

Akasztofa kicsit másképpen

Olvasd el figyelmesen a kiírást, ha kell többször is! A feladat aránylag röviden megoldható, csak a leírás részletes és nem rejtett szándéka, hogy erősítse a feladat értelmezési és a feladatkiírás alapján történő kivitelezési képességet. Csak a pontosan, a kiírás alapján megoldott és beküldött programokat fogadom el! Ez ne tántorítson el senkit. Az "elolvasom egyszer, túl sok és összetett volt a leírás és nem értettem meg semmit" rossz taktika viszont, ajánlott megoldási technika: neki kell ülni, elolvasni a kiírást egyszer teljesen, majd folyamatosan olvasni újra és párhuzamosan megvalósítani az leírtakat pontról pontra. Mindig csak az adott mondatot figyelve. Nem kell az egész feladatot pontosan átlátni, ahogy haladsz előre mindig bővül egy kicsit kódod az adott részfeladatra koncentrálni és egyre jobban átlátod majd, hogy mivel is tudsz dolgozni és mit kellene velük csinálni, onnantól kezdve leegyszerűsödik a dolog. Tehát csak merj nekikezdeni!

Feladatkiírás:

A program megoldásához szabadon használható a *time.h* és a *string.h* header is.

Készítsünk egy struktúra típust (**typedef struct**) melynek a neve **akasztofa**. Kettő adattagja (attribútuma) lesz: az egyik egy 20 nagyságú karaktertömb, melynek a neve **szo**, a másik egy egész értékű változó, melynek neve **testreszek**.

A main függvényen belül készítsünk egy random generátort, mely 0 és 2 között [0,2] sorsol egy számot.

Ezután hozz létre 3 **akasztofa** típusú struktúra elemet. A nevük: **babu1**, **babu2**, **babu3**, valamint egy ugyanilyen típusú 3 elemű tömböt, melynek a neve: **babuk** (Megj.: természetesen mivel az **akasztofa** egy típus a typedef miatt, ezért ugyanúgy hozunk létre egy ilyen típusú változót vagy tömböt, ahogy eddig is tettük az egyszerű típusokkal).

Ezután készíts egy **ertekadas** nevű függvényt, mely void típusú lesz és háromszor kell majd meghívni a main függvényben egymás után. 3 darab bemeneti paramétere legyen: az első az adott bábú [**babu1** majd **babu2** majd **babu3** - mivel 3 hívás lesz], a másik kettő pedig az a két érték amit, majd az így átadott bábúnak állítunk be attribútum értéként az **ertekadas** függvényen belül. Tehát a bábúinknak van egy-egy **szo** és egy-egy **testreszek** attribútuma, ezeknek fogjuk átadni értékül a második és harmadik paramétert az **ertekadas** függvényen belül. Az átadott szó tetszőleges lehet, csak ne haladj meg a 19 karaktert. A testreszek száma kezdetben 0, tehát a harmadik paraméter standard 0 mindhárom függvényhívás esetén.

Ahhoz, hogy ténylegesen az átadott bábúhoz tartozó változók értékét írjuk át, referenciaként kell átadni (&) és az **ertekadas** függvény fejlécében illetve a függvényen belül is dereferenciaként kell hivatkozni az elemekre (*). Egy elemre való hivatkozás a függvényen belül pl.:

(*strukturaPeldany).strukturaAttributum . (Megj.: ezek csak példa nevek voltak)

Miután mind a 3 **akasztofa** példány (**babu1**, **babu2**, **babu3**) átesett az értékadáson, tegyük be őket a már előzőleg létrehozott **babuk** tömbünkbe szép sorban. (Megj.: nem kell ciklus, manuálisan is beletehetjük őket a 0., 1. és 2. helyre, ahogy azt már láttuk.)

Szükségünk lesz még egy 20 nagyságú **eredmeny** nevű karaktertömbre.

A random generátorra azért van szükség, hogy kiválasszuk az egyik **akasztofa** elemet a **babuk** tömbünkből.

Az említett eredmény tömböt töltjük fel '_' karakterekkel olyan hosszban, amilyen hosszú a sorsolt **babuk** tömb eleméhez tartozó szó hossza. Hivatkozás adott elemhez tartozó szó attribútumra pl.: *babuk[randomSorszam].szó* , ahol randomSorszam értéke lehet {0,1,2}, melyet generátorral sorsoltunk.

Ne felejtjük el a 0 karakterrel lezárni az **eredmeny** tömböt.

Mindent előkészítettünk, a következőkben az alábbi műveletet kell megvalósítani:

Egy ciklussal haladjunk végig a sorsolt szón (hivatkozás ugyanúgy lehetséges, mint ahogy a megelőző bekezdésben említve lett). Az eredeti akasztófához képest annyi módosítást teszünk, hogy nem tetszőleges sorrendben kell kitalálni a szóban lévő betűket, hanem sorban haladva, azaz először az első karaktert kell eltalálni, majd ha az meg van akkor a másodikat és ha az meg van léphetünk a harmadikra és így tovább. Minden betű kitalálására van **6** lehetőségünk. Az elhasznált lehetőségek számát mindig frissítjük a babúhoz tartozó **testreszek** attribútumban (hivatkozás ugyanúgy lehetséges mint a **szó** esetén). Tehát, ha az adott karakter nem kerül kitalálásra egyből, akkor minden egyes alkalommal növeljük a **testreszek** értékét. Ha sikerül 6 alatt kitalálni a karaktert, akkor léphetünk a következő karakterre és állítsuk vissza 0-ra a **testreszek** számát, hogy valóban minden egyes karakterre legyen 6 lehetőségünk. Kitalálás alatt azt értjük, hogy bekérünk a játékostól egy karaktert, összehasonlítjuk az adott pozícióban lévő karakterrel és ha egyezés van, akkor eltaláltuk, ellenkező esetben nem. Ha bármelyik karaktert is nem sikerül 6 próbálkozás alatt kitalálni, akkor veszítettünk és a játéknak vége, ellenkező esetben nyertünk és a játéknak vége.

Jelezzük mindig a játékos felé, hogy éppen sikerült-e kitalálni a karaktert és ha nem, akkor mennyi próbálkozása van még az adott karakterre.

A végén meg persze közöljük a játékoskal, hogy nyert avagy sem.

Emellett lehet segítséget is kérni természetesen a pluszpont elűzése nélkül is. Keressetek bátran.

Jó munkát!