóriát foglalni.

A forrásfájl txt formátumú, csak szavakat tartalmazhat az angol abc kis betűivel, sortöréssel elválasztva. Ha üres a fájl, akkor kiírja, hogy nem sikerült memóriát foglalni és kilép -1-es hibakóddal. A hibás fájl nem kezeli a program.